UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE
Sciences et Ingénierie pour l'information (S2I)

Laboratoire Informatique, Image et Interaction (L3i)

THÈSE

présentée par :

François PICARD

soutenue le 11 juillet 2011
pour l'obtention du grade de Docteur de l'Université de La Rochelle

Discipline : Informatique et applications

Contextualisation & Capture de Gestuelles Utilisateur
Contributions à l'Adaptativité des Applications Interactives Scénarisées

---

JURY :

Jean-Pierre JESSEL	Professeur, Université Paul Sabatier Toulouse III, Président du jury
James L. CROWLEY	Professeur, Institut National Polytechnique de Grenoble, Rapporteur
Saida BOUAKAZ	Professeur, Université Claude Bernard Lyon 1, Rapporteur
Didier ARQUES	Professeur, Université de Marne la Vallée, Examinateur
Philippe SOUCHET	Responsable Recherche & Développement, XD Productions, Examinateur
Pascal ESTRAILLIER	Professeur, Université de La Rochelle, Directeur de thèse

Thèse réalisée au ***Laboratoire Informatique, Image, Interaction*** (L3i)
Pôle Sciences & Technologies, Université de La Rochelle
Avenue M. Crépeau
17042 La Rochelle cedex 01

Tél : +33 5 46 45 82 62
Fax : +33 5 46 45 82 42

Web : http://l3i.univ-larochelle.fr

Sous la direction de Pascal Estraillier pascal.estraillier@univ-lr.fr

Financement Convention ***CIFRE***, avec la société XD Productions
145 rue Jean-Jacques Rousseau
92130 Issy Les Moulineaux

Tél : +33 1 41 33 04 50
Fax : +33 1 41 33 04 55

Web : http://www.xdprod.com

Résumé

Depuis 50 ans, une évolution permanente de l'interaction homme-machine a permis qu'aujourd'hui, nous tendions vers des systèmes temps réel proposant une interaction simple et intuitive à l'utilisateur et s'adaptant automatiquement à l'activité observée et interprétée. L'utilisateur peut dorénavant interagir avec un système informatique, volontairement ou de manière non consciente, par le biais de plusieurs modalités, comme il ferait dans la vie courante.
Actuellement, les systèmes les plus développés sont ceux permettant à l'utilisateur d'interagir par le biais de gestuelles, qu'elles soient explicites ou implicites. Ces systèmes sont intégrés de plus en plus dans notre environnement, la plupart du temps invisibles pour nous, et s'adaptent en fonction d'un scénario qui définit les objectifs que nous devons atteindre. Plusieurs domaines d'application sont le cadre du développement de tels systèmes, comme celui du jeu vidéo ou encore de la surveillance vidéo.

Nous proposons, dans ces travaux de thèse, l'architecture d'un système interactif, sur lequel s'exécute une application scénarisée de type jeu vidéo.
L'interactivité est alimentée par les gestuelles du corps d'un unique utilisateur, en temps réel et de manière non invasive. Le système réagit également à divers événements prenant place au sein de la scène réelle, qui sont dus au dynamisme de cette dernière et dont l'utilisateur n'est pas directement responsable (changement d'éclairage, présence de spectateurs, etc.).
L'utilisateur est immergé au sein d'un environnement virtuel qui traduit la réponse interactive du système. Ce dernier répond également, en temps réel, à l'activité observée et interprétée, de manière adaptative, adaptant à la fois sa réponse interactive et son fonctionnement global.

Nos travaux se basent sur l'hypothèse que l'activité observée au sein de la scène est caractérisée par le contexte d'interaction qui l'englobe. Notre système reconnaît ainsi l'activité en modélisant le contexte d'interaction au sein duquel elle prend place. Notre contribution principale se traduit donc par l'introduction de la notion de contexte au sein des processus interactif et adaptatif. La modélisation du contexte d'interaction que nous proposons nous a servi de base pour celle du scénario d'une application. La gestion et l'analyse de ce contexte au cours de l'interactivité permettent au système d'interpréter l'activité observée et de paramétrer les différents mécanismes adaptatifs qui en découlent.
L'activité est capturée et codifiée par un système étendu, adopté en accord avec le cadre industriel de cette thèse (convention ***CIFRE*** avec la société XD Productions). Initialement dédié à la capture, invasive et en milieu contrôlé, des mouvements de l'utilisateur, nos travaux ont permis l'augmentation du processus vers une capture plus générale de l'activité globale, de manière non invasive et en environnement dynamique.
Enfin, nous avons développé une application immergeant l'utilisateur au sein d'une simulation virtuelle d'entrainement au tennis. Par le biais d'études de cas extraites de ce scénario, nous avons implémenté les processus interactif et adaptatif prenant place entre le système et l'utilisateur. L'adaptativité du système, supportée par le scénario de l'application, est concrétisée par la mise en place de mécanismes spécifiques à tous les niveaux de l'architecture, en fonction de l'activité observée et interprétée. Nous mettons en évidence un ensemble de boucles logicielles, appelées « boucles vertueuses », générées par l'accumulation des effets d'adaptation du système et améliorant en permanence l'interactivité.

Les perspectives à ces travaux de thèse concernent la formalisation et la gestion haut niveau de notre modèle de contexte, la complexification de notre scénario et l'amélioration de notre système de capture dans le cadre de nouvelles applications scénarisées.

Mots clés : Interactivité, adaptativité, contexte, temps réel, capture de gestuelles non invasive en milieu non contrôlé, caractérisation et interprétation d'activité, jeu vidéo.

Abstract

For 50 years, human-computer interaction has evolved permanently. Nowadays, real-time systems offer the user simple and intuitive interaction and adapt automatically to the observed and interpreted activity. From now on, the user can interact with computer systems, voluntarily or unconsciously, by several modalities, as he would do it in his everyday life.
Currently, the most developed systems are the ones allowing the user to interact by his explicit or implicit body language. These systems are more and more integrated in our environment, are invisible, and adapt according to the scenario that defines the objectives the user has to achieve. Several application fields are the development frameworks of such systems, like video games or video surveillance.

We define in this work the architecture of an interactive system, upon which runs a scripted application such as a video game.
The interactivity is fed by an only user body language, in real-time and without markers. The system reacts as well to various events taking place within the real scene. These events are due to the scene dynamism for which the user is not directly responsible (illumination changes, spectators, etc.).
The user is immersed within a virtual environment that renders the system interactive response. The system responds adaptively as well, in real-time, to the observed and interpreted activity, adapting at the same time its interactive response and its global functioning.

Our work is based on the following hypothesis: the activity, observed within the scene, is characterized by the interaction context that includes it. Thus, our system recognizes the activity by modelling the interaction context within which it takes place. Our main contribution is then the introduction of the notion of context within the interactive and adaptive processes. We use our interaction context modelling to model the application scenario. The context management and analysis during the interactivity allow the system to interpret the observed activity and to configure the different adaptive mechanisms which follow it.
The activity is captured and codified by an extended system, chosen according to our work industrial framework (Industrial Convention of Formation by Research – ***CIFRE*** – with XD Productions). Initially dedicated to the invasive capture of the user movements in a controlled environment, our work enhances the process towards the general capture of the global activity, without markers and within an uncontrolled environment.
Finally, we have developed an application immersing the user within a virtual simulation of tennis training. Thanks to different case studies extracted from this scenario, we have implemented the interactive and adaptive processes taking place between the system and the user. The system adaptivity, supported by the application scenario, is materialized by the implementation of specific mechanisms at every level of the architecture, according to the observed and interpreted activity. We have brought to light a set of software loops, called virtuous loops, generated by the accumulation of the system adaptation effects and improving the interactivity permanently.

Our work perspectives concern the formalization and the high-level management of our context model, the complexity of our scenario and the improvement of our capture system within the framework of new scripted applications.

Keywords: Interactivity, adaptivity, context, real-time, markerless body language capture in uncontrolled environment, activity characterization & interpretation, video game.

Remerciements

Difficile exercice que celui des remerciements, celui où le 'je' remplace le 'nous', et pourtant tellement attendu, synonyme d'une page qui se tourne et d'une autre qui débute.

J'exprime tout d'abord mes remerciements aux différents membres du jury qui ont accepté d'évaluer ce travail de thèse.
Un grand merci à Saida Bouakaz et à James Crowley d'avoir accepté d'être les rapporteurs de ce long manuscrit, je sais que la tâche a été ardue. Leurs commentaires et interrogations m'ont été très utiles pour la préparation de la soutenance et la mise à jour finale du manuscrit.
Je remercie également Jean-Pierre Jessel, Didier Arquès et Philippe Souchet pour leur participation à mon jury en tant qu'examinateurs. J'en profite pour ajouter un merci très particulier et très chaleureux à Philippe, qui a été mon chef à moi pendant 4 années pleines d'aventures et de rebondissements, un support permanent et précieux en tout temps et une amitié qui est allée bien plus loin que le simple cadre professionnel.

Merci à Jacques Peyrache de m'avoir permis de conduire ma thèse au sein de sa société XD Productions.

Mon infinie gratitude, et ce n'est pas peu dire, à Pascal Estraillier.
Tout d'abord, pour avoir accepté de diriger ma thèse. J'ai pu mener à bien ce travail de thèse grâce à son recul, sa vision et ses conseils avisés. C'est aussi grâce à lui que j'ai pu, littéralement, terminer ma thèse dans une situation professionnelle stable.
Mais ces 4 années sont également beaucoup plus loin que la simple collaboration professionnelle et sa gentillesse, sa générosité et son sens de la famille (famille non moins adorable qui plus est) m'inspireront toujours un profond respect ainsi qu'une reconnaissance éternelle. C'est grâce à lui que j'ai pu apprendre que deux colombiennes insomniaques sont nettement plus efficaces que du café, colombien ou autre, ou encore qu'un réveil matin ☺.
Trop peu de lignes pour tellement de choses à dire. Pascal, merci du fond du cœur.

Je tiens à remercier sincèrement les différentes personnes qui ont rythmé ma vie dans les deux cadres de travail où j'ai travaillé et qui m'ont, explicitement et/ou implicitement, professionnellement et/ou personnellement, aidé au cours de ma thèse.

Concernant XD, un grand merci à Micka et Jérémy, l'un pour son immense gentillesse, l'autre pour son humour décapant (quand on entend ses blagues…) et que j'ai toujours grand plaisir à retrouver aujourd'hui. Ce fut un plaisir, les gars, de travailler avec vous. Je n'oublie bien sûr pas les infographistes, particulièrement Fredo & Nico, qui restent des amitiés chères également. Une mention particulière à Mister B.B. pour des raisons que seules des heures de souvenirs racontés peuvent expliquer (à quand ta crémaillère « rougail-saucisse » !?).

Quant au L3i, la liste est longue et sera certainement incomplète. Quoiqu'il en soit, vous allez tous me manquez.
Je commencerai bien sûr par mes 3 autres fantastiques. Aaaaaahhhh que de souvenirs !! Merci pour ces moments qui n'appartiennent qu'à nous. Merci Romain, pour tout cela (comment résumer ?), pour la vie et les souvenirs que tu m'as offerts à La Rochelle (ils s'en est passé des choses, dans ce port…). Merci également Guillaume et Patoche, mes compagnons fidèles. J'ai toujours pu compter sur vous 3. Sachez que la réciproque sera toujours vraie.

Mes pensées chaleureuses et affectueuses à vous également : Broza & Brozette, pour votre générosité exceptionnelle et votre aide inestimable ; Micka, toujours là, toujours à côté, toujours disponible ; Matthieu, pour ton support précieux, aussi bien professionnel que personnel ; Abdallahi, l'homme de toutes les clés, pour sa générosité sans pareille, on a bien rigolé ☺ ; Mathéo & Gaël, pour les bons moments de franches rigolades, je vous attends sur Paris. Et tous les autres, bien sûr : Antoine, Cyril, Francky, Nath, Sophea, Dounia, Thomas, Dom, Damien, Clément, Olivier, Jérémy. Tuan, je ne t'oublie pas, merci pour ton sourire, ce fut un plaisir de t'enseigner un peu de vocabulaire français ☺.
Enfin, merci à tous les autres membres du laboratoire que j'ai eu l'honneur de côtoyer ces années et qui sont synonymes de bons souvenirs et de sourires : Armelle, Vincent, Ronan, Michel, Kathy (je n'ai toujours pas vu de danse tahitienne…), Arnaud, Rémi, Alain, Frédéric, Christophe, Caroline, Patrick, Jean-Christophe, Jean-Marc, Ismail, Karell, Muriel, Bertrand, Jamal,… j'en oublie assurément, mais ils sont dans ces lignes tout de même.

Je remercie ensuite tous mes compagnons de vie, qui m'ont supporté, d'une manière ou d'une autre, à travers leurs mots, leurs pensées, leurs moments. J'en oublierai et je m'en excuse mais j'essaye de faire que ma vie ne soit pas un long fleuve tranquille et qu'elle rencontre la route de nombreuses personnes, m'accompagnant alors. Toutes ces personnes sont ici.

Un immense merci et ma fidèle amitié à vous : mes vichyssois, certains de moins en moins vichyssois mais tous de plus en plus amis ; mes lourds, sans les juger ; mes CPEiens qui sont bien plus dorénavant ; mes mexicains et mes espagnols, mis carnales ; mes parisiens que je retrouve bientôt avec joie ; mes lyonnais ; mon R6 ; mes compagnons de Lausanne ; tous les autres… impossible de vous citer tous mais chacun se reconnaitra. Vous avez été tous là durant ce travail de thèse et j'espère vous prouver mon amitié autant que vous le faites pour moi à chaque instant.

Enfin, et surtout, un merci toujours insuffisant, à ma famille, à ma tribu, pour leur écoute, leur aide et leur amour, que j'essaye de leur rendre au mieux tous les jours. Particulièrement, Merci à ma môman, sans toi, tout cela n'aurait pas été possible. Il en va de même pour toi, papa. Merci pour tous ces mots et ce soutien. Merci Julie, Merci Cécile, ♥ (je sais faire maintenant) et merci à Mathieu et à Guillaume, mes beaufs préférés.

Je dédie tout cela à mon Papy 'Roanne'…

Table des matières

Table des figures

Chapitre 1. Cadre de la Thèse Applications Interactives à Exécution Adaptative

Chapitre 2. Gestion de contexte

Chapitre 3. Architecture générale de notre système

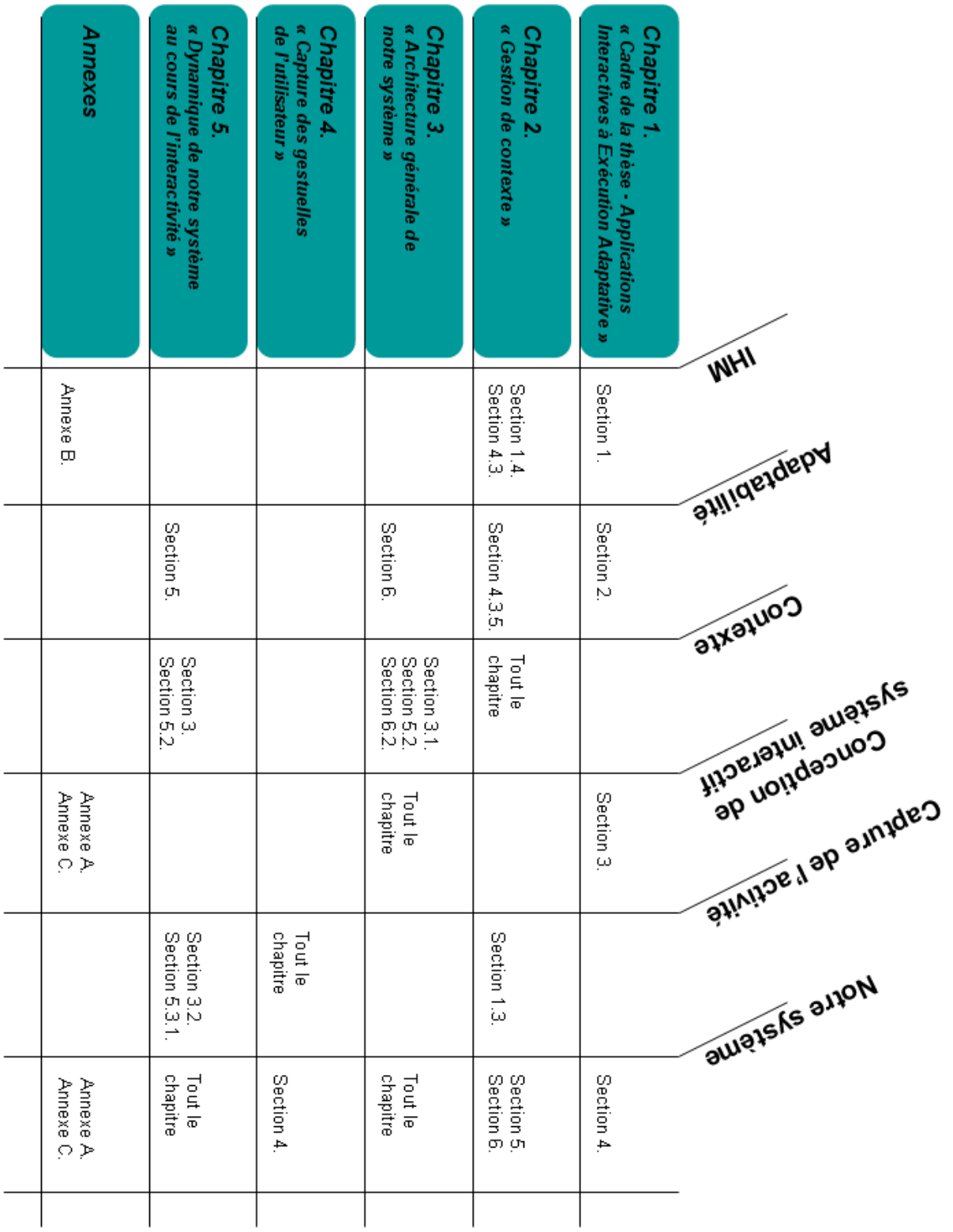
Chapitre 4. Capture des gestuelles de l'utilisateur

Chapitre 5. Dynamique de notre système au cours de l'interactivité

Annexe B. Historique de l'Interaction Homme-Machine

Guide de lecture

Nous proposons au lecteur le guide suivant, dans le but de l'orienter vers les bonnes sections et bons chapitres, en fonction des problématiques qu'il souhaite aborder. Le lecteur peut ainsi parcourir stratégiquement ces travaux de thèse, en fonction de ses intérêts.



Introduction

Cadre scientifique de la thèse

Depuis l'élaboration, dans les années 1960, du système ***SketchPad***, considéré comme l'un des premiers systèmes interagissant avec l'utilisateur par le biais d'une interface graphique, jusqu'aux systèmes actuels, tels que l'***iPhone***, ou encore la console de jeu ***Kinect***, l'interaction entre l'utilisateur et le système a été révolutionnée. Aujourd'hui, **deux constats** sont possibles concernant cette évolution.

Tout d'abord, au fur et à mesure des années, **le besoin d'une interaction plus simple et plus intuitive** s'est fait ressentir.
Pourtant encore omniprésente, **l'interaction standard**, i.e. l'interaction avec un système par le biais d'une souris et d'un clavier, a vite montré ses limites. Elle nécessite en effet une phase d'apprentissage parfois laborieuse et met en œuvre une interaction non naturelle avec l'utilisateur. Cette forme d'interaction s'est pourtant imposée et continue d'être la manière d'interagir la plus usitée de nos jours.
L'interaction instrumentalisée s'est ensuite poursuivie par **la démocratisation des périphériques non standards**, tels que des souris 3D, joysticks ou périphériques adoptant une apparence connue de l'utilisateur (un volant par exemple). Quoique parfois peu intuitifs, ces dispositifs ont permis d'interagir avec les systèmes informatiques de manière plus élaborée et plus naturelle.
Depuis peu, de nouveaux systèmes sont apparus, permettant à l'utilisateur d'interagir **via des gestuelles naturelles**, par le biais de différentes parties de son corps. Les gestuelles qu'adopte l'utilisateur ne nécessitent pas de phase d'apprentissage, permettant ainsi à ce dernier d'interagir avec le système comme il le ferait dans la vie courante. Ce type de systèmes traduit une interaction plus riche et plus adaptée à la tâche. Les premiers systèmes, dits de **'capture'** de **'mouvements'** du **'corps'**, ne furent pas destinés à l'interaction homme-machine mais furent motivés par des applications spécialisées, telles que l'animation d'un avatar (dans un film par exemple) ou encore l'étude d'une gestuelle particulière (golf, danse, etc.). Ces systèmes impliquent que l'utilisateur porte un équipement (électromagnétique, inertiel, optique, etc.) parfois encombrant et fragile. Par la suite, plusieurs systèmes ont proposé **une capture non invasive** des gestuelles de l'utilisateur, dans un contexte applicatif s'intéressant beaucoup plus à l'interactivité entre ce dernier et le système (dans le domaine du jeu vidéo par exemple). D'autres **modalités** sont également de plus en plus envisagées, telles que la voix par exemple, mais nous nous intéresserons, dans ces travaux, uniquement aux systèmes interagissant avec l'utilisateur par le biais de ses gestuelles. Cette classe de systèmes est la plus développée actuellement.

Le deuxième constat concerne **la capacité adaptative des systèmes interactifs**, i.e. leur capacité **à s'adapter**, **automatiquement et en temps réel**, à **l'activité observée**. L'interaction entre le système et l'utilisateur a longtemps été un dialogue convenu, dirigé par le système, imposant à l'utilisateur une gestuelle particulière à adopter, parfois limitée et pauvre au niveau du sens véhiculé et artificielle quant aux gestuelles à effectuer. C'est pourquoi les systèmes actuels sont de plus en plus capables d'analyser **en temps réel** l'activité observée, au cours de laquelle l'utilisateur évolue, pour ensuite réagir **en adéquation avec celle-ci**. A présent, c'est le système qui s'adapte, laissant l'utilisateur adopter,

beaucoup plus librement et naturellement, ses gestuelles, aussi bien **explicites** (i.e. effectuées de manière volontaire et consciente) qu'**implicites** (i.e. effectuées sans désir de la part de l'utilisateur d'interagir avec le système). L'utilisateur ne subit plus l'interaction piloté par le système, et une véritable conversation, compréhensible par les deux parties, s'instaure. L'interactivité système – utilisateur est améliorée en permanence par la capacité adaptative du système.
La réponse adaptative du système peut être mise en œuvre au niveau de la scène réelle observée, de l'interface avec l'utilisateur, du scénario, ou encore du fonctionnement interne du système. Les mesures adaptatives sont mises en place par le système à la suite de **l'interprétation de l'activité**, particulièrement caractérisées par les gestuelles adoptées par l'utilisateur. Ainsi le système adapte sa réponse vis-à-vis de la gestuelle de l'utilisateur **caractérisée** et **identifiée**. Les systèmes ne se bornent dorénavant plus à la reproduction de la gestuelle utilisateur, dans le but d'animer visuellement un personnage virtuel, mais inclut dans le processus interactif une analyse intelligente de celle-ci, pour y répondre en adéquation.
L'évolution des systèmes adaptatifs implique que l'utilisateur interagisse, en temps réel et à tout moment, par le biais de gestuelles naturelles, avec des systèmes qui sont partout et qui peuvent être invisibles pour lui. Nous constatons ainsi l'émergence de **l'ubiquitaire** et du **pervasif**, supportés par des problématiques **d'intelligence ambiante**. L'utilisateur n'interagit pas forcément de manière volontaire et consciente, ses gestuelles n'étant pas imposées par le système adaptatif, en suivant le scénario, plus ou moins élaboré, de l'application, dans le but d'atteindre des objectifs définis soit par lui-même, soit par le scénario. De plus en plus de domaines d'application sont le cadre du développement de systèmes adaptatifs. C'est le cas du domaine du **jeu vidéo** ou encore de la **surveillance vidéo**.

Sujet de thèse *Contextualisation & Capture de Gestuelles Utilisateur : Contributions à l'adaptativité des applications interactives scénarisées*

Ce sujet de thèse a été élaboré dans le cadre de ces deux constats. Nous nous plaçons dans l'hypothèse d'**une application interactive mono-utilisateur** dont la logique d'exécution a été **scénarisée**. Nous souhaitons définir une plateforme permettant de gérer **une interaction gestuelle naturelle et non contrainte** de l'utilisateur. Nous avons fait le choix d'un système de capture **non invasif**. Une des fonctions importantes de cette plateforme repose sur sa capacité à permettre **l'adaptation**, **en temps réel**, de la logique d'exécution de l'application en fonction du comportement **observé** et **interprété** de l'utilisateur, mais respectant un cadre scénaristique prédéfini. Nous plaçons dans un contexte applicatif englobant principalement **le jeu vidéo**.

Nous présentons donc, dans nos travaux, **l'élaboration d'un système interactif**, avec lequel l'utilisateur interagit, **en temps réel et sans contraintes matérielles** (capture non invasive), par le biais des **gestuelles de son corps** et de manière **libre** et **naturelle**. L'utilisateur évolue librement dans un environnement non contrôlé, environnement au sein duquel d'autres événements, dont l'utilisateur n'est pas directement responsable, peuvent survenir. **L'activité**, principalement caractérisée par les gestuelles utilisateur, est capturée et modélisée virtuellement par un système dédié et augmenté, adopté en accord avec le contexte industriel de cette thèse (convention ***CIFRE*** avec la société XD Productions).

Le système interprète l'activité observée en **contextualisant** cette dernière, i.e. en modélisant **le contexte englobant cette activité**, activité particulièrement caractérisée par les gestuelles utilisateur. La modélisation du contexte, au sein duquel l'activité prend place et l'utilisateur

évolue, permet la **détermination** et la **désambigüisation** de la véritable signification des gestuelles utilisateur et des événements autres pouvant survenir au sein de la scène.
Le système répond ensuite **en adéquation à l'activité interprétée**, observée au sein de la scène virtuelle. Cette réponse est à la fois **interactive**, vis-à-vis de l'utilisateur, et **adaptative**, impliquant la mise en place de mécanismes adaptatifs à tous les niveaux du fonctionnement interne du système.

L'application s'exécutant de manière adaptative sur le système interactif est **scénarisée**. Les objectifs que l'utilisateur doit atteindre au cours de l'interactivité sont définis par ce scénario. Nous avons développé une application de type **jeu vidéo**, dans un contexte d'*exertainment* (*exercise + entertainment*). L'utilisateur est immergé au sein d'un entrainement de tennis, par le biais d'un environnement 3D, mis en place à partir du scénario de l'application et de l'interactivité avec l'utilisateur.

Cette thèse a été motivée par une collaboration industrielle, supportée par une convention ***CIFRE***, entre la société parisienne XD Productions et l'équipe ***Images et Interactivité Numérique*** (***ImagIN***) du ***Laboratoire Informatique, Image et Interaction*** (***L3I***) de l'Université de La Rochelle.

Cadre industriel de la thèse

La société XD Productions, fondée en 1998, réalise et produit des émissions de télévision et longs métrages d'animation grand public.
Pour animer un caractère en 3D, dans le cadre d'une émission ou d'un long métrage, la société utilise le principe de **la capture de mouvements** : un acteur évolue dans un environnement de capture spécifique et limité, ses mouvements étant alors observés et codifiés par l'intermédiaire de capteurs. Ces mouvements caractérisés peuvent ainsi être réattribués à un personnage de synthèse, réalisé par des graphistes et designers 3D. Certains équipements propres à la réalité virtuelle, comme par exemple des gants numériques, peuvent alors être utilisés, permettant ainsi d'interagir avec ce personnage 3D, ses expressions faciales, le mouvement de ses yeux, de sa bouche, de ses mains etc.
Les capteurs inhérents à la capture de mouvements peuvent être de différentes natures (électrique, magnétique, optique, etc.). Après avoir longuement utilisé des capteurs magnétiques, la société XD Productions a développé un environnement complet de capture de mouvements, le ***Cyberdôme***, au sein duquel les mouvements d'une personne sont observés et codifiés, grâce à de simples capteurs optiques. Ces capteurs, à savoir 4 morceaux de tissus de couleur, sont facilement manipulables et peu encombrants, ne gênant que modérément la personne capturée, qui peut donc évoluer plus librement. Le processus d'attribution des mouvements à un caractère 3D s'effectue en temps réel, la personne filmée pouvant ainsi voir ses propres mouvements reproduits par un avatar. La contrainte temps réel, respectée par le principe de capture mis en place au sein du ***Cyberdôme***, est une importante valeur ajoutée au produit final délivré par la société, car un réalisateur peut constater, en temps réel, les prémices de l'animation du personnage 3D. Par la suite, une fois les courbes d'animation retravaillées et affinées en *post processing*, les décors, textures, lumières… sont rajoutés en *post production*. Parce que le processus de capture, nécessitant peu de matériel, est simple et peu contraignant pour la personne filmée, la société peut réaliser des films d'animation à faible coût.

Depuis 2008, la société XD Productions élargit son champ d'activités et ses domaines d'application. Une version miniature du système, l'***Animakit***, a été développée. La capture de mouvements s'effectue par le biais du traitement des flux vidéo délivrés par 3 *webcams*, reliées à un ordinateur portable. L'utilisateur doit toujours porter 4 marqueurs colorés autour des bras et des jambes. Deuxièmement, des techniques de reconstruction et de *texturing* vidéo temps réel ont été implémentées au sein du processus de capture, permettant le *cloning* réaliste de la personne filmée. Enfin, plusieurs applications, type jeu vidéo, ont été développées, dont les scénarios impliquent une interaction temps réel entre l'utilisateur et la scène virtuelle dans laquelle il est immergé.

Cadre académique de la thèse

Les travaux de recherche de l'équipe ***ImagIN***, du Laboratoire L3i de l'Université de La Rochelle, reposent sur la définition de modèles, architectures et outils permettant de mettre en œuvre un environnement **paramétrable**, **personnalisable** et **reconfigurable** pour **la conception et l'exécution adaptative d'applications scénarisées interactives**.
L'objectif est de proposer des méthodes permettant de faciliter le développement d'applications interactives adaptatives en s'intéressant, d'une part, aux aspects de spécification amont liée à l'interaction concepteur/système auteur et, d'autre part, aux problématiques d'adaptation au niveau de l'utilisateur, ainsi qu'aux contraintes de mise en œuvre temps-réel de la boucle d'acquisition du comportement – modélisation et interprétation de ce comportement – exécution adaptée de l'application.
Au niveau du concepteur, il s'agit de proposer des approches de génie logiciel permettant à celui-ci d'exprimer, au travers de gabarits génériques, ses compétences métier et de modéliser le déroulement de l'application en fonction de ses objectifs propres. Il lui faut cependant prendre en compte, au plus tôt dans la modélisation, les comportements de l'utilisateur : comportements attendus (qui sont conformes à ce que souhaite le concepteur) ou imprévus (qui doivent néanmoins être pris en compte pour que l'application puisse continuer à s'exécuter de manière cohérente, et sans erreur).
Au niveau de la mise en œuvre de l'interactivité avec l'utilisateur, il s'agit de rendre opérationnels les modèles élaborés au niveau du concepteur, tout en considérant les contraintes liées à l'adaptation de la dynamique d'interaction par la modélisation d'un contexte instantané, lié à l'activité interactive en cours.
Dans le cadre de ses recherches, l'équipe ***ImagIN*** s'est dotée d'un ***Cyberdôme***, dans le but d'étudier différentes gestuelles complexes par le biais de la capture de mouvements. Des applications ont également été développées, le système ainsi d'interagir avec celles-ci par le biais des gestuelles de l'utilisateur.

Problématiques rencontrées

Notre démarche a été d'enrichir progressivement l'architecture du système initialement conçu par la société XD Productions, en prenant en compte les différentes caractéristiques du système final que nous devons concevoir. Ainsi, à partir d'une architecture globale, nous devons préciser la notion de contexte, défini les algorithmes de capture contextualisée et intégrer les mécanismes supportant l'adaptativité de notre système.

Problématique 1 : Définition d'un système interactif sur lequel s'exécutent de manière adaptative des applications scénarisées
Définition de l'architecture de notre système

A l'issue de ces travaux de thèse, nous souhaitons proposer **une architecture globale** de **systèmes interactifs** sur lesquels s'exécutent, de manière **adaptative**, des applications **scénarisées**. Cette architecture met en œuvre principalement deux processus spécifiques : un processus lié à **l'interactivité** se mettant en place entre l'utilisateur et le système et un processus lié à **l'adaptation automatique et dynamique** (adaptativité) du système vis-à-vis de l'utilisateur.

Tout d'abord, les caractéristiques optimales de cette architecture peuvent donc être déterminées à partir de **l'étude des notions d''interactivité'et d''adaptativité'** (problématiques de l'équipe ***ImagIN***). Ces notions sont vastes et englobent de nombreux aspects qu'il nous semble important d'étudier, avant de concevoir notre système. La compréhension de ces deux notions en amont de sa conception nous permettra d'implémenter au mieux **les processus interactif et adaptatif**.
Par la suite, en accord avec notre sujet de thèse, **l'architecture spécifique de notre système** doit être définie précisément. L'architecture finale de notre système constitue une spécialisation de l'architecture générale d'un système élaboré au sein de l'équipe ***ImagIN***.

Problématique 2 : Introduction de la notion de contexte au sein de notre système
Gestion et exploitation du contexte au sein de notre système

L'objectif principal de notre sujet de thèse est donc **d'interpréter l'activité au sein de la scène pour que le système puisse y répondre de manière adaptée**, aussi bien du point de vue de sa réponse à l'utilisateur que de son fonctionnement interne.
Nous émettons l'hypothèse que **le contexte d'interaction**, englobant l'activité, est caractéristique de celle-ci. Ce contexte d'interaction doit donc être **caractérisé**, par un processus dédié du système, pour ensuite être **interprété**. Cette interprétation pilote alors la mise en place de la réponse du système, vis-à-vis de l'activité observée.

Il nous est donc nécessaire, tout d'abord, de définir cette notion d'**'activité'**. Celle-ci, en accord avec nos objectifs, n'est pas uniquement liée **aux gestuelles utilisateur** mais peut être relative à **un dynamisme au sein de la scène**, inhérent à un changement d'illumination, ou encore à la présence de spectateurs.
De plus, nous devons établir **une caractérisation du contexte d'interaction**, traduisant précisément l'activité, et prenant en compte les différentes significations qu'elle peut avoir.

Enfin, de manière à compléter notre *framework* de traitement du contexte d'interaction, nous devons implémenter **un processus d'interprétation** au sein de notre système. Ce processus permet ainsi au système d'identifier efficacement l'activité se déroulant au sein de la scène, avant de répondre en adéquation à celle-ci.
Implicitement, l'implémentation de ces différents processus, exploitant le contexte d'interaction englobant l'activité, implique celle de processus **gérant les différentes informations contextuelles au sein de notre système**. Ces gestionnaires doivent prendre en compte de nombreuses informations hétérogènes, de plus ou moins haut niveau, et sont fonction de la couche sémantique de l'architecture du système à laquelle ils appartiennent.

Problématique 3 : Extension du système initial de capture de mouvements
Elaboration d'un système de capture de l'activité au sein de la scène réelle

Dans le **système commercial de capture de mouvements** (contexte industriel de cette thèse), dont nous disposions au début de la thèse, la scène réelle est interfacée avec le système global. Ce système présente **deux contraintes matérielles** : total contrôle de l'environnement de capture et nécessité de porter des marqueurs de la part de l'utilisateur.
De plus, il ne permet que la capture, dans le sens suivi et codification, **des mouvements** de l'utilisateur, sans prendre en compte la sémantique véhiculée par ces derniers. Il avait été conçu comme un système de **'capture'** de **'mouvements'** du **'corps'**. Ces 3 termes, qui peuvent être ambigus, doivent définis, de manière à nous positionner vis-à-vis du processus particulier de capture de l'activité, que nous voulons développer. Il existe actuellement plusieurs types de 'capture' : détection, suivi, reproduction d'apparence, etc., pour plusieurs types de 'mouvements' (mouvement, geste, action, comportement, gestuelle, etc.), concernant différentes parties du 'corps' (le visage, un membre particulier, le corps dans sa globalité…).

Le système commercial initial doit donc être étendu pour répondre à nos objectifs.

Nous désirons tout d'abord **alléger au maximum** les deux contraintes matérielles que présente le système. Nous cherchons en effet à implémenter un processus de capture **non invasif** (l'utilisateur ne porte aucun marqueur), **en milieu non contrôlé** (dynamisme au sein de la scène réelle que le système doit pouvoir gérer).
De plus, nous voulons étendre le processus à **une capture plus globale de l'activité** (gestuelles de l'utilisateur et dynamisme de la scène réelle, dont il n'est pas directement responsable). Le système doit en effet pouvoir s'adapter non seulement à l'activité relative à l'utilisateur et à ses gestuelles, mais également à d'autres événements (changement de lumière, présence de spectateurs) dont l'utilisateur n'est pas la source.

Problématique 4 : Etude et implémentation du processus adaptatif au sein de notre système

L'interprétation du contexte d'interaction, traduisant l'activité au sein de la scène observée, pilote **l'évolution du scénario** de l'application. Elle est suivie par **la mise en place du processus adaptatif du système**. Ainsi, le système adapte ses réponses, vis-à-vis de l'utilisateur ou relatives à son propre fonctionnement, en adéquation avec l'activité prenant place au sein de la scène.

En accord avec nos objectifs de thèse et le scénario de l'application que nous avons développée, nous devons **étudier et implémenter le processus adaptatif** au sein de notre système.
Parallèlement à sa réponse interactive et visuelle vis-à-vis de la gestuelle utilisateur, le système doit mettre en place, après l'interprétation de l'activité au sein de la scène, un ensemble de **mécanismes adaptatifs**. Ces mécanismes doivent être mis en œuvre en accord avec les différentes couches de l'architecture de notre système.

Contributions – Résultats visés

Pour répondre à ces problématiques, nous orientons nos recherches en vue d'obtenir des résultats complémentaires et cohérents, suivant les différentes visions à prendre en compte :

- **Définition de l'architecture de notre système**

Dans le cadre de ces travaux, nous souhaitons définir **l'architecture finale de notre système**. Cette architecture doit traduire les différentes étapes du processus interactif qui prend place entre le système et l'utilisateur. Elle sera divisée en plusieurs couches, chaque couche correspondant à un niveau sémantique, i.e. à un type de données et de connaissances gérées par le système. Chaque couche de l'architecture comprendra un ensemble de modules de traitement, chacun étant dédié à une étape bien particulière du processus interactif. Deux aspects doivent particulièrement être pris en compte :

- **Modélisation et gestion de contexte**

L'introduction de la notion de contexte, au sein de notre système et du processus interactif, traduira notre contribution principale. Nous émettons en effet l'hypothèse que le **contexte d'interaction**, une fois caractérisé par notre système, traduit **l'activité**, notamment celle liée à l'utilisateur. La gestion et l'analyse du contexte d'interaction au sein duquel l'activité prend place nous permettra ainsi d'atteindre nos objectifs : **interprétation de l'activité** et **réponse adéquate du système** à celle-ci.

Ainsi, une modélisation complète du contexte d'interaction sera proposée. Ce modèle de contexte nous servira de base pour la modélisation précise du scénario d'une application.
Nous allons également implémenter au sein de notre système les modules dédiés à la gestion des informations contextuelles au cours de l'interactivité. En effet, le contexte d'interaction, englobant l'activité au sein de la scène, est observé et caractérisé. Grâce au support du scénario (extraction du contexte d'interaction attendu), le système interprète cette activité, à laquelle il répond de manière adaptée. Cette réponse adaptative, paramétrée par différentes informations contextuelles, se traduit par la mise en place de mécanismes adaptatifs à tous les niveaux de l'architecture. Elle se traduit par l'adaptation à la fois de la réponse visuelle à l'utilisateur et du fonctionnement propre du système.

- **Etude des processus interactif & adaptatif**

Nous proposerons deux études complètes concernant **les processus interactif et adaptatif**, mis en œuvre par notre système. Ces études doivent nous permettre de nous positionner, par le biais d'états de l'art importants, vis-à-vis de nombreux aspects et définitions englobés dans les notions d''interactivité' et d''adaptativité'. Nous cherchons à établir un cahier des charges précis quant à la conception des systèmes développés par l'équipe ***ImagIN***. En suivant ce cahier des charges, nous pourrons définir l'architecture générale des systèmes de l'équipe ***ImagIN***.

- **Extension d'un système commercial de capture de mouvements**

Une grande partie de ce travail s'articulera autour du **système commercial de capture de mouvements**, le ***Cyberdôme***, adopté relativement au contexte industriel de cette thèse. Nous cherchons à augmenter le processus initial de capture de mouvements du corps de l'utilisateur, de manière à capturer non seulement **les gestuelles de l'utilisateur**, i.e. la logique de ses mouvements, gestes, actions et comportements, mais également **l'activité globale au sein de la scène**, relative au dynamisme de la scène réelle. Cette extension du système adopté initialement passera tout d'abord par l'étude suivante :

- **Etude des notions de 'capture', 'mouvement' et 'corps'**

Face aux ambigüités des termes **'capture'**, **'mouvement'** et **'corps'**, nous souhaitons proposer une définition précise de ceux-ci. En nous positionnant vis-à-vis de ces derniers, nous pourrons définir précisément le processus que nous voulons implémenter et identifier les différentes problématiques auxquelles nous allons faire face, pour atteindre nos objectifs (capture de l'activité, de manière non invasive, en milieu non contrôlé).

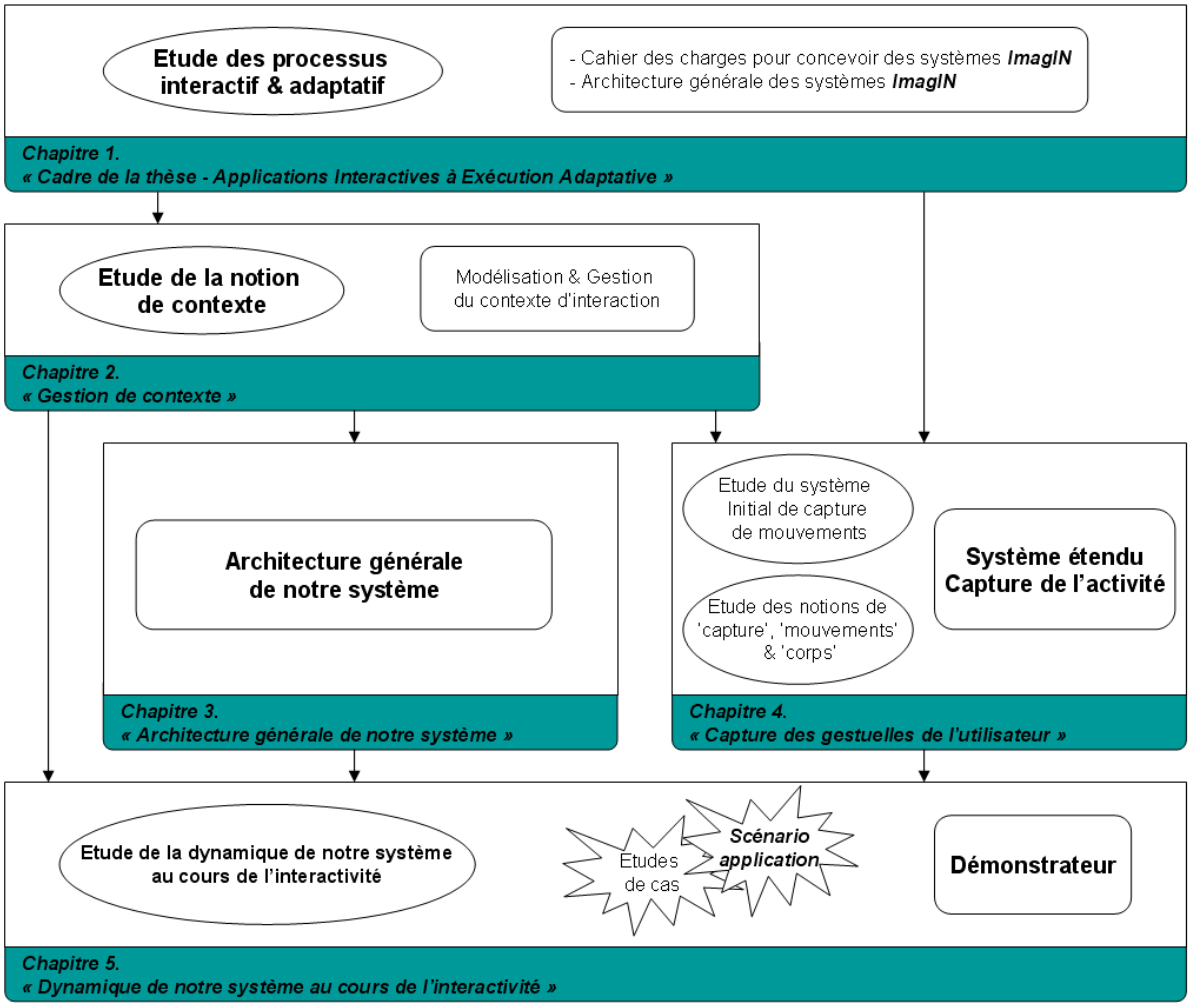
- **Développement d'un démonstrateur**

Dans un contexte d'application particulier (immersion de l'utilisateur au sein d'un entrainement de tennis), les processus **interactif** et **adaptatif** seront implémentés, mis en œuvre en **temps réel**, vis-à-vis de l'activité observée et interprétée au sein de la scène.

Publications

- F. Picard, P. Estraillier (2010) Context-dependent player's movement interpretation: application to adaptive game development, In Proc. of 3D Image Processing & Applications 2010.
- F. Picard, P. Estraillier (2009) Enhancing a motion capture interface by introducing context management, In Proceedings of the International Conference on Advances in Computer Enterntainment Technology (ACE'09).
- F. Picard, P. Estraillier (2008) Motion capture system contextualization, In Proceedings of the 2008 International Conference in Advances on Computer Entertainment Technology (ACE'08).
- F. Picard, P. Estraillier (2008) Motion Capture System Contextualization - Application to game development, In Proc. of Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games 2008.
- F. Picard, P. Estraillier (2008) Extraction contextualisée de silhouettes. In Actes de MajecSTIC 2008.

Organisation du manuscrit



Chapitre 1. Cadre de la Thèse Applications Interactives à Exécution Adaptative

Les expressions **« applications interactives »** et **« exécution adaptative »** recouvrent plusieurs notions qu'il nous semble important d'étudier avant d'élaborer notre système. Nous présentons ainsi tout d'abord le processus interactif scénarisé, du point de vue du domaine de l'***IHM***. Puis nous définissons l'adaptativité d'un système et les différentes dimensions que cette notion englobe. Ces deux études sont illustrées par des états de l'art importants.
Dans la suite du chapitre, nous détaillons le contexte scientifique englobant nos travaux de thèse. Après avoir positionné les approches de l'équipe ***ImagIN*** en matière d'interactivité et d'adaptativité, nous élaborons **le cahier des charges** destiné aux différents concepteurs de l'équipe et établissons **l'architecture générale des systèmes développés** au sein de celle-ci. Nous concluons en positionnant nos travaux de thèse au sein de l'équipe ***ImagIN***.

Chapitre 2. Gestion de contexte

Dans ces travaux de thèse, nous faisons l'hypothèse que **le contexte d'interaction retranscrit l'activité prenant place au sein de la scène observée**. Sa **caractérisation** et son **interprétation** permettront au système de répondre en adéquation avec l'activité observée, que cela soit au niveau de l'utilisateur que vis-à-vis de son propre fonctionnement.
Nous verrons que l'interprétation d'un contexte d'interaction observé peut se faire grâce au support du **scénario de l'application**, à partir duquel sont extraits **les contextes d'interaction attendus** à différents moments clés. De plus, les informations contextuelles permettent **le paramétrage des mécanismes adaptifs**, mis en place par le système au cours de sa réponse, orientant et cadrant les différents traitements.
Ce chapitre propose donc **l'étude de la notion de contexte**. Nous définissons précisément cette notion, aussi bien du point de vue de l'humain que du système interactif. Par la suite, nous passons en revue les différentes approches pour modéliser un contexte au sein d'un système. Nous constatons ainsi qu'il existe plusieurs sous contextes que nous identifions. Nous présentons également plusieurs méthodologies pour gérer un modèle de contexte au sein d'un système. Le chapitre se termine par un positionnement vis-à-vis des différentes études exposées, et par la présentation de nos contributions. Nous proposons un modèle de contexte complet, nous permettant la modélisation précise du scénario d'une application.

Chapitre 3. Architecture générale de notre système

Les deux chapitres précédents nous ont permis d'étudier l'ensemble des caractéristiques que doit intégrer **l'architecture finale de notre système**.
Notre architecture est divisée en **4 couches sémantiques spécifiques** : support opérationnel, gestion de la scène virtuelle, analyse du dialogue interactif et logique concepteur. Nous détaillons dans ce chapitre les différents modules de traitement composant chaque couche. Ces modules constituent une étape bien particulière de l'interaction se déroulant entre l'utilisateur et le système. Nous détaillons nos contributions vis-à-vis de notre architecture.

Chapitre 4. Capture de gestuelles de l'utilisateur

Le point de départ du processus interactif est **la capture de l'activité au sein de la scène réelle**. Cette capture permet la modélisation des indices caractéristiques de l'activité au sein de la scène virtuelle.
Le processus de capture de l'activité est **la généralisation** et **l'extension du processus de capture des mouvements de l'utilisateur**, mis en œuvre par le ***Cyberdôme***. Le ***Cyberdôme*** est le point de départ autour duquel s'articulent ces travaux de thèse et constitue le système initial adopté en accord avec le contexte industriel de cette thèse.
Notre objectif fut donc de passer d'un processus de capture des mouvements du corps de l'utilisateur à celui de **l'activité de manière plus globale au sein de la scène réelle**, activité englobant particulièrement celle relative aux gestuelles de l'utilisateur.
L'expression « capture des mouvements du corps » présente plusieurs significations et ambigüités. Nous définissons tout d'abord les différentes facettes de cette expression. Ces définitions nous permettent de nous positionner et de préciser les aspects du processus de capture, que nous implémentons dans le cadre de ces travaux.
Nous présentons ensuite un état de l'art non exhaustif des différents systèmes de capture actuels, industriels et académiques, permettant ainsi de souligner les différents aspects d'un processus de capture, ainsi que les problématiques que ce processus englobe.

Nous présentons par la suite le ***Cyberdôme***, développé par la société XD Productions. Nous exposons **nos solutions pour alléger les deux contraintes qu'il présente** (contrôle du fond de la scène et nécessité de marqueurs colorés).

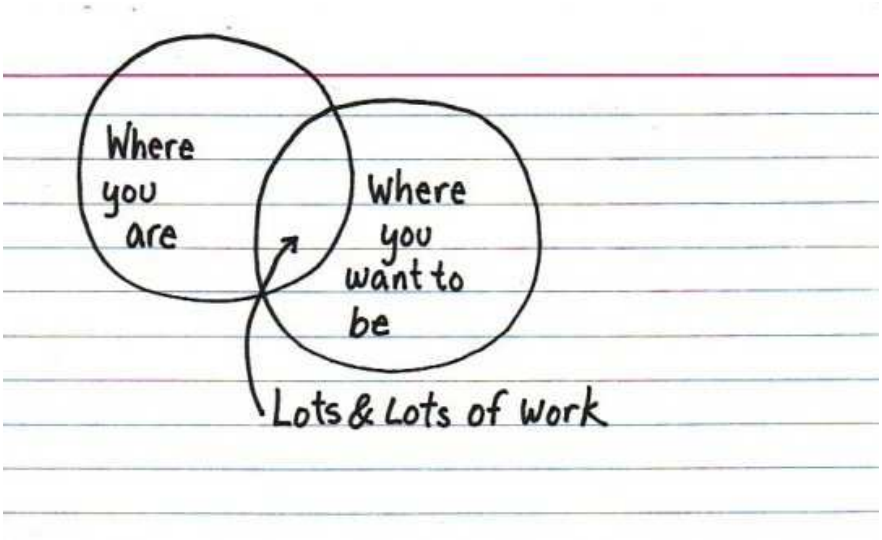
Chapitre 5. Dynamique de notre système au cours de l'interactivité

Le Chapitre 5. présente **la dynamique du processus interactif**, après la retranscription de l'activité au sein de la scène virtuelle.

Les différentes connaissances permettant la caractérisation du contexte d'interaction au sein duquel l'activité prend place sont tout d'abord identifiées. Ces différentes connaissances sont **extraites de la scène virtuelle** par le biais **de mécanismes d'observation** particuliers.
Il est alors possible **de caractériser ce contexte d'interaction** et les sous contextes qu'il regroupe. Nous présentons, dans le cadre d'un contexte applicatif bien spécifique, notre modélisation.
Un module dédié se charge par la suite **d'interpréter ce contexte d'interaction**, en évaluant **la distance** pouvant exister entre **le contexte d'interaction observé** et celui **attendu** par le scénario.
Cette distance permet l'évolution du scénario et donc **la mise en place des réponses interactives et adaptatives du système**. Nous nous arrêtons particulièrement sur l'étude du processus adaptatif au sein de notre système.

Pour illustrer nos travaux de thèse, nous avons développé une application, immergeant l'utilisateur au sein d'**un entrainement de tennis**. Nous illustrons nos travaux par le biais de 3 études de cas bien particulières.

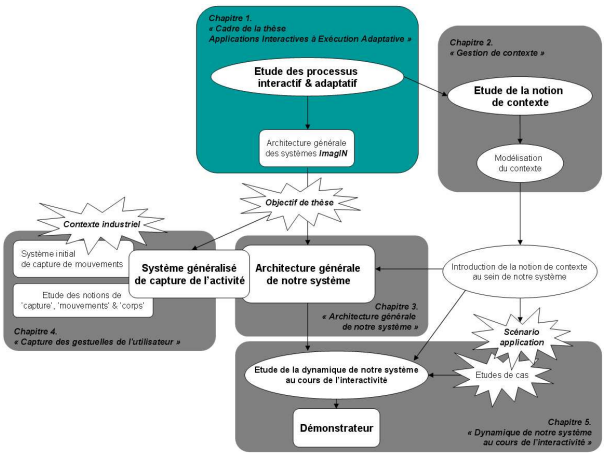
Chapitre 1. Cadre de la Thèse
Applications Interactives à Exécution Adaptative



Start with the things that you know and the things that are unknown will be revealed to you
- Rembrandt -

Et sinon, vous, ça va ?
- RR -

Résumé



Les expressions **« applications interactives »** et **« exécution adaptative »** en avant plusieurs notions, qu'il nous semble important d'étudier avant d'élaborer notre système. Nous présentons ainsi tout d'abord le processus interactif scénarisé, du point de vue du domaine de l'***IHM***. Puis nous définissons l'adaptativité d'un système et les différentes dimensions que cette notion englobe. Ces deux études sont illustrées par des états de l'art importants.

Dans la suite du chapitre, nous détaillons le contexte scientifique englobant nos travaux de thèse. Après avoir positionné les approches de l'équipe ***ImagIN*** en matière d'interactivité et d'adaptativité, nous élaborons **le cahier des charges** qui cadre les différents concepteurs de l'équipe et établissons **l'architecture générale des systèmes développés** au sein de celle-ci. Nous concluons en positionnant nos travaux de thèse au sein de l'équipe ***ImagIN***.

Plan du chapitre

Ces travaux de thèse s'inscrivent dans un contexte de recherches scientifiques effectuées par l'équipe ***« Images et Interactivité Numérique »*** (***ImagIN***) du Laboratoire ***Informatique, Image, Interaction*** (***L3i***), de l'Université de la Rochelle. Cette équipe étudie les modèles, architectures et outils permettant de mettre en œuvre un environnement paramétrable, personnalisable et reconfigurable pour l'**exécution adaptative d'applications scénarisées interactives**.

L'objectif final de ce chapitre est de proposer l'architecture globale d'un **système interactif** complet, sur lequel s'exécuteront **de manière adaptée des applications scénarisées**, dans le contexte de travail de l'équipe ***ImagIN***. L'architecture du système que nous élaborons dans ces travaux de thèse sera une spécialisation de celle proposée dans ce chapitre, en accord avec les objectifs que nous nous sommes donnés.

Pour définir l'architecture d'un système développé par l'équipe ***ImagIN***, nous établissons, au cours de ce chapitre, un **cahier des charges** intuitif, sous la forme d'un ensemble de **règles générales** (**RG**) de conception. Ce cahier des charges aura pour objectif de guider et cadrer les concepteurs du système et de l'application. Il leur permettra d'élaborer un système efficacement et de manière optimale, à tous les niveaux et en prenant en compte tous les aspects de l'interactivité avec l'utilisateur, au cours de laquelle le système répondra de manière adaptée. Ce cahier des charges doit donc décrire les différentes caractéristiques d'un tel système.

Ce cahier des charges est établi à partir de l'étude des notions d'**« application scénarisée interactive »** et d' **« exécution adaptative d'applications »**. Les différents aspects et concepts scientifiques qui s'articulent autour de ces notions sont relatifs à l'étude du processus d'interaction scénarisé entre l'utilisateur et le système et de la faculté du système à s'adapter automatiquement à l'activité observée et interprétée, particulièrement celle relative à l'utilisateur. Nous allons présenter ces deux notions et les concepts qu'elles englobent par le biais de deux états de l'art détaillés.

L'expression **« application scénarisée interactive »**, ainsi que les différents concepts qu'elle englobe, sont étudiés au cours de la Section 1. Ces notions sont relatives à l'étude du processus d'interaction prenant place autour de l'utilisateur, **processus piloté par le scénario de l'application** et **orchestré par le système interactif**. Nous positionnons par la suite les travaux de l'équipe ***ImagIN***, vis-à-vis des différents concepts étudiés.
Pour concevoir un système sur lequel s'exécutent des applications scénarisées, il est tout d'abord indispensable de bien étudier ce qu'est l'**interactivité** supportée par les **connaissances scénaristiques**. Les travaux effectués dans le cadre du domaine de **l'*Interaction Homme-Machine*** cherchent à étudier toutes les facettes du processus d'interaction scénarisée, permettant l'élaboration finale d'un système les englobant et les illustrant. Nous présentons ce domaine d'étude.
L'interactivité peut se concevoir de différentes façons, selon plusieurs visions. Ces différentes philosophies sont appelées des **paradigmes d'interaction**. Les caractéristiques globales d'un système, et par la suite du scénario d'une application, dépendront de ce paradigme. Nous établissons un état de l'art autour de la notion de paradigme d'interaction.
Le processus d'interaction scénarisée entre le système et l'utilisateur, effectuant une tâche, est modélisé, en accord avec le paradigme d'interaction élu. Les concepteurs peuvent extraire de cette modélisation, appelée un **modèle d'interaction**, un cahier des charges précis et détaillé,

pilotant l'élaboration du système. Après avoir proposé une définition d'un modèle d'interaction, nous proposons un état de l'art autour de ce concept.
Enfin, la conception des **interfaces** entre l'utilisateur et le système, les **interfaces homme-machine**, ou **interfaces utilisateur-système**, est une étape importante lors de l'élaboration du système interactif. Elles constituent en effet le cadre des communications entre l'utilisateur et le système, communications pilotées par le scénario de l'application.

L'expression **« exécution adaptative applications »** introduit la notion d'adaptativité au sein du processus interactif scénarisé. Dans le cadre de notre démarche de conception, nous souhaitons développer un système s'adaptant à l'activité observée, particulièrement **aux gestuelles de l'utilisateur**, et nous pensons donc que la mise en place du processus adaptatif au sein du système nous permettra d'atteindre cet objectif. Il nous semble donc important d'étudier la notion d'**adaptativité** (Section 2.). Nous positionnons les travaux de l'équipe ***ImagIN***, relativement à cette étude.
L'**adaptativité** est une facette particulière de l'adaptation. Au cours de cette section, Nous définissons le processus d'adaptation au niveau de l'être humain. Puis nous survolons les différentes classes de systèmes au sein desquels la notion d'adaptation a été introduite. Parmi ces classes se trouve celle des **systèmes adaptatifs** que nous étudions dans le cadre de ces travaux de thèse.
L'adaptativité d'un système ayant été préalablement définie, nous énumérons et caractérisons par la suite les différentes **dimensions** de l'adaptativité, qui nous semble pertinentes dans le cadre des travaux de l'équipe ***ImagIN***. Ainsi, nous cherchons à savoir **pourquoi** le système s'adapte, **quand** et **où** et enfin, **à quoi** le système doit s'adapter et **que** doit-il adapter.

Au cours de la Section 3., nous présentons les différents aspects des travaux de l'équipe ***ImagIN***, relatifs à l'étude des notions d' **« applications scénarisées interactives »** et d' **« exécution adaptative d'applications »**. Cette étude nous permet d'établir le cahier des charges qui oriente et cadre les différents concepteurs de l'équipe, lors de l'élaboration d'un système interactif sur lequel s'exécutent de manière adaptée une application scénarisée. Nous proposons enfin l'architecture générale d'un tel système, dans le cadre scientifique de l'équipe ***ImagIN***.

En guise de conclusion de ce chapitre, nous positionnons nos travaux de thèse au sein du contexte scientifique défini par les travaux de l'équipe ***ImagIN***.
Nous élaborons dans ces travaux un système **mono-utilisateur** sur lequel s'exécute une application de type **jeu vidéo**, dans un contexte d'*exertainment* (*exercise* + *entertainment*). L'utilisateur interagit, **en temps réel** et **sans contraintes matérielles**, par le biais de ses **gestuelles**. Ces gestuelles peuvent être **explicites**, i.e. adoptées de manière consciente et volontaire pour interagir avec le système, ou **implicites**, l'utilisateur n'interagissant pas consciemment avec le système. **L'activité** est également marquée par divers événements liés **au dynamisme de la scène réelle**, dont l'utilisateur n'est pas le responsable directe.
En accord avec le scénario de l'application, l'utilisateur est immergé au sein d'une **scène virtuelle interactive**. Ses interactions au sein de la scène permettent au processus dédié du système de caractériser **le contexte d'interaction** dans lequel il évolue. Un processus d'interprétation de l'activité est alors mis en œuvre, grâce au support du scénario de l'application, à partir duquel sont extraits **les contextes d'interaction attendus** à un instant donné. Le système répond alors, en fonction de cette interprétation. La réponse du système est double : à la fois **interactive**, par le biais de la mise à jour de la scène virtuelle en accord avec l'activité, et **adaptative**, adaptant ainsi l'ensemble de son fonctionnement en accord avec l'activité interprétée.

La terminologie employée dans ce chapitre, concernant l'**architecture** du système, la **logique concepteur** et le **scénario**, est précisée au cours de l'Annexe A.

1. Application scénarisée interactive

L'expression « application scénarisée interactive » motive un **état de l'art** important autour de la notion d'**interactivité**, **guidée par les connaissances scénaristiques apportées par l'application** et **mise en œuvre par un système interactif**.

L'étude de l'interaction entre l'utilisateur et un système est un domaine de recherche à part entière. C'est le domaine de **l'*Interaction Homme-Machine* (*IHM*)**, au départ cantonné à l'**étude de l'interface entre l'homme et le machine**, et qui est devenu au fil des ans l'**étude des différents aspects de l'interaction entre l'homme et la machine**. Ce domaine de l'informatique est présenté au cours de la Section 1.1., que le lecteur intéressé pourra compléter avec le bref **historique**, proposée en Annexe B.
Le domaine de l'***IHM*** étudie les différentes facettes du processus d'interaction. Mais ce processus n'est-il pas toujours scénarisé ? Autrement dit, une application comprend-elle à chaque fois un scénario ? C'est en exposant **les différentes connaissances qu'apporte le scénario de l'application au sein d'un système interactif** et en mettant en évidence **l'influence du scénario à tous les niveaux du processus**, que nous essayons de répondre à ces questions. Nous définissons ainsi la notion de processus d'interaction scénarisé, que la communauté de l'***IHM*** étudie implicitement.
L'apport du scénario au sein du processus d'interaction est indéniable. Il n'en reste pas moins que **c'est le système qui supporte et orchestre ce processus**. De manière à concevoir un système au sein duquel le processus d'interaction scénarisé a été profondément étudié, **il nous semble nécessaire d'étudier certains aspects de ce dernier** : **paradigme d'interaction**, **modèle d'interaction** et **interfaces**. Ce sont ces différentes facettes qui sont étudiées au cours des Sections 1.2., 1.3. et 1.4.

Le processus d'interaction scénarisé peut être considéré de différentes manières. Cette vision, appelée un **paradigme d'interaction**, est la philosophie, suivie par les différents concepteurs, qui pilotera le développement d'un **système interactif** et d'une **application scénarisée** s'exécutant sur ce dernier.
Il existe de nombreux méta-paradigmes et paradigmes d'interaction. Nous répertorions et présentons, au cours de la Section 1.2., les différents paradigmes d'interaction qui ont orientés la conception de l'ensemble des systèmes interactifs depuis l'avènement de l'***IHM***.

Les concepteurs d'un système interactif ne peuvent pas ne suivre qu'une vision de l'interactivité pour développer un système interactif. Ils doivent pouvoir se référer à un **cahier des charges**, un cadre d'études et de développements, aussi précis et détaillé que possible, les guidant au cours des différentes étapes de la conception.
Ce cahier des charges peut être établi à partir d'un **modèle d'interaction**. Un modèle d'interaction décrit le **processus d'interaction scénarisée** entre le **système** et l'**utilisateur** cherchant à réaliser une **tâche**, en accord avec un **paradigme d'interaction** particulier. Il existe plusieurs solutions pour modéliser l'utilisateur, le système, la tâche et l'interaction type qui peut avoir lieu entre l'utilisateur et le système.

C'est pourquoi, la notion de modèle d'interaction est complexe. Plusieurs définitions ont été proposées et plusieurs modèles ont été élaborés en accord avec un paradigme d'interaction donné, pour être par la suite appliqués à la conception logicielle et matérielle d'un système particulier. Pour être sûr que ces trois premières règles rentrent dans le cadre d'un modèle d'interaction, il est nécessaire de proposer une définition de la notion de modèle d'interaction, conjointement avec celles déjà proposées et les modèles déjà établis dans la littérature (Section 1.3.).

Enfin, l'utilisateur et le système ont besoin de communiquer par le biais d'interfaces, les **interfaces homme-machine** (ou interfaces utilisateur-système). Ces interfaces, situées entre l'utilisateur et le système global, sont les **cadres comportemental**, **matériel** et **logiciel** de l'interactivité.
En accord avec le modèle d'interaction, le type d'applications développées pour le système interactif, le budget et les moyens techniques du projet global, la plateforme sur laquelle est développé le système, etc. les concepteurs doivent faire plusieurs choix décisifs quant aux **interfaces logicielles** (comment les rendre plus riches ?) et **matérielles** (quelles interfaces matérielles ?), ainsi qu'à la manière d'interagir de l'utilisateur avec le système (quelles **techniques d'interaction** adoptées et comment les évaluer ?). Ces différentes idées sont discutées au cours de la Section 1.4.

1.1. Etude de l'interaction entre l'homme et la machine

La signification de l'acronyme ***« IHM »*** est multiple. Historiquement, le premier sens de ***« IHM »*** fut ***Interface Homme-Machine***. Le domaine de l'***IHM*** était donc cantonné à **l'étude des interfaces entre l'utilisateur et le système**, c'est-à-dire le cadre comportemental (technique d'interaction), matériel (interface matérielle) et logiciel (interface logicielle) permettant aux deux parties de communiquer l'une avec l'autre. A l'époque, le cadre matériel se résumait aux **périphériques standards** : souris, clavier et écran ; et le cadre logiciel aux **interfaces de type *WIMP*** (*Window, Icons, Menus & Pointers* – « Fenêtre, Icônes, Menus et Pointeurs »), avec des techniques d'interaction qui se réduisant à la désignation et au déplacement d'objets. Il est à noter que ces interfaces sont toujours employées de nos jours. Cependant, très vite, il s'est avéré que pour offrir à l'utilisateur une meilleure interactivité, l'***IHM*** ne devait pas s'arrêter à l'unique étude des interfaces avec le système.

Comme exposé dans [Beaudouin-Lafon 04, Olsen & Klemmer 05], nous pensons donc que l'***IHM*** doit devenir l'étude de l'***Interaction Homme-Machine***, auquel nous lui préférons les expressions **« Interaction Homme-Système »** ou **« Interaction Utilisateur-Système »**, de manière beaucoup plus globale et complète. Il s'agira d'étudier l'élaboration, le développement et l'évaluation des mécanismes généraux mis en œuvre au cours du processus d'interaction, aussi bien au niveau de l'utilisateur que du système. L'***IHM*** doit être centrée sur l'utilisateur, dans le sens où l'interaction globale est étudiée pour que ce phénomène soit aussi **naturel** et **adapté** (à son profil, à ses besoins, etc.) que possible. L'évolution de l'***IHM*** au cours du temps est illustrée par l'historique proposé en Annexe B.

Une des hypothèses majeures de ce travail est de considérer que les applications que nous développons sont **scénarisées** et **interactives**. Il est clair que le scénario est propre à une application et qu'il apporte **un ensemble de connaissances qui pilotent et guident le processus d'interaction**, ce dont nous parlerons au cours de la Section 1.1.1.

Cependant, l'interactivité en elle-même est **mise en œuvre effectivement et matériellement par le système**, qui sera de toute façon conçu avant l'application. L'objectif de ce chapitre étant de proposer une architecture de système interactif sur lequel s'exécute des applications scénarisées, nous exposerons les différentes facettes du phénomène interactif qu'il nous juge pertinent d'étudier en amont de la conception du système (Section 1.1.2.).

1.1.1. Un processus d'interaction piloté par le scénario de l'application

Le système interactif que nous concevons dans le cadre de ces travaux est le siège de l'exécution d' **« applications scénarisées »**. Mais cette expression n'est-elle pas un pléonasme ? **Une application ne serait-elle pas tout le temps scénarisée ?** Le domaine de l'***IHM*** n'étudierait-il pas implicitement un processus d'interaction qui serait piloté par le scénario d'une application ?

Pour pouvoir répondre à ces questions, il est tout d'abord nécessaire d'identifier les **connaissances** qu'apporte le scénario d'une application. Le scénario d'une application décrit une **histoire** qui est proposée à l'utilisateur par le biais du système. Il comporte un certain nombre de **tâches** que l'utilisateur doit accomplir pour atteindre la fin de ce récit.

Un scénario peut s'exprimer de différentes manières.
Il peut véritablement se concrétiser sous la forme d'un récit complexe et non linéaire, comme dans un jeu vidéo par exemple. L'utilisateur doit effectuer plusieurs tâches, imposées par le scénario, dont une tâche finale pour que l'application prenne fin.
Mais un scénario peut être aussi un ensemble de scénarios, chacun comportant une ou plusieurs tâches que l'utilisateur choisit d'effectuer dans le but d'atteindre ses propres objectifs. Par exemple, une application permettant d'effectuer divers traitements sur une image numérique permettra à l'utilisateur de transformer une image qu'il a choisie, par le biais de tâches qu'il effectue volontairement, tâche rendue possible par les différents scénarios de l'application. L'utilisateur, dans ce cas, suit plusieurs scénarios définis par l'application. S'il appuie sur tel bouton ou s'il sélectionne telle zone de l'image, l'application réagit de telle manière, etc.

Le scénario d'une application apporte donc au système les connaissances relatives à l'**environnement interactif** au sein duquel l'utilisateur évolue ; aux **événements que le système doit reconnaître pour comprendre la gestuelle de l'utilisateur** ; et enfin **aux réponses que le système doit établir en fonction des gestuelles de l'utilisateur**. A noter que l'expression « environnement interactif » est utilisé dans un sens large, pouvant s'agir aussi bien d'une interface graphique en 2D que d'un monde interactif en 3D. Il est donc certain que le scénario influera sur le processus d'interaction puisqu'il **pilote véritablement** ce phénomène. En fonction du scénario, les objectifs de l'utilisateur, ses motivations évolueront, ce qui modifiera ses capacités de perception, de cognition et d'action. Quant au système, il reconnaitra les gestuelles de l'utilisateur en les confrontant à celles attendues par le scénario. Il répondra alors en modifiant, en accord avec les directives du scénario, l'environnement interactif dans lequel l'utilisateur évolue.

Pour conclure, nous pensons, dans le cadre de ces travaux de thèse, que **toute application peut être considérée scénarisée**. Et donc, en conséquence, que le processus d'interaction

utilisateur-système, tel qu'il est étudié dans le domaine de l'***IHM***, est scénarisé. Nous adopterons par la suite l'expression **« processus d'interaction scénarisée »**.

1.1.2. Mais un processus d'interaction orchestré par le système interactif

Toutefois, malgré la contribution majeure du scénario, **c'est le système interactif qui orchestre le processus d'interaction**. Autrement dit, le scénario pilote le processus d'interaction mais c'est le système interactif qui le **structure** et l'**exécute**. Quoiqu'il en soit, un système interactif n'est pas conçu en fonction d'une application scénarisée, c'est l'application qui est développée en fonction du système. Un système est donc développé pour une classe, plus ou moins grande, de scénarios.
En accord avec les objectifs de l'équipe ***ImagIN***, il nous est donc nécessaire d'étudier les différentes facettes de ce processus en amont de la conception du système de manière à l'implémenter au sein de celui-ci efficacement et de manière optimale.

Tout d'abord, la première tâche du concepteur est d'élire un ou plusieurs **paradigmes d'interaction**. Ces paradigmes décrivent **une façon d'envisager globalement le processus d'interaction**. Ce choix définira non seulement les grandes lignes directrices de conception du système, mais également la nature des applications scénarisées qui s'exécuteront sur ce dernier. Autrement dit, le scénario d'une application devra refléter les traits caractéristiques du ou des paradigmes d'interaction choisis par le concepteur. Les paradigmes d'interaction seront présentés au cours de la Section 1.2.

Une fois le ou les paradigmes d'interaction adoptés, le concepteur doit étudier le phénomène interactif type, tel qu'il se déroule entre un utilisateur désireux d'accomplir une tâche donnée et le système. Cette étude pourra nécessiter la **modélisation de l'utilisateur** (processus perceptuels, cognitifs et moteurs mis en jeu au cours de l'interaction…), **du système** (les différents états de celui-ci, ainsi que la séquence d'opérations qu'il effectue au cours de l'interaction…), **de la tâche** (le processus en plusieurs étapes de réalisation d'une tâche…) et **de l'interaction entre les deux participants en elle-même** (précision des canaux de communication utilisés…). Ces modélisations, très dépendantes les unes des autres et regroupées en un seul modèle, appelé **modèle d'interaction** (Section 1.3.), permettent d'établir le **cahier des charges** que guidera le concepteur pour élaborer son système. Selon leur degré de généricité et d'abstraction, ces modèles d'interaction peuvent s'appliquer à un ou plusieurs systèmes interactifs et peuvent orienter le développement d'applications d'une certaine classe, plus ou moins grande, de scénarios. L'élaboration d'un modèle d'interaction est une étape majeure dans le processus de conception d'un système interactif.

Pour conclure, l'étude du processus d'interaction passera forcément par l'étude des **interfaces** que l'utilisateur a à sa disposition pour interagir avec le système (Section 1.4.). Une interface comporte trois aspects : un aspect **comportemental** qui se traduit par un ensemble de techniques d'interaction que l'utilisateur doit adopter pour communiquer avec le système ; un aspect **logiciel** définissant l'environnement interactif au sein duquel l'utilisateur évolue ; et un aspect **matériel** qui décrit un éventail d'outils physiques avec lequel l'utilisateur interagit. C'est par ces interfaces que la gestuelle de l'utilisateur sera acquise par le système et que la réponse du système sera restituée à l'utilisateur. A l'instar des modèles d'interaction, les interfaces peuvent orienter le développement de certaines applications scénarisées.

1.2. Paradigmes d'interaction

Dans le cadre de l'étude du processus d'interaction scénarisé, le premier aspect de ce processus à considérer est celui des **paradigmes d'interaction**.

Un paradigme d'interaction traduit une **conception**, une **vision haut niveau**, **du processus d'interaction scénarisé**. Un paradigme décrit les traits caractéristiques et notables du processus d'interaction scénarisé qui se déroule entre l'utilisateur et le système. L'interaction peut évidemment adopter plusieurs paradigmes d'interaction de manière combinée.
Un paradigme d'interaction décrit également les **caractéristiques globales** d'un **système** et du **type d'applications scénarisées** qui s'exécuteront sur ce système. Le concepteur pourra donc établir, à partir d'un paradigme d'interaction, les grandes lignes directrices, la philosophie de conception d'un système interactif et de développement d'applications scénarisées.

Il existe de nombreux paradigmes d'interaction qui se complètent et se complémentent les uns avec les autres, que nous identifions et cartographions par le biais de la Fig. 1.1. Seront tout d'abord présentés dans la section suivante, plusieurs paradigmes, dont le **paradigme de la manipulation directe** et le **paradigme de l'interaction standard**, qui peuvent être qualifiés de **méta-paradigmes d'interaction** (voire de méta-méta-paradigmes d'interaction). Ces paradigmes ont effet inspirés la définition de nouveaux paradigmes.
Quelques-uns de ces paradigmes, les **paradigmes d'interaction *Post-WIMP***, seront alors présentés au cours de la Section 1.2.2.

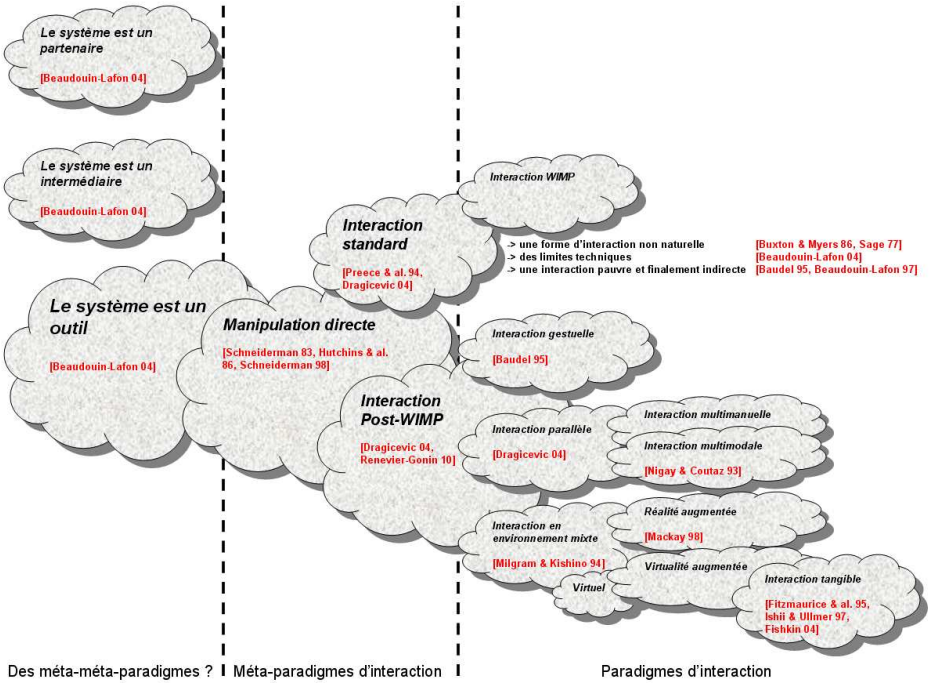


Figure 1.1. : Paradigme d'interaction

1.2.1. Méta-paradigmes d'interaction

Un **méta-paradigme d'interaction** sera défini comme étant un paradigme d'interaction à partir duquel d'autres méta-paradigmes ou paradigmes d'interaction pourront être élaborés. Nous faisons même référence à certains paradigmes, à un très niveau d'abstraction, qui pourraient être considérés comme des **méta-méta-paradigmes** (Section 1.2.1.1.).
Nous considérons que les **paradigmes de la manipulation directe** et **de l'interaction standard** sont des méta-paradigmes (Section 1.2.1.2. et Section 1.2.1.3.) car ils furent la source de nouveaux paradigmes d'interaction. En effet, le paradigme de l'interaction standard définit une interaction scénarisée que nous connaissons bien, pour l'expérimenter tous les jours : celle proposée par les interfaces de type ***WIMP***. Quant au paradigme de la manipulation directe, il décrit une forme d'interaction scénarisée que l'ensemble des paradigmes d'interaction existants cherchent à atteindre.

1.2.1.1. Des méta-méta-paradigmes ?

[Beaudouin-Lafon 04] énumère 3 méta-paradigmes (méta-méta-paradigmes) que nous pensons pouvoir être considérés comme les paradigmes les plus hauts niveaux. Ces méta-paradigmes positionnent l'utilisateur au centre du processus d'interaction scénarisé et traduisent l'image qu'il peut se faire d'un système informatique.
Etudié par la communauté ***IHM***, le premier méta-paradigme considère le système **« comme un outil »**, permettant à l'utilisateur d'étendre et d'enrichir ses capacités et ses compétences. Les paradigmes d'interaction de la manipulation directe, ***WIMP*** et ***Post-WIMP***, décrits dans les chapitres suivants, sont des exemples de paradigmes élaborés à partir de ce méta-paradigme.
Le deuxième méta-paradigme analyse le système interactif **« comme un partenaire, un collègue de travail »** qui se substitue aux moyens humains de communication, comme la parole par exemple. Autrement dit, l'utilisateur délègue au système les tâches relatives à la communication, comme un traducteur automatique par exemple.
Le dernier méta-paradigme envisage le système **« comme un intermédiaire, un médium »** rendant possible les communications inter-utilisateurs. Il s'agit par exemple des emails, des visioconférences, des chats, etc.

1.2.1.2. Le paradigme de la manipulation directe

Le paradigme de la manipulation directe (*Direct Manipulation*), [Schneiderman 83, Hutchins & al. 86, Schneiderman 98], inspiré par les idées de Douglas Engelbart, l'inventeur de la souris, peut également être considéré comme un méta-paradigme d'interaction [Baudel 95]. En effet, de nombreux paradigmes sont élaborés à partir de ce dernier car il traduit une interaction parfaite de l'utilisateur vers le système. Associée aux interfaces graphiques, l'interaction caractérisée par le paradigme de la manipulation directe donne l'illusion à l'utilisateur de **manipuler directement les objets d'intérêt**, comme il ferait dans la vie courante. Ce méta-paradigme d'interaction constitue donc une véritable **métaphore du monde réel**. Les règles qui guideront la conception d'un système obéissant à ce paradigme seront décrites au cours de la Section 1.3.2.

1.2.1.3. Le paradigme de l'interaction standard

Bien connue du grand public et largement utilisée de nos jours, l'**interaction standard** [Preece & al. 94, Dragicevic 04] regroupe l'ensemble des paradigmes d'interaction permettant un **contrôle cohérent** des applications scénarisées[1], aussi bien du point de vue de l'utilisateur que du développeur, et impliquant l'usage de périphériques génériques dits **« standards »** : l'écran, le clavier et la souris.

Les **interfaces *WIMP*** obéissent particulièrement à ce paradigme. Le paradigme de l'interaction standard ayant été élaboré à partir du paradigme de la manipulation directe, l'utilisateur a l'illusion, par le biais d'interfaces ***WIMP***, de manipuler directement les éléments informatiques relatifs à la tâche qu'il doit accomplir. Les interfaces de type ***WIMP*** sont composées de boites de dialogues, des *widgets*, des menus, des barres d'outils, etc.

Au départ considérée comme idéale, l'interaction standard a atteint ses limites au fil des années. Tout d'abord, il fut mis en avant que cette forme d'interaction n'avait **rien de naturelle**. De plus, les systèmes interactifs se diversifiant grandement, **plusieurs limitations techniques** ont été mises en évidence. Enfin, il fut démontré que cette interaction **n'obéissait pas véritablement au paradigme de la manipulation directe**.

Dès son avènement, il a été prouvé, [Buxton & Myers 86] à partir des travaux de [Sage 77], que le paradigme de l'interaction standard impose sur les applications **un contrôle limité et absolument non naturel**. Ces travaux modélisent et décrivent la physiologie et la perception d'un utilisateur dans un avenir lointain, dans le cas où le paradigme de l'interaction standard serait toujours adopté (voir Fig. 1.2.).
Les aberrations sont nombreuses et évidentes : un œil unique qui s'accorde avec la petite taille de l'écran ; une audition qui ne peut percevoir que de simples bips ; deux bras asymétriques qui ne sont pas situés au même niveau du corps, le bras destiné au clavier étant pourvu de 5 doigts peu précis et le bras destiné à la souris, plus long, plus mobile et plus précis, étant terminé par un doigt unique ; une incapacité à produire des sons et à agir par le biais d'autres parties du corps, telles que les pieds par exemple.

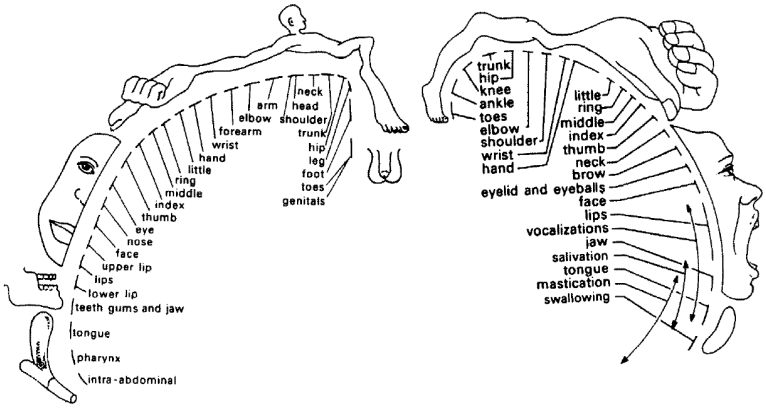


Figure 1.2. : Homonculus sensitif (à gauche) & moteur (à droite) ([Buxton & Myers 86, Sage 77])

[1] C'est-à-dire que d'une application à une autre, ce contrôle reste inchangé.

De plus, l'interaction standard a atteint ses limites au fur et à mesure que les systèmes interactifs se sont spécialisés [Beaudouin-Lafon 04] : gestion de données de plus en plus importantes et diversifiées, que ce soit par l'utilisateur ou par le concepteur du système ; plateformes sur lesquelles ces informations sont distribuées (ordinateurs, téléphones mobiles, PDA, etc.) de plus en plus complexes et variées ; et utilisateurs de plus en plus nombreux, aux profils, aux capacités, aux besoins et aux attentes de plus en plus différents. En effet, bien qu'**adaptable à tout type d'applications**, l'interaction standard n'est souvent pas **la mieux adaptée**.

Enfin, le paradigme de l'interaction standard reste une **version appauvrie et standardisée du paradigme de la manipulation directe** [Baudel 95]. L'aspect 'direct' de ce paradigme est loin d'être toujours atteint, les manipulations de la part de l'utilisateur étant alors plutôt indirectes [Beaudouin-Lafon 97]. En effet, la manipulation des objets de la tâche se fait dans la plupart des cas par le biais d'intermédiaires logiciels tels que des boites de dialogue par exemple, focalisant l'attention de l'utilisateur sur ces derniers plutôt que sur les objets d'intérêt eux-mêmes.

Il a été mis en évidence que, bien largement adoptée depuis plus de 40 ans, l'interaction standard traduit une forme d'interaction limitée non naturelle, la plupart du temps non adaptée à la tâche que l'utilisateur doit accomplir, et non directe. Malgré tout, l'interaction standard reste aujourd'hui le paradigme **le plus utilisé** sur le marché.
Cependant, de nouveaux paradigmes d'interaction, **les paradigmes d'interaction *Post-WIMP***, dont quelques exemples sont donnés au cours de la section suivante, ont été élaborés pour pallier ces limites, dans le but d'offrir à l'utilisateur une interaction naturelle, adaptée et directe.

1.2.2. Quelques paradigmes d'interaction Post-WIMP

Devant les limites de l'interaction standard mises en avant au cours de la section précédente, les besoins toujours plus nombreux des utilisateurs et les progrès technologiques incessants, de nouveaux paradigmes ont vu le jour pour **faciliter et rendre plus naturelle et efficace** (dans le sens concise et directe [Baudel 95, Dragicevic 04]) l'interaction de l'utilisateur avec le système. Nous exposons dans cette section quelques **paradigmes d'interaction *Post-WIMP***, étudiés également dans [Dragicevic 04, Renevier-Gonin 10].
Tout d'abord, **le paradigme de l'interaction gestuelle (**Section 1.2.2.1.) est présenté, traduisant une interaction par le biais des gestes de l'utilisateur.
Par la suite, **le paradigme de l'interaction parallèle (**Section 1.2.2.2.) décrit une interaction en parallèle par le biais de deux canaux de communication, paradigme au sein duquel nous distinguons **le paradigme de l'interaction multimodale** et nous définissons **le paradigme de l'interaction multimanuelle**.
Nous concluons cette section en introduisant **le paradigme de l'interaction en environnement mixte**, dont la vision conduit au développement d'applications impliquant de **la virtualité augmentée** et/ou de **la réalité augmentée (**Section 1.2.2.3.).

Les **interfaces** liées à ce paradigme sont donc qualifiées de ***Post-WIMP***. Il est également possible de rencontrer dans la littérature la notion de **systèmes *Post-WIMP*** ou encore celle de **systèmes mixtes**. D'après nos recherches, le qualificatif **'mixte'** peut être employé pour signifier **1.** que ces systèmes prennent en compte plusieurs modalités d'interaction

hétérogènes ; **2.** qu'ils permettent les interactions entre le monde réel et le monde 3D (il sera alors question de réalité mixte, paradigme d'interaction ***Post-WIMP*** défini par la suite) ; **3.** qu'ils ont été élaborés suivant l'association de plusieurs paradigmes d'interaction ***Post-WIMP***. Nous choisissons ici d'employer cet adjectif en accord avec la deuxième proposition, sujet de la Section 1.2.2.3.

Il est à prendre en compte que ces paradigmes sont **récents** et **très étudiés**. Il en résulte un nombre très important de travaux riches et complexes, qu'il nous est impossible de référencer totalement dans cette thèse. De plus, ces travaux génèrent des outils complexes, inadéquats, limités et non adaptés pour les utiliser, pour l'heure, comme des standards car trop spécialisés et dédiés à une tâche, une façon d'interagir, etc. En se rapportant au **modèle d'évolution d'une technologie au cours du temps** (voir Fig. 1.3., [Gaines 91, Nigay & Gray 06]), il est possible de dire que les technologies relatives aux nouveaux paradigmes d'interaction se situent majoritairement aux **étapes de réplication et d'empirisme**, et de plus en plus vers l'**étape de la théorie**.

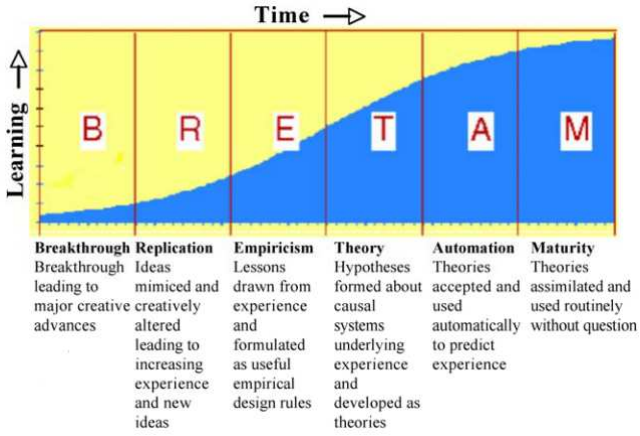


Figure 1.3. : Evolution d'une technologie au cours du temps [Gaines 91, Nigay & Gray 06]

1.2.2.1. Paradigme de l'interaction gestuelle

Dans le cas des systèmes adoptant le **paradigme de l'interaction gestuelle**, l'utilisateur emploie des **gestes**, qu'il a appris ou tout simplement intuitifs, de la vie courante pour interagir (désignation d'objet, sélection, défilement, gestes sportifs, etc.). Il peut interagir, plus ou moins conventionnellement, avec le système par le biais de périphériques matériels (les gants de type ***Data Gloves*** pour le système ***Charade*** [Baudel 95] et la fameuse ***Wiimote***[1] de Nintendo) mais les interfaces matérielles élaborées à partir de ce paradigme d'interaction **tendent à disparaître**, tout particulièrement lorsqu'il s'agit des domaines de **l'ubiquitaire** ou du **jeu vidéo** (le ***Life Wall***[2] de Panasonic, ***Kinect***[3] de Microsoft). La gestuelle étudiée varie d'une application à une autre, et peut aller de la simple détection de mouvement au suivi précis des gestuelles explicites et implicites de l'utilisateur.

[1] http://www.nintendo.com/wii/console/controllers
[2] http://www.youtube.com/watch?v=pekz2XH69CY
[3] http://www.xbox.com/en-US/kinect

Dans le cas d'une interaction gestuelle, il est possible de s'intéresser à la dynamique globale du mouvement : le début et la fin du mouvement, la position de la zone désignée par l'utilisateur, la trace du mouvement, sa vitesse, la force qu'il génère, etc. Ainsi, **une gestuelle plus variée, enrichie sémantiquement et importante** peut être considérable et interprétable par le système obéissant à ce paradigme [Baudel 95].
Cependant, en adoptant ce paradigme d'interaction, les concepteurs se heurtent constamment à des **problèmes de vitesse et précision** de capture des gestuelles de l'utilisateur ainsi que des **problèmes d'interprétation** de ces derniers s'ils présentent des ambiguïtés de sens. Ceci résulte en une trop importante simplification ou complexification des gestes compris par le système et donc en une **perte**, respectivement, **de naturel** au niveau des gestuelles adoptées par l'utilisateur ou **d'efficacité** au niveau des commandes interprétables (concision et 'directitude').

1.2.2.2. Paradigme de l'interaction parallèle

Le terme d' « **interaction parallèle** » a été employé par [Dragicevic 04] pour définir l'ensemble des paradigmes permettant une interaction en parallèle de l'utilisateur avec le système par le biais **de plus d'un dispositif** d'entrée, matériel ou assimilé à une partie du corps de l'utilisateur.
Ce terme pourra être rapproché de celui d' « **interaction multimodale** » si un dispositif d'entrée est assimilé à un **canal de communication** pris en compte et compris par le système et si plusieurs canaux de communication homogènes sont considérés comme étant chacun un canal de communication à part entière (le geste de la main droite considéré comme un canal de communication distinct de celui du geste de la main gauche).

Cependant, **le paradigme de l'interaction multimodale** est généralement considéré comme un paradigme à part entière [Nigay & Coutaz 93], englobé par celui de l'interaction parallèle, définissant une interaction de l'utilisateur avec le système **sur plusieurs canaux de communication hétérogènes**, indépendants ou combinés (la voix, le geste, la manipulation d'objets, le regard, etc.). L'utilisateur interagit par le biais de ses cinq sens, tel qu'il pourrait le faire avec une autre personne. Il est possible de se confronter à des systèmes multimodaux principalement dans les domaines de l'aéronautique, militaire, médical, du jeu vidéo, etc.

L'interaction multimodale est **très naturelle** car plus 'humaine' et les systèmes adoptant ce paradigme peuvent être très rapidement appréhendés par un grand nombre d'utilisateurs non expérimentés. De plus, une interaction multimodale présente **moins d'ambiguïtés de sens** (une commande gestuelle pourra être confirmée par une commande vocale) et, si l'ordre des modalités n'est pas important, peut **être exécutée de plusieurs manières** (redondance des commandes, la geste avant la voix ou la voix avant le geste donnent le même résultat).
Les problématiques rencontrées lors de la conception de systèmes multimodaux sont : **l'étude des mécanismes cognitifs** mis en jeu chez l'humain pour traiter des informations de nature différente; **la conception des interfaces** entre l'utilisateur et le système (ergonomie, facilité d'utilisation, etc.) ; **la capacité du système à gérer, à fusionner et interpréter des données issues de modalités hétérogènes** par le biais de différents dispositifs matériels (avec prise en compte des redondances, des erreurs et des ambiguïtés de sens). Les systèmes multimodaux sont en général complexes et mettent en jeu des architectures multi-agents pour paralléliser les traitements. Le lecteur intéressé pourra se référer à [Dumas & al. 09] qui propose un état de l'art sur l'ensemble de ces problématiques.

Au sein de l'interaction parallèle, il est nous semble également possible de définir le **paradigme de l'interaction multimanuelle**, regroupant les techniques d'interaction à deux ou plusieurs mains et/ou doigts. Cette définition complète celle du **paradigme de l'interaction bimanuelle** de [Dragicevic 04], en ajoutant l'ensemble des interactions impliquant une collaboration d'au moins deux utilisateurs et d'au moins trois mains pour exécuter une commande, comme c'est le cas pour le ***Kinect*** de Microsoft. Ce paradigme comprend aussi celui de **l'interaction multidigitale**, particulièrement adopté par l'ensemble des **systèmes tactiles** : l'***iPhone***[1] d'Apple, ***Surface***[2] de Microsoft, [Buxton 09].

Le paradigme de l'interaction multimanuelle est supporté par l'idée qu'une personne interagit constamment par le biais de ses deux mains, soit en collaboration, soit de manière asymétrique, mais toujours de manière complémentaire [Guiard 87]. De plus, il a été prouvé que l'interaction multimanuelle permettait une interaction plus rapide dans le cas de commandes spécifiques [Buxton & Myers 86]. Enfin, les ambiguïtés de sens au niveau d'une commande pourront être résolues, en prenant en compte la dominance ou la non-dominance d'une main ou d'un doigt [Guiard 87].

L'interaction multimanuelle est **très naturelle** pour le ou les utilisateurs car l'humain interagit constamment de cette manière dans la vie courante. Mais le concepteur de systèmes obéissant à ce paradigme devra faire face à des problématiques nombreuses telles que **la détection, le suivi et l'interprétation des mouvements** d'un ou plusieurs doigts et/ou d'une ou plusieurs mains et **la gestion synchrone et asynchrone temps réel** des événements interactifs que ces mouvements génèrent.

1.2.2.3. Paradigme de l'interaction en environnement mixte

Le continuum de la **« réalité mixte »** a été introduit par [Milgram & Kishino 94] et décrit les différentes combinaisons possibles, dans des proportions variables, entre **réalité** et **virtualité** (Fig. 1.4.). **Le paradigme d'interaction en environnement mixte** est adopté par l'ensemble des systèmes permettant l'échange d'informations entre **un environnement réel** et un **environnement virtuel** dont la combinaison est décrite par le continuum de la réalité mixte. Cet échange est contrôlé soit par l'utilisateur, soit par le système. Il s'agira de **virtualité augmentée** lorsque des éléments physiques, réels, sont intégrés dans un environnement virtuel et de **réalité augmentée** lorsque des éléments virtuels sont intégrés dans un environnement réel.

Figure 1.4. : Continuum de la réalité mixte [Milgram & Kishino 94, Dragicevic 04]

[1] http://www.apple.com/fr/iphone/
[2] http://www.microsoft.com/surface/

Dans le cas d'une application de **virtualité augmentée**, l'utilisateur manipule réellement un ensemble d'éléments physiques, reconnus par le système et permettant à la fois l'acquisition de la gestuelle de l'utilisateur et l'échange d'informations avec ce dernier. Ces interfaces physiques ont tout d'abord été qualifiées de **saisissables** [Fitzmaurice & al. 95] puis de **tangibles** [Ishii & Ullmer 97, Fishkin 04], générant alors des études sur l'interaction physique, par le biais d'outils matériels.

[Coutrix & Nigay 09] listent quelques exemples de **systèmes mixtes**, impliquant des applications de virtualité augmentée.

Les éléments, manipulés par l'utilisateur, pourront être soit **passifs**, c'est-à-dire seulement déclencheur d'un événement au sein de l'environnement virtuel ; soit **actifs**, c'est-à-dire à la fois déclencheur d'événements virtuels et pouvant restituer à l'utilisateur par le support matériel qu'ils constituent ces événements (***Surface*** de Microsoft).

De manière à rendre l'interaction avec l'utilisateur encore plus naturelle, ces éléments physiques pourront **inviter l'utilisateur à adopter une gestuelle particulière**, par le biais d'une forme spécifique, d'un signe, d'un son, etc. Plus les caractéristiques d'un élément interactif seront similaires à celles d'un objet de la vie courante, plus l'utilisateur interagira naturellement avec celui-ci.
Généralement, les applications de virtualité augmentée mettent en place **un ensemble de tâches collaboratives** entre plusieurs utilisateurs et les systèmes sur lesquels s'exécutent ces applications sont conçus en adoptant également le **paradigme d'interaction multimanuelle** décrit dans la Section 1.2.2.2.
Les problématiques rencontrées lors de la conception de systèmes obéissant à ce paradigme de virtualité augmentée seront **la reconnaissance** (leur forme, les signes décrits sur ces derniers, etc.) **et le suivi** des éléments physiques manipulés par l'utilisateur ainsi que **la gestion synchrone et asynchrone temps réel** des événements interactifs qu'ils génèrent. A ces problématiques s'ajouteront celles relatives au **paradigme de l'interaction multimanuelle** (Section 1.2.2.2.).

Le terme de **réalité augmentée** a été introduit par [Mackay 96]. Les systèmes sur lesquels s'exécutent des applications de réalité augmentée comprennent une interface d'acquisition, manipulée par l'utilisateur et mettant en place l'observation d'un environnement réel ; et une interface de restitution, restituant donc à l'utilisateur, l'environnement réel observé au sein duquel sont intégrés un ensemble d'informations virtuelles.

[Mackay 98] précise les différentes caractéristiques de ce paradigme et décrit quelques exemples adoptant ce dernier.

La plupart du temps, ces informations sont rattachées à un ou plusieurs éléments réels compris dans la scène. Ceci nécessite **de fortes hypothèses sur le contenu de la scène** que va observer l'utilisateur : composition de la scène, forme de certains éléments, couleur, géo-localisation de l'utilisateur, etc. C'est pourquoi les applications de réalité augmentée sont très spécialisées, orientées particulièrement **Divertissement** ou **Service** pour l'utilisateur.
Les interfaces d'acquisition et de restitution sont en généralement **combinées en un unique dispositif** matériel mobile, tel qu'un téléphone portable ou des lunettes immersives, mais

peuvent bien sûr être **séparées** respectivement en une caméra ou une webcam et un écran d'ordinateur.
Il s'agira dans le cas de ce paradigme de prendre en compte des problèmes de **reconnaissance et de suivi temps réel** d'éléments réels supposés observés (environnement bruité et capacités limitées au niveau de l'interface d'acquisition) auxquels s'ajouteront **les problématiques rencontrées** lors du développement d'applications **pour les systèmes mobiles** : gestion des ressources matérielles et logicielles limitées, robustesse et précision des algorithmes, gestion de la puissance de calcul du système, etc.

1.3. Modèle d'interaction

Dans notre démarche de conception optimale de notre système interactif, il est tout d'abord nécessaire d'étudier les paradigmes d'interaction adoptés. Cette étude nous permettra de mieux cerner le processus d'interaction scénarisé se déroulant entre l'utilisateur et le système, de manière à l'implémenter correctement et efficacement au sein du système.
Cependant, un paradigme d'interaction est une vision de l'interaction qui peut s'avérer trop vague et abstraite, permettant de concevoir le phénomène interactif à un niveau d'abstraction trop élevé et insuffisant pour la réalisation d'un système interactif complet.
C'est pourquoi, la définition ou l'adoption d'un paradigme particulier doit s'accompagner de l'étude du processus d'interaction scénarisé, se déroulant entre l'utilisateur effectuant une tâche, et le système. De cette étude, qui sera synthétisée sous la forme d'un **modèle d'interaction**, sera établi un cahier des charges spécifique, précisant, cadrant et régissant le processus d'interaction. Ce cahier des charges doit être compréhensible par les différents concepteurs du système et doit leur fournir un cadre d'études et de développements précis et détaillé. Ainsi, il les guide au cours des différentes étapes de la conception du système. L'établissement de ce modèle d'interaction constitue donc une étape essentielle et en amont de la conception du système (voir Fig. 1.5.).

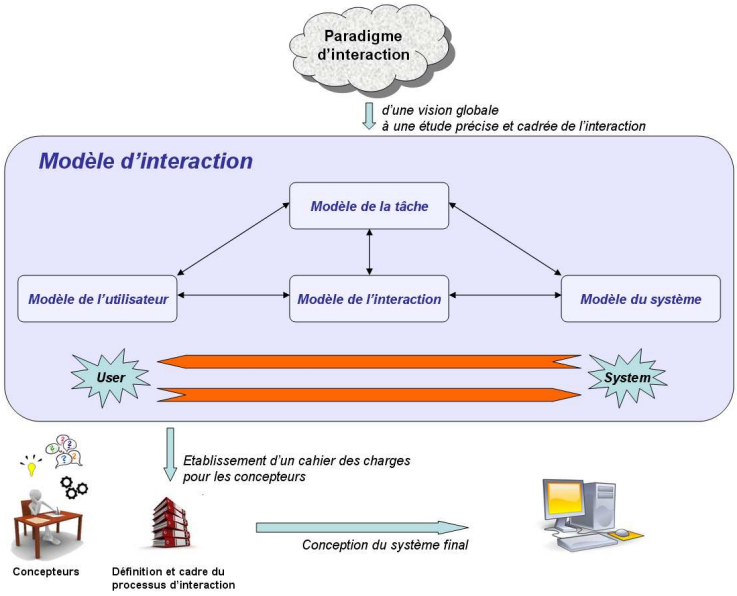


Figure 1.5. : Modèle d'interaction

Il n'existe pas de formalismes spécifiques pour décrire le modèle d'interaction suivi par un paradigme donné. C'est pourquoi plusieurs modèles d'interaction peuvent convenir à la définition d'un paradigme d'interaction donné. La notion de modèle d'interaction a été cependant définie dans plusieurs travaux, chaque modèle étant alors illustré par la conception d'un système interactif et/ou d'une application.

Pour [Baudel 95], un modèle d'interaction **décrit structurellement les communications entre l'utilisateur et le système** (aspects lexicaux, syntaxiques et pragmatiques), **indépendamment des périphériques d'entrée du système et de l'application**. Une fois l'application pensée et les périphériques choisis, l'ensemble des actions que l'utilisateur est défini, en prenant en compte les contraintes articulatoires du corps humain, et chaque action est alors liée sémantiquement au modèle conceptuel de l'application. Cette approche, très bien détaillée, est illustrée par trois types d'applications pour différents choix de périphériques d'entrée. Limité aux actions intentionnelles de l'utilisateur, ce modèle d'interaction peut être utilisé dans le cas du **paradigme de l'interaction standard** et aux **périphériques non standards**, mais peut s'avérer peu adapté aux paradigmes d'interaction récemment définis (interaction gestuelle, interaction multimanuelle, interaction multimodale, interaction environnement mixte, etc.).

[Beaudouin-Lafon 00a] décrit le modèle d'interaction comme **« un ensemble de principes, de règles et de propriétés qui guide le design de l'interface »**, dans l'optique de faire particulièrement ressentir le processus d'interaction à l'utilisateur. Il doit être évalué suivant 3 critères [Beaudouin-Lafon 04] : sa capacité à décrire une variété importante d'interfaces existantes (**puissance descriptive**) ; sa capacité à proposer des alternatives souples de conception du système (**puissance évaluative**) ; et sa capacité à proposer à l'utilisateur des outils pour créer de nouvelles interfaces (**puissance générative**). Dans ces travaux est décrit et défini le ***Modèle de l'Interaction Instrumentale***, *Instrumental Interaction Model*, perfectionnant le **paradigme de la manipulation directe**, dans la mesure où ce modèle introduit la notion d'**instruments** en tant que médiateurs entre l'utilisateur et les éléments informatiques relatifs à la tâche qu'il doit accomplir. Ces instruments sont manipulés à la manière d'outils ou d'accessoires, communs et rencontrés dans la vie de tous les jours. Ce modèle couvre un grand nombre de styles d'interaction, de l'interaction standard (interfaces ***WIMP***) aux interactions ***Post-WIMP*** (interaction multimanuelle et interaction en environnement mixte). L'application ***CPN2000/CPN Tools*** [Beaudouin-Lafon & al. 01], qui permet l'édition et la simulation de réseaux de Petri colorés, a été développée dans le but d'illustrer ce modèle d'interaction.

Pour [Bortolaso & al. 09], le modèle d'interaction se distingue et est indépendant du modèle de l'utilisateur et du modèle de la tâche. Il permet de **faciliter « la prise en compte des dimensions pertinentes pour concevoir l'interaction » de l'utilisateur avec un système**. Il doit également **structurer l'interaction**. Le modèle d'interaction assiste les concepteurs à comprendre et considérer les aspects importants et déterminants que présente l'interaction au niveau du système. Il doit être commun et compréhensible pour l'ensemble des concepteurs et développeurs du système et des applications.

A noter que dans le domaine du jeu vidéo, la modélisation de l'interaction peut être à rapprocher du *game design*. De nombreuses définitions existent et diffèrent. Le *game designer*, à partir du scénario d'un jeu vidéo donné, est chargé d'étudier l'ensemble **des éléments du jeu qui vont rendre l'expérience que va vivre le joueur intéressante, divertissante et agréable**. De plus, il doit porter son attention sur le fait que ces éléments

doivent être **cohérents les uns avec les autres, avec le scénario, le type du jeu** (combat, action, aventure, etc.) **et le profil du joueur visé**.
Les éléments étudiés du jeu sont relatifs au personnage que le joueur représente (histoire, graphisme, compétences, etc.) ; aux niveaux (leur composition, leur graphisme, l'univers, les personnages que les composent, tous les éléments qui vont donner au joueur l'envie de les explorer) ; au *gameplay* (périphériques utilisés, techniques d'interaction, ergonomie des commandes, etc.) ; au scénario en lui-même (narration au cours de la progression du joueur, difficulté des niveaux, les énigmes et aventures proposées, les 'règles du jeu', etc.)…

Devant les différences que peuvent présenter ces définitions, il nous est nécessaire de synthétiser ces différentes propositions de manière à ce que nous puissions nous positionner par rapport à celles-ci (voir Fig. 1.6.).

[Baudel 95]	• Description des communications entre le système & l'utilisateur • Indépendance vis-à-vis des périphériques & de l'application • Liaison avec le vocabulaire d'actions intentionnelles de l''utilisateur	→ *En fonction de la tâche, ça ne peut pas se limiter qu'à cela* → *Il est préférable d'étudier une communication interactive dans son contexte, en fonction de la tâche et des périphériques utilisés* → *Les nouveaux paradigmes ne s'arrêtent pas qu'aux entrées intentionnelles de l'utilisateur*
[Beaudouin-Lafon 00, 04] [Beaudouin-Lafon & al. 01]	• Guide pour la conception d'interface utilisateur-système • Etude du processus d'interaction avec l'utilisateur • Mise en œuvre d'outils, des instruments, pour manipuler directement les objets de la tâche	→ *L'interface est dépendante de l'application, de la tâche et des périphériques* → *La vision est globale mais se limite à l'interface* → *S'applique bien aux interfaces WIMP mais est moins évidente dans le cadre de jeux vidéo*
[Bortolaso & al. 09]	• Indépendance vis-à-vis de la tâche & de l'utilisateur • Prise en compte des dimensions pertinentes du processus d'interaction • Mise en relief des aspects importants & déterminants du processus d'interaction	→ *Le processus d'interaction ne peut pas être indépendant et de la tâche et de l'utilisateur* → *C'est une bonne vision du processus d'interaction mais la définition reste tout de même un peu vague*
Game Design	• Etude de l'ensemble des éléments du jeu qui va rendre l'expérience vécue par le joueur intéressante, divertissante et agréable • Etude de la cohérence de ces éléments les uns avec les autres, avec le scénario, avec le type du jeu et avec le profil du joueur cible	→ *De nombreuses définitions existent mais c'est exactement tel que nous concevons la notion de modèle d'interaction*

Figure 1.6. : Synthèse : définition de la notion de modèle d'interaction

Un modèle d'interaction doit permettre aux différents concepteurs de comprendre aussi précisément que possible le processus d'interaction scénarisé qui s'établit entre le système et l'utilisateur. Or, nous pensons que ce phénomène prend place au niveau de l'**utilisateur**, au niveau du **système** et **entre l'utilisateur et le système**. De plus, il est fonction de la **tâche** que doit accomplir l'utilisateur et qui est définie soit par le scénario de l'application, soit par l'utilisateur lui-même. C'est pourquoi, nous ne pensons pas ici qu'un modèle d'interaction doive être indépendant des canaux de communication qu'exploitent l'utilisateur et le système [Baudel 95], ou être indépendant du modèle de tâche et du modèle de l'utilisateur [Bortolaso & al. 09].
De plus, le modèle d'interaction doit **piloter les différents concepteurs au cours de chaque étape de l'élaboration du système**. Sa définition doit être assez précise pour leur permettre de modéliser le système dans sa globalité, pas seulement au niveau de son interface [Baudel

95, Beaudouin-Lafon 00], mais également au niveau de l'architecture du système, au sein de laquelle le processus d'interaction scénarisé prend place.
Enfin, la définition d'un modèle d'interaction doit être **commune à l'ensemble des paradigmes d'interaction**. [Baudel 95] se limite aux entrées intentionnelles de l'utilisateur. Or, les nouveaux paradigmes peuvent prendre en compte aussi bien les gestuelles explicites de l'utilisateur que ses gestuelles implicites. Quant au ***Modèle de l'Interaction Instrumentale*** [Beaudouin-Lafon 00a], il est adapté à la majeure partie des paradigmes d'interaction ***Post-WIMP*** mais parait peu adéquat pour des applications impliquant l'évolution de l'utilisateur dans un environnement 3D, comme un jeu vidéo par exemple.
Un bon *game design*, à l'image des modèles définis par [Beaudouin-Lafon 00a] et [Bortolaso & al. 09], reflète l'étude extensive de tous les **aspects pertinents et essentiels de l'interaction qui enrichissent l'expérience de l'utilisateur**. Une fois les dimensions importantes de l'interaction mises en avant, l'ensemble des concepteurs peuvent en permanence se référer au modèle d'interaction pour élaborer le système interactif.

Nous proposons dans la section suivante **une définition de la notion de modèle d'interaction**, en tentant d'englober les définitions exposées précédemment et de nous positionner par rapport aux différences qu'elles présentent les unes avec les autres (voir Fig. 1.7.). De plus, notre définition doit être en accord avec les différents modèles d'interaction déjà existants, que nous présentons au cours de la Section 1.3.3. C'est pourquoi le modèle d'interaction est défini ici comme étant **la modélisation de l'architecture de l'interaction** se déroulant entre le système et l'utilisateur, sous la forme d'**opérations**, d'**états** et de **transitions**, au niveau de l'utilisateur, du système ou entre les deux. Un modèle d'interaction peut requérir **plusieurs modélisations complémentaires :** utilisateur, système, tâche et interaction entre l'utilisateur et le système.
Par la suite, plusieurs modèles d'interaction sont présentés (voir Fig. 1.7.). Le paradigme de l'interaction standard et les paradigmes d'interaction ***Post-WIMP***, décrits précédemment, ont été élaborés à partir du **paradigme de la manipulation directe**. C'est pourquoi le modèle d'interaction cadrant ce paradigme est décrit précisément dans la Section 1.3.2.
Au cours de la Section 1.3.3., **quelques exemples de modèles d'interaction** sont donnés relativement aux paradigmes d'interaction ***Post-WIMP*** décrits dans la Section 1.2.2.

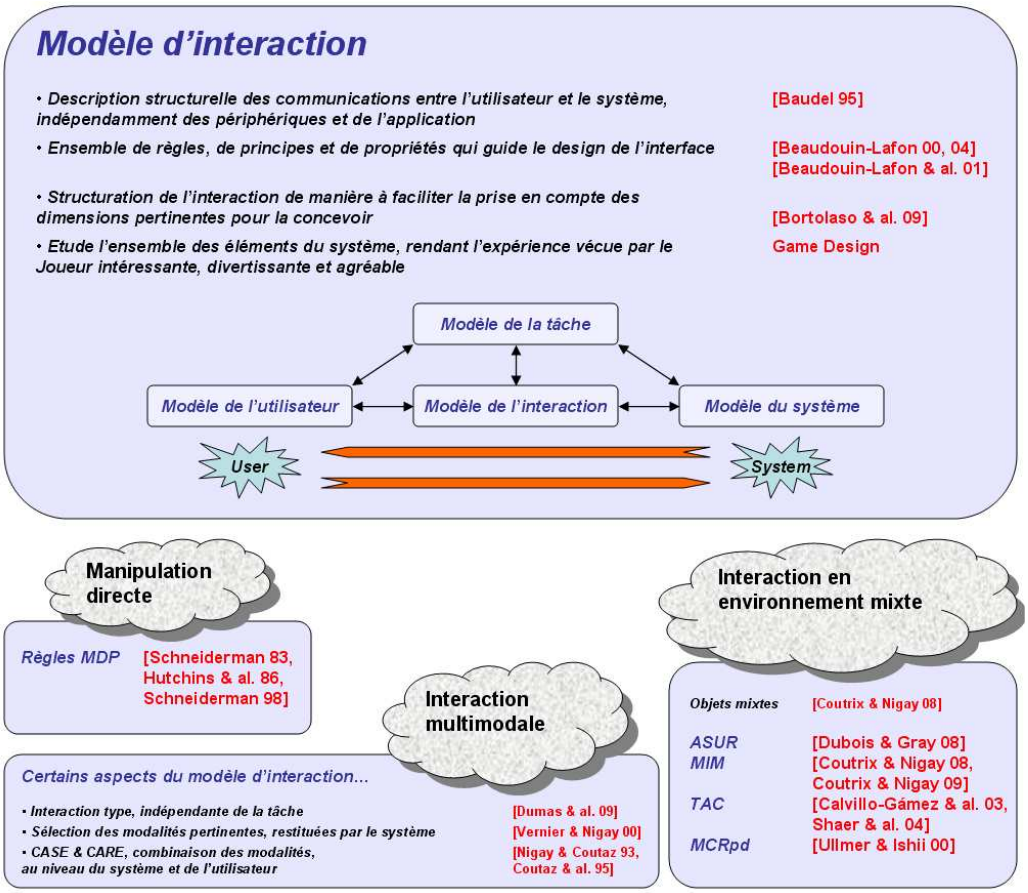


Figure 1.7. : Synthèse des travaux présentés au cours de la Section 1.3.

1.3.1. Définition et description d'un modèle d'interaction

Sur les différentes observations établies précédemment, nous proposons de définir la notion de modèle d'interaction comme étant la **modélisation de l'interaction scénarisée en tant que processus** ayant lieu entre **l'utilisateur**, effectuant une **tâche** et cherchant à atteindre un **objectif** (défini par lui-même ou par le scénario de l'application), et le **système**, répondant à la gestuelle de l'utilisateur, en accord avec le scénario de l'application. Il décrit les **étapes de l'interaction** comme une **séquence d'opérations et d'états, organisée par un ensemble de transitions**, au niveau de l'utilisateur, au niveau du système ou au niveau intermédiaire traduisant les communications entre l'utilisateur et le système au cours d'une interaction scénarisée type.

Un modèle d'interaction est différent de l'architecture matérielle et logicielle d'un système. En effet, un modèle d'interaction décrit plutôt l'architecture d'une interaction type qui peut avoir lieu entre l'utilisateur et un système donné. Aucune spécificité matérielle ou logicielle du système n'est représenté au sein d'un modèle d'interaction. C'est pourquoi ce dernier, selon son degré de généricité et d'abstraction, peut s'appliquer pour plusieurs systèmes interactifs ayant les mêmes fonctionnalités. De plus, un modèle d'interaction peut orienter le développement de certaines applications, et donc de scénarios, particulières.

Pour être complet et exhaustif, nous proposons qu'un modèle d'interaction implique :

- **la modélisation de l'utilisateur** (Section 1.3.1.1.),
- **la modélisation du système** (Section 1.3.1.2.),
- **la modélisation d'une tâche** (Section 1.3.1.3.),
- **la modélisation des étapes de l'interaction entre l'utilisateur et le système** (Section 1.3.1.4.).

Ces modélisations, liées les unes avec les autres, sont soit comprises au sein du modèle d'interaction, soit fortement rattachées à ce dernier. Il existe de nombreuses approches pour élaborer ces différents modèles.

L'élaboration d'un modèle d'interaction constitue une étape majeure, située en amont de la conception d'un système interactif. En effet, par le biais d'un modèle d'interaction, les concepteurs peuvent alors définir un **ensemble de principes et de règles**, plus ou moins précises et rigoureuses (de la philosophie de conception à suivre aux différentes unités de traitements qui doivent composer le système), qui les guideront, chacun ou ensemble, lors de l'élaboration logicielle et matérielle du système interactif. C'est pourquoi l'élaboration d'un bon modèle d'interaction a pour conséquence la conception efficace et optimale d'un système interactif.

1.3.1.1. Modélisation de l'utilisateur

Cette modélisation doit permettre de décrire les différentes étapes de **perception**, de **raisonnement** et d'**action** qui ont lieu au niveau de l'utilisateur lors de l'interaction. Il existe de nombreuses modélisations possibles de l'utilisateur (voir Fig. 1.8.) mais ces dernières sont clairement inspirées du ***Modèle du Processeur Humain***, *Model Human Processor* (***MHP***), défini par [Fitts & Posner 67, Card & al. 83], récemment abordé par [Sears & Jacko 09] et qui modélise le processus de traitement d'informations au niveau de l'utilisateur. Ce processus est divisé en 3 sous-processus, qui interagissent les uns avec les autres et chacun couplé avec un ensemble de mémoires dédiées :

- **Le processeur de perception**, *Perceptual Processor*, qui stocke et code les informations (visuelles, sonores, etc.) provenant de l'environnement extérieur ;
- **Le processeur regroupant les différents mécanismes cognitifs**, *Cognitive Processor*, responsables de la prise de décision sur la réponse à émettre, en prenant en compte les informations perçues et les différentes connaissances (par exemple, relatives au scénario d'une application) que possède l'utilisateur ;
- **Le processeur relatif aux capacités motrices de l'utilisateur**, *Motor Processor*, qui traduit la réponse de ce dernier, vis-à-vis des informations perçues.

Il est difficile de décrire un modèle utilisateur indépendamment d'un **modèle de la tâche** et d'un **modèle de l'interaction**. En effet, pour une tâche donnée, l'utilisateur percevra, raisonnera et agira spécifiquement au cours de l'interaction. Les canaux de perception et d'action sont alors en général particulièrement détaillés, en prenant en compte les différents canaux de communication et les limites des capacités sensori-motrices d'un être humain.

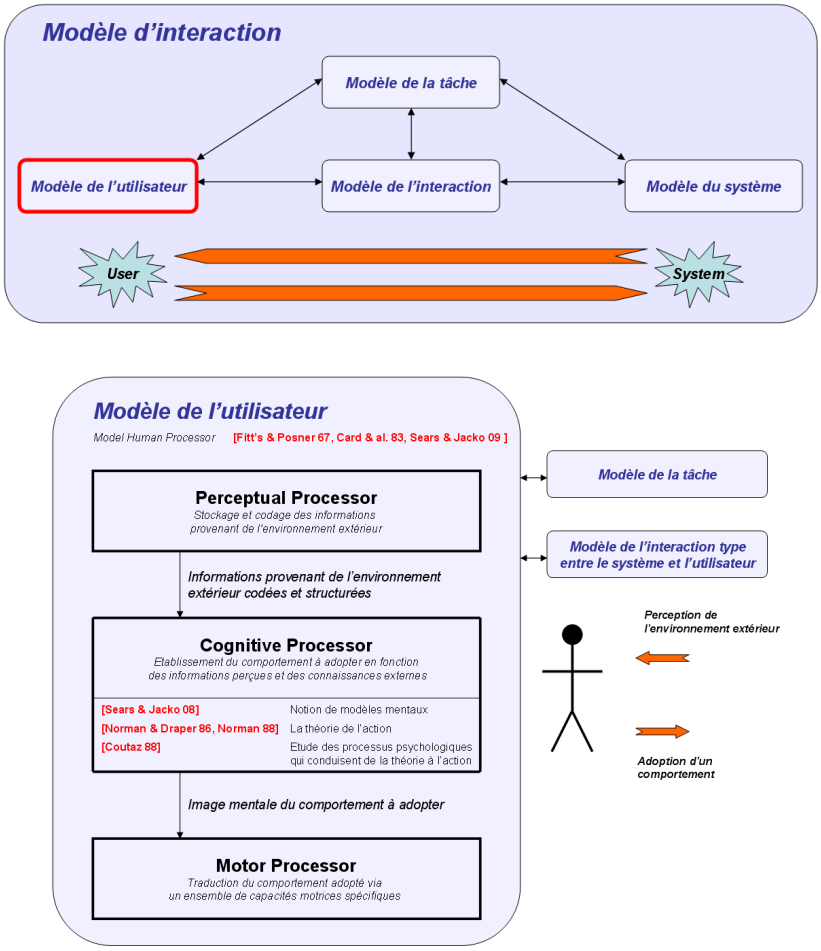


Figure 1.8. : Modèle de l'utilisateur

Au fil du temps, de nombreux travaux ont étudié ces processus, à différents niveaux d'abstraction, mais c'est le **processeur cognitif** qui a principalement intéressé les scientifiques. En effet, ce processeur est **responsable de la gestuelle adoptée par l'utilisateur**. Cette adoption est fonction de nombreuses **connaissances** (sur l'environnement, sur l'état interne de l'utilisateur, sur ses objectifs définis par le scénario ou par lui-même, sur ses connaissances sur le système, sur le scénario de l'application, etc.) et des **mécanismes cognitifs** spécifiques, comme la sélection d'informations pertinentes relativement à une tâche à accomplir par exemple (notion de modèles mentaux, abordée dans [Sears & Jacko 08]).

Pour ne citer qu'une approche fameuse, avant l'adoption d'une gestuelle donnée, la ***théorie de l'action*** [Norman & Draper 86, Norman 88] établit que l'utilisateur construit des modèles conceptuels, sur la base de ses connaissances et de la situation d'interaction courante. S'inspirant de ces travaux, [Coutaz 88] propose une analyse des processus psychologiques qui conduisent de la théorie à l'action, en définissant les distances d'exécution et d'évaluation qui peuvent exister entre l'objectif tel qu'il est modélisé dans le modèle utilisateur et l'objectif traduit et présenté par le système à l'utilisateur par le biais des interfaces.

1.3.1.2. Modélisation du système

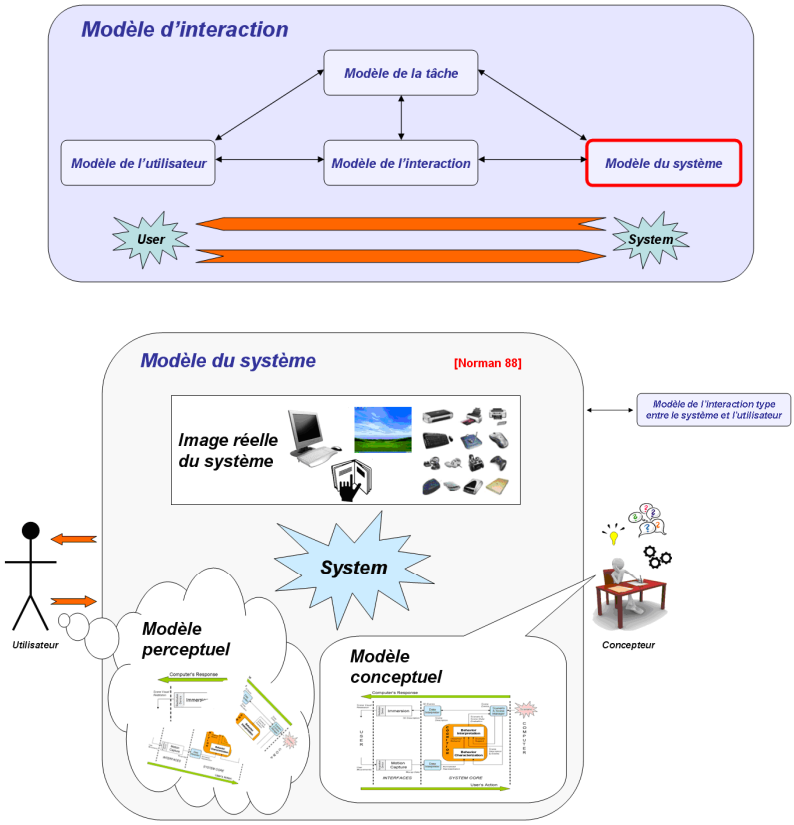


Figure 1.9. : Modèle du système

A l'image du modèle utilisateur, il existe plusieurs modèles du système (voir Fig. 1.9.). Il ne s'agit pas ici de modélisation d'architecture logicielle mais d'une **représentation du processus global du traitement de l'information** et de **la gestion de l'interaction scénarisée** par le système.

Au cours de l'interaction, il existe trois modélisations du système [Norman 88] :

- **L'image réelle du système**, i.e. tous les éléments matériels et physiques relatifs au système (de l'interface au manuel d'utilisation) en plus de toutes les opérations qu'il permet, que ce soit du point de vue de l'utilisateur que du concepteur ;
- **Le modèle conceptuel du système qu'élabore le concepteur**, décrivant les différentes étapes du fonctionnement du système au cours de l'interaction avec l'utilisateur ;
- **Le modèle perceptuel du système qu'élabore l'utilisateur**, et qui se modifie et se complète au cours de l'interaction. Ce modèle prend en compte également les différentes expériences interactives antérieures avec d'autres systèmes.

Un modèle de système pourra être dépendant de celui de l'interaction type qui peut se dérouler entre ce dernier et l'utilisateur. En effet, le système pourra restituer sa réponse, au cours de l'interaction, suivant différents canaux de communication, qui seront détaillés dans le modèle de l'interaction.

1.3.1.3. Modélisation de la tâche

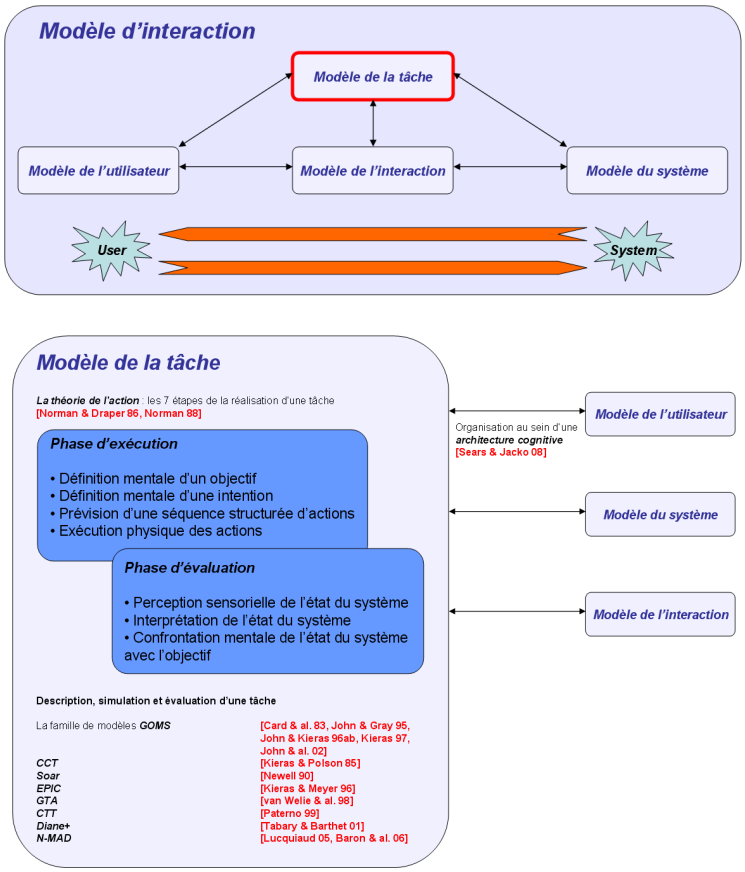


Figure 1.10. : Modèle de la tâche

Une tâche est accomplie par l'utilisateur en fonction de ses objectifs, qui sont définis soit par le scénario de l'application, soit par lui-même. Toujours d'après la théorie de l'action, [Norman & Draper 86, Norman 88], **la réalisation d'une tâche** par l'utilisateur se décrit en **7 étapes**, regroupées en **2 phases**, dépendantes des capacités sensorielles, cognitives et motrices de l'utilisateur et de l'interface entre ce dernier et le système :

Phase d'exécution

- Définition mentale d'un **objectif**, pouvant être décomposé en plusieurs sous-objectifs ;
- Définition mentale d'une **intention** ;
- Définition mentale d'une **séquence structurée d'actions** que l'utilisateur prévoit d'effectuer ;
- **Exécution physique des actions** définies dans la séquence, par le biais des capacités motrices de l'utilisateur.

Phase d'évaluation

- **Perception** de l'état du système, par le biais des capacités sensorielles de l'utilisateur ;

- **Interprétation** de l'état du système, en fonction des attentes de l'utilisateur ;
- **Confrontation** de l'**état du système** avec l'**objectif** défini au cours de la phase d'exécution.

L'ordre de ces étapes n'est pas établi (une étape peut se répéter plusieurs fois de suite et il est possible de revenir sur n'importe quelle étape à n'importe quel moment) et il n'est pas nécessaire de passer par toutes ces étapes pour accomplir une tâche.
Un modèle de la tâche est à rapprocher du modèle de l'utilisateur défini dans la Section 1.3.1.1. En effet, les 3 premières étapes, ainsi que les 2 dernières, sont propres au processeur cognitif ; la 4ème étape au processeur moteur ; et la 5ème au processeur perceptuel du ***Modèle du Processeur Humain***. Le modèle de la tâche prend également en compte certaines caractéristiques du système (le scénario, son fonctionnement, etc.) et le déroulement d'une interaction type entre l'utilisateur et le système (l'utilisateur va interagir par le biais de canaux de communication spécifique par exemple). Le modèle de la tâche est donc lié au modèle du système défini au cours de la Section 1.3.1.2. et celui de l'interaction entre l'utilisateur et le système défini dans la section suivante.

Sur ces bases ont été établis des modèles (voir Fig. 1.10.) permettant la **description**, la **simulation** et l'**évaluation** de tâches spécifiques, dans le cadre d'une interface donnée. Ces modèles permettent de **définir** et d'**affiner** précisément une tâche, de manière à la rendre **évidente** et facilement **réalisable** pour l'utilisateur. De plus, ils constituent un moyen **d'étude des gestuelles adoptées par l'utilisateur**. En effet, pour une tâche donnée, certains modèles prédisent les gestuelles de l'utilisateur pour l'accomplir. Ainsi, en modifiant les propriétés de la tâche, il est possible de **simuler** les différentes gestuelles que peut adopter l'utilisateur de manière à étudier les **incohérences** qu'elles peuvent présenter. Dans [Sears & Jacko 08], ces modèles sont englobés au sein d'une **architecture cognitive**, *cognitive architecture*, modélisation de plus haut niveau, impliquant l'usage des différentes mémoires, détaillant l'ensemble des communications entre les différents mécanismes, etc.

Tout d'abord introduits par [Card & al. 83] puis particulièrement étudiés par [John & Kieras 96ab], **les modèles *GOMS*** permettent la description d'une tâche en termes de ***G****oals* (les objectifs à atteindre), ***O****perators* (actions permises par l'interface et que l'utilisateur doit effectuer pour accomplir sa tâche), ***M****ethods* (séquence d'*operators* et de sous-*goals* à réaliser pour atteindre un objectif donné) et ***S****election Rules* (règle de sélection permettant d'élire une *method* parmi un choix de *methods* concurrentes pour atteindre un objectif donné). Par le biais de ces modèles, il est possible d'analyser une tâche et donc les mécanismes cognitifs mis en jeu lors de la réalisation de cette dernière. De plus, certains modèles prédisent les temps d'exécution, d'apprentissage, les erreurs, tout en identifiant les différentes facettes de l'interface qui mènent à ces prédictions.
Les modèles *CMN-GOMS* (***Card, Moran & Newell GOMS***) et ***KLM*** (***Keystroke-Level Model***), en plus d'être les premiers définis, sont les plus basiques. Par le biais de ces modèles, le temps d'exécution final est calculé en additionnant les temps d'exécution des différents *operators* mis en jeu lors de la réalisation d'une tâche. **Les modèles *NGOMSL*** (***Natural GOMS Language*** - [Kieras 97]) et ***CPM-GOMS*** (***Cognitive-Perceptual-Motor GOMS*** - [John & Gray 95, John & al. 02]) sont plus élaborés. ***NGOMSL*** permet la représentation et la construction d'un modèle ***GOMS*** grâce au langage naturel. Ce modèle peut le temps d'apprentissage par l'utilisateur et le temps d'exécution d'une *method* donnée. ***CPM-GOMS***,

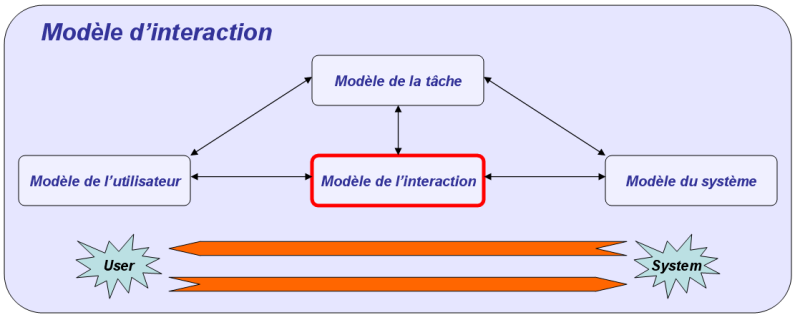
quant à lui, prend en compte les capacités sensori-motrices de l'utilisateur, ainsi que ses capacités à accomplir plusieurs tâches en parallèle.
Malgré le nombre d'études sur le sujet, un modèle ***GOMS***, quel qu'il soit, reste difficile et long à élaborer.

Depuis l'introduction de la famille de modèles ***GOMS***, **d'autres modèles de tâches** ont été établis : ***CCT*** (***Cognitive Complexity Theory***) [Kieras & Polson 85], ***Soar*** [Newell 90], ***EPIC*** [Kieras & Meyer 96], ***GTA*** [van Welie & al. 98], ***CTT*** [Paterno 99], ***Diane+*** [Tabary & Barthet 01] et plus récemment ***N-MAD*** (***Noyau du Modèle de Description de l'Activité***) [Lucquiaud 05, Baron & al. 06] qui définit et lie étroitement deux modules, dédiés à la description formelle, respectivement, de la tâche que l'utilisateur doit accomplir et des éléments constituant l'environnement de ce dernier. En général, chaque modèle est éditable par un outil logiciel dédié. Par exemple, dans le cas du modèle ***N-MAD***, l'outil ***K-MADe*** (***Kernel of Model for Activity Description environment***) offre la possibilité de décrire et d'analyser les tâches d'un utilisateur face à un système interactif.

1.3.1.4. Modélisation de l'interaction entre l'utilisateur et le système

Le modèle de l'interaction (voir Fig. 1.11.) décrit **la logique de communication, décrite par l'interaction** (les différents canaux de communication utilisés, l'ordre des participants, etc.) **; les différents états de l'information au cours de l'interaction ;** et **les mécanismes de traitements de l'information** qui surviennent lors d'un échange 'une action – une réaction' entre l'utilisateur et le système.

Ce modèle peut donc prendre en compte la totalité ou une partie du **modèle utilisateur** (en particulier, les composants dédiés à la perception et à l'action) et la totalité ou une partie du **modèle système**. De plus, ce modèle peut décrire une interaction générique se déroulant typiquement entre le système et l'utilisateur, sans prendre en compte le modèle de la tâche, ou peut exposer et détailler une interaction plus spécialisée, car effectuée dans le cadre de tâches bien particulières.



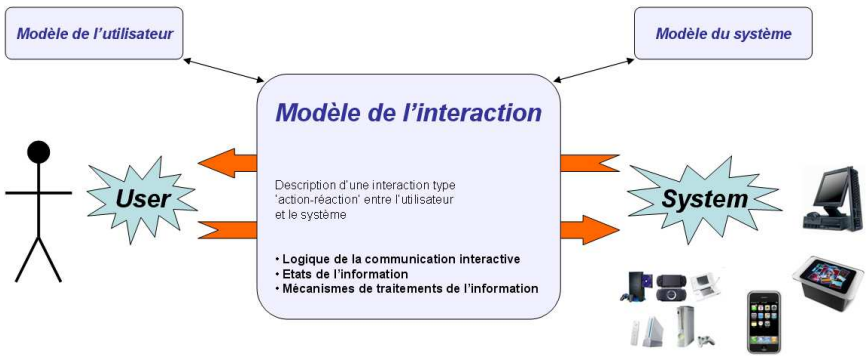


Figure 1.11. : Modèle de l'interaction entre l'utilisateur et le système

1.3.2. Paradigme de la manipulation directe

Le **modèle d'interaction** associé au paradigme de la manipulation directe [Schneiderman 83, Hutchins & al. 86, Schneiderman 98] décrit un ensemble de règles de conception dans le but de permettre à l'utilisateur de manipuler directement et naturellement les éléments relatifs à la tâche qu'il doit accomplir. **Les techniques d'interaction**, définies à partir de ces règles mais aucunement imposées, donnent à l'utilisateur les sentiments de s'engager physiquement dans l'action et d'agir directement sur un environnement informatique.

Les principes (**MDP** pour « Manipulation Directe, Principes ») définis par ce modèle d'interaction sont les suivants :

- **MDP1 : L'utilisateur agit physiquement et directement sur les objets d'intérêt**, plutôt que d'exprimer des actions par le biais d'une commande syntaxique. Idéalement, chaque action de la part de l'utilisateur doit entraîner un changement d'état du système et/ou de l'application.
  L'action directe de l'utilisateur peut se faire soit sur une représentation informatique de l'objet d'intérêt par l'intermédiaire d'une interface entre l'utilisateur et le système ; soit directement sur cette interface physique si celle-ci constitue l'objet d'intérêt en lui-même (dans le cas d'une application de virtualité augmentée par exemple).
  Plus l'utilisateur oubliera l'interface physique qui le sépare des objets d'intérêt, plus son action sera directe. Cette élimination cognitive des intermédiaires physiques définit la notion de **transparence d'une interface**. La transparence d'une interface sera plus importante si cette dernière invite l'utilisateur, par le biais d'une forme, d'un icône (utilisation d'affordances [Gibson 77]), d'un signe, etc. à adopter une gestuelle spécifique [Hutchins & al. 86]. Enfin, la transparence d'une interface sera **maximale** si l'utilisateur interagit directement avec les membres de son corps, sans intermédiaire matériel.
- **MDP2 :** L'utilisateur est capable d'**appréhender rapidement** une nouvelle application, **de manière naturelle**, et **sans pour autant posséder des connaissances particulières et spécialisées sur la tâche** qu'il doit accomplir. Chaque technique d'interaction est apprise rapidement, par étapes, et obéit aux mêmes principes d'une application à une autre.
- **MDP3 :** Chaque opération est **simple** et **rapide**, dénote une **progression incrémentale** vers le but à atteindre, et est **réversible** (possibilité de tests et d'erreurs).
- **MDP4 :** Les objets d'intérêt et les influences des actions de l'utilisateur sur ces derniers doivent être **visuellement observables**, **immédiatement et de manière continue**. Ainsi,

grâce à ce retour constant d'information, une image de l'état courant du système et de son fonctionnement est transmise à l'utilisateur.

Le modèle d'interaction du paradigme de la manipulation directe décrit **une interaction parfaitement naturelle** entre l'utilisateur et le système. Il n'est cependant pas sans inconvénients. En effet, ce modèle reste **abstrait** et ne cadre pas suffisamment les concepteurs et développeurs qui doivent faire face à une infinité de possibilités quant à l'élaboration d'un système et d'une application. Ces derniers optent alors pour un ensemble de techniques d'interaction **simplistes** et souvent confinées à la **désignation**, la **sélection** et au **déplacement** des objets d'intérêt, dans le but de permettre à l'utilisateur non expérimenté d'interagir plus rapidement et très facilement avec le système. En revanche, en optant pour des techniques plus longues et élaborées, seul l'utilisateur **expérimenté** pourra explorer l'ensemble des fonctionnalités que lui offre l'interface.

1.3.3. Des modèles d'interaction adaptés aux nouveaux paradigmes

Comme il l'a été précisé précédemment, les paradigmes d'interaction décrits dans la Section 1.2.2. étant particulièrement récents, les modèles d'interaction relatifs à ceux-ci sont **en cours de formalisation**. Pour un paradigme donné, l'élaboration du modèle d'interaction se heurte à **deux problèmes majeurs**. D'une part, les systèmes adoptant ce paradigme doivent **répondre à un besoin de l'utilisateur**, auquel un autre système existant ne peut pas répondre, ou, tout du moins, plus difficilement, moins naturellement, etc. D'autre part, un choix déterminant doit être fait quant aux **technologies employées** par le système. En effet, ces nouvelles technologies évoluent constamment et rapidement. C'est pourquoi les systèmes interactifs adoptant ces nouveaux paradigmes sont **complexes** et présentent **une importante variabilité** à différents niveaux.

Les travaux des différentes communautés, nombreux, fournissent donc un éventail infini de solutions pour les différents concepteurs d'un système. Les modèles d'interaction sont riches mais il devient impossible pour les concepteurs du système d'analyser l'ensemble des facettes de ce dernier. Suite à ce constat, les modèles d'interaction relatifs aux **paradigmes d'interaction multimodale et en environnement mixte** sont particulièrement étudiés. Quelques travaux concernant ces paradigmes sont exposés dans la suite de cette section.

1.3.3.1. Interaction multimodale

[Dumas & al. 09] propose dans ses travaux une modélisation d'une interaction multimodale entre l'utilisateur et le système (voir Fig. 1.12.), pouvant être la base d'un modèle formalisé d'interaction multimodale. De manière générale, les modèles **bas niveau** s'attaqueront aux aspects techniques de l'interaction, tandis que les modèles **haut niveau** étudieront les mécanismes globaux mis en jeu lors de l'interaction.

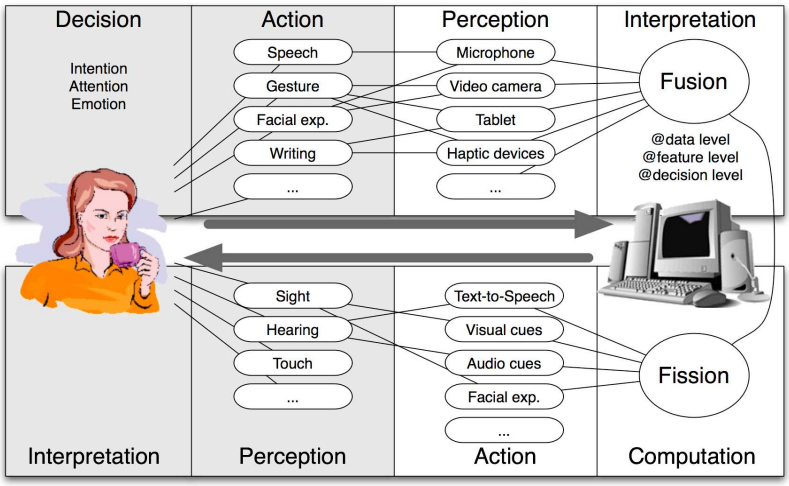


Figure 1.12. : Interaction multimodale entre l'utilisateur et le système [Dumas & al. 09]

Dans un premier temps, l'utilisateur **(1)** décide de l'action à réaliser, en fonction de ses objectifs et de son état interne, puis **(2)** agit en transmettant un certain nombre d'informations par le biais de multiples canaux de communication hétérogènes. Le système **(3)** perçoit et capte ces informations pour **(4)** les interpréter par la suite, indépendamment les unes des autres et/ou en les combinant (notion de **fusion** de données).
Ensuite, le système **(5)** génère (notion de **fission** des données) et **(6)** transmet sa réponse multimodale. L'utilisateur **(7)** perçoit cette réponse suivant plusieurs modalités dédiées. De la même façon que le système, l'utilisateur **(8)** interprète la réponse du système en considérant chaque modalité indépendamment les unes aux autres et/ou en les combinant.

Les **modalités d'interaction** se retrouvent au centre du problème, lors de l'élaboration d'un modèle d'interaction multimodale. En effet, du point de vue de l'utilisateur et du système et aussi bien en entrée qu'en sortie, le concepteur doit tout d'abord **choisir** et **caractériser les modalités pertinentes** parmi le choix infini de modalités d'interaction qu'il peut considérer ; puis **établir les différentes combinaisons de données**, issues de canaux de communication hétérogènes, qu'il peut envisager ; pour enfin **décider des technologies** qui **capteront** et **restitueront** les informations selon les modalités choisies.

Chaque problématique est en général un sujet d'études à part entière. Par exemple, [Vernier & Nigay 00] définit, uniquement pour les modalités d'interaction restituées à l'utilisateur, un espace de caractérisation, permettant la sélection des modalités pertinentes relativement au système développé, et un espace pour analyser les combinaisons de ces modalités. Ces travaux font suite à ceux de [Nigay & Coutaz 93] et [Coutaz & al. 95] qui définissaient respectivement les modèles ***CASE*** et ***CARE*** pour analyser les possibilités de combinaisons de modalités, au niveau du système et de l'utilisateur.

1.3.3.2. Interaction en environnement mixte

Les modèles d'interaction relatifs au **paradigme d'interaction mixte** sont récents et font l'objet de nombreux travaux très complets. Ces modèles sont centrés sur la notion d'**objets**

mixtes [Coutrix & Nigay 08], éléments du système qui comprennent **une facette physique**, avec laquelle l'utilisateur interagit, et **une facette numérique**, avec laquelle le système interagit. Trouver **un compromis efficace et cohérent** concernant **la part de réel** et **la part de virtuel** dans la modélisation d'un objet mixte constitue, pour le concepteur, une des problématiques majeures, à laquelle doit répondre le modèle d'interaction. Ces objets mixtes sont étroitement **liés à la tâche** que l'utilisateur doit accomplir et permettent les **communications**, suivant des modalités spécifiques (les *linking modalities*, dont la définition est inspirée des modalités d'interaction définies dans [Vernier & Nigay 00]) **entre les mondes physiques et virtuels**.

Depuis leur élaboration, **deux modèles** sont largement étudiés et évalués : **Le modèle *ASUR*** (***Adapter, System, User, Real object***), précisément décrit et perfectionné dans [Dubois & Gray 08], et **le *Modèle d'Interaction Mixte* (*MIM*)** [Coutrix & Nigay 08, Coutrix & Nigay 09].

ASUR a été développé dans le but d'**analyser les différentes combinaisons possibles entre les mondes réels et virtuels**, relativement aux tâches que l'utilisateur doit accomplir. Par le biais de ce modèle, le système interactif est conçu en mettant en jeu les objets mixtes, les plus efficaces et pertinents, qui permettront à l'utilisateur d'accomplir sa tâche. Considérant l'interaction du point de vue de l'utilisateur, ce modèle définit d'une part l'ensemble des **entités, physiques et virtuelles, participant à l'interaction** (les éléments, les *adapters*, permettant le passage entre les mondes réels et virtuels ; les différents outils réels et virtuels liés à la tâche ; et le ou les utilisateurs). D'autre part, il décrit les différents **flux d'informations**, réelles ou virtuelles, et les **associations** entre les différentes entités.
MIM décrit le système interactif comme **un ensemble structuré d'objets mixtes**. Ce modèle permet la conception et l'évaluation de systèmes mixtes. Dans ce modèle, un objet mixte est décrit, au sein d'un espace de caractérisation, par ses propriétés **intrinsèques**, i.e. ses caractéristiques qui sont pas modifiées d'un contexte d'interaction à un autre, et **extrinsèques**, i.e. ses caractéristiques qui sont modifiées d'un contexte d'interaction à un autre. Les propriétés d'un objet mixte peuvent être **physiques** (la taille, la couleur, etc.) ou **numériques** (représentation virtuelle, état interne, etc.). Les *linking modalities* lient les propriétés réelles d'un objet avec ses propriétés virtuelles. Le modèle ***MIM*** a été utilisé pour comparer et évaluer un ensemble de systèmes mixtes existants, impliquant des interfaces tangibles, de la réalité augmentée et de la virtualité augmentée. De plus, complété par la prise en compte de différentes modalités d'interaction, définies par [Vernier & Nigay 00], ce modèle a permis d'enrichir le ***Modèle de l'Interaction Instrumentale***, défini par [Beaudouin-Lafon 00a]. Enfin, il a été le cadre de conception d'au moins deux systèmes mixtes : le système ***ORBIS***, sur lequel peut s'exécuter des applications impliquant la manipulation de médias (photos, musiques et vidéos) ; et le système ***Roam***, un système mobile d'enregistrement d'images et de sons.

Les modèles ***ASUR*** et ***MIM*** ont été évalués notamment par [Bortolaso & al. 09], qui a mis en place une méthode, le ***Focus-Group Instrumenté*** (***FGI***), consistant à **associer un modèle d'interaction** avec **une méthode informelle** spécifique, le ***Focus Group*** [Kontio & al. 04]. Le but de cette approche est d'aider les concepteurs à bien évaluer les attentes et les besoins des utilisateurs et à choisir le modèle d'interaction le plus pertinent, avant d'élaborer le système interactif.
Une méthode informelle définit un cadre d'études pour un certain nombre de participants aux compétences et besoins divers, dans le but, d'une part, d'**évaluer et de perfectionner les dimensions de l'interaction considérée**, et, d'autre part, de **préciser les besoins des**

utilisateurs. Le ***Focus-Group*** [Kontio & al. 04] regroupe, sous la tutelle d'un médiateur, plusieurs participants aux expériences et compétences différentes. Il existe bien sûr d'autres méthodes informelles, telles que le ***Brainstorming*** [Wilson 06], technique regroupant plusieurs participants et permettant la génération d'un maximum d'idées, « à la volée ». Dans les travaux de [Bortolaso & al. 09], le modèle d'interaction, ***ASUR*** ou ***MIM***, associé au ***Focus-Group***, permet de cadrer et de structurer le dialogue et d'évaluer la faisabilité des idées émises par les participants, en les replaçant dans un contexte d'interaction précis.

D'autres modèles d'interaction permettant de concevoir des systèmes mixtes existent. Par exemple, **les modèles *TAC* (*Token And Constraints*)** [Calvillo-Gámez & al. 03, Shaer & al. 04] et ***MCRpd* (*Model Control Representation (physical and digital)*)** [Ullmer & Ishii 00], tous deux destinés à la conception d'**interfaces tangibles**.

1.4. Interface Homme-Machine

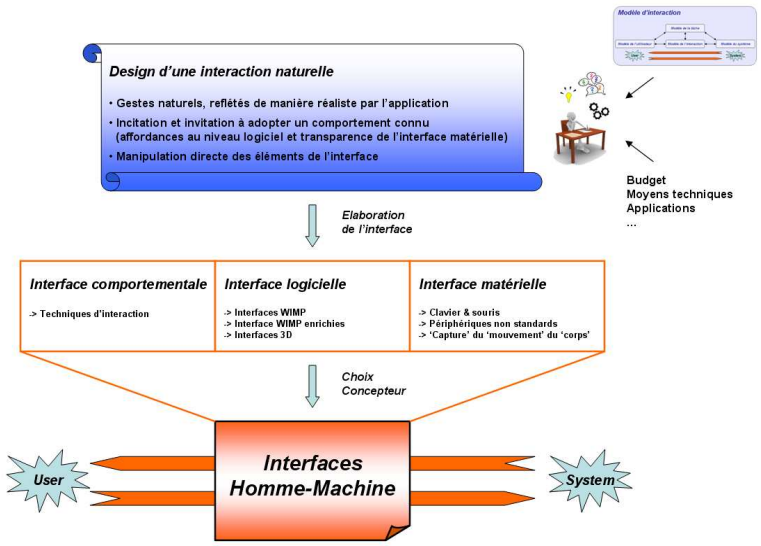


Figure 1.13. : Interface Homme-Machine

L'étude des interfaces constitue une étape majeure pour comprendre comment se déroule le processus d'interaction scénarisé entre le système et l'utilisateur. Il est nécessaire de bien assimiler ce concept avant de développer l'interface d'un système interactif donné.

L'**interface** entre l'utilisateur et système, appelée **interface homme-machine** ou **utilisateur-système**, regroupe l'ensemble des **techniques comportementales** et des **outils logiciels et matériels**, permettant les **communications** entre les deux partis, communications pilotées par le scénario de l'application. En effet, c'est grâce à l'interface que la gestuelle de l'utilisateur sera acquise et que la réponse du système sera restituée. De plus, elle constitue pour l'utilisateur un moyen de **contrôler** l'application s'exécutant sur le système. Enfin, l'interface, située entre le système et l'utilisateur, définit le **cadre comportemental, logiciel et matériel** du processus d'interaction scénarisé et reflète l'image du système et de l'application à l'utilisateur (voir Fig. 1.13.).

L'élaboration d'une interface dépend de nombreux paramètres : du modèle d'interaction établi, du type d'applications qui seront développées pour le système, du budget, des moyens techniques mis en œuvre, etc. Lors de cette élaboration, le principal objectif du concepteur est de rendre l'interaction du système avec l'utilisateur **la plus naturelle possible**. Pour cela, il doit faire en sorte que cette interface **disparaisse sur le plan cognitif** de l'utilisateur, qu'elle devienne **transparente**. Pour cela, le concepteur dispose de trois moyens :

1. Plus l'interaction impliquera que l'utilisateur adopte des gestuelles **intuitives**, **apprises**, **empruntées à la vie de tous les jours**, plus cette interaction sera naturelle. Cela d'autant plus si la réaction du système liée à la gestuelle de l'utilisateur est fortement similaire à celle que l'utilisateur prévoit et attend. Dans un jeu vidéo par exemple, si sa gestuelle est similaire à celui d'une situation réelle et qu'il est parfaitement reproduit par l'application, le joueur sera totalement impliqué et immergé dans l'action.
2. L'interface doit **inciter** et **inviter** l'utilisateur à **adopter une gestuelle particulière**, par le biais d'une forme particulière, une couleur, un signe, etc.
   Au niveau logiciel, un élément informatique peut mettre en œuvre une affordance spécifique [Gibson 77], voire représenter virtuellement un objet de la vie courante, se comportant et réagissant comme il le ferait dans la vie courante, suite à une interaction avec l'utilisateur.
   Au niveau matériel, plus l'interface sera proche d'un objet connu, plus l'utilisateur la manipulera naturellement et facilement, sans apprentissage préalable, en fonction de ses habitudes et de ses connaissances [Norman 88]. Cette remarque est particulièrement pertinente si le paradigme d'interaction adopté est celui de la virtualité augmentée (Section 1.2.2.3.). L'interface sera d'autant plus transparente, s'il existe une relation directe entre l'interface matérielle et l'élément virtuel avec lequel elle est liée, ce qui est très souvent le cas dans les jeux vidéo par exemple. Enfin, le concepteur pourra opter pour la disparition totale de l'interface matérielle (notion d'**interface non intrusive**), l'utilisateur interagissant alors uniquement par le biais de son corps.
3. Une interaction naturelle se traduit enfin par **une manipulation directe** des éléments de la scène, comme une personne le ferait dans la vie de tous les jours. L'image de l'application renvoyée par l'interface doit intégrer les différents éléments avec lesquels l'utilisateur interagit, et doit pouvoir être explorée directement par ce dernier. Encore une fois, cette façon d'interagir est très employée dans le domaine du jeu vidéo.

Il est bien évident que les interfaces dites standards ne poussent par l'utilisateur à adopter une gestuelle naturelle. C'est pourquoi de nouvelles interfaces ont été développées. [Turk 00] introduit une classification de ces interfaces. Même si cette classification n'est pas forcément très claire, la dernière classe, *Perceptual User Interface* (PUI), ou, plus communément aujourd'hui, *Natural User Interface* (NUI), décrivent des interfaces permettant à l'utilisateur d'interagir, naturellement et de manière transparente et non invasive, avec le système, de la même façon qu'il le ferait avec une autre personne ou avec l'environnement physique qui l'entoure. Aussi bien l'utilisateur que le système peut interagir par le biais de plusieurs canaux de communication. Le système peut raisonner à partir de ce qu'il perçoit et interpréter les gestuelles, explicites et implicites, de l'utilisateur. Ces réactions seront alors adaptées aux gestuelles de l'utilisateur et pourront impliquer l'usage de multimédias. L'utilisateur, quant à lui, apprend rapidement, par le biais d'essais/erreurs, et intuitivement comment utiliser l'interface. Comme il n'existe plus de métaphore, comme celle du 'Bureau', entre l'utilisateur

et le système, l'utilisateur s'engage véritablement dans l'action et manipule directement le contenu de la scène dans laquelle il est immergé. Les interfaces basées uniquement sur l'observation visuelle de la scène, *Vision-based Interfaces*, sont typiquement des exemples de NUI. Les NUI sont associées à des paradigmes d'interaction ***Post-WIMP***, tels que l'interaction en environnement mixte, l'interaction multimodale, etc.

Dans l'optique de rendre l'interaction avec l'utilisateur la plus naturelle possible, nous pensons que les concepteurs du système et de l'application doivent étudier et élaborer les **trois aspects** majeurs de l'**interface** :

- **L'interface comportementale** (Section 1.4.1.). Cette interface constitue l'ensemble des techniques d'interaction par le biais desquelles l'utilisateur interagit avec le système. La pertinence d'une technique d'interaction peut être évaluée avant son implémentation dans une interface.
- **L'interface logicielle** (Section 1.4.2.). Nous donnons ici une vision succincte des différentes interfaces logicielles qu'il est possible d'adopter, d'outils semi-transparents pour améliorer et enrichir l'interaction par le biais d'interfaces ***WIMP***, aux interfaces immergeant l'utilisateur dans un environnement en 3D.
- **L'interface matérielle** (Section 1.4.3.). Depuis la souris et le clavier, de nombreux périphériques non standards sont apparus. Il est possible de classer ces différents périphériques et d'évaluer de la pertinence d'un périphérique donné pour une tâche spécifique. La majeure partie des périphériques existants est dédiée à la capture des mouvements du corps de l'utilisateur. Nous traiterons particulièrement de ces derniers, dans la perspective des travaux présentés dans cette thèse.

La Section 1.4.4. positionnera les travaux de l'équipe ***ImagIN*** par rapport aux concepts et aux états de l'art exposés au cours des sections suivantes

1.4.1. Interface comportementale

L'**interface comportementale** regroupe l'ensemble des techniques d'interaction qui permettent à l'utilisateur de communiquer avec le système. Une **technique d'interaction** (voir Fig. 1.14.) permet à l'utilisateur d'**accomplir une tâche** spécifique en se servant d'une **interface matérielle**, un périphérique, pour agir sur une **interface logicielle**. L'ensemble des techniques d'interaction implémentées dans un système interactif donné définit donc le vocabulaire de gestuelles compréhensibles par ce dernier.
Une technique d'interaction est élaborée ou choisie par le concepteur, de manière à être particulièrement cohérente avec le modèle d'interaction auquel obéit l'élaboration du système, qui peut définir les différentes contraintes au niveau de l'utilisateur, ainsi que les différents canaux de communication considérés entre le système et l'utilisateur.

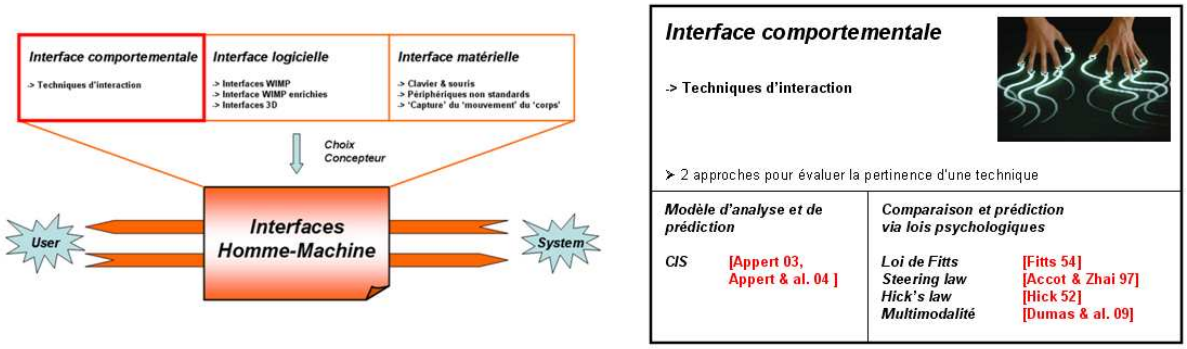


Figure 1.14. : Interface comportementale

L'efficacité et la pertinence d'une technique d'interaction peut être **évaluée de deux manières**. Il est possible d'une part de **modéliser une technique d'interaction au sein d'un espace de caractérisation**, pour l'analyser et la comparer à d'autres techniques. Mais, certaines **lois**, empruntées au domaine de la **psychologie**, sont capables de **prédire les performances** de certaines techniques d'interaction dédiées à des tâches spécifiques. Ainsi, les performances observées d'une de ces techniques d'interaction peuvent être comparées à celles prévues par une loi psychologique associée.

Une technique d'interaction peut être évaluée par le biais d'un modèle, comme le modèle ***CIS*** (***Complexity of Interaction Sequences***) [Appert 03, Appert & al. 04], avant de la considérer dans une interface donnée. Ce modèle permet de décrire et de combiner des techniques d'interaction, en prenant en compte les capacités sensori-motrices de l'utilisateur, dans le but de les analyser et de prédire leur cohérence et leur efficacité, dans le cadre d'un contexte d'interaction donnée. En effet, d'après ces travaux, deux techniques d'interaction ne peuvent être effectivement comparées que si elles le sont dans un contexte d'interaction particulier.
Mais les performances d'une technique d'interaction, dans le cadre de certaines tâches particulières, peuvent également être comparées à celle calculées par le biais de lois spécifiques et souvent empiriques, empruntées au domaine de la psychologie. Par exemple, la très connue ***loi de Fitts*** [Fitts 54] permet de calculer le temps du mouvement nécessaire qu'effectuera l'utilisateur pour atteindre et pointer une cible. Cette loi est applicable à un très grand nombre de techniques de pointages, et de nombreux travaux [Guiard & al. 01, McGuffin & Balakrishnan 02, Blanch & al. 04] ont tenté d'élaborer des techniques d'interaction qui battrait celle-ci. Il existe d'autres lois permettant d'évaluer une technique d'interaction dédiée à une tâche donnée : la ***Steering Law*** [Accot & Zhai 97], la loi de direction[1], pour évaluer le déplacement d'un curseur le long d'une direction contrainte ; la ***loi de Hick*** [Hick 52], pour évaluer le temps mis par l'utilisateur pour déterminer la réponse d'une cible parmi un choix de réponses répertoriées connues (cette loi est utilisée pour évaluer le temps de recherche visuelle d'une cible) ; [Dumas & al. 09] répertorie plusieurs lois utilisées pour évaluer des techniques d'interaction multimodale… Il existe un grand nombre de lois, mais celles-ci ne sont dédiées qu'à une seule tâche particulière, sans oublier le fait qu'il n'existe pas de lois pour évaluer un grand nombre de techniques d'interaction.

[1] Il existe d'autres traductions possibles : la loi du mouvement canalisé, la loi de tâche trajectorielle de navigation, la loi de trajectoire…

1.4.2. Interface logicielle

L'interface logicielle traduit la **surface de l'application** et regroupe **l'ensemble des outils logiciels** permettant **les communications entre l'utilisateur et le système** (voir Fig. 1.15.). Ces outils peuvent donner à l'utilisateur la possibilité de sélectionner une commande, de naviguer au sein d'un environnement informatif, d'organiser des données, etc. L'interface logicielle fixe **les frontières de l'environnement numérique**, définie par l'application et restituée à l'utilisateur, et constitue donc le **cadre délimitant les gestuelles** de ce dernier.

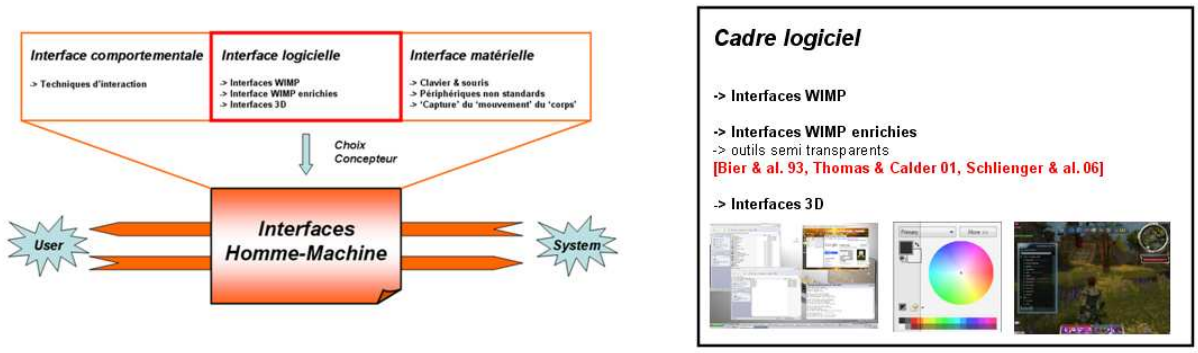


Figure 1.15. : Interface logicielle

A partir de l'avènement de l'interaction standard, les interfaces logicielles sont devenues **graphiques**. Les interfaces de type ***WIMP*** en sont l'exemple le plus populaire. Les applications, développées autour de ce paradigme, fournissent à l'utilisateur un ensemble d'outils tels que des **menus déroulants**, des **fenêtres**, des **boites de dialogue**, etc.
Lorsque les interfaces non standards et les nouveaux paradigmes d'interaction sont apparus, de nouveaux outils logiciels, appelés **outils semi-transparents**, ont été pensés, adaptés aussi bien aux interfaces ***WIMP*** qu'aux interfaces ***Post-WIMP***. L'objectif de la conception de ces nouveaux outils est toujours de rendre l'interaction avec l'utilisateur la plus naturelle possible, en respectant véritablement les règles définies par le **paradigme de la manipulation directe** (Section 1.3.2.).

Les 2 outils les plus exposés sont la ***Toolglass*** et la ***Magic Lens***, définis dans [Bier & al. 93] et parfaitement expliqués dans [Baudel 95]. Ces deux outils se révèlent particulièrement puissants lorsque les applications les utilisant sont développées autour du paradigme d'interaction bimanuelle. Plusieurs autres outils semi-transparents sont présentés dans [Baudel 95].

La ***Toolglass*** est une couche logicielle, transparente et composée d'une sélection d'outils semi-transparents, qui peut s'intercaler entre le curseur et les éléments informatiques définis par l'application (voir Fig. 1.16.). Cette opération est facilement réalisable si l'utilisateur se sert de ses deux mains pour contrôler la ***Toolglass*** et le curseur. Ce dernier sélectionne, sur la ***Toolglass***, un outil spécifique, qui exécutera l'opération à laquelle il est lié, sur le ou les éléments de l'application qui se trouvent en dessous. Ces outils peuvent permettre de modifier les propriétés graphiques (couleur, police, épaisseur des traits, etc.) et les propriétés

géométriques (position, rotation, déformation) des éléments de l'application, et d'éditer ces derniers (création, suppression, sélection, organisation, etc.).

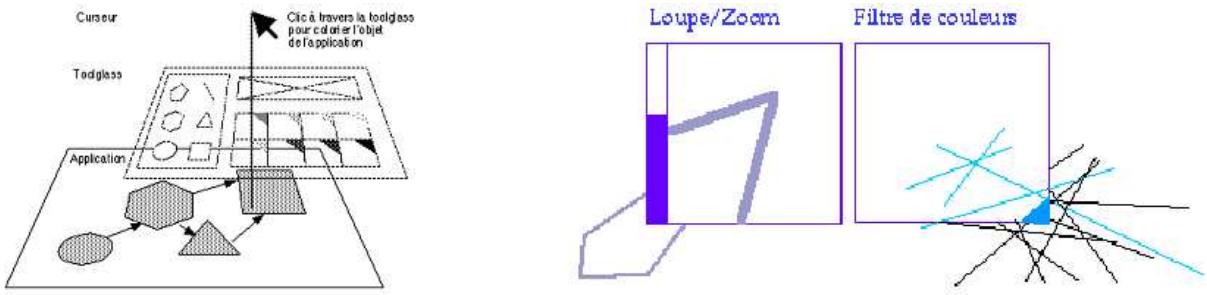


Figure 1.16. : ***Toolglass*** (à gauche) et ***Magic Lens*** (à droite), exemples et figures tirés de [Baudel 95]

La ***Magic Lens*** est un outil semi-transparent qui peut complémenter l'utilisation d'une ***Toolglass***. Cet outil peut être compris sur la ***Toolglass*** et agit comme un filtre vis-à-vis des éléments de l'application au dessus desquels il se trouve (voir Fig. 1.16.). Par le biais de ce filtre, la représentation graphique des éléments, restituée à l'utilisateur, est modifiée localement. Ce filtre peut être un zoom, un filtre de couleur, etc.
Ces deux outils obéissent indéniablement aux principes définis par le paradigme de la manipulation directe. Cependant, au niveau de l'utilisateur, ils ne peuvent pas toujours être évidents à manipuler (coordination des mains nécessaires) et peuvent présenter un manque de lisibilité au niveau des superpositions des couches. Le concepteur, quant à lui, pourra se heurter à des problèmes d'implémentation.
Les outils semi-transparents peuvent être considérés comme un paradigme d'interaction [Dragicevic 04] et comme un modèle d'interaction [Baudel 95]. Dans [Beaudouin-Lafon 00a], les outils semi-transparents sont plus perçus comme une nouvelle technique d'interaction, technique compatible avec le ***Modèle de l'Interaction Instrumentale*** [Beaudouin-Lafon 00a] et utilisée dans l'application ***CPN2000/CPN Tools*** [Beaudouin-Lafon & al. 01].

Il existe bien sûr d'autres outils logiciels, enrichissant l'interaction avec l'utilisateur. Il est commun, par exemple, d'emprunter des **techniques d'animation** (distorsion des objets, interpolation des propriétés d'un objet au cours d'une transition d'état, transitions visuelles spécifiques à l'objet manipulé, etc.), pour augmenter l'illusion de manipulation directe des éléments de l'application [Thomas & Calder 01, Schlienger & al. 06]. Enfin, les **interfaces logicielles 3D**, largement adoptées, et depuis longtemps ! dans le cadre des jeux vidéo, sont de plus en plus utilisées (paradigme d'interaction en environnement mixte, voir Section 1.2.2.3.).

1.4.3. Interface matérielle

Contrairement à l'interface logicielle qui représente l'interface avec l'utilisateur, interne au système, l'**interface matérielle** constitue l'interface **externe**, ensemble de **dispositifs**, ou de **périphériques**, **matériels**. Il est d'usage de distinguer l'**interface d'acquisition**, qui observe et capte les informations provenant de l'environnement extérieur au système, et l'**interface de restitution**, qui transmet à l'environnement extérieur la réponse du système (voir Fig. 1.17.).

Aujourd'hui, les interfaces matérielles tendent à devenir **invisibles**, observant uniquement l'environnement extérieur, et **non-intrusives** vis-à-vis de l'utilisateur. Autrement dit, l'utilisateur ne manipule plus physiquement un dispositif mais **interagit directement avec son corps**, par le biais de différentes **modalités**.

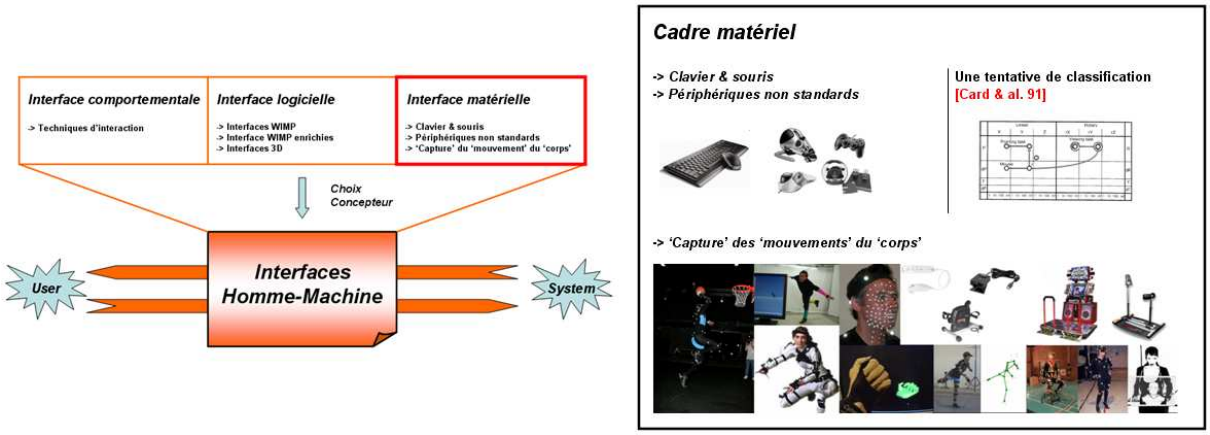


Figure 1.17. : Interface matérielle

Il existe de très nombreux périphériques matériels. Les premiers et les plus populaires, largement utilisés dans le cadre d'interface ***WIMP***, sont bien sûr la **souris**, utilisée pour pointer, sélectionner et tracer, et le **clavier**, utilisé pour écrire un texte, rentrer une commande et complémenter les actions de la souris.
Puis de **nouveaux périphériques** et de nouvelles façons d'interagir ont vu jour : de la souris 3D à la borne interactive, en passant par tous les dispositifs d'observation (caméra, webcam, etc.) et permettant une interaction sur plusieurs modalités (microphones, hauts parleurs, périphérique à retour haptique, simulation de goût, etc.).

Devant l'infinité de périphériques qui peut exister, il n'existe pas de classification standard et formalisée, capable de répertorier l'ensemble de ces dispositifs. Il est toutefois commun de citer [Card & al. 91], qui introduisit pour la première fois une classification des périphériques d'entrée en fonction de leur conception matérielle.

Dans l'espace de classification et d'analyse défini dans ces travaux, chaque point, un opérateur, représente un élément physique, constituant une partie ou la totalité du périphérique, sur lequel l'utilisateur, ou un autre opérateur, peut agir. Cet opérateur est positionné dans cet espace en fonction **(1)** du type d'action que l'utilisateur ou l'opérateur doit effectuer (translation et/ou rotations, forces et/ou couples), **(2)** du type de correspondance (directe/absolu ou indirecte/relative) entre cette action et ce qu'elle contrôle, et **(3)** du nombre d'états que peut prendre cet opérateur (2 états pour un bouton et une infinité pour une souris). Un opérateur peut être fusionné, associé ou connecté avec un autre opérateur. Une fusion (*merge composition*) regroupe plusieurs opérateurs en un même élément physique du périphérique. Une association (*layout composition*) décrit la disposition physique des éléments du périphérique. Une connexion (*connect composition*) traduit l'influence d'un opérateur sur un autre opérateur.
Une fois représenté, un périphérique peut être mis en relation avec l'environnement dans lequel il évolue et avec la tâche qu'il doit permettre d'accomplir. La pertinence d'un

périphérique pour une tâche donnée peut donc être étudiée. La Figure 1.18. représente l'espace de classification et d'analyse défini dans ces travaux. Le périphérique 'souris' y est caractérisée et ce dernier est connecté aux tâches de pointage et de visualisation par le biais d'une caméra virtuelle. Ce schéma montre que ce périphérique est adéquat pour ces deux tâches mais qu'il permet plus grande précision et efficacité pour la tâche de pointage.

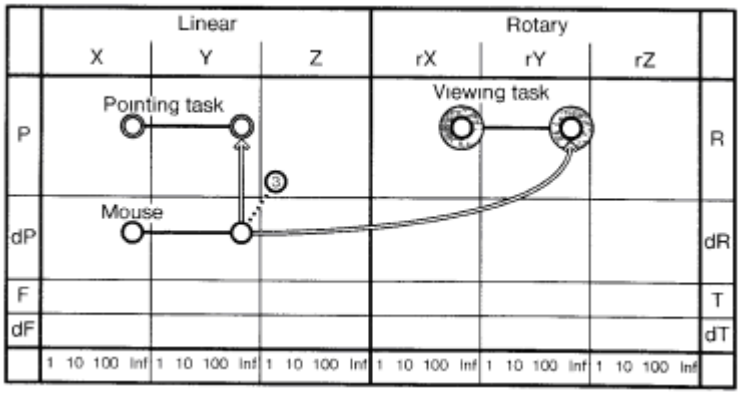


Figure 1.18. : Analyse de la souris vis-à-vis des tâches de pointage et de visualisation [Card & al. 91]

Cette classification est très efficace pour analyser et comparer les périphériques d'entrée standards et moins standards. En revanche, cette approche peut se révéler laborieuse voire impossible dans le cas de périphériques obéissant aux paradigmes ***Post-WIMP***. En effet, l'utilisateur peut dorénavant interagir avec un système par le biais d'objets courants, de gants électroniques, voire même par le biais de parties corporelles ! qui peuvent se révéler complexes à modéliser.

La plus grande majorité des périphériques d'entrée sont dédiés à la **capture** des **mouvements** du **corps** de l'utilisateur. Ici, les termes **'capture'**, **'mouvements'** et **'corps'** sont à prendre au sens large et nécessitent d'être précisés. En effet, même l'interaction par le biais d'une souris et d'un clavier peut être considérée comme une capture des mouvements du corps de l'utilisateur. Ce dernier interagit avec ses deux mains, obéissant au paradigme d'interaction bimanuelle, par le biais de ses mouvements coordonnés, obéissant au paradigme d'interaction gestuelle, et ses mouvements influencent le système et l'application, qui y répondent alors.
Nous parlerons plus largement de la 'capture' des 'mouvements' du 'corps' au cours du Chapitre 4.

1.5. Positionnement des travaux de l'équipe ImagIN

Nous positionnons les travaux de l'équipe ***ImagIN*** relativement aux différents concepts abordés au cours de la Section 1.

L'expression « applications scénarisées interactive » implique la mise en œuvre du processus d'interaction scénarisé par un type de système bien particulier. Par **« processus d'interaction scénarisé »**, nous faisons l'hypothèse que le processus d'interaction est piloté par le scénario de l'application, en considérant qu'une application est toujours scénarisée. Le processus

d'interaction scénarisé n'en est pas moins mis en œuvre et supporté par le système. Nous proposons, dans ces travaux, de développer un tel système.
En amont de la conception d'un système, nous jugeons pertinent d'étudier les différents aspects englobés dans le processus d'interaction scénarisé. Cette étude nous permettra de comprendre et d'implémenter au sein du système interactif les différents facettes, et ce, à tous les niveaux, que peut présenter ce processus. Ces différentes facettes sont le ou les paradigmes d'interaction à adopter (Section 1.5.1.), le modèle d'interaction qui en découle et à partir duquel il est possible d'établir un cahier des charges orientant les concepteurs (Section 1.5.2.), et les différentes interfaces (Section 1.5.3.). L'étude de ces concepts constitue une base théorique et pratique solide, qui permettra aux concepteurs du système d'élaborer, tant sur le plan matériel que logiciel, **un système interactif final optimal** et **un meilleur processus d'interaction scénarisé pour l'utilisateur** (voir Fig. 1.19.).
Sur la base de nos positionnements au sein de l'équipe, nous pensons que le domaine du jeu vidéo est un excellent terrain d'expérimentations et de développements. D'autres systèmes sont développés au sein du laboratoire, relativement à diverses applications scénarisées : simulation de trafic urbain, enseignement à distance et classe virtuelle, etc.

Dans ces travaux, nous avons choisi de ne pas aborder plusieurs aspects de l'interaction scénarisée entre l'utilisateur et le système, comme, par exemple, les modélisations relatives aux émotions de l'utilisateur ; les mécanismes cognitifs prenant place lors d'une erreur de la part de l'utilisateur au cours de l'interaction ; les mécanismes sociaux et collaboratifs mis en jeu lors d'une interaction entre plusieurs utilisateurs et/ou plusieurs agents, etc. Ces aspects, bien qu'importants dans le cadre de l'***IHM***, sont des sujets d'étude à part entière et mettent en jeu l'étude de mécanismes psychologiques au niveau de l'être humain, qui sortent du cadre de cette thèse.

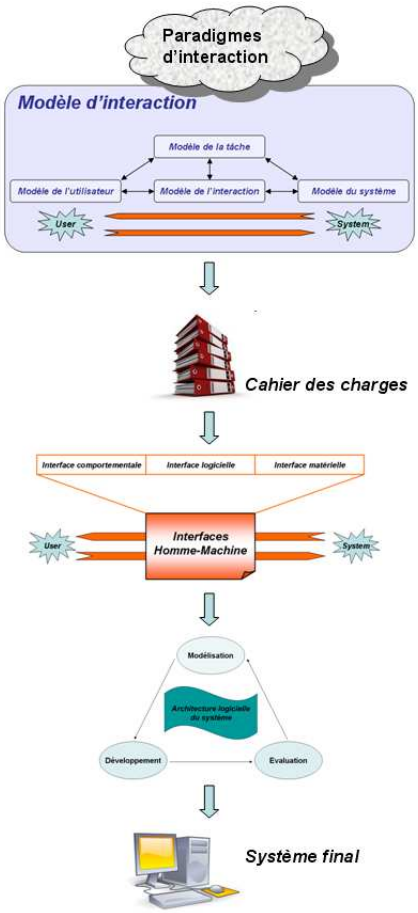


Figure 1.19. : Interaction Homme Machine – Conception de systèmes interactifs

1.5.1. Paradigme d'interaction

Nous pensons que la notion de paradigme d'interaction est la première facette du processus d'interaction scénarisé à étudier. La bonne définition de cette vision globale de l'interactivité entre l'utilisateur et le système constituera les premières directives globales de conception du système interactif.

La grande majorité des applications développées dans le cadre de l'équipe ***ImagIN*** utilise l'ordinateur comme un outil, comme la majeure partie des systèmes interactifs. Cependant, dans le cadre du projet d'enseignement à distance ***FOAD*** (***Formation Ouverte et A Distance***) par exemple, le système est vu également aussi bien comme un intermédiaire permettant les communications entre utilisateurs, et comme un partenaire, qui peut se substituer aux moyens de communication de l'utilisateur (projets ***Autistic*** et ***RobAutistic***, destinés à des enfants autistes). En effet, dans le cadre de ce projet, les utilisateurs, professeur et élèves, sont représentés au sein d'un environnement visuel 3D par des avatars (représentation 3D). Non seulement, les utilisateurs communiquent les uns avec les autres par le biais de cet espace virtuel (le système comme un intermédiaire), mais en plus, la gestuelle d'un avatar est adaptée à la gestuelle capturée de l'utilisateur qu'il représente (le système comme un partenaire).

Un système développé au sein l'équipe ***ImagIN*** peut tout aussi bien obéir aux paradigmes de l'interaction standard qu'aux nouveaux paradigmes ***Post-WIMP***. Nous ferons toutefois

attention à préciser que le processus d'interaction décrit par les paradigmes adoptés est scénarisé.
Cependant, le système doit interpréter et répondre, en accord avec le scénario de l'application, aux gestuelles implicites et explicites de l'utilisateur. De plus, le processus d'interaction scénarisé entre le système et l'utilisateur est un véritable dialogue interactif, implicatif, constant et temps réel. Ces caractéristiques sont donc plus en adéquation avec les paradigmes d'interaction ***Post-WIMP***.
Cette orientation permet également de positionner les travaux de l'équipe ***ImagIN*** par rapport au paradigme de la manipulation directe, puisque les paradigmes ***Post-WIMP*** ont été élaborés dans le but de mieux obéir à ce méta-paradigme.
Pour finir, dans le cadre du développement des applications, l'ensemble des paradigmes ***Post-WIMP*** peut être adopté. Cependant, l'ensemble des projets étudiés dans le cadre de l'équipe ***ImagIN*** ont tendance à adopter les paradigmes d'interaction gestuelle (nous employons le terme de ***« gestuelle »*** pour qualifier une interaction de la part de l'utilisateur, voir Chapitre 4.), bimanuelle et en environnement virtuel (capture des gestuelles corporelles globales de l'utilisateur et jeux vidéo).

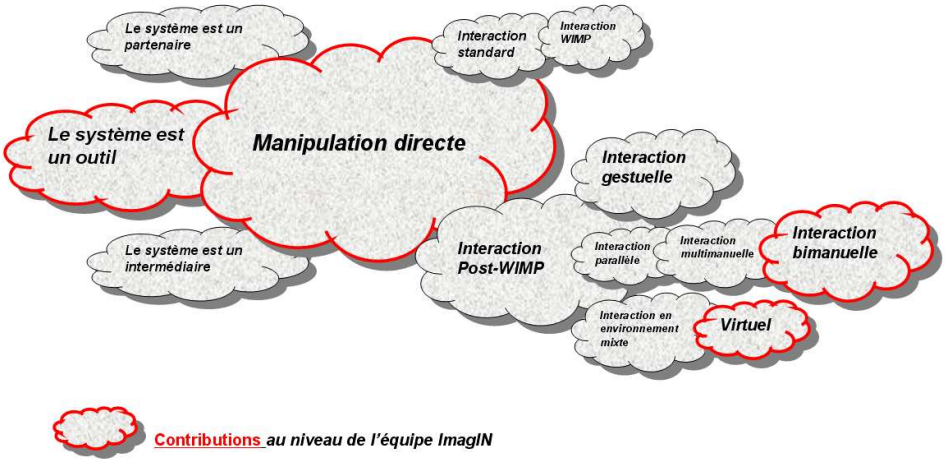


Figure 1.20. : Paradigme d'interaction : Positionnement des travaux de l'équipe ***ImagIN***

La Figure 1.20. illustre notre positionnement par rapport à la notion de paradigme d'interaction, relativement aux travaux de l'équipe ***ImagIN***.

1.5.2. Modèle d'interaction

Dans le cadre de l'équipe ***ImagIN***, l'accent est particulièrement mis sur la conception d'architectures logicielles et matérielles. Du modèle d'interaction ne seront retenus que les aspects qui permettront l'élaboration aussi bien logicielle que matérielle du système. C'est pourquoi, les travaux de l'équipe ***ImagIN*** se positionnent plutôt autour de la modélisation du système et de l'interaction entre l'utilisateur et le système, bien que tout autre modèle soit compatible (voir Fig. 1.21.).

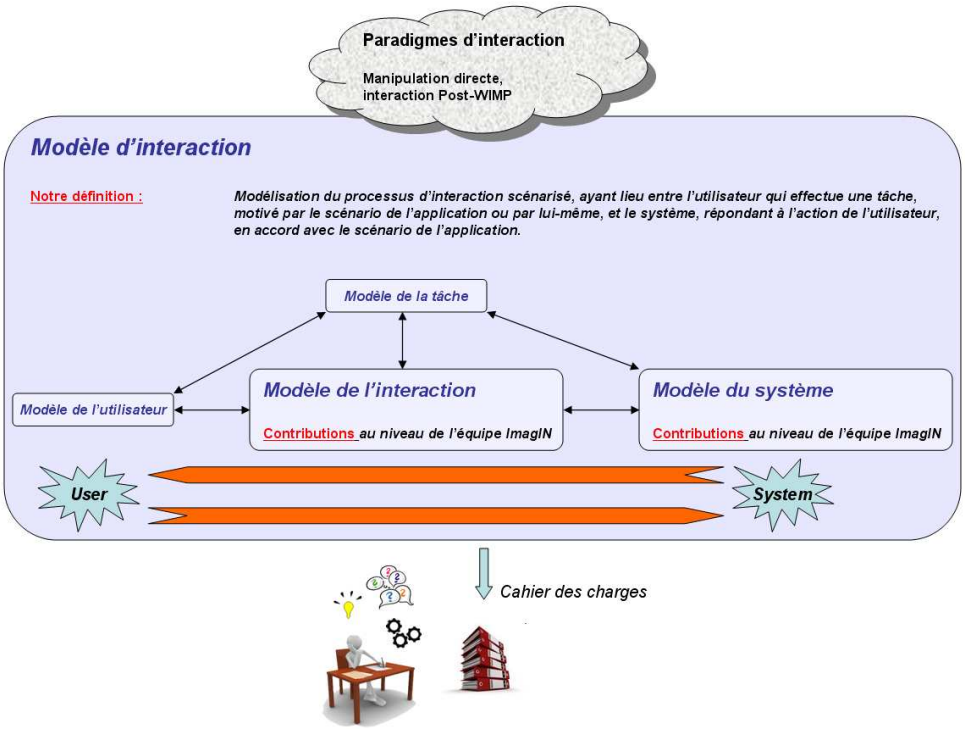


Figure 1.21. : Modèle d'interaction : Positionnement des travaux de l'équipe ***ImagIN***

En revanche, les travaux de l'équipe ***ImagIN*** visent exactement à développer des processus interactifs scénarisés en accord avec le modèle d'interaction relatif au paradigme de la manipulation directe : implication physique et directe de l'utilisateur dans l'action (**MDP1**) ; transparence (voire transparence maximale) de l'interface (**MDP1**) ; gestuelles simples et naturelles adoptées par l'utilisateur, en accord avec ses connaissances et suggérés par l'environnement interactif (**MDP2** et **MDP3**) ; immersion de l'utilisateur dans l'environnement interactif (**MDP4**) ; compréhension permanente de l'utilisateur de ce qu'il se passe au sein de l'environnement interactif (**MDP4**).

Pour finir, aucun paradigme d'interaction n'est particulièrement adopté *a priori* lors du développement de systèmes et d'applications dans le cadre de l'équipe ***ImagIN***, avec toutefois une orientation notable vers les paradigmes d'interaction ***Post-WIMP***. Encore une fois, il ne s'agit pas d'étudier à proprement parler le phénomène interactif mais d'instaurer les règles qui guideront la conception des architectures logicielle et matérielle du système. Ainsi, tous les modèles d'interaction présentés, qui focalisent leur modélisation sur les aspects architecturaux du système, pourraient être potentiellement étudiés et applicables. Quoiqu'il en soit, l'ensemble des modèles d'interaction existants assurent un cadre d'étude essentiel et nécessaire.

1.5.3. Interfaces

Le concepteur du système doit donc effectuer trois choix décisifs lors de l'élaboration de l'interface homme-machine, concernant l'aspect comportemental (techniques d'interaction), l'aspect logiciel (interface logicielle) et l'aspect matériel (interface matérielle) de l'interface. Les caractéristiques d'une bonne interface sont à rapprocher étroitement du paradigme de la

manipulation directe, paradigme traduisant une interaction parfaitement naturelle entre l'utilisateur et le système : gestuelles naturelles, suggérées par l'interface logicielle (**MDP2**) ; transparence de l'interface matérielle qui doit être autant que possible non intrusive, tant sur le plan cognitif que physique, au niveau de l'utilisateur (**MDP1**) ; implication directe et physique de l'utilisateur dans l'interaction (**MDP1**).
De manière générale, au sein de l'équipe ***ImagIN***, nous tentons de nous diriger autant que possible vers la conception de NUI.

Concernant les techniques d'interaction, les lois empruntées au domaine de la psychologie semblent limitées et contraignantes. En revanche, l'évaluation des techniques d'interaction via le modèle ***CIS*** [Appert 03, Appert & al. 04] semblent tout à fait pertinents et adéquats. En effet, ce modèle est adapté à l'évaluation de techniques d'interaction en accord aussi bien avec le paradigme de l'interaction standard qu'avec les paradigmes d'interaction ***Post-WIMP***. Toutefois, dans le cadre des travaux de l'équipe ***ImagIN***, les interfaces sont élaborés de manière à ce que l'utilisateur interagisse par le biais de gestuelles naturelles, empruntées à la vie de tous les jours. Cette forme d'interaction étant la plus naturelle qui soit, l'évaluation des techniques d'interaction n'est pas forcément nécessaire et obligatoire.

L'interface logicielle est développée au même moment que l'application et les choix qui en découlent dépendent grandement du type d'application développée. Elle est également fonction du modèle d'interaction établi et donc du ou des paradigmes d'interaction adoptés. Dans le cadre des travaux de l'équipe ***ImagIN***, aucun paradigme d'interaction n'est *a priori* préféré. Quoiqu'il en soit, il est évident que les moyens de développer une interface logicielle sont riches et nombreux. Le développement d'une interface logicielle mettant en jeu un environnement en trois dimensions semble sans limites et contraintes particulières.

Le constat est le même pour l'interface matérielle. Cette dernière sera choisie ou conçue en fonction du type d'application développée. Cependant, dans le contexte de l'équipe ***ImagIN***, la capture des gestuelles (le regard, le corps, la main, etc.) de l'utilisateur est particulièrement étudiée. Cette interface matérielle doit également dénotée une transparence importante, voire maximale. Cela dit, toutes les solutions matérielles sont envisageables *a priori*.

2. Exécution adaptative d'applications

Nous poursuivons notre démarche de **conception**, concernant les systèmes étudiés au sein de l'équipe ***ImagIN***. Notre objectif est donc de développer un système s'adaptant à l'activité au sein de la scène observée et particulièrement aux gestuelles de l'utilisateur. Ainsi, un véritable dialogue interactif et intelligent s'instaure entre l'utilisateur et le système. Nous désirons que ce dialogue ne soit pas subi par l'utilisateur et nous pensons que la mise en place d'un processus d'**adaptativité** au cours de l'interactivité est le meilleur moyen de le faire. Cela d'autant plus dans un contexte de jeux vidéo, l'adaptativité permettant alors de rendre beaucoup plus intéressante et divertissante l'expérience du joueur.

A l'instar du processus interactif, le processus d'adaptativité est **scénarisé**. Ce processus est amorcé à partir de la reconnaissance d'événements répertoriés au sein de la scène, particulièrement ceux relatifs aux **gestuelles interprétées** de l'utilisateur. Il se matérialise sous la forme d'une **double réponse**, vis-à-vis de l'utilisateur et vis-à-vis du système lui-même.

L'adaptativité au sein d'un système implique l'introduction d'une sensibilité au **contexte d'interaction**, dans lequel l'activité prend place. Le contexte d'interaction, une fois modélisé, permet de caractériser et de désambigüiser une activité observée. Il était donc nécessaire d'étudier ces différents concepts, de manière à élaborer notre système efficacement.

A l'instar de l'interactivité, il nous semble obligatoire d'étudier ce processus, avant de l'introduire au sein d'un système interactif. C'est pourquoi nous allons, dans la suite de ce chapitre, **définir la notion d'adaptativité** et **préciser les principales dimensions qui la caractérisent**. Nous positionnerons ensuite les travaux de l'équipe ***ImagIN***.

La Section 2.1. **définit la notion d'adaptativité**. Comme nous le verrons, l'adaptativité est une **forme d'adaptation**. Il nous a donc semblé pertinent, avant de proposer notre définition, de définir ce qu'est l'**adaptation pour un être humain**. Cela nous semble en effet primordial, dans le sens où nous considérons l'être humain comme un système parfait, qui s'adapte en permanence à toute situation.
Nous verrons que même chez l'être humain, le processus d'adaptation est **scénarisé**, et qu'il est étroitement lié à la notion de **contexte**. En comprenant comment l'être humain s'adapte à une situation donnée, une description du processus d'adaptation peut être établie pour les systèmes interactifs.
Il existe actuellement six formes d'adaptation (au cours de l'exécution de l'application) pour un système. L'adaptativité est une de ces formes d'adaptation, aujourd'hui vastement étudiée et réalisable. Elle est à différencier de la notion d'adaptabilité, avant d'être précisément définie.
La section se conclura avec une brève présentation des paradigmes d'interaction et de programmation propices au développement de systèmes adaptatifs.

La Section 2.2. énumère et caractérise les **dimensions de l'adaptativité**, qui nous semblent importantes, antérieurement à la conception du système.
Nous explorons tout d'abord le **« Pourquoi ? »** de l'adaptativité, que cela soit autour de l'utilisateur ou du système.
Puis, nous survolons brièvement le **« Quand ? »** et le **« Où ? »**, permettant la mise en œuvre des mécanismes d'adaptation à des instants spécifiques au cours de l'interactivité, et par le biais de composants internes ou externes au système.
Enfin, nous répondrons aux questions importantes **« A quoi ? »** et **« Quoi ? »**, mettant en relief la nécessité de la modélisation du contexte d'interaction au sein duquel l'activité prend place, et les différents aspects du système qui peuvent s'y adapter.

2.1. Définition de l'adaptativité d'un système

Avant d'introduire cette aptitude au sein d'un système, il nous semble important de définir ce qu'est précisément la **faculté d'adaptation** au niveau de l'**être humain**. En étudiant cette faculté, que nous considérons comme parfaite, nous serons plus à même de développer cette dernière au sein d'un système interactif (Section 2.1.1.). Pour l'être humain qui cherche à atteindre un objectif particulier, l'adaptation à une situation est supportée par un ensemble de connaissances qui permet d'élaborer :

> **(1)** un **scénario**, mettant en œuvre cette situation et se terminant lorsque l'objectif est atteint ;

(2) un **contexte**, englobant la situation et lui donnant une signification vis-à-vis du scénario élaboré.

C'est ainsi que l'être humain interprète une situation, lui permettant alors de choisir parmi l'ensemble des réactions possibles, celle qui est la plus adaptée à la situation, en accord avec son objectif. Nous voulons introduire ce processus au sein d'un système interactif dont le but est permettre à l'utilisateur d'**atteindre ses objectifs** tout en vivant une **expérience interactive intéressante**.
La faculté d'adaptation, telle qu'elle prend place au niveau de l'être humain, a été implémentée au sein d'un système à partir du moment où l'interaction standard a atteint ses limites. En se restreignant au domaine de l'adaptation au niveau de l'exécution de l'application, **six niveaux d'adaptation** peuvent être distingués pour un système. L'**adaptativité** du système, le **Niveau 2** de l'adaptation, est aujourd'hui une des formes d'adaptation les plus étudiées et réalisables (Section 2.1.2.).
L'**adaptabilité** et l'**adaptativité** d'un système sont deux formes d'adaptation que la littérature ne différencie pas tout le temps. Bien que ces deux formes d'adaptation soient effectivement proches, il est cependant clair qu'elles sont bien distinctes. Il est donc important de les différencier pour définir correctement l'adaptativité d'un système. C'est l'objet de la Section 2.1.3.
Enfin, une fois ces différences établies et la notion d'adaptativité définie, la Section 2.1.4. donnera un aperçu des nouveaux **paradigmes d'interaction et de programmation**, propices à la conception de systèmes adaptatifs.

2.1.1. Définition de la capacité d'adaptation

L'introduction de la faculté d'adaptation au sein d'un système est un domaine d'étude récent, qui en est à ses tous premiers pas. Il suffit de quelques minutes de recherche pour se rendre compte que la notion d'adaptation est complexe et qu'il y a autant de définitions de cette notion que de systèmes développés : système adaptable, système adaptatif, système adaptif, etc. Alors, **quelles sont les différences entre toutes ces déclinaisons ?** Pour caractériser ces différences, il nous semble important de **définir ce qu'est l'adaptation au niveau de l'être humain**, le considérant comme un système parfait, s'adaptant en permanence et en toutes circonstances.

L'adaptation est définie[1] comme étant l'**action d'adapter ou de s'adapter**. C'est un ensemble structuré de transformations, exécutées par une entité dans le but de répondre de manière stable et harmonieuse à de nouvelles situations. Ces transformations peuvent être **externes**, i.e. lorsqu'une entité adapte, elle agit sur son environnement extérieur, elle l'ajuste, l'accorde et l'arrange en fonction de ses objectifs. Ou ces transformations peuvent être **internes**, i.e. l'entité s'adaptant modifie sa propre structure (génétique, physique/physiologique, mentale, architecturelle, etc.) et ses propres gestuelles vis-à-vis d'une situation donnée. Elle réalise son adaptation vis-à-vis de cette situation. **Nous nous intéressons ici la capacité de s'adapter**, donc de manière interne, notre système n'agissant pas explicitement sur le monde extérieur.

Nous, l'être humain, ne répondons que très rarement par un simple réflexe à un événement donné, **l'humain s'adapte en permanence à toute situation qu'il perçoit**. Nous faisons cela

[1] Le Nouveau Petit Robert de la Langue française, 2000

tous les jours, à tout instant, en toutes circonstances. L'action humaine est flexible. Face à une situation qui se présente à nous, nous établissons un ensemble de réponses possibles et nous choisissons notre gestuelle, de manière à réagir efficacement ou, tout du moins, pour découvrir de nouvelles façons de répondre plus efficacement. Mieux, notre gestuelle s'améliore au fur et à mesure que nous adaptons à la situation.
La faculté d'adaptation est supportée par un ensemble de connaissances, sur notre passé, nos expériences, l'environnement extérieur… A partir de ces connaissances, nous construisons, d'une part, un **scénario**, plus ou moins élaboré, autour de la situation que nous percevons, et la mettant en œuvre. Ce scénario se termine lorsque nous avons atteint nos objectifs. D'autre part, nous englobons la situation au sein d'un **contexte**, lui attribuant alors une signification au sein de notre scénario. Nous lui déterminons donc un passé, un début, une évolution en fonction d'une gestuelle que nous pourrions adopter, une fin et un futur. **Nous l'interprétons**. Plusieurs gestuelles sont évidemment possibles, c'est pourquoi nous nous adaptons à la situation, en choisissant la meilleure gestuelle, en accord avec notre scénario, et donc nos objectifs.
Plus nous nous adaptons à une situation donnée, plus nos connaissances sont remodelées et reconstruites, mettant de côté celles qui ne servent à rien et ne considérant seulement que les plus pertinentes. Ce sont ces connaissances qui vont nous permettre de mieux percevoir et mieux interpréter une situation, affinant, chaque fois un peu plus et un peu mieux, le contexte qui entoure cette situation et le scénario qui la met en œuvre. Notre perception et notre interprétation, orientées par le scénario et surtout par le contexte de la situation courante, n'en seront que plus précises et rapides. Ainsi, plus nos connaissances seront précises, fiables et vraies, mieux nous nous adapterons à une situation donnée.

Dans ces travaux, c'est la faculté d'un système à s'adapter aux gestuelles de l'utilisateur qui nous intéresse. Il est donc question de l'aptitude du système à effectuer une série de transformations sur la scène virtuelle et sur son état interne, de manière adaptée vis-à-vis des gestuelles de l'utilisateur. Nous voulons, comme l'être humain, que le système identifie la situation qu'il perçoit, au sein du scénario de l'application et qu'il s'adapte à celle-ci, tout d'abord en la contextualisant, puis en choisissant parmi un ensemble de réponses possibles la meilleure réaction, en accord avec ce scénario. L'utilisateur cherche à atteindre les objectifs du scénario de l'application et le système s'adapte pour que celui-ci les atteigne en vivant une expérience interactive intéressante.

2.1.2. Adaptation du système à l'utilisateur

Il est à noter, en remarque préalable, que nous ne parlerons que de la faculté d'adaptation du système **au cours de l'exécution de l'application**. Dans ces travaux, nous ne nous intéresserons pas au processus d'adaptation au cours du développement, de la compilation ou du chargement du programme.

Dans le domaine de l'***Interaction Homme-Machine***, la capacité d'un système à s'adapter à l'utilisateur a été étudiée dès que le processus d'interaction global a été repensé et qu'il fut nécessaire de se détacher de la métaphore du 'Bureau' et de sortir de l'interaction standard [Dourish 04].

Très vite, il s'est avéré qu'un système interactif pouvait s'adapter de différentes façons à l'utilisateur. C'est pourquoi, aujourd'hui, il est possible de distinguer six formes, **six niveaux**, d'adaptation pour un système [Totterdell & Rautenbach 90, Thevenin 01] :

- **Niveau 0 : le système câblé**, *designed system*.
  Cette classe de systèmes regroupe la majorité des systèmes avec lesquels nous interagissons actuellement. Le système ne s'adapte pas, ses réactions sont définies une fois pour toute au niveau de sa conception. C'est l'utilisateur qui s'adapte, quitte à ne pas interagir de manière très naturelle…
- **Niveau 1 : le système adaptable**, *adaptable system*.
  L'utilisateur peut configurer le système, avant le processus d'interaction, en fonction de ses besoins et de ses objectifs, dans les limites du degré de personnalisation du système fixé par le concepteur.
- **Niveau 2 : le système adaptatif**, *adaptive system*.
  Le système perçoit et interprète la situation d'interaction courante, et réagit alors en conséquence. Ces mécanismes de perception, d'interprétation et d'action, vis-à-vis d'une situation clairement identifiée, sont fixes et définis au moment de la conception du système.
- **Niveau 3 : le système autorégulateur**, *self-regulating adaptive system*.
  Le système est doté de la capacité d'auto-évaluer, *a posteriori*, sa réaction vis-à-vis de l'utilisateur.
- **Niveau 4 : le système automédiateur**, *self-mediating system*.
  En plus d'être autorégulateur, le système est capable de planifier son adaptation et peut évaluer sa réaction, *a priori*, au cours de cette planification.
- **Niveau 5 : le système automodificateur**, *self-modifying adaptive system*.
  Ce type de système est capable d'apprendre de nouvelles situations et de nouvelles réactions à celles-ci.

Un système combinant plusieurs niveaux d'adaptation peut être appelé un **système à contrôle mixte**. Par exemple, dans un système à contrôle mixte de **Niveau 1** (système adaptable) et de **Niveau 2** (système adaptatif), le processus d'adaptation est à la fois mis en œuvre par l'utilisateur et le système lui-même.

Ce sont les systèmes de **Niveau 2** qui nous intéressent dans le cadre de ces travaux de thèse. Cette forme d'adaptation est aujourd'hui une des plus étudiées et réalisables.
Nous ne parlerons que brièvement de l'**adaptabilité** d'un système (**Niveau 1**), au cours de la prochaine section, dans un souci de précision et de différenciation vis-à-vis de la notion d'**adaptativité** (**Niveau 2**). Les autres formes d'adaptation (**Niveaux 3**, **4**, **5**), bien que vastement étudiées aujourd'hui, nous semblent encore des domaines de recherches plus récents et relativement éloignés de nos objectifs. C'est pourquoi elles ne rentrent pas dans le cadre des problématiques de ces travaux de thèse. Elles pourront cependant être traitées davantage dans les recherches à venir de l'équipe ***ImagIN***.

2.1.3. Adaptabilité vs Adaptativité

Avant de définir précisément la notion d'**adaptativité** au sein d'un système, il nous semble primordial de définir celle d'**adaptabilité**, de manière à pouvoir faire clairement la distinction entre ces deux concepts.

En effet, ces deux notions décrivent toutes deux la capacité d'un système à s'adapter. Toutefois, les définitions de ces notions sont bien distinctes. En résumé, le système peut s'adapter **avant l'interactivité avec l'utilisateur** (système **adaptable**, l'utilisateur personnalise l'interaction à venir par le biais du système) ou **pendant celle-ci** (système **adaptatif**, le système adapte ses réactions au cours de l'interactivité).
Cette différence n'est pas toujours clairement mise en évidence, c'est pourquoi la distinction est parfois difficile. De plus, ces deux notions sont parfois utilisées sans aucunement les différencier et, pour ne rien simplifier, ont les mêmes synonymes. Enfin, du fait que les termes techniques en français et en anglais sont souvent similaires, les traductions sont parfois un peu hasardeuses. Ainsi, le terme '*adaptativity*', traduction immédiate *a priori* du terme français 'adaptativité', n'existe pas, bien que certains dictionnaires en ligne le traduiront de la sorte. De même, '*adaptive system*' est traduit en français par 'système adaptatif', les termes 'adaptif' et 'adaptivité' n'existant *a priori* pas. Il est toutefois possible de rencontrer ces traductions directes, trop évidentes, dans la littérature.

L'**adaptabilité** d'un système (système **adaptable**) est la capacité de ce dernier à pouvoir être modifié facilement, **avant l'interactivité** avec l'utilisateur à proprement parler. Cette modification, qui pourra être également appelé personnalisation, paramétrage ou configuration, n'est pas automatisée et est effectuée par l'utilisateur du système ou une personne tierce, supervisant le système. Cette faculté est prévue lors de la conception du système et du développement de l'application, elle est donc scénarisée. Elle est introduite au sein du système en fonction des connaissances des concepteurs sur l'utilisateur, sur les tâches qu'il va devoir accomplir, sur l'environnement… sur le contexte d'interaction au sein duquel il va évoluer.
Cette capacité permet donc de donner à accès à différents paramètres, que l'utilisateur peut modifier lors d'une phase d'initialisation, et dont les variations se répercuteront sur les réactions du système au cours de l'interactivité. Chaque paramètre, accessible directement par l'utilisateur, constitue donc une alternative d'interaction, que ce dernier choisit en fonction de ses contraintes, de ses compétences & habitudes, de ses connaissances propres et de ses objectifs. Cette accessibilité est généralement définie et facilitée par un ensemble d'outils logiciels, mis à la disposition de l'utilisateur, qui n'a pas forcément de connaissances en programmation.
Traduction anglaise : *adaptability*, *adaptableness*, *adaptable system*.
Synonymes : élasticité, flexibilité, malléabilité, plasticité, souplesse, variabilité.
système altérable, ajustable, changeable, configurable, convertible, customisable, élastique, malléable, modifiable, modulable, paramétrable, plastique, spécialisable, souple.
Il est évident que l'adaptabilité est un facteur important pour faciliter et améliorer l'interactivité avec l'utilisateur. Ce dernier profite de la capacité d'adaptation du système pour personnaliser son environnement, ses techniques d'interaction, ses périphériques, etc. Mais, ce n'est pas parce qu'un système est adaptable qu'il permet de rendre l'interaction naturelle avec l'utilisateur. C'est généralement loin d'être le cas et, au final, c'est l'utilisateur qui s'adapte au système, qui se l'approprie en fonction de ses connaissances. Devant ce constant, plusieurs travaux ont cherché à améliorer et enrichir la capacité d'un système à être utilisé dans des contextes pour lesquels il n'était pas conçu au départ (sa *reinterpretability*, en anglais [Beaudouin-Lafon 04]). Cette propriété suppose que l'utilisateur puisse modifier, ajouter ou supprimer et définir directement et facilement certaines réactions du système, certains périphériques, certaines techniques d'interaction, etc. [Beaudouin-Lafon 00b] définit le paradigme d'Informatique Située (*Situated Computing*) pour permettre le développement de systèmes personnalisables par l'utilisateur à très haut niveau et qui puissent évoluer avec les

usages, sans que les concepteurs aient *a priori* anticipé ceux-ci. L'application ***CPN2000/CPN Tools*** [Beaudouin-Lafon & al. 01] illustre cette approche. Cet éditeur graphique de réseaux de Petri colorés a été développé en se basant sur l'observation et l'analyse de l'interactivité avec l'utilisateur. L'interface la plus adéquate et la plus pertinente pour ce type d'applications a donc été implémentée, permettant à l'utilisateur de choisir parmi un large choix de techniques d'interaction. [Dragicevic 04] pose le problème de l'adaptabilité en entrée et définit cette notion comme étant la combinaison des propriétés de contrôlabilité (capacité d'un système à prendre en compte des entrées enrichies), d'accessibilité (capacité d'un système à prendre en compte des entrées appauvries) et de configurabilité (capacité d'un système à pouvoir être personnalisé du point de vue des entrées par l'utilisateur). Après avoir présenté un modèle original d'interaction en entrée, le système ***ICON*** (Input Configurator) est présenté. Ce système permet le développement d'applications interactives, le développeur pouvant configurer et tester des dispositifs d'entrée et des techniques d'interaction, standards ou non, en se plaçant dans des contextes spécifiques d'interaction en entrée, et l'utilisateur choisissant sa configuration en entrée en fonction de son profil, de son expérience, de son matériel, etc. Il est à noter que dans le domaine des jeux vidéo, il existe un phénomène à rapprocher de ces différentes remarques, le *gameplay* émergent [JeuxVideo.com 06], qui décrit comment les différents joueurs s'approprient le jeu vidéo et inventent de nouvelles façons de jouer qui n'ont pas été anticipées par les concepteurs.

Dans le cadre de ces travaux de thèse, c'est l'**adaptativité** du système (système **adaptatif**, **Niveau 2**) qui est étudiée. Ce n'est pas l'utilisateur qui modifie le système, au cours d'une phase antérieure à l'interactivité, mais bien **le système qui adapte ses réactions**, automatiquement, dynamiquement, et de manière continue, au cours de l'interactivité, sans intervention explicite de la part de l'utilisateur [Rouillard 08]. L'objectif d'un système adaptatif n'est pas de détourner l'utilisateur de sa tâche mais bien au contraire de l'assister, de plus en plus efficacement et précisément, au fur et à mesure de l'interactivité.
Le système, s'il en détecte le besoin, s'adapte au **contexte d'interaction** qu'il perçoit et interprète. Ce contexte est fonction de l'utilisateur, de ses gestuelles, de l'environnement réel, de l'état du système en lui-même, des interactions passées, etc. De plus, son adaptation dépend des connaissances du système sur l'utilisateur, sur l'environnement, etc. et sur celles qu'il acquiert et construit au cours de l'interactivité. La capacité d'un système à percevoir et à interpréter ce contexte d'interaction définit sa **sensibilité** à celui-ci. Il sera alors question de sensibilité au contexte utilisateur, au contexte environnement, etc. Les différentes informations contextuelles, collectées automatiquement par le système, permettent à ce dernier de mieux percevoir, de mieux interpréter et de mieux s'adapter à l'activité observée. La notion de contexte sera l'objet du prochain chapitre.
L'adaptation d'un système peut être **explicite**, au niveau de l'interface logicielle restituée à l'utilisateur par exemple, soit **implicite**, au niveau de sa gestion des ressources matérielles et logicielles. Un système adaptatif peut être également capable d'anticiper, à court terme, les interactions à venir et ses réactions qui en découlent. Ces mécanismes d'adaptation, ainsi que les contextes d'interaction qui les activent, sont **scénarisés**. Ils sont en effet prévus au moment de la conception du système.
L'étude de l'**adaptativité** est un domaine plus récent que celui de l'étude de l'**adaptabilité** et il existe aujourd'hui encore peu de standardisation dans ce domaine. Ce processus est cependant décomposé en **4 phases** distinctes [Dietrich & al. 93] :

- la phase d'**initiative**, au cours de laquelle le système suggère une adaptation ;

- la phase de **suggestion**, le système sélectionnant plusieurs réactions parmi l'ensemble des réactions possibles ;
- puis il choisit une réaction particulière au cours de la phase de **décision** ;
- enfin exécute cette réaction lors de la phase d'**exécution**.

Traduction anglaise : *adaptivity, adaptiveness, adaptive system.*
Synonymes : élasticité, flexibilité, plasticité.
système adaptable (parfois…), ajustable, élastique, malléable, modulable, plastique, spécialisable.

2.1.4. Nouveaux paradigmes d'interaction et de programmation

Les systèmes adaptatifs et sensibles au contexte sont des systèmes capables de réagir intelligemment et de manière cohérente face au contexte d'interaction qu'ils perçoivent. Avec eux ont été alors étudiés de **nouveaux paradigmes d'interaction et de programmation**. Ces paradigmes, ici au nombre de 3, ont permis la conception de systèmes de plus en plus capables de percevoir le contexte d'interaction en cours, comprenant l'environnement, l'utilisateur, d'autres dispositifs, etc.
Ces paradigmes ont été classés par [Balme 08], à partir de [Lyytinen & Yoo 02], **suivant deux dimensions** qui traduisent leurs capacités à permettre la conception **(1)** de systèmes intégrés au sein de notre environnement de tous les jours et **(2)** de systèmes mobiles. Ces paradigmes sont à comparés à celui de l'**informatique traditionnelle**, symbolisé par notre ordinateur de bureau. Cette classification est exprimée par la Figure 1.22.

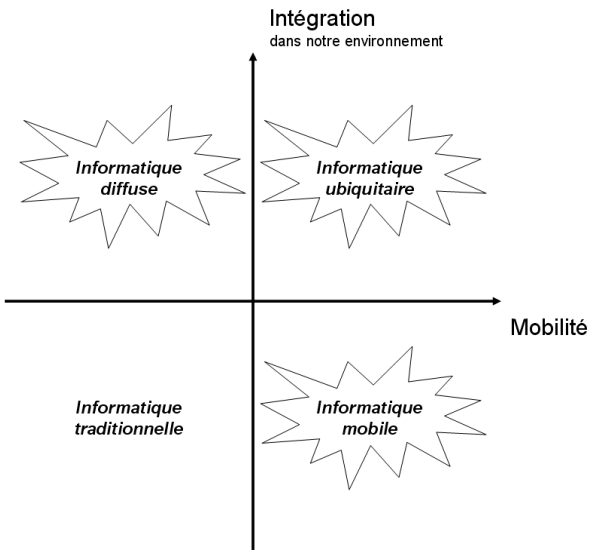


Figure 1.22. : Classification des nouveaux paradigmes d'interaction et de programmation
Schéma proposé par [Balme 08], traduit de [Lyytinen & Yoo 02]

Les systèmes **mobiles**, ou nomades (téléphone portable, PDA, etc.), conçus à partir du **paradigme de l'informatique mobile**, sont des systèmes accompagnant constamment l'utilisateur, qui les utilise n'importe où et n'importe quand. Outre le fait qu'ils doivent permettre à l'utilisateur d'accéder à n'importe quelle information, n'importe où et n'importe

quand, ces systèmes doivent adapter leurs réactions vis-à-vis des gestuelles de l'utilisateur. L'exemple souvent cité est celui d'un téléphone qui se mettrait automatiquement en mode silence lorsqu'il détecterait que son utilisateur est en réunion.
Ces systèmes sont particulièrement sensibles au contexte utilisateur et sont sensés détecter automatiquement les ressources matérielles qui se situent aux alentours et qui pourront se révéler nécessaires au cours de l'interactivité. De plus, ces systèmes se veulent de plus en plus sensibles au contexte environnement. [Chen & Kotz 00], par exemple, propose un état de l'art sur les systèmes mobiles adaptatifs.
Ce paradigme d'interaction et de programmation illustre bien la faculté d'adaptativité d'un système. Cependant, les problématiques sont nombreuses (particulièrement liées aux limitations du dispositif mobile en lui-même) et la conception d'un vrai système mobile adaptatif n'est pas encore complètement une réalité.

Le **paradigme de l'informatique ubiquitaire** [Weiser 91] (*Ubiquituous computing*, ou *Calm computing* du même auteur, ou encore *Pervasive computing*, *Situated computing* [Hull & al. 97], *Sentient computing* [Hopper 99], intelligence ambiante, informatique ambiante) n'est pas si récent que cela. En 1991, Mark Weiser décrit sa vision des systèmes informatiques au 21ème siècle, *Ubiquituous Computing*, ou ***Ubicomp***. Selon cette vision, l'information est accessible n'importe où et n'importe quand, par n'importe qui, par le biais d'ordinateurs et d'écrans. Cet accès à l'information est adapté aux gestuelles de l'utilisateur, elles-mêmes interprétées par le système.
Ce paradigme définit les principes de conception de systèmes, à la fois mobiles et intégrés au sein de l'environnement. Ces systèmes tendent à se fondre dans tous les équipements quotidiens, de manière à être invisibles pour l'utilisateur, qui n'a alors pas forcément conscience d'interagir avec eux. Ils doivent pouvoir s'adapter à n'importe qui, en toutes circonstances et à tout moment. Le **paradigme de l'informatique diffuse** décrit une manière de concevoir des systèmes adaptatifs très similaire à celui de l'informatique ubiquitaire, la différence étant que les systèmes diffus ne comprennent pas de notion de mobilité (ce qui n'est pas toujours le cas suivant les définitions qu'il est possible de rencontrer…).
Idéalement, l'utilisateur se trouve dans un environnement commun et connu, un *smart environment* (un bureau, un salon, etc.) et interagit explicitement, i.e. en manipulant des dispositifs, de la vie de tous les jours et/ou dédiés au contrôle du système, et/ou implicitement, le système observant par le biais de capteurs son activité et le contexte d'interaction dans lequel il se trouve. Quoiqu'il en soit, le système considère l'utilisateur en permanence et ce dernier n'interagit que par le biais de gestuelles naturelles, impliquant le large usage de NUI.
Un des grands atouts d'un système ubiquitaire concerne le fait que les services offerts à l'utilisateur ne le distraient pas, ni ne l'interrompe. C'est pourquoi ces systèmes sont capables de percevoir et d'interpréter l'activité humaine, dans le but d'y répondre de manière adaptée. Ces systèmes peuvent également apprendre de nouvelles situations d'interaction, à partir des gestuelles, implicites et explicites, de l'utilisateur, observées au cours de l'interactivité.
Ces systèmes ubiquitaires (*Ubiquituous system*, ou encore *Invisible Computer* [Norman 98], *Disappearing computer* [Wejchert 00]), **sensibles aux contextes**, collectent les informations nécessaires pour interpréter les gestuelles de l'utilisateur et les changements de contextes d'interaction. Ces informations contextuelles concernent l'utilisateur (identité de l'utilisateur, localisation spatiale dans l'environnement, type d'activité, etc.), l'environnement (éclairement, bruit, etc.), le système en lui-même (ressources, dispositifs disponibles), etc. L'utilisateur interagissant naturellement et pas de manière consciente systématiquement, le contexte d'interaction dans lequel il se trouve est considéré comme variable et imprévisible. Une fois que le système a interprété ce qu'il se passait dans la scène réelle observée, il s'assure de proposer à l'utilisateur le service dont ce dernier a besoin, à l'endroit opportun et à

l'instant opportun. Sa réponse peut être implicite, l'utilisateur ne se rendant pas compte de celle-ci, et/ou explicite, en faisant intervenir l'utilisateur dans l'interaction. Les systèmes ubiquitaires sont généralement hautement adaptatifs, cela d'autant plus que la gestuelle de l'utilisateur peut être complexe et complètement imprévisible.
Il est à noter que pour la conception de tels systèmes, les problématiques sont très nombreuses. Les concepteurs doivent en permanence faire des compromis entre une interaction explicite de la part de l'utilisateur par le biais d'interfaces dédiées, et une interaction implicite grâce au support de capteurs observant la scène. Ces interfaces matérielles, dispositifs ou capteurs, peuvent être très nombreuses, mobiles, sont disséminées au sein de l'environnement d'interaction global et sont connectées les unes aux autres. Le concepteur devra pouvoir déployer, contrôler à distance, modifier, supprimer ou ajouter facilement une nouvelle interface ou un module de traitement au sein de ce système. C'est pourquoi les systèmes ubiquitaires sont des systèmes nécessairement distribués [Zaidenberg & al. 09]. Les systèmes ubiquitaires sont complexes, comprennent un certain nombre de dispositifs hétérogènes, mettent en jeu des environnements dynamiques et des applications diversifiées, et sont destinés à un grand nombre d'utilisateurs. De plus, une interaction ubiquitaire implique que le système adaptatif soit capable de raisonner à partir d'événements incertains. Enfin, les concepteurs devront se heurter à des problèmes de miniaturisation de dispositifs matériels, de réseau, de systèmes embarqués, de fusion de modalités, d'éthique et de sécurité, etc. la liste est longue. Certaines de ces problématiques sont discutées dans [Pascoe & al. 99].
L'évaluation d'un système ubiquitaire est discutée dans [Schmidt 02], à différents niveaux de l'architecture du système. Le constat est qu'il n'y a pas une solution meilleure qu'une autre. Il est possible d'effectuer différentes évaluations basées sur différentes approches, mais ces évaluations sont empruntées à d'autres domaines scientifiques et ne sont pas toujours adaptées, ni satisfaisantes. Cependant, [Balme 08] propose une taxonomie, sous la forme d'un espace problème, dont les 7 dimensions permettent d'identifier et de caractériser les différents aspects de l'adaptation (adaptabilité, Niveau 1, et adaptativité, Niveau 2) à l'exécution dans un système ubiquitaire (voir Fig. 1.23.). Le 1er axe, « Moyens d'adaptation », définit si le système se reconfigure et/ou se redistribue pour s'adapter ; le 2ème axe, «Granularité des composants de l'IHM », indique à quel niveau de l'interface le système s'adapte ; le 3ème axe, « Granularité de l'état de reprise », donne une mesure de la continuité de l'interaction au moment de l'adaptation ; le 4ème axe, « Déploiement de l'IHM », indique à quel moment de l'exécution, l'adaptation a lieu ; le 5ème axe, « Couverture du contexte d'interaction », définit à quoi s'adapte le système ; le 6ème axe, « Couverture des espaces technologiques » caractérise la capacité du système à couvrir un ou plusieurs espaces technologiques ; le 7ème axe, « Méta-IHM », mesure le degré d'autonomie du système, au cours de l'adaptation, par rapport à l'utilisateur.

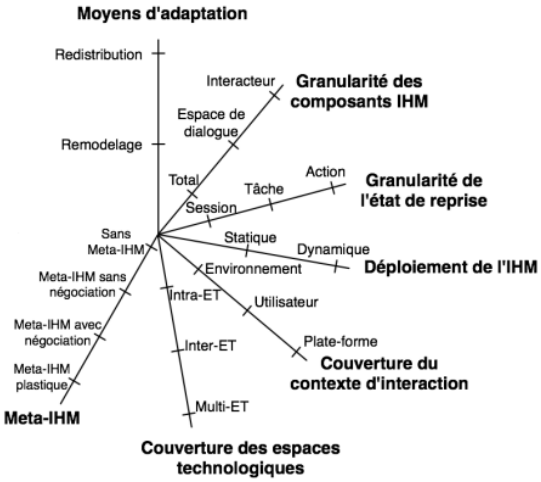


Figure 1.23. : Classification caractérisant l'adaptation à l'exécution d'un système ubiquitaire [Balme 08]

Ces paradigmes d'interaction et de développement, **mobile**, **diffus** et **ubiquitaire**, sont encore jeunes et de nombreux champs de recherche sont encore à explorer. Ces paradigmes sont propices à la conception de systèmes adaptatifs et les concepteurs adoptent largement ces paradigmes en combinaison avec les paradigmes d'interaction ***Post-WIMP***, présentés au cours de la Section 1.2.2.

2.2. Dimensions de l'adaptativité

L'étude de l'adaptativité peut se faire selon plusieurs dimensions. Nous allons reprendre, au cours des prochaines sections, les **dimensions principales de l'adaptativité**, énumérées dans [Rouillard 08]. Il nous semble nécessaire de répondre à ces questions avant de concevoir un système adaptatif. Ces questions sont :

- **Pourquoi s'adapter** (Section 2.2.1.) **?** Il s'agit ici de développer une logique qui tient compte de l'utilisateur. Le système adaptatif répond alors de manière adéquate et naturelle aux besoins et objectifs de l'utilisateur si cette logique est positive, ou au contraire le déstabilise si cette logique est d'opposition. Les mécanismes adaptatifs peuvent prendre différentes formes et prennent place à tous les niveaux du système.
- **Quand et où adapter** (Section 2.2.2.) **?** Au cours de l'interactivité, l'adaptativité peut se mettre en place de manière continue ; après la détection de déclencheurs spécifiques ou à la demande de l'utilisateur. De plus, les mécanismes d'adaptation peuvent être mis en œuvre soit de manière interne, soit de manière externe.
- **A quoi s'adapter et quoi adapter** (Section 2.2.3.) **?** A l'image de l'être humain, le système modélise le contexte d'interaction au sein duquel l'utilisateur évolue et le confronte au scénario de l'application. Ainsi, il peut répondre de manière adéquate et peut mettre en place certains mécanismes d'adaptation à différents niveaux de son architecture logicielle et matérielle, de son scénario et de son interface. Nous mettrons en relief le besoin de modéliser le contexte d'interaction et nous détaillerons les différents aspects du système qui peuvent s'y adapter.

D'autres dimensions sont traitées plus en détails dans [Rouillard 08].

2.2.1. Pourquoi s'adapter ?

Les concepteurs définissent l'adaptativité d'un système **autour de l'utilisateur**, à l'instar de l'interactivité. Le système doit s'adapter aux gestuelles de l'utilisateur, pour répondre à ses besoins en toute circonstance, et par le biais d'une interaction la plus naturelle qui soit. L'utilisateur ne subit plus l'interaction mais il la construit, en accord avec ses objectifs. L'interaction adaptée vis-à-vis de l'utilisateur devient plus simple, plus directe, plus rapide et plus transparente. Elle lui permet d'utiliser des systèmes complexes, nécessitant peu d'apprentissage de sa part.

Le processus adaptatif met en place une logique qui est fonction de l'utilisateur. Un système s'adapte pour satisfaire ou tout du moins pour fournir un support, une aide, à l'utilisateur, si celui-ci est en accord avec un scénario spécifique donné (logique positive). Mais ce même système cherchera à déstabiliser l'utilisateur et à remettre en question ses gestuelles, si ce dernier s'écarte du scénario élu par le système à un instant donné (logique d'opposition).
Du point de vue de l'utilisateur, pourquoi s'adapter ? **Quelles formes l'adaptativité peut-elle prendre autour de l'utilisateur ?**

Tout d'abord et avant tout, un système adaptatif a l'objectif d'**améliorer la vie quotidienne de l'utilisateur**, de lui apporter ce dont il a besoin et en toutes circonstances. Il doit également anticiper ses besoins et ses attentes. Ces systèmes, diffus ou ubiquitaires, tendent à être totalement intégrés dans l'environnement de l'utilisateur, et de manière invisible. Ils capturent et réagissent particulièrement aux gestuelles implicites de l'utilisateur, ce dernier n'ayant alors pas conscience d'interagir véritablement.
Les principaux systèmes adaptatifs développés et étudiés permettent un **affichage d'informations pertinent et intelligent vis-à-vis de l'utilisateur**. Les problématiques sont alors d'adapter le contenu informatif au profil de l'utilisateur, aux dispositifs qu'ils manipulent, à l'interface qui lui est restituée, etc. Le ***Life Wall*** de Panasonic en est un exemple parfait. L'utilisateur interagit avec un mur entier, qui lui restitue un contenu adaptée, en fonction de son profil, de sa localisation par rapport au mur, de ses interactions passées. Ce système permet également à l'utilisateur d'interagir via une surface tactile. [Rouillard 08] présente deux systèmes dans cette optique. Le premier permet, par le biais d'un appareil mobile, la lecture de codes barres spécifiques, les ***QR Code*** (*Quick Response* code), et l'affichage personnalisé des informations associées. Le second comprend un assistant numérique intelligent qui peut interpréter les éléments du contexte courant et qui modifie l'interface logicielle restituée en fonction. Dans ses travaux, [Gutiérrez Alonso 05] définit un Environnement Virtuel Sémantique (*Semantic Virtual Environment*, voir Fig. 1.24.), représentation sémantique décrivant les fonctions et les caractéristiques des différents objets virtuels peuplant un monde virtuel, ainsi que et les relations entre eux. En se basant sur cet environnement, il propose un nouveau paradigme d'interaction et de visualisation. L'utilisateur peut interagir par le biais de nombreux dispositifs et le monde 3D dans lequel il est immergé est restitué de manière adaptée vis-à-vis du dispositif de restitution. Ce modèle est centré sur le rôle de l'objet virtuel au sein d'un contexte d'interaction et est inspiré des standards ***MPEG 7***[1] et ***MPEG 21***[2]. Il utilise plusieurs ontologies pour représenter cet

[1] http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm
[2] http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm

environnement virtuel sémantique, facilitant ainsi le développement de tels environnements et leurs utilisations dans de nombreux dispositifs. Enfin, [Plesca & al. 08] présente un service de navigation vidéo sur appareils mobiles, qui adapte la police de caractères en fonction de la bande passante du réseau (diminution) et de l'intérêt de l'utilisateur (augmentation).

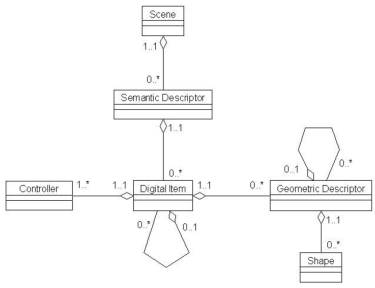


Figure 1.24. : Semantic Virtual Environment [Gutiérrez Alonso 05]

Un système adaptif permet également d'**étudier la gestuelle de l'utilisateur**, pour la corriger ou pour simplement en établir un certain nombre de statistiques. Ces systèmes se basent sur l'observation, le suivi fin et sur l'interprétation précise des gestuelles de l'utilisateur. Ce dernier effectue généralement une gestuelle technique (dans le domaine sportif, médical), hautement identifiable et ne présentant pas d'ambigüités de sens, et le système étudie alors cette gestuelle, pour le corriger par exemple.
Un système adaptatif peut être développé dans le but de **supporter les communications et les collaborations entre différents utilisateurs**. De tels systèmes s'occupent d'analyser les gestuelles d'une personne et de les restituer à d'autres personnes en accord avec leur contexte d'interaction. Cela suggère une analyse très fine du mouvement de l'utilisateur. Ces systèmes sont particulièrement développés pour traduire les gestes d'une personne communiquant par le biais de langages des signes [Oh & Jung 09]. Ces systèmes peuvent permettre également la reproduction intelligente des gestuelles de l'utilisateur, par le biais d'un robot, ou d'une représentation virtuelle dans un contexte d'enseignement à distance par exemple.
De nombreux systèmes adaptatifs ont été développés dans le but de **surveiller l'activité humaine**, pour contrôler un travail, une communication ou un flux de personnes, dans le but d'avertir d'un danger, d'empêcher de commettre une erreur, de filtrer, modérer, censurer des informations, etc. Ces systèmes sont généralement diffus et observent une gamme limitée de gestuelles facilement identifiables. Il est possible ainsi de rencontrer un certain nombre de dispositifs chargés d'observer une scène (une rue, un quai de métro, une autoroute, une sortie de magasin, un parking, etc.) pour détecter d'éventuels dangers et accidents. Dans le domaine médical également, pour surveiller des patients dans un hôpital ou des personnes âgées dans une maison de retraite [Bouchard & al. 08, Noury & al. 09]. [Perreira Da Silva & al. 08ab, Perreira Da Silva & al. 09] développent des systèmes qui estiment depuis plusieurs périphériques les actions et l'attention (à partir de son regard) d'un enfant autiste, dans le but de comprendre ses réactions et pour lui répondre de manière adaptée et en temps réel. Ces systèmes ont pour but d'aider les enfants au cours du processus de réhabilitation, par le biais de jeu éducatif personnalisé.
Enfin, de nombreux systèmes adaptatifs sont développés dans le **domaine du jeu vidéo**, particulièrement pour des jeux vidéo cherchant à faire faire à l'utilisateur des exercices physiques, tout en le divertissant (*exergames : exercise and games*). Typiquement, les concepteurs redoublent d'efforts pour développer des interfaces innovantes et divertissantes.

[Buttussi & al. 07] proposent d'ajouter un moteur d'adaptation aux jeux vidéo qu'ils développent. Ce moteur d'adaptation considère les informations personnelles de l'utilisateur (son âge, son sexe, son poids, sa taille), les données physiologiques capturés par le biais de capteurs dédiés (les battements du cœur, l'oxygène dans le sang), les mouvements capturés par le biais d'accéléromètres et les informations capturées au cours des sessions d'exercices précédentes. Ainsi, le système, par le biais de ce moteur, ajuste l'intensité des exercices physiques, en adaptant le *gameplay* et le monde 3D restitué. Les mécanismes d'adaptation sont définis par un simple automate à états finis (voir Fig. 1.25.). Beaucoup d'autres systèmes de ce type existent, comme celui présenté par [Hämäläinen & al. 05], ***Kick ass Kung fu***, qui agit de manière adaptée sur la reproduction visuelle des mouvements de combat du joueur.

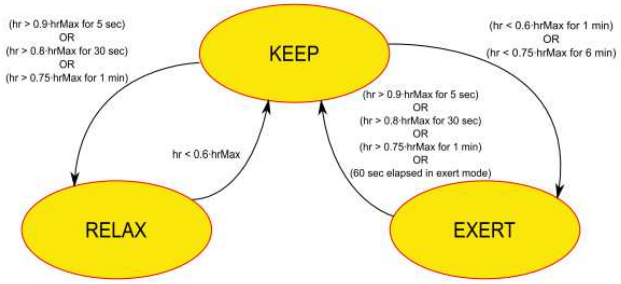


Figure 1.25. : Exemple de mécanismes d'adaptation par le biais d'un automate [Buttussi & al. 07]

Il est à noter, en remarque, que l'adaptativité peut s'articuler **autour du système** lui-même, dans le but d'améliorer ses performances et ses résultats. L'adaptation se fait alors au niveau de son architecture logicielle (migration, reconfiguration, etc.), ou des modules et processus qui la composent (distribution, fusion de modalités, etc.), de son fonctionnement global (considération des ressources environnantes, etc.) etc. [Robertson & Brady 99, Hall & al. 06] proposent un système de suivi auto-adaptatif, capable d'évaluer les résultats du suivi, et de modifier alors les paramètres des différents processus (ce qui fait de ce système, un système autorégulateur, Niveau 3, voir Section 2.1.2.). Ce système réclame peu de supervision et est destiné à la surveillance vidéo d'une scène extérieure. Son architecture logicielle est présentée par la Figure 1.26.

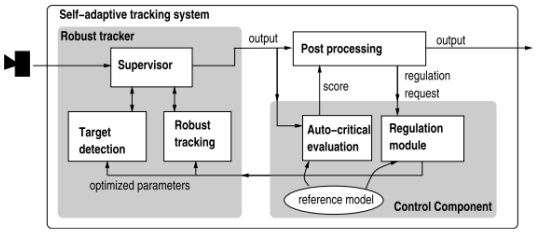


Figure 1.26. : Architecture logicielle d'un système de suivi auto-adaptatif [Hall & al. 06]

2.2.2. Quand et où ?

Répondre à ses questions dépendra bien sûr du type de systèmes et d'applications développés, des sujets et activités observés, etc. Nous reprenons ici en partie les propos de [Rouillard 08].

En remarques préliminaires pour répondre à la question **« Quand ? »**, nous reprécisons que nous parlons de l'adaptation du système **au cours de l'exécution de l'application**, et non pas au cours du développement, de la compilation ou du chargement de celle-ci. De plus, nous considérons l'adaptation du système **au cours de l'interactivité avec l'utilisateur**, et non pas lors d'une phase antérieure d'initialisation et de configuration du système. Nous ne considérons donc l'adaptation que sous un point de vue **dynamique**, même si les autres formes d'adaptation sont tout à fait envisageables. Le système s'adapte vis-à-vis des gestuelles observées de l'utilisateur. Il peut s'adapter **en permanence**, ce qui suppose un suivi continu et précis des gestuelles de l'utilisateur. Ces gestuelles sont constamment interprétés, processus complexe et qui peut rendre sensible le système aux nombreuses ambigüités de sens. Le système peut également s'adapter **à certains instants clés**, après la détection d'une gestuelle de la part de l'utilisateur ou d'une réaction de lui-même, d'une situation du scénario, d'un contexte d'interaction particulier. Ainsi, le système ne s'adapte pas tout le temps et les mécanismes d'adaptation se mettent en place après la détection de déclencheurs spécifiques. Enfin, le système peut s'adapter **à la demande explicite de l'utilisateur** au cours de l'interactivité.

[Oreizy & al. 99] répond à la question **« Où ? »** en précisant que l'adaptation peut se faire soit de manière **interne** au système, les mécanismes étant intégrés à ce dernier dès la conception du système, soit de manière **externe**, via un autre système ou service, permettant ainsi l'ajout de réactions non prévues lors de la conception du système (c'est le cas des systèmes ubiquitaires par exemple).

2.2.3. Quoi adapter et à quoi ?

Nous allons premièrement répondre brièvement à la question **« A quoi ? »**. L'humain, à tout instant, perçoit la situation dans laquelle il se trouve, par le biais d'un certain nombre de stimuli extérieurs. Puis, il l'interprète en prenant en compte le **contexte** dans lequel il se trouve : qui il est et quels sont ses objectifs ? que connaît-il de l'environnement ? que manipule-t-il et comment ? Il **contextualise** la situation pour mieux la comprendre. De cette opération, il en adoptera la réaction adéquate ou déterminera une nouvelle façon de réagir.
Comme l'humain, un système adaptatif collecte un certain nombre de données, potentiellement par le biais de différents canaux de communication, et construit alors à partir de celles-ci le **contexte d'interaction** dans lequel l'utilisateur évolue. Ce contexte d'interaction modélisé, plus ou moins complet, et comprenant certaines caractéristiques spécifiques propres à l'utilisateur, à l'environnement, au système, aux interactions précédentes, etc., permet au système d'interpréter justement et précisément les gestuelles de l'utilisateur. Le contexte est confronté au **scénario** de l'application et, si certains déclencheurs sont détectés, permet la réponse adéquate du système vis-à-vis de l'utilisateur et la mise en place de mécanismes particuliers d'adaptation. Il est donc nécessaire de définir et de modéliser ce qu'est le contexte, en accord avec l'application développé, l'utilisateur considéré, etc. En effet, la bonne modélisation du contexte d'interaction, dans lequel l'utilisateur évolue, permettra d'**améliorer l'interactivité** et d'**introduire des mécanismes d'adaptativité efficaces** (orientation des processus, minimisation des erreurs d'interprétation, etc.) au sein du dialogue interactif.

Notre objectif étant d'élaborer des systèmes adaptatifs réagissant aux gestuelles de l'utilisateur, il nous faut donc estimer le contexte dans lequel l'utilisateur interagit. C'est pourquoi la notion de contexte sera étudiée plus exhaustivement au cours du chapitre suivant.

En revanche, nous allons présenter ici les différents aspects du système qui peuvent s'adapter aux gestuelles de l'utilisateur. Ces formes d'adaptation peuvent évidemment se combiner ou être dépendantes les unes des autres, et nous ne prétendons pas ici énumérer tout ce qui peut s'adapter au sein d'un système. Nous essayerons cependant d'être aussi complets que possible.

Tout d'abord, dans un système adaptatif, le comportement de chaque module de traitements composant l'architecture logicielle du système est dépendant de ceux des autres modules. Ces modules s'influencent les uns les autres, les résultats de l'un orientant les comportements des autres. Un mécanisme d'adaptation peut donc **guider** un module de traitements vers un résultat particulier, dans le but d'influer sur le comportement global du système. De plus, l'adaptation d'un système peut se concrétiser par la **réorganisation dynamique** de ces modules, la **mise en priorité** de certains modules, l'**ajout** ou la **suppression** d'un module, la **modification** et le **redéploiement** d'un module, la **reconfiguration des communications** entre ces modules, etc. dans le but d'assurer un fonctionnement optimal du système, sans aucune intervention de la part de l'utilisateur ou d'un superviseur. Enfin, l'adaptation d'un système peut se concrétiser, sur un laps de temps limité, par **l'anticipation des ressources logicielles et matérielles**, nécessaires à son futur bon fonctionnement.

Le **scénario** de l'application peut également s'adapter aux gestuelles de l'utilisateur [Champagnat 11]. Généralement, les scénarios d'applications, en tout cas pour les plus complexes, ont une ligne narrative principale et plusieurs autres lignes narratives optionnelles que l'utilisateur peut décider, ou non, de suivre. L'adaptation d'un scénario permettra par exemple de remettre l'utilisateur sur la ligne directrice, s'il s'en est trop écarté ou pendant trop longtemps. L'adaptation d'un scénario se fait par la **réorganisation**, voire la **construction dynamique**, **des situations d'interaction et des dialogues interactifs**, supposant alors une narration non linéaire.

Le principal aspect traité lors de l'étude du mécanisme adaptatif au sein d'un système est celui de **l'interface**. De nombreux travaux ont été effectués dans le but d'adapter l'interface comportementale, matérielle et logicielle, en fonction des gestuelles de l'utilisateur. Ce type d'interfaces est parfois nommé ***Interfaces Utilisateur Intelligentes*** (*Intelligent User Interfaces*, IUI).

Premièrement, un système peut adapter les **techniques d'interaction** mises à la disposition de l'utilisateur en fonction des gestuelles de ce dernier. Cette forme d'adaptation est particulièrement pertinente lorsque le système peut considérer en entrée plusieurs modalités. En effet, ce type de système devra reconfigurer certains de ses modules lors de la prise en compte de nouvelles modalités [Nigay & Gray 06], ou pourra encore sélectionner et combiner les modalités pertinentes pour une situation d'interaction particulière [Chalmers & Galani 04]. Dans ses travaux de thèse, [Delmas 09] propose une architecture logicielle, permettant le pilotage adaptée d'un jeu vidéo, en se basant sur l'observation et l'analyse du déroulement du scénario. Le but est de proposer au joueur les événements qui permettront de faire évoluer le scénario. Au cours de ce scénario, le joueur ne doit pas contraint dans ses gestuelles ; il doit

évoluer dans un univers cohérent, du point de vue de ses connaissances et du type de jeu proposé ; et le scénario du jeu doit être structuré et interactif. Pour illustrer ces travaux, plusieurs applications ont été développées (tetris, labyrinthe, etc.) et cette forme de pilotage a même été implémentée au sein d'un robot ***AIBO***[1] de Sony.

L'adaptation d'une **interface matérielle**, au cours de l'interaction, est possible, même si cette forme d'adaptativité est peu rencontrée. Cela parait normal puisqu'il faut que le matériel s'y prête, le concepteur du système se heurtant alors à des problèmes de faisabilité, de mécanique, de robustesse de l'équipement et de coût. De plus, il faut que les mécanismes d'adaptation soient bien élaborés au moment de la conception, les modifications/ajouts/suppressions de mécanismes n'étant pas toujours possibles. La forme d'adaptativité au niveau de l'interface matérielle la plus évidente est celle de l'adaptation automatique des positions et rotations des caméras en fonction des gestuelles de l'utilisateur.

Enfin, du point de vue de l'**interface logicielle**, il s'agira d'adapter le contenu de la scène 3D au sein de laquelle l'utilisateur évolue. Au niveau de l'interface globale, celle-ci pourra s'afficher avec plus ou moins de détails en fonction du profil de l'utilisateur et/ou du dispositif qui la restitue. Quant au contenu à proprement parler, en fonction de la gestuelle de l'utilisateur, certains outils logiciels, ou certains éléments de la scène 3D, pourront être mieux mis en évidence et suggérés leur utilisation. Enfin, dans le but de faire évoluer le scénario de l'application, le système pourra orienter l'utilisateur en activant certaines fonctions spécifiques à un instant donné ou en affichant tout simplement un message visuel, au contenu plus explicite.
[Thevenin & Coutaz 99, Thevenin 01] définissent la **plasticité d'une interface** comme étant la capacité d'adaptation (adaptabilité et adaptativité) d'une interface interactive aux changements de contexte d'interaction (système et environnement, l'utilisateur étant la motivation pour s'adapter), dans les limites de son cadre d'utilisation, tout en préservant l'utilisateur au centre de l'interactivité. Un point important et nécessaire est le fait que l'adaptation de l'interface soit observable par l'utilisateur pour qu'il puisse prendre en compte ce changement. L'étude des interfaces plastiques est devenu un domaine d'étude à part entière, dont l'objectif est des générer des IUI, utilisables par n'importe qui et en toutes circonstances. Le terme 'plasticité' est inspiré de la plasticité de matériaux sous contraintes, qui se dilatent et se contractent sans rupture. A l'instar de ces matériaux, une interface plastique peut se structurer et se restructurer sous des formes différentes.
Dans le cas des interfaces de type ***WIMP***, [Thevenin 01] propose une approche d'***Ingénierie Dirigée par les Modèles*** (IDM), *Model-Driven Engineering* (MDE). Le modèle de l'utilisateur, de la tâche et des concepts métier sont transformés de manière successive, suivant différents niveaux d'abstraction, tout d'abord en IHM abstraite, puis en IHM concrète et enfin en IHM finale. L'objectif est de définir une fois l'IHM, à un niveau d'abstraction, et de la décliner, par transformations de modèles, en une IHM spécifique à un contexte d'interaction donné. Dans le cadre de ces travaux, un démonstrateur, le ***MMS*** (*Minimal MediaSpace*), dont le but est de mesurer le taux d'activité des personnes connectées à l'application, illustre la plasticité dynamique de l'IHM au cours de l'exécution.
Les interfaces plastiques ont également été étudiées dans le cadre de l'informatique ubiquitaire. Le concepteur d'interfaces plastiques, dans ce cadre, doit alors faire face à deux problèmes majeurs, qui n'existent pas en informatique traditionnelle. Tout d'abord, pour un système ubiquitaire, les contextes d'interaction peuvent être très variables et ne sont pas toujours prévisibles. Deuxièmement, les systèmes ubiquitaires utilisent largement des

[1] http://support.sony-europe.com/aibo/

interfaces ***Post-WIMP***. Selon [Calvary & al. 08], l'adaptation ne se limite plus à la portabilité et à la translation de l'interface. L'interface doit pouvoir être remodelée et redistribuée (potentiellement sur un ensemble de dispositifs hétérogènes), totalement ou partiellement, suivant le contexte d'interaction estimé. Il est possible d'imaginer l'adaptation d'une interface ***WIMP*** en une interface ***Post-WIMP*** par exemple. L'approche par transformations de modèles se révèle donc insuffisante en elle-même. [Balme 08] propose un modèle haut niveau à composants dynamiques, ***Ethylene***, pour reconfigurer et redistribuer dynamiquement, au cours de l'exécution, les composants d'une interface plastique. Dans ces travaux, un langage de description de ce modèle, ***EthyleneXML***, et un *framework* pour faciliter le développement d'interfaces plastiques, ***EthyleneFramework***, ont été implémentés. [Calvary & al. 08] propose une méthode pour développer des interfaces plastiques au sein d'un système ubiquitaire, en combinant une approche d'IDM et d'***Architectures Orientées Services***, *Service-Oriented Architecture* (SOA). Cette approche permet de créer des interfaces comprenant une partie interfaces de type ***WIMP*** et une partie, interfaces ***Post-WIMP***. Les interfaces de type ***WIMP*** sont décrites par un langage déclaratif haut niveau dédié (*User Interface Description Language*) puis sont générées plus ou moins automatiquement au cours de la compilation du programme. Un graphe de modèles permet la représentation de ces interfaces à différents niveaux d'abstraction. Ces modèles sont élaborés au moment de la conception et sont utilisés pour générer les interfaces lors du processus d'adaptation (approche IDM). A ces interfaces sont connectés des composants orientés services (SOA), qui représentent les parties ***Post-WIMP***. Ces composants sont développés par le biais de langages très bas niveau, car il n'est actuellement pas possible d'avoir des langages déclaratifs haut niveau pour générer automatiquement des interfaces ***Post-WIMP***.
Pour une application qui nécessite une **scène virtuelle**, l'adaptation du système sera traduite par celle des différents éléments composant cette scène : représentation de la scène immergeant l'utilisateur, représentation et réactions d'éléments interactifs, d'agents autonomes [Heguy & al. 01], règles régissant le monde 3D, activation/désactivation en fonction des besoins et gestuelles de l'utilisateur, etc.

2.3. Positionnement des travaux de l'équipe ImagIN

Une des problématiques majeures de l'équipe ***ImagIN*** est l'étude de l'adaptativité d'un système (système adaptatif, Niveau 2). L'adaptabilité au sein de nos systèmes est basique et ne constitue pas un sujet d'étude à part entière. Nous nous intéressons à l'adaptation du système au cours de l'exécution de l'application et au cours de l'interactivité avec l'utilisateur.

En nous rapportant à [Trigg & al. 87, Rouillard 08], nous cherchons à développer au sein de l'équipe des systèmes :

(1) **flexibles**, ces systèmes pouvant interprétés des gestuelles spécifiques, effectuées de différentes manières, par différents utilisateurs, dans le cadre de tâches données,
(2) parfois **intégrables**, ces systèmes pouvant s'interfacer avec d'autres systèmes (un moteur d'adaptation pourra se connecter à différents modules de capture de gestuelles, par exemple).

Ces systèmes ne sont en revanche pas modifiables par l'utilisateur, seules les applications peuvent être personnalisables, de manière basique, par ces derniers.

Nos systèmes sont **sensibles au contexte** (*context sensitive system*, *context aware system*). Nous nous rapportons à la définition de [Dey 01], qui décrit un système sensible au contexte comme **un système utilisant le contexte pour fournir des informations et/ou services à l'utilisateur, pertinemment vis-à-vis de la tâche effectuée par celui-ci**. Nous la complétons en ajoutant que le système doit également adapter ses réactions, de manière à optimiser les différents processus le composant et à assurer un fonctionnement optimal. L'étude de la gestion du contexte au sein d'un système interactif est traitée au cours du Chapitre 2.

Nous développons des **applications temps réel scénarisée**, de type NUI, impliquant l'interprétation et l'étude l'activité ayant lieu au sein de la scène réelle, particulièrement caractérisée par les gestuelles de l'utilisateur. Nous utilisons majoritairement des interfaces d'observation visuelle de la scène, avec le moins de contraintes intrusives du point de vue de l'utilisateur. Nous pouvons donc parler de systèmes diffus, sans prendre en compte la notion de mobilité pour le moment. Nos applications sont des jeux vidéo, peuvent entrer dans le cadre d'enseignement à distance, permettent l'analyse de mouvements sportifs, etc.

Au sein de l'équipe ***ImagIN***, l'adaptativité au sein du système se traduit par l'affichage pertinent d'informations, l'étude intelligente des gestuelles utilisateur, la surveillance d'activité (plutôt à des fins médicales) et le divertissement dans le cadre de jeux vidéo. Les systèmes adaptatifs mettent en place des mécanismes pour adapter le scénario, l'interface logicielle, généralement des scènes en 3D (ce qui rend difficile l'emploi de techniques relatives à la plasticité des interfaces), et le fonctionnement global de ces systèmes, toujours dans le but d'améliorer l'interactivité et d'augmenter les performances de ces derniers.

3. Notre approche au sein de l'équipe ImagIN

Nous désirons proposer, au cours de cette section, l'architecture d'un système type, tel qu'il serait développé au sein de l'équipe ***ImagIN***.
En accord avec les positionnements établis, vis-à-vis des axes de recherche concernant l'interactivité et l'adaptativité (Section 1.5. et 2.3.), nous établissons, au cours de la Section 3.1., un cahier des charges, dont le but est d'orienter et de cadrer les différents concepteurs d'un système.
Ce cahier des charges guide notre démarche de conception, au cours de la Section 3.2., nous permettant ainsi de définir l'architecture d'un système interactif, sur lequel s'exécutent de manière adaptée des applications scénarisées.

3.1. Cahier des charges

Au cours de cette section, nous établissons notre cahier des charges de conception de systèmes, à partir de l'étude des expressions **« application scénarisée interactive »** et **« exécution adaptative »**.
Notre cahier des charges est un ensemble de **7 Règles Générales de travail** (**RG**, Règle Générale). La première règle concerne aussi bien l'interactivité que l'adaptativité. Les règles RG 2., 3. et 4. concernent le processus interactif en tant que tel, et les trois dernières règles cadrent les capacités adaptatives du système au cours de l'interactivité. Il est à noter que ces trois dernières règles ne concernent que la conception du système autour de l'**Observé**. En effet, l'interactivité au niveau de l'**Utilisateur** reste conventionnelle, standard. Comme nous

le concevons au sein de l'équipe, le système **adapte uniquement ses réactions vis-à-vis des gestuelles de l'Observé**.
Le lecteur peut se référer à tout moment à l'Annexe C. qui énumère ces règles.

RG 1. Le processus d'interaction scénarisé mis en œuvre par le système peut s'établir à la fois avec une personne observée par ce dernier, l'**Observé**, et avec une personne le commandant, l'**Utilisateur**.

RG 2. L'application s'exécutant sur le système comprend **deux scénarios** : un scénario destiné à la commander, suivi par l'Utilisateur ; et un scénario l'animant, immergeant l'Observé et le faisant évoluer au sein d'une narration.

RG 3. Un **dialogue interactif**, implicatif, constant et temps réel doit s'établir **entre l'Utilisateur et le système** et **entre l'Observé et le système**. Ce dialogue est piloté par le scénario que suit chacun des interactants.

RG 4. L'application répond par le biais d'une **interface de restitution** aux **commandes** explicites de l'**Utilisateur**, reconnus par le biais du scénario qu'il suit, acquises par le biais d'une **interface d'acquisition**. Elle répond par le biais d'une nouvelle **interface de restitution** à l'activité, relative à l'**Observé**, reconnue par le biais du scénario, observée et capturée par une **interface d'acquisition**.

RG 5. L'**activité**, relative à l'Observé, est **observée**, **caractérisée** et **interprétée**. Ces trois processus sont **pilotés** par le **scénario** de l'application.

RG 6. L'activité, relative à l'Observé, si elle est **attendue** par le **scénario** de l'application, **déclenche le processus d'adaptativité scénarisée** au sein du système.

RG 7. Guidé par les connaissances scénaristiques, le système s'adapte **en temps réel** à l'activité, relative à l'Observé, **aussi bien au niveau de sa réponse vis-à-vis de ce dernier qu'au niveau de son fonctionnement global**.

3.1.1. Règle Générale RG 1.

Le processus d'interaction scénarisé mis en œuvre par le système peut s'établir à la fois avec une personne observée par ce dernier, l'Observé, et avec une personne le commandant, l'Utilisateur.

Jusqu'à présent, nous n'avons parlé que d' « utilisateur » mais c'est une notion que nous devons dorénavant préciser, comme le stipule la règle **RG 1.** En effet, le processus d'interaction peut prendre place entre le système et une personne qui le commande, que nous appellerons l'**Utilisateur**, ou une personne qui est guidé par celui-ci, l'**Observé**. Il peut y avoir plus d'un Utilisateur/Observé et une même personne peut être à la fois Utilisateur et Observé.

Un Utilisateur/Observé est associé à une **entité informatique virtuelle**. Cette entité modélise la personne réelle associée et peut donc posséder un profil, une représentation virtuelle, etc. Sa gestuelle reflète celle de la personne réelle associée, interagissant avec le système selon les différentes modalités considérées par ce dernier. Il est possible qu'un Utilisateur/Observé soit uniquement virtuel, par exemple dans le cas respectivement du pilotage automatique d'une application ou du besoin d'agents autonomes au sein de la scène virtuelle (Personnage Non-Joueur, PNJ, ou les bots dans les jeux vidéo). En revanche, seul l'Utilisateur peut être

purement réel. Nous faisons effectivement le choix qu'un Observé doit, même de manière minimaliste, être représenté virtuellement. Dans la suite de cette section, nous considérerons majoritairement qu'un Utilisateur ou un Observé est une personne physique, associée à une entité informatique virtuelle, les autres cas étant considérés comme des exceptions. Cela dit, ces exceptions demandent peu de modifications dans notre raisonnement.

Une **scène virtuelle** constitue le monde virtuel des Observés. Ces derniers évoluent au sein de cette scène et leurs interactions, entre eux ou avec différents éléments interactifs, ne sortent pas du cadre de celle-ci. Le contenu de la scène constitue donc la source de l'interactivité entre le système et les Observés. La scène virtuelle est englobée dans un **espace virtuel de simulation** dans lequel évoluent les différents Utilisateurs virtuels. C'est de cette espace que les Utilisateurs 'observent' les gestuelles des Observés au sein de la scène virtuelle. Cette scène et cet espace sont deux environnements interactifs, avec lesquels les Observés et les Utilisateurs interagissent. Ces deux environnements peuvent être représentés visuellement, que cela soit par une interface graphique de type ***WIMP*** en deux dimensions que par un monde complet et gigantesque en trois dimensions.

3.1.2. Règle Générale RG 2.

L'application s'exécutant sur le système comprend deux scénarios : un scénario destiné à la commander, suivi par l'Utilisateur ; et un scénario l'animant, immergeant l'Observé et le faisant évoluer au sein d'une narration.

La règle **RG 2.** met en évidence l'apport essentiel du **scénario** de l'application dans l'interactivité, **structurée** et **dirigée** par celui-ci.
Ce processus est guidé par des connaissances spécifiques apportées par le **scénario de l'application** et est orchestré par le **système interactif**. Ces connaissances portent sur des événements particuliers répertoriés, impliquant une **réponse précise du système vis-à-vis de l'activité observée reconnue**.

L'application peut comporter **deux scénarios** distincts, l'un dédié aux Observés et l'autre aux Utilisateurs. Ces scénarios interviennent aussi bien au niveau de la personne réelle observée que de l'entité informatique à laquelle elle est associée.
Au niveau des Observés réels, le scénario constitue un apport essentiel de connaissances, qui définira les objectifs que ces derniers doivent remplir, les tâches qu'ils doivent effectuer, ainsi que les gestuelles qu'ils doivent adopter pour ce faire. De plus, ce scénario définit la scène virtuelle et son contenu (éléments interactifs, entités informatiques virtuelles le composant) telle qu'elle est perçue par l'entité, représentant l'Observé, évoluant au sein de celle-ci.
L'Observé suit ce scénario au cours d'une simulation mise en œuvre par l'application. Au cours de la simulation, les gestuelles de l'Observé modifient la scène virtuelle, adoptées dans le but d'atteindre certains objectifs ou d'accomplir certaines tâches, définis par le scénario. Une fois ces objectifs ou tâches réalisés, le scénario évolue. Cette évolution donne de nouveaux objectifs et de nouvelles tâches à l'Observé, et modifient la scène virtuelle, telle qu'elle est perçue par son entité virtuelle, au sein de laquelle cette dernière évolue. L'Observé adoptera alors de nouvelles gestuelles et le cycle recommence. Une fois le scénario déroulé, la simulation se termine et les résultats de cette simulation, traduisant l'ensemble des interactions ayant eu lieu au sein de la scène virtuelle, sont envoyés à l'Utilisateur.
Pour ce qui est des Utilisateurs, leur scénario leur apporte plus des solutions pour atteindre les objectifs qu'ils se sont définis ou pour effectuer les tâches qu'ils veulent réaliser. Ce scénario

est plus simple et est constitué d'une série schématique d'actions-réactions de type 'commandes de l'utilisateur – réponse du système'. L'Utilisateur interagit avec le système pour contrôler la simulation au sein de laquelle les Observés évoluent. Ses commandes impliquent donc un retour d'informations, qui expriment un résultat répondant aux objectifs ou tâches de l'Utilisateur.

3.1.3. Règle Générale RG 3.

Un dialogue interactif, implicatif, constant et temps réel doit s'établir entre l'Utilisateur et le système et entre l'Observé et le système. Ce dialogue est piloté par le scénario que suit chacun des interactants.

Le **« dialogue interactif »**, défini par **RG 3.**, décrit entre deux ou plusieurs participants, un ensemble d'échanges. Tout d'abord, ces échanges, de type action-réaction, sont **implicatifs**, c'est-à-dire que les participants s'impliquent directement et personnellement dans le dialogue interactif. Deuxièmement, les échanges sont **constants** dans le sens où c'est un dialogue à proprement parler, à une action d'un des participants correspond en permanence une réaction de l'autre participant, elle-même fonction de cette action. Le dialogue interactif est également **scénarisé**, le scénario, d'une part permettant la reconnaissance par un des participants des actions d'un autre participant et d'autre part supportant un participant à établir une réaction, face aux actions d'un autre participant. Enfin, ces échanges s'effectuent en **temps réel**, dans la mesure où la réponse d'un participant à l'action de l'autre est immédiate, ou tout du moins dont le délai n'est pas perceptible.
Une action d'un des participants vis-à-vis de l'autre, est appelée une **interaction**. La logique de ces interactions définit l'**interactivité** d'un système. Il peut être également question d'interactivité relationnelle.

Deux dialogues interactifs, implicatifs, constant et temps réel se mettent en place, entre le système et les Observés, et le système et les Utilisateurs.
Dans un premier temps, les Observés perçoivent la scène virtuelle au sein de laquelle ils évoluent, puis agissent alors sur cette dernière en fonction des objectifs et tâches définis par leur scénario. Par la suite, si ses objectifs et tâches sont réalisés, le scénario évolue et modifie alors la scène virtuelle telle qu'elle est perçue par les Observés, en accord avec de nouveaux objectifs et de nouvelles tâches à réaliser. Le dialogue reprend alors par l'adoption de nouvelles gestuelles de la part des Observés.
Les Utilisateurs, quant à eux, agissent sur le système, en envoyant des commandes pour initialiser, paramétrer, lancer et contrôler la simulation. Ces commandes dépendront des solutions apportées par le scénario, des objectifs personnels de l'Utilisateur et éventuellement des simulations précédentes (déroulement et résultats). Une fois la simulation terminée, le système répond en leur transmettant les résultats de la simulation. Chaque commande, rendue possible par les connaissances scénaristiques de l'utilisateur, implique donc une réponse scénarisée de la part du système.

3.1.4. Règle Générale RG 4.

L'application répond par le biais d'une interface de restitution aux commandes explicites de l'Utilisateur, reconnus par le biais du scénario qu'il suit, acquises par le biais d'une interface d'acquisition. Elle répond par le biais d'une nouvelle interface de restitution à l'activité,

relative à l'Observé, reconnue par le biais du scénario, observée et capturée par une interface d'acquisition.

La règle générale **RG 4.** décrit le cadre de conception de l'interface du système. Les interactions, entre un participant et le système, sont observées et acquises par une **interface d'acquisition** dédiée. Cette interface doit pouvoir capturer, selon les différentes modalités[1] considérées (le geste, la voix, etc.), les indices spécifiques caractérisant les actions du participant. En accord avec un scénario particulier et de la réaction passée du système, le participant interagit avec l'interface d'acquisition par le biais de **techniques d'interaction** spécifiques. Une fois l'interaction du participant confrontée aux attentes courantes du scénario et reconnue, le système répond à celle-ci par le biais d'une **interface de restitution**, immergeant le participant dans un espace interactif par le biais duquel il pourra répondre à son tour. Les modifications au sein de cet espace constituent la réponse globale du système, véritable image de la compréhension de l'action observée du participant.

Encore une fois, la **distinction** Observé-Utilisateur doit être mise en évidence et quelques **précisions** supplémentaires doivent être faites.
L'activité, particulièrement celle liée aux gestuelles des Observés, est acquise à deux niveaux : au niveau de la scène réelle (événements divers, dont l'Observé n'est pas responsable) et de la personne physique (interactions avec le système par le biais de l'interface matérielle) et au niveau de l'entité informatique qui représente cette dernière (interactions au sein de la scène virtuelle). L'Observé réel se voit restituer la scène virtuelle, soit au sein de laquelle il est représenté en tant qu'entité à part entière, soit telle qu'elle est perçue par cette entité virtuelle. L'image de la scène virtuelle constitue l'interface logicielle restituée à la personne réelle par le biais de l'interface de restitution. Cette interface logicielle représente donc indirectement la situation d'interaction scénarisée courante.
Pour ce qui est de l'Utilisateur, qu'il soit réel ou virtuel, seules ses commandes explicites sur le système sont acquises mais, à la différence de l'Observé, ses gestuelles ne sont ni observées ni interprétées. L'espace de simulation global lui est restitué, lui permettant d'analyser et d'interpréter la simulation mise en œuvre par l'application. Comme pour l'Observé, l'espace de simulation, tel qu'il lui est restitué, est l'interface logicielle pour l'Utilisateur. Cette interface logicielle reflète les possibilités d'interaction accordées par le scénario de l'application.

3.1.5. Règle Générale RG 5.

L'activité, relative à l'Observé, est observée, caractérisée et interprétée. Ces trois processus sont pilotés par le scénario de l'application.

C'est l'activité, particulièrement celle liée aux gestuelles d'un Observé, qui entraine la mise en place des mesures adaptatives par le système. Le scénario de l'application évolue en fonction de cette activité. Il est donc bien normal d'**observer** cette activité, via l'interface d'acquisition et de manière aussi transparente que possible ; de la **caractériser** pour le rendre compréhensible par le système et comparable avec l'activité attendue par le scénario ; et de l'**interpréter** pour savoir si oui ou non, c'est l'activité, et surtout la gestuelle de l'Observé, que le scénario attend.

[1] Une modalité est un canal de communication homme – machine : la voix, l'audition, la vision, le geste, les expressions du visage, etc.

Toutes les gestuelles de l'Observé sont prises en compte, aussi bien les gestuelles **explicites** qu'effectuent l'Observé pour interagir volontairement avec la scène virtuelle, que les gestuelles **implicites**, l'Observé ne sachant pas forcément pas qu'il interagit avec le système (interaction non consciente et non intentionnelle). D'autres événements, prenant place au sein de la scène réelle et caractérisant l'activité de manière globale, peuvent être considérés par le système. Ces événements ne sont pas directement liés à l'Observé et à ses gestuelles mais peuvent influer sur le comportement du système.
Les gestuelles explicites et implicites de l'Observé et les autres événements, dont l'Observé n'est pas la source, sont caractéristiques du **contexte d'interaction** dans lequel se croit être l'Observé.

Le scénario de l'application, en attendant certaines gestuelles clés, pilote, de manière contextualisée, les processus d'observation, de caractérisation et d'interprétation de l'activité. En effet, connaissant la gestuelle à venir à un instant donné, le système sait quoi, où, quand et comment observer ; il sait ce qu'il est important de caractériser et comment le faire ; et il peut comparer ce qu'il a observé avec ce qu'il attend, pour interpréter la gestuelle de l'Observé.
L'observation est **double** : **une observation focalisée**, adaptée en fonction des différentes connaissances (état courant du scénario, objectifs définis par ce dernier, gestuelle attendue en fonction, etc.) ; **une observation globale**, de l'Observé, de la scène en général, de manière à faire face à toute gestuelle non attendue par le scénario et tout événement prenant place au sein de la scène réelle et dont l'Observé n'est pas responsable.

3.1.6. Règle Générale RG 6.

L'activité, relative à l'Observé, si elle est attendue par le scénario de l'application, déclenche le processus d'adaptativité scénarisée au sein du système.

L'application dans laquelle est immergé l'Observé est **scénarisée**. Ce scénario et son évolution sont contrôlés par l'application, et par le biais de ce scénario, certaines gestuelles et événements particuliers, caractérisant l'activité globale, sont **attendus** au sein de la scène réelle (de la part de l'Observé ou non) à un instant donné. Cette activité spécifique, une fois observée (mise en place d'**observateurs**), est alors responsable de l'évolution du scénario. Guidé par le scénario, le système met alors en place le **processus d'adaptativité**, enrichissant alors celui de **l'interactivité**.

Certes, certaines gestuelles clés sont attendues, mais **aucune gestuelle particulière n'est imposée** à l'Observé. Par le biais de ses réactions adaptées, le système ne peut que l'inciter et l'inviter à adopter une gestuelle particulière, en le plaçant dans des **contextes d'interaction adéquats**.
En effet, comme pour l'être humain, nous englobons l'activité au sein d'un **scénario**, celui de l'application. Ce scénario définit les différentes situations d'interaction auxquelles devra faire face l'Observé (et donc définit ses différents objectifs). Ces situations sont contextualisées et chaque contexte d'interaction sera fonction de l'Observé, de sa gestuelle, de la scène au sein de laquelle il est immergé, des situations précédentes, etc. Par le biais du scénario, il est donc possible de modéliser un certain nombre de **contextes d'interaction** qui seront attendus par le système.
Ces contextes permettent alors la mise en place de **déclencheurs spécifiques**. Nous cherchons à ce que ces déclencheurs soient uniquement détectés visuellement, sans équipements

spécifiques portés de la part de l'Observé. Nous appuyons notre interprétation de l'activité sur la caractérisation continue de celle-ci et/ou sur la détection de déclencheurs précis à instant donné. Ces informations nous permettent de modéliser le contexte d'interaction observé, qui sera comparé à celui attendu par le scénario. Une fois l'activité reconnue, le scénario évolue, le système répond en adéquation avec les gestuelles de l'Observé et des mécanismes spécifiques d'adaptation, extraits également du scénario, sont mis en place.
Les contextes d'interaction attendus au cours de l'interaction et les mécanismes d'adaptation sont établis avec le scénario, au moment de la conception.

Les gestuelles de l'Observé ne sont donc pas prévisibles *a priori*. Elles peuvent être sujettes à des exceptions, non attendues et non prévues lors de la conception du système et de l'écriture du scénario de l'application. L'Observé peut donc tout à fait réagir spontanément et irrationnellement. Enfin, les gestuelles de l'Observé peuvent être ambigües au niveau de leur signification.

3.1.7. Règle Générale RG 7.

Guidé par les connaissances scénaristiques, le système s'adapte en temps réel à l'activité, relative à l'Observé, aussi bien au niveau de sa réponse vis-à-vis de ce dernier qu'au niveau de son fonctionnement global.

Le déroulement du scénario (et donc la reconnaissance de l'activité au sein de la scène réelle, particulièrement celle relative à l'Observé) met en œuvre le **processus d'adaptativité**, supporté par le **processus d'interactivité**. En effet, l'interactivité est complémentée et enrichie par l'adaptativité du système. Un **dialogue interactif intelligent** est dorénavant établi entre le système et l'Observé, qui ne subit plus l'interaction. L'adaptativité, comme l'interactivité, est conçue au moment de l'écriture du scénario et de la conception du système.

Le système répond de manière adaptée et en temps réel aux gestuelles implicites et explicites de l'Observé, ainsi qu'à d'autres événements répertoriés, dont l'Observé n'est pas la source et caractérisant l'activité globale. **Cette réponse est double**. Tout d'abord, elle se situe **au niveau de la scène virtuelle** immergeant l'Observé (logique application). Les modifications au sein de celle-ci entraîneront l'Observé à adopter une nouvelle gestuelle. D'un autre côté, le système adapte également **l'ensemble de son fonctionnement interne**, orientant les différents processus le composant en accord avec la gestuelle de l'Observé et la gestuelle qu'il est supposé avoir (logique système). Cette double réponse, caractérisant le processus d'adaptativité, est décrite au sein du scénario, et est donc prévue *a priori*.
L'adaptativité se traduira donc au niveau de la scène virtuelle au sein de laquelle l'Observé évolue (logique application) et au niveau des différents processus composant le système (logique système). Selon [Oreizy & al. 99], elle sera soit interne, soit externe, la gamme de systèmes étudiés au sein de l'équipe étant assez diversifiée.
Il est à noter que le processus d'adaptativité n'est pas supporté que par les connaissances scénaristiques, mais également par celles que le concepteur introduit *a priori* sur l'Observé, sur l'environnement réellement observé, etc. Enfin, un ensemble de connaissances se construit dynamiquement au cours de l'interactivité et permet de guider le processus d'adaptativité. Ces connaissances se répercuteront non seulement sur la scène virtuelle mais également sur les processus d'observation, de caractérisation et d'interprétation de l'activité.

Les mécanismes d'adaptation sont un ensemble de règles conditionnelles. Nous cherchons en permanence le compromis entre le nombre de mécanismes et leur complexité. Un mécanisme complexe sera plus adapté à une situation mais sera plus long et coûteux à mettre en place, alors qu'un mécanisme simple sera plus facile à mettre en œuvre mais sera peut être trop basique. De plus, les mesures adaptatives ne doivent pas être trop redondantes, se limitant par exemple au recommencement incessant de la même situation d'interaction.

Le processus d'adaptativité permet de rentrer dans une **boucle constante d'amélioration de l'interactivité**. Le processus d'adaptativité permet d'entamer un **cercle vertueux d'interaction**, une meilleure réponse du système orientant mieux l'Observé et une meilleure gestuelle de l'Observé guidant mieux le système. Une réponse adaptée du système impliquera l'adoption par l'Observé d'une gestuelle attendue par la suite par le scénario. Cette gestuelle ayant de fortes chances d'avoir lieu, le système pourra mieux l'observer, mieux la caractériser, mieux l'interpréter et, bien sûr, mieux y répondre. Et le cycle recommence (voir Fig. 1.27.).
Une bonne adaptativité influence l'ensemble de l'interactivité avec l'Observé. Nous jugerons également de l'adaptativité d'un système par sa tolérance aux erreurs d'interprétation et sa sensibilité aux ambigüités de sens.

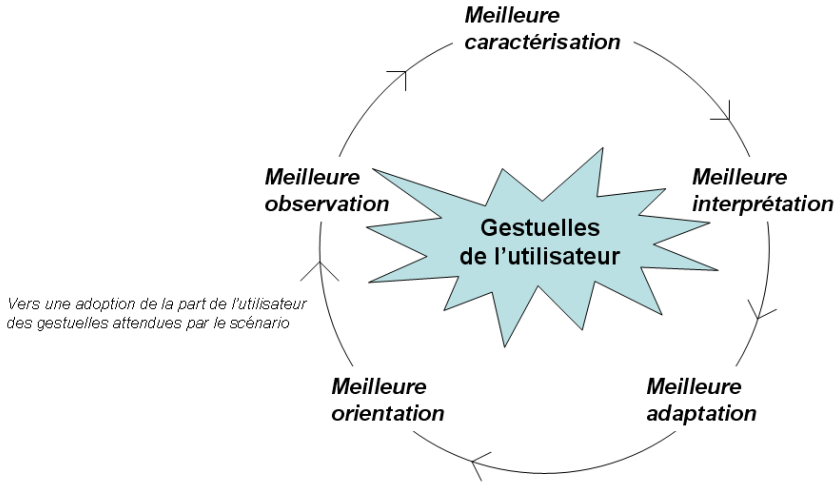


Figure 1.27. : Cercle vertueux d'interaction mis en place par l'adaptativité du système

3.2. Architecture d'un système au sein de l'équipe ImagIN

Au sein de l'équipe, l'attention est tout particulièrement portée sur la conception de l'architecture logicielle d'un système, puisque les différents travaux et projets n'apportent aucune contribution particulière au niveau matériel, du matériel standard et grand public étant toujours utilisé. L'architecture logicielle est définie en suivant le cahier des charges précédemment établi et les objectifs du projet de recherche en cours.

Nous adoptons au cours de cette section une démarche de conception, telle que nous la concevons au sein de l'équipe ***ImagIN***, en étudiant successivement les différentes facettes que peut présenter le processus d'interaction scénarisé adaptatif.

Pour un système donné, la **première étape** est de choisir ou d'élaborer un ou plusieurs paradigmes d'interaction (Section 3.2.1.). Les travaux de l'équipe n'apportent pas de contributions en la matière.
En revanche, au cours de la **deuxième étape** de l'étude du processus d'interaction scénarisé, un modèle d'interaction est proposé au cours de la Section 3.2.2.
La Section 3.2.3., marquant la **dernière étape** de l'étude, abordent les thèmes des interfaces, sans contributions notables de la part de l'équipe ***ImagIN***.
Enfin la Section 3.2.4. présente l'architecture logicielle des systèmes interactifs au sein de l'équipe ***ImagIN***.

3.2.1. Première étape de conception : paradigme d'interaction

Nous pensons que la première étape de conception consiste à élire un ou plusieurs paradigmes d'interaction, vision globale du processus d'interaction scénarisé, à partir de laquelle certaines caractéristiques globales du système pourront être extraites. Relativement aux méta-paradigmes définis par [Beaudouin-Lafon 04], nous percevons le système principalement **comme un outil**. Cependant, certains travaux au sein de l'équipe ***ImagIN*** peuvent adopter les **autres méta-paradigmes** (le système est un partenaire ; le système est un intermédiaire).

Ensuite, nos systèmes obéissent majoritairement aux paradigmes d'interaction ***Post-WIMP*** (interaction gestuelle, bimanuelle et dans un environnement virtuel principalement), soulignant notre volonté d'adopter des paradigmes plus fidèle à celui de la **manipulation directe**.

3.2.2. Deuxième étape de conception : modèle d'interaction

En accord avec le ou les paradigmes d'interaction adoptés, un modèle d'interaction est par la suite élaboré au cours de la deuxième étape de l'étude du processus d'interaction scénarisé.

Les interactants avec le système sont l'Utilisateur et l'Observé. Dans le cadre des travaux de l'équipe ***ImagIN***, l'interactant est toujours considéré comme imprévisible et libre d'adopter n'importe quelle gestuelle. Le modèle de l'interactant exprime uniquement ses capacités de perception et d'action, sans détailler les mécanismes cognitifs qui rentrent en jeu relativement à ces capacités. Nous ne modélisons pas non plus la tâche et l'objectif à proprement parler, si ce n'est comme étant une connaissance spécifique liée à l'interactant. Nous exprimons tout de même le lien qui peut exister entre cette tâche ou cet objectif, et le scénario de l'application. Toute approche relative à la modélisation de l'interactant et de la tâche peut éventuellement être employée. En revanche, notre modèle d'interaction doit permettre par la suite d'élaborer un cahier des charges pour concevoir l'architecture logicielle d'un système. C'est pourquoi il doit comprendre les **modèles du système** et **de l'interaction** qui se déroule entre l'interactant et le système.

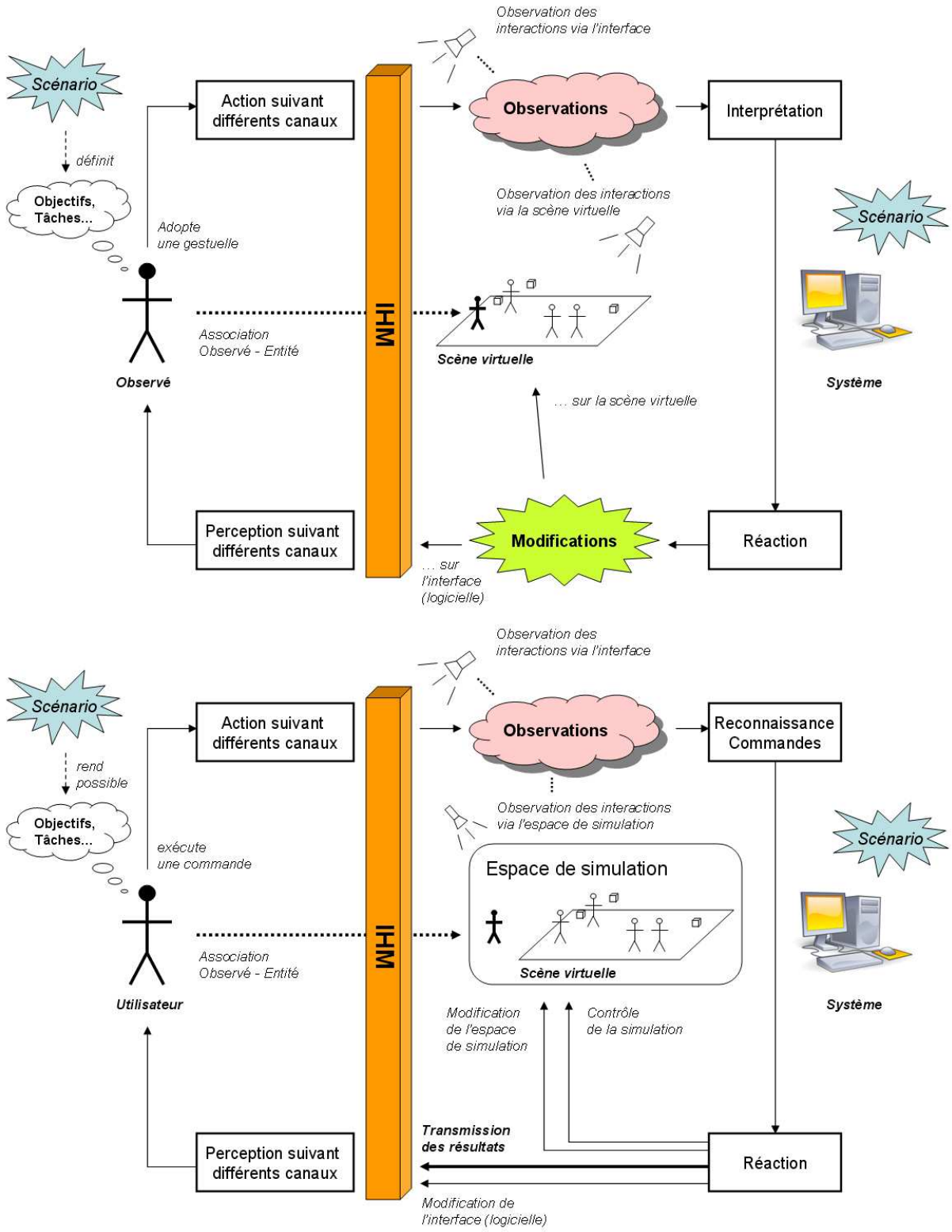


Figure 1.28. : Systèmes développés au sein de l'équipe ***ImagIN*** - Modèle d'interaction

Nous précisons donc comment l'interactivité est envisagée au sein de l'équipe ***ImagIN***. La description du processus d'interaction scénarisé nous permet d'élaborer notre **modèle**

d'interaction (voir Fig. 1.28.). L'élaboration d'un modèle d'interaction va de paire avec l'établissement de notre cahier des charges.

3.2.3. Troisième étape de conception : interface

Généralement, dans le cadre des travaux de l'équipe ***ImagIN***, les interfaces prenant place entre le système et l'Utilisateur, lorsqu'il n'est pas également un Observé, sont majoritairement tirées du **paradigme de l'interaction standard**. Cette forme d'interaction est connue et largement étudiée. Typiquement, l'utilisateur peut observer la simulation par le biais d'une fenêtre graphique dédiée. La représentation de cette simulation peut se faire de manière plus ou moins détaillée. Elle peut en effet s'effectuer aussi bien de manière complètement invisible et accélérée pour l'Utilisateur que de manière détaillée graphiquement et réelle d'un point de vue temporel. L'Utilisateur contrôle la simulation par le biais de *widgets* standards propres aux interfaces de type ***WIMP***.

Les interfaces entre un Observé et le système sont nettement plus intéressantes, en tout cas au niveau des interfaces d'acquisition, puisque ces dernières doivent permettre la **capture des gestuelles naturelles de la personne observée**, ainsi que certains événements spécifiques répertoriés, caractérisant l'activité globale au sein de la scène De plus, la majeure partie des interfaces étudiées au sein de l'équipe ***ImagIN*** se veulent aussi **transparente** que possible. La tendance est donc de développer des NUI. Pour ce qui est des interfaces de restitution, elles doivent permettre l'**immersion** de l'Observé, au sein de la représentation visuelle de la scène virtuelle mise en jeu par une **application multimédia** ou par un **jeu vidéo**. Ces deux types d'applications sont en effet majoritairement développés au sein de l'équipe.

Il est à noter que pour l'Observé et pour l'Utilisateur, l'interface logicielle est l'image indirecte du scénario, ou tout du moins de la situation d'interaction scénarisée courante. En effet, pour l'Observé, cette interface logicielle constitue la représentation visuelle de la scène virtuelle au sein de laquelle il est lui-même modélisé, ou celle de la scène virtuelle telle qu'elle est perçue par son entité virtuelle. Il en est de même pour l'Utilisateur avec l'espace de simulation. Toute gestuelle adoptée de la part des interactants sera mis en évidence par le biais de l'interface logicielle.

3.2.4. Architecture des systèmes de l'équipe ImagIN

A ce stade, le concepteur a adopté un ou plusieurs paradigmes d'interaction, établi le modèle d'interaction qui en découle et élaboré précisément ses interfaces. Le processus d'interaction et le processus d'adaptativité ont donc été étudiés en détails et le concepteur dispose alors d'un ensemble de solutions logicielles permettant l'étude et le développement des différents aspects de ces processus au sein d'un système.

Les systèmes interactifs sont aujourd'hui de plus en plus complexes et doivent répondre, par le biais d'une interaction aussi naturelle que possible, aux besoins toujours croissants des utilisateurs. Il n'est donc plus question **de réduire le développement d'un système à celui de ses interfaces**.
C'est pourquoi, tous les aspects du système doivent tout d'abord être pensés puis conçus, pour enfin être évalués. C'est particulièrement le cas pour l'**architecture du système**, qui décrit les différents modules de traitement du système, ainsi que les relations qu'ils ont entre eux. Elle

reflète également tous les aspects de l'interactivité et de l'adaptativité avec l'interactant, qui sont à prendre en compte et qui sont décrits par le modèle d'interaction correspondant. C'est donc véritablement le plan du système, auquel les différents concepteurs se réfèrent pour l'élaborer concrètement.

La Figure 1.29. présente l'architecture logicielle des systèmes développés au sein de l'équipe ***ImagIN***. Cette architecture a été établie depuis les analyses exposées précédemment, particulièrement grâce au modèle d'interaction et au cahier des charges que nous avons proposés.

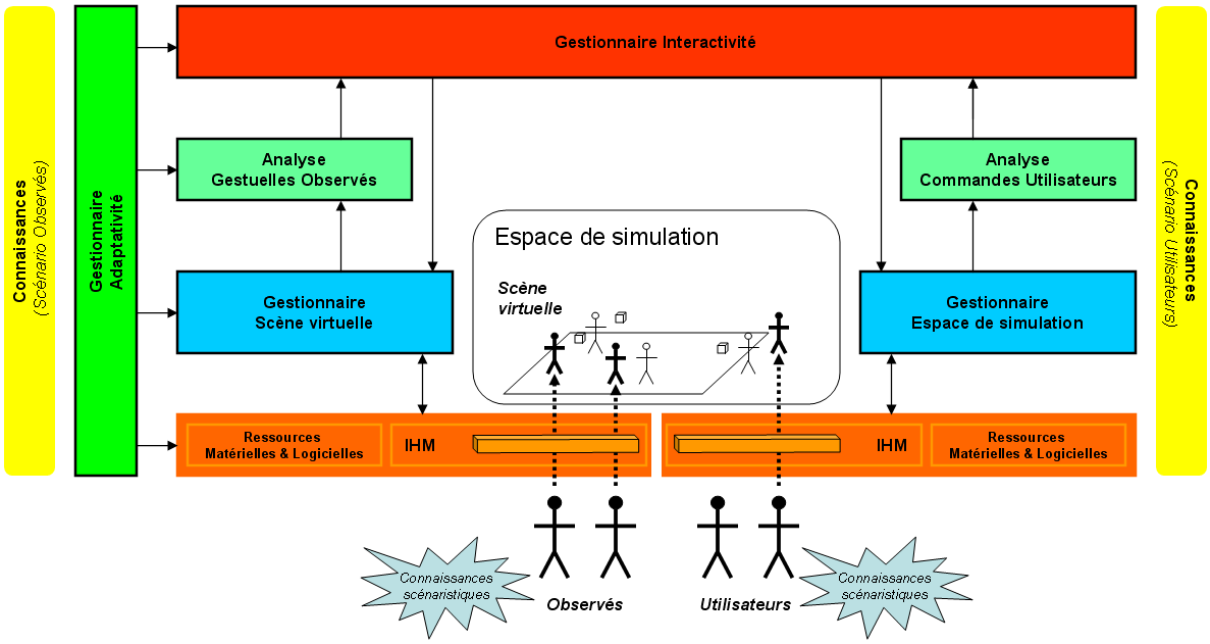


Figure 1.29. : Architecture des systèmes développés au sein de l'équipe ***ImagIN***

4. Conclusion : Positionnement de ces travaux de thèse

Nous proposons dans ces travaux de thèse de développer un **système interactif** sur lequel s'exécute, de manière adaptée, une application scénarisée de type **jeu vidéo**.

Nous souhaitons élaborer un système, avec lequel **un unique utilisateur** interagit, **en temps réel et sans contraintes matérielles** (capture non invasive), par le biais des **gestuelles** de son corps, en accord avec le **scénario** d'une application de type jeu vidéo, dans un contexte d'*exertainment* (*exercise* + *entertainment*).
L'activité au sein de la scène regroupe les **gestuelles de l'utilisateur** et divers **événements réels**, **dont l'utilisateur n'est pas responsable** (modification visuelle de la scène réelle par des mouvements de spectateurs, par exemple). Elle est acquise à partir de la modélisation de la **scène réelle capturée** et de celle d'une **scène 3D interactive**, au sein de laquelle l'utilisateur est immergé. A partir de ces modélisations est mise à jour la **scène virtuelle**, système d'informations virtuelles caractérisant l'état du système à un instant donné.
La scène virtuelle est donc modifiée par l'interaction de l'utilisateur avec le système, ainsi que par la codification d'événements réels dont l'utilisateur n'est pas responsable. A partir de

l'observation de cette scène virtuelle, le **contexte d'interaction** qui englobe **l'activité** est caractérisé. C'est à partir de ce contexte et des indices implicites et explicites qu'il contient que le système reconnaitra l'activité, par le biais d'un processus dédié.
La réponse du système est alors établie en adéquation avec son **interprétation** de l'activité observée au sein de la scène. Dans le cadre du système étudié dans ces travaux de thèse, nous considérons que le scénario de l'application est **fixe**. Au cours de l'interactivité, il est possible d'extraire de ce scénario, les contextes d'interaction attendus à un instant donné. Au cours du processus d'interaction, le contexte d'interaction observé est confronté à celui attendu par le scénario. Nous interprétons donc l'activité à partir de la distance qui peut exister entre ces deux contextes.
En fonction de cette distance, le système met en place sa réponse. Cette réponse alimente **l'interactivité** avec l'utilisateur, mais traduit également la capacité **adaptative** du système, qui adapte à la fois sa réponse interactive à l'utilisateur et son propre fonctionnement global. Notre système doit être tolérant aux erreurs d'interprétation et doit pouvoir notamment prendre en compte des gestuelles utilisateur ayant plusieurs sens possibles, en fonction du contexte dans lequel elles ont lieu.

Comme il le ferait dans la vie réelle, l'utilisateur interagit avec le système, par le biais d'une gestuelle qu'il connaît et qui lui est donc naturelle. Ces caractéristiques traduisent l'adoption des **paradigmes de la manipulation directe**, **de l'interaction gestuelle** et **de l'interaction en environnement virtuel**.

Nous nous situons également par rapport au modèle d'interaction établi au sein de l'équipe ***ImagIN***. Dans ces travaux de thèse, **un utilisateur unique est à la fois l'Utilisateur et l'Observé**. Conventionnellement, nous l'identifierons par la suite par les termes 'utilisateur', ou 'joueur' dans la mesure où nous nous orientons vers le développement de jeux vidéo.
La scène virtuelle est restituée visuellement à l'utilisateur, sous la forme d'une scène 3D interactive, par le biais d'un dispositif d'immersion dédié. L'utilisateur réel est associé à une représentation virtuelle, représentation qui peut lui être restituée visuellement (au sein de la scène 3D) sous la forme d'un avatar. La représentation virtuelle de l'utilisateur évolue au sein de la scène virtuelle, fusionnée avec l'espace de simulation. La scène virtuelle n'est constituée que d'éléments virtuels interactifs avec lesquels le joueur interagit. Autrement dit, la scène virtuelle ne comprend pas d'agents autonomes. Cette notion de scène virtuelle est définie plus en détails au cours du Chapitre 3.

Du côté des interfaces, le joueur adopte un ensemble de gestuelles, empruntées à un vocabulaire qu'il connaît. L'ensemble de ces gestuelles, véritables **techniques d'interaction**, constituent l'**interface comportementale** de notre système. Dans nos travaux, la gestuelle utilisateur englobe les mouvements, gestes, actions et comportements de l'utilisateur, qu'ils soient explicites et intentionnels ou implicites et non voulus. Nous définirons plus précisément cette notion de gestuelle au cours du Chapitre 4.
Le joueur est immergé et évolue au sein d'une scène 3D, avec laquelle il interagit. La scène 3D restituée est extraite de la scène virtuelle, en accord avec le scénario de l'application. Elle constitue l'**interface logicielle** de notre système.
La gestuelle du joueur est acquise par le biais d'un système commercial de capture, le ***Cyberdôme***, de la société XD Productions. L'interface étant intrusive, nous verrons, au cours du Chapitre 3., comment il est possible de rendre cette interface autant que possible transparente. Nos contributions ont également permis de généraliser ce système à une capture de l'activité plus globale au sein de la scène réelle. Le ***Cyberdôme***, augmenté de nos

contributions, constitue l'**interface d'acquisition** de notre système. L'**interface de restitution** est un dispositif d'immersion, restituant à l'utilisateur la scène 3D, image visuelle de la réponse du système vis-à-vis de la gestuelle utilisateur. Ces interfaces constituent l'**interface matérielle** de notre système.

Le système met en œuvre vis-à-vis du joueur une interaction **'tour de rôle'**, impliquant la succession permanente d'une action du joueur avec une réaction du système (voir Fig. 1.30.), sachant que cette réponse du système peut ne pas être perceptible de la part du joueur (traduisant un état d'attente ou de sommeil du système par exemple).
Au cours de l'interactivité, le joueur adopte la gestuelle qu'il juge pertinente vis-à-vis du scénario de l'application. Il interagit avec la scène 3D qui lui est restituée de manière immersive. L'état de la scène virtuelle est alors modifié par la gestuelle capturée de l'utilisateur, et, éventuellement, par des événements prenant place au sein de la scène réelle et dont l'utilisateur n'est pas responsable (dus aux mouvements de spectateurs, par exemple). L'état de la scène virtuelle permet la caractérisation et l'interprétation, par des processus dédiés du système, de cette activité. Le système y répond alors en modifiant l'état de la scène virtuelle, ses connaissances et son état interne. C'est à partir de l'état de la scène virtuelle qu'est mise à jour la scène 3D interactive, qui est restituée à l'utilisateur, lui transmettant ainsi une image visuelle de la réponse du système à sa gestuelle antérieure.

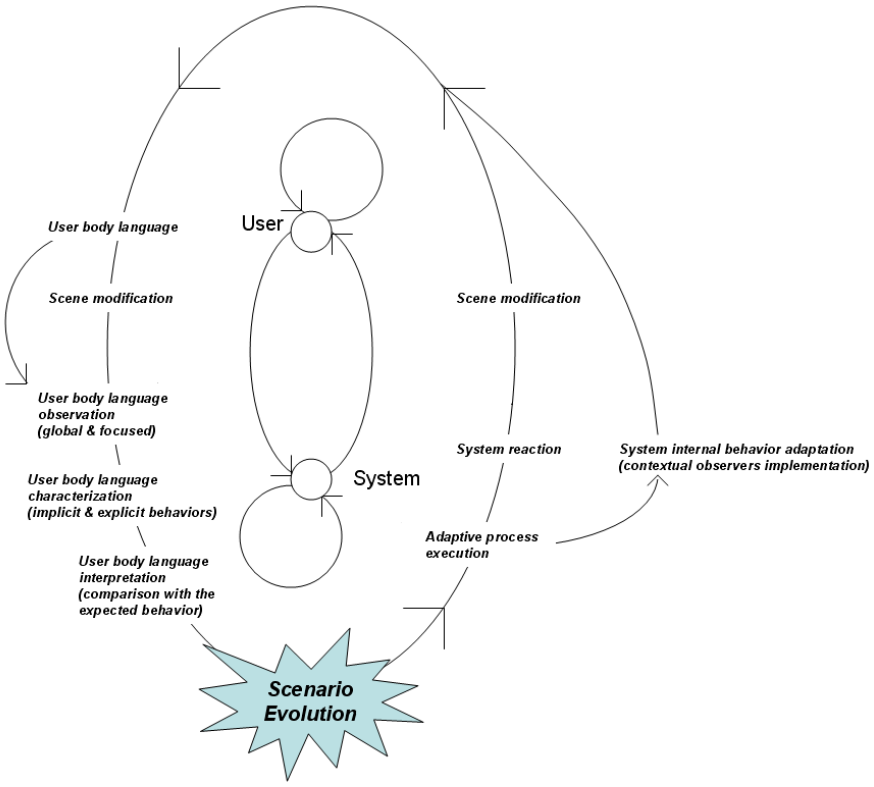


Figure 1.30. : Interaction « tour de rôle »

L'adaptativité se traduit au niveau de la **scène virtuelle** et du **fonctionnement interne du système**, par des mécanismes simples. L'adaptation est à la fois interne et externe [Oreizy &

al. 99]. Ce processus est dirigé par le scénario de l'application. La logique application permet de mettre en place une réponse adaptée du système à l'utilisateur. Cette adaptation se déclinera sur les différents niveaux sémantiques du système, jusqu'à la restitution de la réponse visuelle à l'utilisateur. Ainsi, par le biais de la logique système, chaque processus mis en jeu dans cette réponse pourra adapter son comportement.
Au niveau de la scène virtuelle, l'adaptativité se traduira par la gestion efficace et anticipée des ressources interactives au sein de la scène (ajout/suppression, activation/désactivation, modification de leur apparence visuelle dans la scène 3D, de leur comportement, etc.). Au niveau du système, nous chercherons à orienter les différents processus mis en jeu au cours de l'interactivité : capture de la scène réelle ; gestion de la scène 3D ; gestion de la scène virtuelle ; caractérisation du contexte d'interaction observé ; interprétation de l'activité en fonction de la distance existant entre le contexte d'interaction observé et le contexte d'interaction attendu, extrait du scénario. L'adaptativité au niveau du système se traduira également par une gestion optimisée et anticipée des ressources logicielles et matérielles.
Que ce soit au niveau de la scène virtuelle ou du système, ces mécanismes permettent également de mettre en place un certain nombre d'**observateurs**, symbolisant le contexte d'interaction au sein duquel l'utilisateur est censé interagir. Si certains événements spécifiques ont lieu dans les cadres d'étude observés, le contexte est reconnu et le scénario peut évoluer. Les informations contextuelles fournissent un cadre de vision de la scène. La gestion de contextes permet donc de clarifier de désambigüiser l'activité. Aucune gestuelle de sa part n'est obligatoire. Mais le système adaptera ses réactions de manière à placer constamment l'utilisateur dans le contexte d'interaction attendu par le scénario. De plus, il est parfois difficile de suivre finement et en permanence les gestuelles utilisateur. La détection de déclencheurs simples spécifiques à un instant particulier, au niveau de la scène 3D, peut permettre de palier cette difficulté.

Les **connaissances** supportant l'interactivité et l'adaptativité sont multiples, et peuvent être statiques, c'est-à-dire définies avant l'exécution de l'application et introduites par les concepteurs du système, et/ou dynamiques, évoluant et se construisant au cours des interactions entre l'utilisateur et le système. Une partie de ces connaissances permettent de déterminer le contexte d'interaction au sein duquel l'activité est censée prendre place.
Le **scénario** de l'application est la source principale de connaissances. Il définit, d'une part, les objectifs de l'utilisateur, ainsi que la scène virtuelle au sein de laquelle il évolue, et dans laquelle il est représenté. D'autre part, il décrit les contextes d'interaction que le système devra attendre et observer. En fonction du contexte d'interaction observé, le scénario définit également la réponse adaptée du système, traduite par le biais de la scène virtuelle, et les mécanismes adaptatifs associés.
L'**utilisateur** représente également un ensemble de connaissances, introduites au sein du système au moment de sa conception, en tant qu'hypothèses. Ces connaissances, allant du fait que nous observons un être humain, à la morphologie et au profil (culture, préférence, etc.) de ce dernier, sont nécessaires au bon fonctionnement des différents processus mis en jeu au cours de l'interactivité. Par exemple, nous savons *a priori* que nous capturons les gestuelles d'un être humain, dont nous connaissons la morphologie.
Il en est de même pour l'**environnement physique réel**, au sein duquel l'utilisateur évolue. Par exemple, il sera nécessaire, avant l'interactivité à proprement parler, de calibrer cet espace, c'est-à-dire de calculer les matrices permettant de faire la correspondance entre une image caméra en 2D et la scène 3D.
Il existe également un ensemble de connaissances, qui font office de **paramètres** pour les différents processus mis en jeu au cours de l'interaction, particulièrement les algorithmes de traitement d'images et de caractérisation et d'interprétation de l'activité.

Enfin, **l'interactivité** en elle-même génère dynamiquement une accumulation de connaissances (historique, notion de passé et de futur), à court terme, qui pourront être considérés par les différents processus du système. Les interactions de l'utilisateur, détectées à partir de la scène 3D, sont généralement explicites et non ambigües, tandis que ses gestuelles réelles capturées ne fournissent pas systématiquement une intention claire de sa part. Les connaissances établies à partir de cette capture peuvent cependant être utilisées pour enrichir la sémantique du contexte d'interaction caractérisé.
Ces différentes connaissances pourront être fusionnées, pour atteindre de nouveaux niveaux de connaissances. Tout d'abord, à partir de ces différentes connaissances, et à la suite de l'observation et de l'interprétation de l'activité, il est alors possible de modéliser les informations numériques et sémantiques qui seront utilisées par les mécanismes adaptatifs. Ensuite, ces mécanismes adaptifs permettent d'orienter l'utilisateur et les différents processus actifs au cours de l'interactivité. Suivant les connaissances disponibles et l'état courant de ces processus, de nouvelles connaissances sont modélisées dans le but d'améliorer le processus d'interprétation.

Nous établissons par le biais de la Figure 1.31., l'architecture générale de notre système. Celle-ci est une spécialisation de l'architecture générale des systèmes développés au sein de l'équipe ImagIN, en accord avec nos objectifs de thèse et notre contexte industriel.

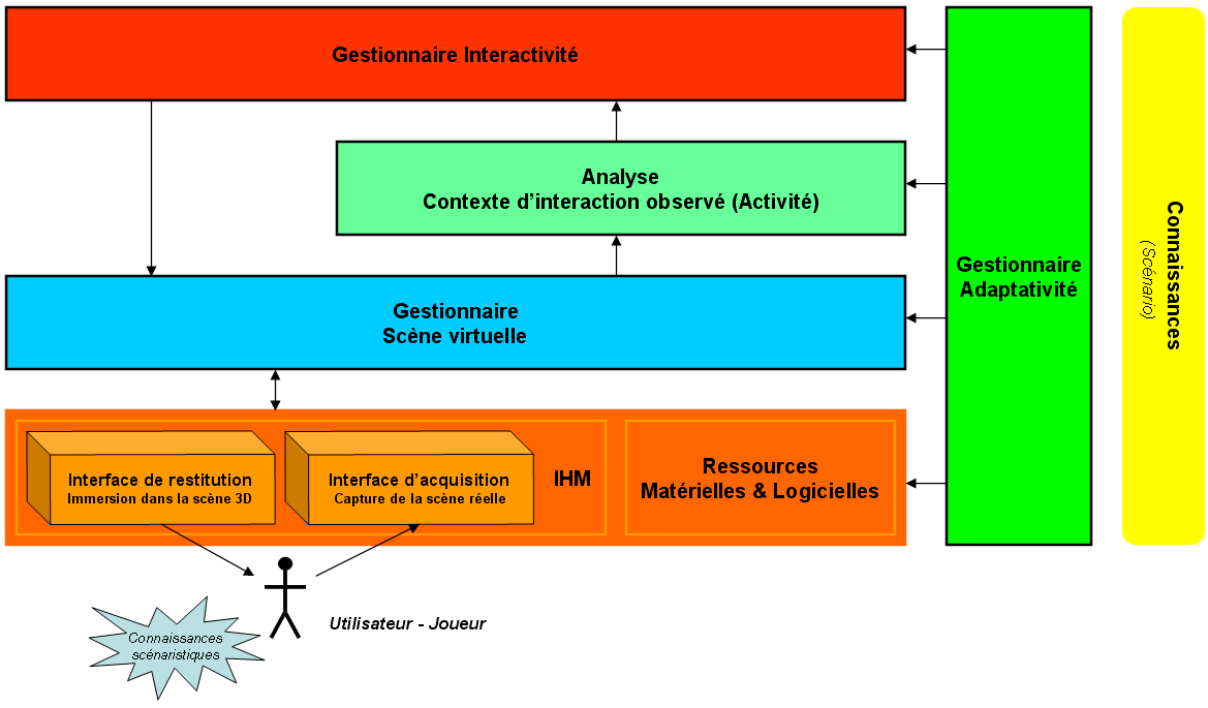
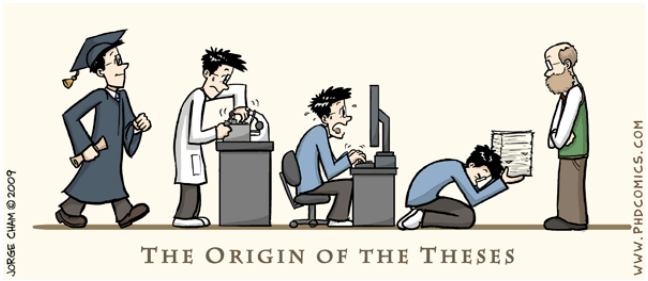


Figure 1.31. : Architecture générale de notre système

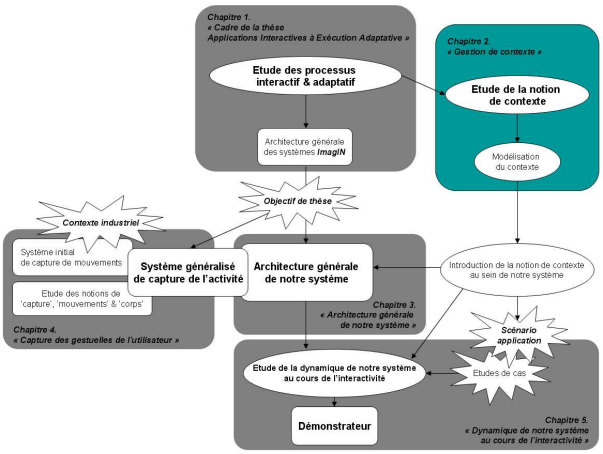
Chapitre 2. Gestion de contexte



La clé de recherche requise n'a été trouvée dans aucun contexte d'activation actif
- Message d'erreur, Windows XP -.

La tarte au chocolat, c'est comme le mouton, ça perd ses plumes
- AOM -

Résumé



Dans ces travaux de thèse, nous faisons l'hypothèse que **le contexte d'interaction retranscrit l'activité prenant place au sein de la scène observée**. Sa **caractérisation** et son **interprétation** permettront au système de répondre en adéquation avec l'activité observée, que cela soit au niveau de l'utilisateur que vis-à-vis de son propre fonctionnement.

Nous verrons que l'interprétation d'un contexte d'interaction observé peut se faire grâce au support du **scénario de l'application**, à partir duquel sont extraits **les contextes d'interaction attendus** à différents moments clés. De plus, les informations contextuelles permettent **le paramétrage des mécanismes adaptifs**, mis en place par le système au cours de sa réponse, orientant et cadrant les différents traitements.
Ce chapitre propose donc **l'étude de la notion de contexte**. Nous définissons précisément cette notion, aussi bien du point de vue de l'humain que du système interactif. Par la suite, nous passons en revue les différentes approches pour modéliser un contexte au sein d'un système. Nous constatons ainsi qu'il existe plusieurs sous contextes que nous identifions. Nous présentons également plusieurs méthodologies pour gérer un modèle de contexte au sein d'un système. Le chapitre se termine par un positionnement vis-à-vis des différentes études exposées, et par la présentation de nos contributions. Nous proposons un modèle de contexte complet, nous permettant la modélisation précise du scénario d'une application.

Plan du chapitre

Dans le cadre de ces travaux de thèse, nous voulons développer un système interactif s'adaptant à l'activité observée au sein de la scène. Une étape préalable à l'adaptation du système est donc celle de l'**interprétation cette activité**. Dans nos travaux, l'activité au sein de la scène regroupe :

(1) les **gestuelles réelles** de l'utilisateur, capturées par le biais d'un système commercial de capture de gestuelles, augmenté de nos contributions. Ces gestuelles peuvent traduire les **interactions caractérisées**, intentionnelles ou non, de l'utilisateur, au sein de la scène 3D immersive ;
(2) certains **événements** caractérisés, prenant place au sein de la scène réelle, générés par un **dynamisme** dont l'utilisateur n'est pas responsable.

Cependant, ce problème n'est pas si simple à résoudre, et l'interprétation d'une activité reste aujourd'hui un problème ouvert, même s'il est largement étudié, comme nous le verrons dans la suite de ce chapitre. L'interprétation d'une activité nécessite en effet une **capture** fine, précise et robuste, une **codification** appropriée et compacte, ainsi qu'une **classification** de celle-ci au sein d'un espace restreint d'activités répertoriés. De plus, une activité peut avoir **un grand nombre de sens et de buts différents**, dans des **contextes d'interaction distincts**.

Dans tout processus, la notion de **contexte** est présente à plusieurs niveaux. En effet, une équation mathématique est appliquée – une gestuelle est interprétée ou attendue – une interaction prend place dans le cadre d'un contexte spécifique. Nous pensons que la modélisation du contexte d'interaction, au sein duquel l'activité prend place, constitue un **support indispensable** à l'interprétation cette activité et nous permettra d'améliorer et de fiabiliser ce processus. Les gestuelles, spontanées ou réfléchies, de l'utilisateur reflètent sa compréhension du contexte d'interaction dans lequel il évolue. La modélisation de ce contexte permettra de replacer la gestuelle observée au sein de celui-ci. Ainsi, il est possible aussi bien de limiter les plus que nombreuses possibilités d'interprétation des gestuelles de l'utilisateur, que de résoudre les ambigüités de sens que ces dernières peuvent présenter.

La **modélisation** et la **gestion** du **contexte d'interaction** s'avère possible grâce au support du **scénario de l'application**, clairement défini au moment de son développement. En effet, un scénario peut être considéré comme un livre déjà lu. Comme son évolution est prévisible, il est possible d'attendre de la part de l'utilisateur des **gestuelles spécifiques** (aussi bien implicites qu'explicites) à un **instant donné**. Autrement dit, le scénario évolue lorsque certaines étapes particulières sont atteintes. Et ces étapes sont atteintes une fois que l'utilisateur a adopté des gestuelles particulières, pour réaliser les objectifs définis dans le contexte d'interaction courant.
Le **scénario** doit donc non seulement définir les **gestuelles** que l'utilisateur doit adopter à des moments spécifiques, mais également préciser le **contexte d'interaction** au sein desquels ils ont lieu : quelle a été la gestuelle précédente ? Quelle sera la gestuelle suivante ? Où et quand observer la gestuelle actuelle ? Il décrit le **contenu de l'environnement avec lequel** l'utilisateur interagit à un instant donné, **la manière** dont il interagit avec celui-ci (d'un point de vue gestuel) et **des événements qui en découlent**. Il peut également prendre en compte d'autres événements, qui peuvent éventuellement influer sur les gestuelles de l'utilisateur, les observations pouvant être faites de la scène, etc.
Les contextes d'interaction, évoluant en même temps que le scénario de l'application, sont déterminés **au moment du développement de l'application**. Les connaissances qu'apportent le scénario, et donc ces contextes d'interaction, s'ajoutent à celles **introduites au sein du**

système par les concepteurs de ce dernier, ainsi qu'à celles **se construisant dynamiquement** au cours de l'interactivité.

Le contexte d'interaction fournit donc un ensemble d'informations **décisives** à la bonne interprétation de l'activité qui a lieu au sein de celui-ci. Ces informations permettent de mettre en place des **mécanismes de capture et d'observation** spécifiques de la scène (voir Fig. 2.1.). Les événements qui ont lieu au sein de ces cadres de capture et d'observation déterminent si le contexte d'interaction attendu par le scénario a bien été observé. L'activité peut alors être analysée en détail.

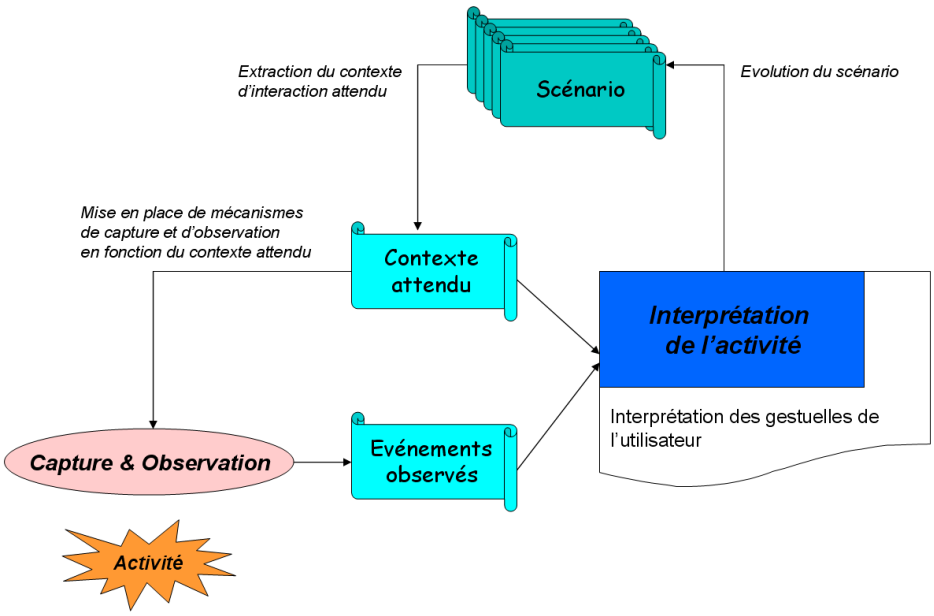


Figure 2.1. : Mise en place d'observateurs en fonction du contexte attendu

Une fois le contexte d'interaction observé décrypté et l'activité, particulièrement la gestuelle utilisateur, analysée, il est possible de mettre en place la **réponse adaptative** du système, toujours grâce au support du scénario. Si l'utilisateur ne se trouve pas dans le bon contexte d'interaction ou s'il n'effectue pas les gestuelles attendues par le scénario, le système se chargera de répondre en adéquation, de manière à placer constamment l'utilisateur dans le bon contexte d'interaction. De plus, pour un objectif donné, que l'utilisateur doit atteindre, le système peut être capable d'évaluer la séquence des gestuelles supposées être adoptées par celui-ci, grâce au scénario de l'application. Ce processus, qui peut être considéré comme une forme de prédiction, peut permettre au système d'optimiser à court terme les ressources logicielles et matérielles dont il aura besoin dans la suite de l'interactivité et d'anticiper différents traitements spécifiques mis en jeu au cours du processus d'analyse.

Ce chapitre se propose au départ de présenter un état de l'art autour du concept du **contexte d'interaction**.
Bien que largement étudié, et dans tous les domaines, le contexte reste difficile à définir. La Section 1. se propose d'étudier cette notion, aussi bien au niveau de l'**être humain** que d'**un système interactif**. Nous verrons que le contexte est mis en jeu lors du processus de

contextualisation d'une situation observée, processus permettant l'explication de cette dernière. Le contexte est donc fortement lié aux notions de **situation** et d'**activité**. Nous éclaircirons ces différents concepts au cours de la section. Puis, nous conclurons sur un **historique** et un **état de l'art** de l'étude du contexte dans le domaine de l'***IHM***.
La Section 2. passera en revue différentes approches pour **modéliser informatiquement un contexte d'interaction**. Ce contexte doit englober de nombreux aspects de l'interaction et doit être facilement et rapidement gérable au sein d'un système interactif. La section débute par l'énumération des raisons pour lesquelles la modélisation du contexte peut s'avérer indispensable. **Mis en œuvre au sein d'un scénario**, un contexte d'interaction permet de mettre en place des mécanismes d'interprétation de la situation observée. Les contextes d'interaction sont caractérisés par le biais de **données bas niveau**, captées par le système. Cette section se conclura par la présentation des **propriétés** d'un bon modèle de contexte.
Il est donc nécessaire de caractériser le contexte d'interaction d'une situation donnée, de manière à l'expliquer. Cependant, il est bien évident qu'**il n'existe pas qu'un seul contexte**, et que chaque contexte est caractérisé par un nombre important et une grande diversité de paramètres **captés** par le système. La Section 3. s'occupe d'énumérer **les différents contextes** qu'il nous parait utile de caractériser pour une bonne **interprétation de la situation observée**. Les différents paramètres, qui permettent la caractérisation de chaque contexte, sont également partiellement présentés. La section se conclue sur une discussion autour du **travail du développeur de l'application**, qui cherche à déterminer de quels contextes il peut avoir besoin, et à partir de quelles informations il peut construire ces derniers.
La Section 4. traite du problème de la **gestion du modèle de contexte au sein d'un système interactif**. Nous mettrons en évidence le besoin de couches logicielles intermédiaires, dédiées à la gestion du contexte, au sein d'un système. Ces couches, appelés des **serveurs de contexte**, se chargent de récupérer les données des capteurs et de les interpréter, pour les transmettre au reste du système. Nous terminerons la section en présentant différentes **problématiques spécifiques**, auxquelles le concepteur devra faire face lors de la conception de systèmes sensibles au contexte.
Enfin, la Section 5. positionne ces travaux de thèse vis-à-vis de l'état de l'art précédent. La dernière section, la Section 6., expose alors notre contribution relativement à la gestion du contexte au sein d'un système informatique. Nous proposons un modèle se basant sur la modélisation de l'univers, et des **acteurs de l'interaction** le composant, par un **vecteur d'états**. Ce vecteur évolue au cours d'un scénario, modélisé par le biais de **situations** et de **contextes d'interaction**. Nous verrons que notre approche est fiable et peut s'appliquer à de nombreux cas théoriques.

L'étude du contexte d'interaction et de son utilisation au sein d'un système peut faire face à des problèmes de protection de contenu, de perte ou de vol de données privées, etc. En effet, certains systèmes, pour accomplir leurs objectifs, doivent pouvoir localiser une personne à tout moment, déterminer son activité, n'importe quand et n'importe où, et communiquer l'ensemble de ces informations. **Ces problèmes de sécurité et de vie privée ne seront cependant pas abordés dans le cadre de ces travaux de thèse.** Nous étudions le contexte de manière à étudier le comportement de l'utilisateur et à mettre en place un processus d'adaptativité, dans un système privée, peu important en termes de taille et de matériel et pour des applications type jeux vidéo. Ces problèmes ne seront donc pas pris en compte dans le cadre de notre étude. Toutefois, le lecteur intéressé pourra se référer à [Chen & Kotz 00], qui traitent partiellement du sujet.

1. Etude de la notion de contexte

La modélisation et la gestion du contexte est un sujet largement étudié dans les domaines de **l'interaction** et de **l'interprétation de l'activité, particulièrement l'activité humaine**. « *Context is key* » [Coutaz & al. 2005] : le contexte est la clé, le contexte est partout et explique tout. Car le contexte influence et impacte sur notre façon de percevoir, d'interpréter et donc de faire. D'un contexte à un autre, une même gestuelle n'a plus la même signification.

Cependant, la notion de contexte est difficile à définir et tend à rester vague et confuse. Elle intervient à tous les niveaux et dans tous les domaines. Elle est la source de nombreuses interrogations, auxquelles nous ne répondrons pas exhaustivement.
Le terme **'contexte'** vient du latin *contextere* : tisser avec, entrelacer avec. Il est défini comme étant l'ensemble des circonstances qui entourent un fait, au sein desquelles ce dernier s'insère[1]. Le processus de contextualisation est un processus actif, dans lequel un certain nombre d'**observateurs** sont impliqués. Ces derniers 'tissent' leurs connaissances (expérience, connaissances acquises ou intuitions, etc.) autour d'un fait, pour lui donner un sens. Il sera question de contexte familial, de contexte politique, de contexte actuel ou en encore du contexte de l'économie mondiale. Il a pour synonymes, toujours discutables, les mots **'situation'**, **'conjoncture'**, **'circonstances'** ou encore **'environnement'**.

Il est impossible de parler de la notion de contexte, sans parler de celles de **situation**, d'**activité**, d'**acteurs**, de **scène**…
Une **scène** est observée, d'un ou de plusieurs points de vue particuliers, c'est un espace physique délimité, au sein de laquelle les **acteurs** évoluent. Un acteur peut être l'utilisateur, le système, un processus du système, un périphérique, etc.
L'**activité** prend donc place au sein de la scène et décrit la logique des actions effectuées par les différents acteurs. L'activité laissera la place à l'interactivité lorsqu'il s'agira d'échanges (ou interactions) entre plusieurs acteurs. De cette activité en résultent un certain nombre d'**événements**, modifiant alors les états des acteurs et de la scène.
Une **situation** décrit une étape déterminée de l'activité, englobant cette dernière ainsi que la scène et les acteurs qui en sont le support. Elle est donc le cadre scénarisé de l'activité, dans la mesure où la mise en situation des acteurs définit leurs objectifs, et donc leurs comportements. Le **contexte** est une notion plus subtile à définir car il peut englober une partie de l'activité au sein d'une situation ; la situation peut être considérée comme le contexte de l'activité ; et la situation peut être englobée au sein d'un contexte particulier.

La notion de contexte est donc étroitement liée à celle de situation. Ces deux notions sont pour nous bien distinctes et nous considérons qu'une situation, que cela soit pour un être humain ou un système informatique, est une portion délimitée et échantillonnée, de l'univers, qui est lui infiniment descriptible (Section 1.1.).
Nous définirons alors ensuite la notion de **contexte**, tout d'abord au niveau de l'être humain (Section 1.2.). Notre approche est progressive et s'appuie sur plusieurs travaux, aussi bien dans le cadre d'étude des mécanismes cognitifs chez l'**être humain** que dans le domaine de l'**Interaction Homme-Machine**. Nous verrons que le contexte est un ensemble d'informations paramétrant le **processus de contextualisation**, processus s'opérant dans le but de permettre de donner un sens à une situation observée.

[1] Le Nouveau Petit Robert de la Langue française, 2000

La Section 1.3. met en évidence **la sensibilité de l'activité vis-à-vis de la situation expliquée à travers le contexte d'interaction**, démontrant ainsi combien il est important de modéliser le contexte pour interpréter la gestuelle d'une personne.
La Section 1.4. présente un **historique** de l'étude du contexte dans le domaine de l'**Interaction Homme-Machine**.

1.1. La situation

Plaçons-nous dans un univers infiniment descriptible que nous, êtres humains, voulons observer. Il n'est pas possible pour nous de considérer l'ensemble des stimuli extérieurs, déjà car nous ne pouvons en mesurer ni l'ampleur, ni la portée [Greenberg 01], mais également parce que nous le pouvons pas physiquement, nos capacités sensorielles ne nous le permettant pas.

Pour un être humain, une **situation** représente sa **perception de l'état de l'univers à un instant donné**. C'est un échantillon, un sous-ensemble de l'univers, délimité, descriptible et énumérable. Cette définition est en accord avec celle de [Schmidt 02], qui décrit la situation comme étant l'état du monde réel, à un instant et endroit donnés. Une partie de ces états sera alors relatif à la scène, tandis qu'une autre aux acteurs.

Une même situation est différente d'une personne à une autre, chaque personne pouvant en plus la décrire selon différents niveaux d'abstraction. En effet, la description d'une situation est dépendante du profil, du passé, de l'expérience, des émotions… des **connaissances** au sens plus général, d'un être humain. C'est ce qu'est une situation au niveau de l'être humain, mais cette définition est également vraie pour des processus actifs plus simples. Une fonction informatique, par exemple, prendra en entrée un certain nombre de paramètres, à un instant donné, qu'elle utilisera au cours du traitement qu'elle décrit. L'ensemble de ces variables au cours du temps constitue l'univers, tel que la fonction peut le percevoir. Cet exemple est pertinent car pour cette fonction, le monde s'arrête là, elle ne perçoit que ce qui lui est présenté et n'ira pas plus loin.

1.2. Le contexte

Au cours de cette section, nous allons étudier les différentes approches définissant la notion de contexte. Cette notion est à discuter avec celle de situation, le contexte permettant de donner une signification à une situation observée. Ce processus est appelé le **processus de contextualisation** d'une situation donnée. Nous allons tout d'abord constater que le processus de contextualisation ne se résume pas qu'à un simple processus de comparaison entre les informations contextuelles acquises lors d'une expérience passée, avec les états enregistrés de la situation courante observée (Section 1.2.1.). **Le processus est plus complexe**, aussi bien au niveau de l'être humain que des systèmes sensibles au contexte. Il permet la **reconstruction de la scène**, par le biais du contexte modélisé à partir de la situation observée, des connaissances introduites par le concepteur au sein du système et des objectifs de l'utilisateur (Section 1.2.2.). Nous conclurons en ouvrant une parenthèse sur la **confusion possible entre la notion de situation et celle de contexte**, parenthèse que nous refermerons en mettant en évidence leur forte interdépendance (Section 1.2.3.).

1.2.1. La contextualisation : un simple processus de comparaison ?

Dans un premier temps, une des approches pour définir le contexte pourrait être celle de [Schmidt 02]. Dans ces travaux, le contexte est une **description générique d'un type de situation**. Cette description comprend une valeur d'identification et un ensemble de conditions et de paramètres, chacun étant associé à un intervalle de valeurs. Un contexte décrit donc une classe de situations. Une situation observée appartient alors à un contexte donné, si ses états respectent l'ensemble des conditions et des intervalles de valeurs définis par le contexte. La contextualisation de la situation observée s'opère donc en vérifiant si ses états obéissent aux conditions contextuelles ou s'ils sont compris dans les intervalles de valeurs définis par le contexte.
Pour l'être humain, le processus est similaire. Nous pouvons identifier une situation donnée en nous référant à une expérience passée, en reconnaissant cette situation pour l'avoir vécue antérieurement. Mais le contexte et le processus de contextualisation chez l'être **humain ne peuvent pas se résumer qu'à une simple comparaison** entre une situation observée et un patron contextuel de celle-ci. En effet, il est tout à fait possible de se trouver face à une situation non vécue dans le passé et de savoir tout à fait de quoi il s'agit.

1.2.2. Un processus plus complexe

Le contexte en soi n'est pas le plus important, c'est le processus qui l'utilise. Un contexte est en effet un ensemble d'informations, dites **contextuelles**. C'est le processus de contextualisation, actif et dynamique, qui permet de caractériser et d'expliquer une situation observée, processus utilisant les informations contextuelles sélectionnées, regroupées et structurées. La **situation contextualisée** n'est pas modifiée, elle est simplement mieux décrite et donc mieux expliquée.
Dans [Dey 01], le contexte est « l'ensemble des informations qui peuvent être utilisées pour caractériser la situation d'une entité », une entité étant définie comme étant une personne, un environnement ou un objet. Une situation devient alors la description des états de l'ensemble des entités impliquées, à un instant donné et la situation contextualisée, la description des états pertinents d'entités particulières. Le processus de contextualisation devient donc une visualisation filtrée à travers un contexte donné. C'est ainsi que cela se passe au niveau de l'être humain.
Le processus de contextualisation (voir Fig. 2.2.) a été étudié par de nombreux auteurs, sur le plan du **mécanisme cognitif** s'opérant chez l'être humain. [Cohen & al. 98] établit que le processus de contextualisation, paramétré par un ensemble d'informations contextuelles, modère le flux de traitements prenant place de l'acquisition des stimuli extérieurs à la réponse physique. Typiquement, relativement à l'objectif de la personne et aux connaissances contextuelles, la fonction principale de ce processus est de pondérer les différentes propriétés d'un stimulus extérieur. Ainsi, les stimuli pertinents (vis-à-vis du contexte), pour l'accomplissement de la tâche envisagée, contribuent plus fortement lors de l'établissement de la réponse motrice. Le processus de contextualisation a également été étudié par [Hommel & al. 00], le définissant comme un processus automatique, permettant à l'être humain de réagir de manière adéquate face à une situation donnée. [Van Oers 98] rajoute que plus une entité participe à une interaction donnée, plus cette entité construit vite et précisément un modèle de contexte pour la situation correspondante. Dans [Suchman 87], il est établi que l'activité d'un

être humain dépend de la situation dans laquelle il se trouve. Nous lui préférons l'expression **« situation contextualisée »**.

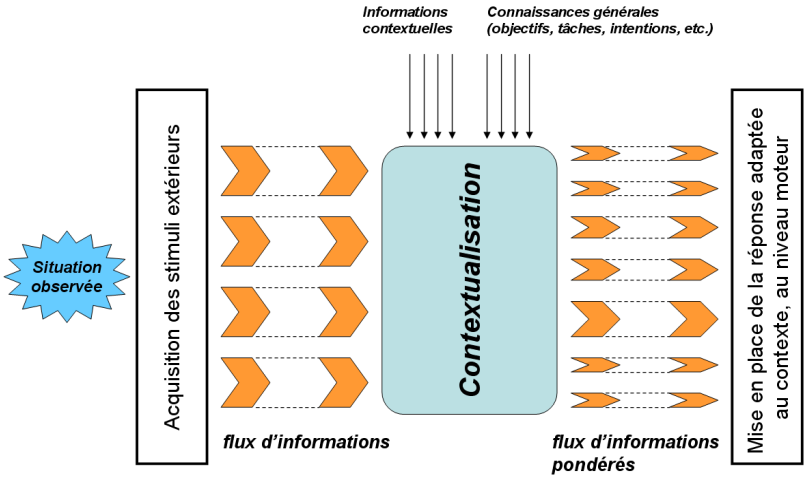


Figure 2.2. : Processus de contextualisation chez l'être humain

Dans le cas de la conception d'un **système interactif sensible au contexte**, la scène est observée par le biais de capteurs et de périphériques. Une situation à un instant donné est un ensemble de stimuli, perçus par le biais de ces interfaces. Il n'est bien sûr pas question de modéliser l'ensemble de l'univers, c'est pourquoi les systèmes sont conçus dans le cadre d'une application spécifique ou un ensemble de tâches particulières, et prennent en compte un environnement clos et bien délimité. Le contexte devient **une image restreinte, caractéristique et discriminante de la situation observée**, au sein de laquelle prend place **l'activité**, en accord avec les différentes **connaissances** introduites au sein du système par les concepteurs et les **objectifs** de l'utilisateur. Etudier le contexte et le modéliser reviendra donc à reconstruire la scène perçue, sous différents aspects (spatiaux, temporels, etc.) en accord avec ces connaissances et ses objectifs, comme c'est par exemple le cas dans [Bremond 97].

1.2.3. Situation vs Contexte

Il est possible de rencontrer dans la littérature, une **confusion** entre la notion de situation et celle de contexte. Cette confusion est du au fait que la notion de contexte est présente à **trois niveaux** vis-à-vis de la situation.
Premièrement, **le contexte peut être englobé par une situation**, concernant alors une partie de l'activité (le contexte d'une action dans une situation donnée). Dans ce cas, la situation est considérée comme étant l'univers, infiniment descriptible. Le contexte devient alors l'échantillonnage en états discrets de cette situation, échantillonnage que le système pourra traiter.
Deuxièmement, **un contexte peut englober plusieurs situations, au sein duquel elles sont structurées**. Le contexte regroupe alors un ensemble d'informations pour l'ensemble des situations, ensemble statique dans le sens où la nature de ces informations n'évoluera pas tant que le contexte est valide. Cependant, un ensemble donné d'informations sera considéré et

modifiée au cours d'une situation donnée. D'une situation à une autre, ces informations, sous-ensemble de l'ensemble des informations contextuelles, ne sont donc pas les mêmes. Le contexte peut être alors considéré comme notion dynamique, dans le sens où les informations contextuelles associées évoluent au fur et à mesure que les situations contextualisées se déroulent.
Et enfin, troisièmement, **la situation peut être considérée comme le contexte local de l'activité**.

Dans nos travaux, la situation est caractérisée par l'ensemble des états bruts captés lors de l'observation de la scène. Comme établi dans [Dey 01], le contexte est l'ensemble des informations permettant d'expliquer la situation. Ce peut être une modélisation restreinte et pertinente de la scène, mais également un ensemble de processus.
Toujours est-il qu'aussi bien chez l'être humain que pour un système informatique, **les notions de situation et de contexte sont très interdépendantes**. [Hommel & al. 00] montre que la perception d'une situation dépend des informations contextuelles considérées mais que ces informations, leur sélection, leur regroupement et leur structuration, sont fonction de la situation perçue. Dans ces travaux, autant le processus de contextualisation est un mécanisme cognitif automatique, autant les informations contextuelles qui modèrent la perception de la situation sont évaluées au moment de celle-ci. [Roussarie 06] corrige légèrement ce constat en établissant qu'une partie des stimuli extérieurs sont pertinents *a priori* et ne sont pas remis en cause au moment de la contextualisation de la situation. Bien que ces travaux concernent les mécanismes cognitifs chez l'être humain, le constat semble être le même dans le cas d'un système interactif sensible au contexte.

1.3. Interdépendance de l'activité humaine avec le contexte

Pourquoi définir et modéliser le contexte d'une situation est-il si important ? Parce que cette notion est grandement liée à celle d'activité humaine, et donc de l'**activité de l'utilisateur**.

Il est établi que l'utilisateur agit en fonction de la situation dans laquelle il se trouve [Suchman 87]. Plusieurs théories s'affrontent pour déterminer si le contexte est la source de l'activité au sein duquel elle prend place (**le contexte génère l'activité**), si l'activité définit le contexte [Dourish 04] (**le contexte évolue en même temps que l'activité**), ou si l'activité est partie intégrante du contexte (**les informations contextuelles décrivent l'activité**).

Toujours est-il que **l'activité**, **l'action**, est une notion fortement interdépendante avec celle du contexte. L'être humain va agir en fonction de la **situation courante** et des **informations contextuelles disponibles**. La situation et le contexte sont donc responsables de la prise de décisions et d'actions de l'être humain. Au cours de l'activité, la situation et les informations contextuelles évoluent, engendrant alors un nouvel état perceptible de l'univers. Une fois l'activité terminée, l'être humain considère alors une nouvelle situation et de nouvelles informations contextuelles.
Pour [Hommel & al. 00], il est clair que la représentation cognitive de l'activité est fortement liée aux informations du contexte dans lequel elle a lieu. Autrement dit, l'activité dépend du contexte. Le contexte apporte du sens et une cohérence à l'activité. En fait, plus exactement, la situation telle qu'elle est perçue par l'humain a déjà un sens, immédiat, mécanique. **Le contexte colore la situation**, lui donne un nouveau sens, enrichit certains de ses aspects ou n'engendre que la considération de certains autres. Sans contexte, l'humain réagit

machinalement et parfois **involontairement** (c'est le cas des réflexes). Avec, il réagit de manière **adaptée**, sa réaction reflétant sa compréhension du contexte dans lequel il se trouve. Face à une situation contextualisée, un être humain adopte des comportements davantage pertinents, dans la mesure où ils s'inscrivent en cohérence avec le contexte. Autrement dit, **il exécute mieux les bonnes actions**.
En revanche, [Van Oers 98] stipule que le processus de contextualisation, chez un être humain, est paramétré par :

(1) un ensemble de **connaissances déjà existantes**, relatives à son profil, son état interne, son passé, etc. ;
(2) la **situation** d'interaction en elle-même ;
(3) l'**activité**, l'interaction, des différents protagonistes de la situation.

Il est donc clair que le processus de contextualisation, et donc la modélisation du contexte d'interaction, dépend de l'activité en cours.

Si nous voulons concevoir des systèmes capables de s'adapter aux gestuelles de l'utilisateur, il nous faut comprendre ces gestuelles, les interpréter. Il faut comprendre le sens de l'activité, de manière plus générale. Or, ce sens, c'est le contexte qui l'apporte. L'activité, au sein de la scène observée, peut donc être mieux interprétée **en estimant le contexte au sein duquel elle a lieu**. D'un autre côté, pour mieux estimer le contexte d'interaction courant, il nous faut comprendre l'activité. Un **cercle vertueux** apparaît alors : une activité comprise permettra une meilleure modélisation du contexte (et donc une meilleure adaptation), et un bon modèle de contexte permettra de mieux comprendre l'activité (et donc de mieux s'y adapter). La notion de contexte est donc centrale lors de la conception d'un système sensible au contexte et c'est pourquoi, au fil des années, elle fut étudiée, dans le but de mieux répondre à l'activité, notamment celle de l'utilisateur.

1.4. Etude du contexte dans le domaine de l'Interaction Homme-Machine

La notion de contexte a depuis longtemps été étudiée dans le domaine de l'**Interaction Homme-Machine**. L'historique suivant est partiellement inspiré de [Crowley & al. 06].

Tout d'abord, le contexte a été abordé du point de vue de l'**Intelligence Artificielle**, l'objectif étant d'élaborer des techniques de raisonnement contextuel. L'approche typique fut d'essayer d'enrichir les techniques de raisonnement déjà existantes en y introduisant la notion de contexte. Les travaux du domaine se sont alors largement inspirés de l'étude des **communications entre deux personnes** : partage des connaissances, quiproquos, ambigüités, compréhension d'une erreur d'interprétation, situation et contexte, etc.
[Quillian 68] introduit la notion de **réseau sémantique**, permettant la compréhension du langage naturel. Dans [Minsky 75] sont définies les **'frames'**, permettant de passer d'une image visuelle à une description de cette dernière en langage naturel. [Schank & Abelson 77, Schank 99] introduit les **'scripts'**, ensemble de connaissances pour comprendre un récit textuel. Le contexte est alors plus modélisé sous la forme d'un scénario, un 'script' ou une 'frame'. Un script est activé si un certain nombre de conditions d'entrée sont vérifiées et reconnues. Ces conditions d'entrée définissent le ou les contextes, dans lesquels le script doit se dérouler. Mettant en jeu plusieurs **acteurs**, le script se déroule alors, enchainement ordonné

d'événements et de scènes. Ce travail constitue une référence majeure mais bien que très complète, cette approche n'est pas facile à implémenter au sein d'un système. Dans le cas de scénarios plus complexes, elle génère facilement des coûts et des temps de calcul prohibitifs, éloignant largement les traitements du temps réel. Schank basera ensuite ses théories sur la notion de **'plan'**. Un plan, décrivant de manière plus générique comment un acteur peut atteindre son objectif, sera utilisé lorsque les scripts atteindront leurs limites. Dans [Bobrow 77], la notion de **'schéma'** modélise les informations contextuelles, permettant la compréhension d'une image, d'un son, d'une parole ou d'un texte. La modélisation du contexte permet donc de représenter et de structurer un ensemble de connaissances d'un individu, sur l'univers. [Demko 92] utilise la **théorie des graphes conceptuels**, pour représenter ces connaissances, et rend opérationnel cette théorie pour résoudre une ambigüité de sens dans une expression textuelle. Cette approche permet également l'apprentissage de nouvelles informations contextuelles.

Dans le domaine de l'**Intelligence Artificielle**, la problématique concernant la reconnaissance du contexte a été définie comme le **« *Frame Problem* »**, qui n'a pas pu encore être résolu. En effet, toutes les approches présentées précédemment sont partis d'études linguistiques, alors qu'il aurait été préférable de partir d'études concernant **la perception et l'action au niveau de l'être humain**.
Dans le domaine de l'**Interaction Homme-Machine**, il est nécessaire d'avoir « un modèle simple, explicite et unifié, pour rassembler en une seule représentation plusieurs contextes individuels » [Bolchini & al. 07]. Ce modèle est alors construit à partir de la scène, observée par le biais de différents canaux. Un comportement, représentation haut niveau de l'activité, est associé à un ensemble de descripteurs contextuels bas niveau. Ces descripteurs, une fois observés, nous permettent de caractériser le contexte, au sein duquel le comportement de l'utilisateur prend place.
Le système, tout d'abord interactif, devient dorénavant **sensible au contexte**. Cette notion de **sensibilité au contexte**, *context awareness*, a été introduite pour la première fois par [Schilit & al. 94]. Il sera alors question de *context-aware system* ou *application* (concepts corrélés : *adaptative system*, *reactive system*, *situated system*, etc.). Un nouveau paradigme de programmation apparaît, *context-aware computing*, permettant le développement de nouveaux outils logiciels, mettant en œuvre différents mécanismes automatiques d'extraction d'informations contextuelles, au cours de l'exécution et de l'interaction, en temps réel. Historiquement, la sensibilité au contexte a tout d'abord été étudiée dans le cadre de systèmes majoritairement mobiles, puis l'étude s'est élargie à tous systèmes interactifs.

Tout d'abord ne regroupant seulement que des informations de localisation spatiale, le contexte est par la suite divisé en **trois catégories** [Schilit & al. 94] :

- **le contexte physique**, *where you are* (éclairement, bruit, conditions de trafic, température, etc.) ;
- **le contexte utilisateur**, *who you are* (profil, localisation, entités proches de lui, situation sociale actuelle) ;
- **le contexte informatique**, *what resources are nearby* (connectivité au réseau, coût de communication, bande passante, ressources informatiques disponibles, proches de l'utilisateur, telles qu'une imprimante, un écran ou une station de travail).

[Chen & Kotz 00] complétera cette division par une dernière catégorie, **le contexte temporel** (moment de la journée, mois, année, saison, interactions et contextes d'interaction passés,

etc.). Dans [Schilit & al. 94], il est stipulé qu'il est alors possible de développer 4 catégories d'applications sensibles au contexte, dans le domaine de l'informatique mobile :

- la catégorie dite de *Proximate Selection*, technique consistant à mettre en valeur et à rendre évidente à choisir dans le cadre d'une tâche donnée, les ressources proches de l'utilisateur ;
- la deuxième catégorie regroupe les applications impliquant la reconfiguration automatique d'architectures de serveurs mobiles et des canaux de communication avec les clients (*distributed computing system*) ;
- la troisième catégorie concerne les applications transmettant des informations à l'utilisateur ou des commandes aux clients, en accord avec le contexte d'interaction courant ;
- la dernière catégorie définit l'ensemble des applications mettant en œuvre des mécanismes d'adaptation au sein du système, en accord avec les événements observés.

Dans sa thèse, en 1995, Schilit se focalisera massivement sur l'adaptation des ressources du système, en accord avec le contexte d'interaction observée. Les systèmes développés doivent :

(1) détecter les différentes entités participant à la situation et extraire leurs différentes caractéristiques (**phase de découverte**) ;
(2) décider quelles ressources vont être utilisées, en accord avec le contexte d'interaction (**phase de sélection**) ;
(3) utiliser les ressources (**phase d'utilisation**).

[Bremond & Thonnat 96] représentent globalement le contexte, par le biais d'un formalisme défini, **pour un processus d'interprétation de situation observée**, dans le but d'améliorer ses performances par le biais d'un ensemble d'informations contextuelles. Les auteurs cherchent à interpréter les comportements de personnes et de véhicules dans des scènes extérieures, telles qu'un quai de métro ou une autoroute, observées à partir d'une simple caméra. Ces travaux sont intéressants car ils définissent un nouveau type de contexte, **le contexte calculatoire d'un processus donné**. Un processus particulier utilise, pour effectuer ses traitements, un ensemble de **connaissances principales**, couplées avec un ensemble d'**informations contextuelles**. Les connaissances principales sont globales, générales, vérifiables au cours d'un intervalle temporel important et obligatoirement nécessaires au bon fonctionnement du processus. Les informations contextuelles, extraites à partir des connaissances globales, sont des connaissances non obligatoires, pas forcément complètes et bien définies, et fortement dépendantes de l'application. Ces informations permettent d'améliorer l'efficacité et les performances du processus. Dans le cadre de cette étude, les comportements d'une entité ne sont pas analysés en continu, seuls certains événements particuliers, bas niveau, permettent d'arriver au résultat final, i.e. la caractérisation sémantique d'un comportement observé. Dans ces travaux, le processus est clairement divisé en plusieurs étapes, chaque étape étant associée à une liste exhaustive d'informations contextuelles. Ces travaux présentent **deux contraintes**. Tout d'abord, la liste très importante d'informations contextuelles à gérer suppose une intervention, parfois longue et fastidieuse, d'un superviseur. Ce dernier doit en effet inclure manuellement les différentes informations au sein du processus d'interprétation. [Brémond 97] propose un logiciel d'acquisition du contexte, ***MARES*** (Module d'Acquisition et de Représentation d'Environnements Statiques) dans le but d'alléger le travail de cet opérateur. Deuxièmement, l'interprétation d'une nouvelle situation nécessite l'analyse de cette dernière, manuellement et en amont de son

interprétation, dans le but d'identifier les différentes informations contextuelles nécessaires au processus.
[Schmidt & al. 99a, Schmidt & al. 99b, Schmidt 02] sont également des travaux de référence dans la mesure où, contrairement à une large partie de leurs prédécesseurs, les auteurs affirment que **le contexte d'interaction va bien au delà de la localisation spatiale ou géographique**. Ils identifient alors les différentes caractéristiques contextuelles relatives à l'utilisateur et à son environnement (voir Fig. 2.3.). Le contexte est divisé en **2 catégories** : les facteurs humains, ou **le contexte utilisateur**, et l'environnement physique, ou **le contexte environnement**. Le contexte utilisateur regroupe les informations sur l'**utilisateur** (habitudes, états émotionnels, conditions biophysiologiques, etc.), sur l'**environnement social** de celui-ci (dynamique de groupe, interactions sociales, localisation spatiale des autres personnes, etc.), et sur les **tâches de l'utilisateur** (activités effectuées spontanément, tâches commencées, objectifs généraux, etc.). Le contexte environnement, quant à lui, décrit les caractéristiques **physiques** de celui-ci (éclairement, pression, bruit, température, etc.), les **caractéristiques du système** en lui-même (ressources système environnantes, communication, performances, etc.), les **informations de localisation** de celui-ci (position absolue, position relative, co-localisation, etc.). Dans ces travaux, le contexte relatif au système informatique et interactif est compris dans le contexte environnement. Ces travaux présentent plusieurs systèmes sensibles au contexte.

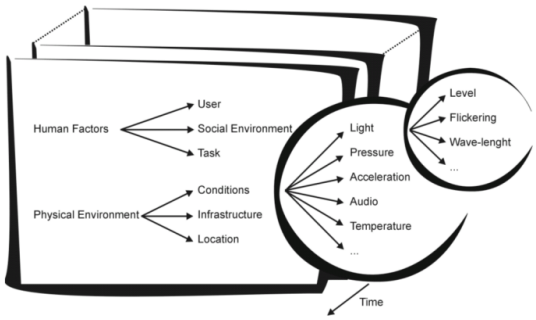


Figure 2.3. : Modélisation du contexte, tiré de [Schmidt 02]

Selon [Thevenin & Coutaz 99, Thevenin 01], le contexte d'interaction est défini par le triplet < **Plateforme, Utilisateur, Environnement** >. **Le contexte 'Plateforme'** regroupe l'ensemble des informations caractérisant les supports matériels et logiciels utilisés, les ressources de calcul et de communication, mis en jeu au cours de l'interaction avec l'utilisateur. **Le contexte 'Utilisateur'** définit ses capacités sensori-motrices et cognitives, ses caractéristiques socioculturelles, sa localisation ou celle de certaines parties de son corps, ses objectifs, etc. Enfin, le contexte **'Environnement'** décrit toutes les dimensions physiques et sociales de l'espace réel où se déroule l'interaction avec l'utilisateur.
[Chen & Kotz 00] définit le contexte en informatique mobile. Pour les auteurs, le contexte présente **deux aspects**. Le premier aspect, actif, inclut les **caractéristiques de l'environnement** (états et configurations) qui déterminent le comportement du système mobile. Le deuxième aspect, passif et non critique, regroupe un **ensemble d'informations pertinentes pour le système**. Il n'est pas nécessaire au système de s'adapter relativement aux informations contextuelles non critiques pour le système, si ce n'est pour les transmettre aux utilisateurs intéressés. Le fait qu'une information contextuelle soit active ou passive dépend

de la façon dont elle est utilisée dans l'application. Ces travaux répertorient plusieurs applications, en précisant et caractérisant chaque information contextuelle, en accord avec leur définition.
Enfin, pour terminer, entre les années 1999 et 2000, le projet ***GUIDE***[2] a été mené par Davies, Cheverst et Michell. Dans le cadre de ce projet, un système mobile sensible au contexte a été développé, dans le but **de transmettre à l'utilisateur des informations contextuelles pertinentes**, au cours d'une visite historique de la ville de Lancaster. Le contexte comprend **deux facettes** : **une facette utilisateur** (profil de l'utilisateur, budget de l'utilisateur, etc.) et **une facette environnement** (moment de la journée, localisation géographique, etc.). En fonction de l'évolution de la visite, le système transmet à l'utilisateur les horaires d'ouverture d'un lieu touristique proche, les prédictions météorologiques, etc. Ce système est souvent cité comme le premier système sensible au contexte véritablement opérationnel.

2. Modélisation du contexte

Nous nous proposons dans cette section de présenter un **état de l'art** relatif à la **modélisation du contexte d'interaction au sein d'un système interactif**. Comme nous l'avons vu dans la section précédente, le contexte est une notion difficile à définir, qui dépend de nombreux éléments, comme une tâche particulière, un environnement d'interaction donné, l'application développée, etc. Il doit englober de nombreux aspects s'articulant autour de l'interaction, comme, par exemple, ses caractéristiques spatiales et temporelles. Par conséquent, cette représentation doit pouvoir **évoluer dynamiquement au cours de l'interaction** et se référer soit à **l'utilisateur**, soit au **système interactif**.
Un modèle de contexte est nécessaire au **système adaptatif**, à tout instant et pour tous les processus qui le composent. C'est pourquoi, la représentation d'un modèle de contexte est un problème difficile et il n'existe pas, à ce jour, une modélisation standard et formalisée. Plusieurs modèles, ainsi que plusieurs de leurs aspects, ont été étudiés dans le cadre de nombreux travaux.

Un modèle de contexte est élaboré, dans le but d'atteindre certains objectifs spécifiques. La Section 2.1. commence cette section en énumérant **les différentes raisons pour lesquelles un modèle de contexte est utilisé** au sein d'un système interactif.
Plusieurs travaux cherchent à modéliser le contexte **au sein d'un scénario**, mettant en jeu différentes situations d'interaction. Il est à noter que ces scénarios ne concernent pas forcément que l'utilisateur, mais peuvent s'appliquer à un processus système, à un périphérique du système… La Section 2.2. présente plusieurs travaux traitant de ce sujet.
Les contextes sont modélisés à partir de **données bas niveau**, collectés à partir de l'observation de la scène, par le biais de divers capteurs. La Section 2.3. expose quelques travaux définissant le contexte à partir de données bien identifiées. Ces travaux mettent en évidence la **dépendance du modèle de contexte et des données permettant sa construction**, vis-à-vis de l'application et des tâches à accomplir par les entités participant à la situation d'interaction courante.
Enfin, la Section 2.4. identifie les différentes **caractéristiques** que doit présenter un modèle de contexte, pour être efficace au sein d'une architecture logicielle de système interactif.

[2] http://www.guide.lancs.ac.uk/overview.html

2.1. Pourquoi modéliser le contexte ?

Nous allons énumérer dans cette section **les différentes raisons pour lesquelles un modèle de contexte est développé et introduit au sein d'un système interactif**. Ces raisons dépendent fortement de l'application, des tâches à accomplir par l'utilisateur, des objectifs de ce dernier, etc.

1. *Le contexte est modélisé dans le but de mieux gérer et de mieux exploiter les connaissances au sein du système.*
   Tout d'abord, les données contextuelles peuvent être nombreuses et hétérogènes, c'est pourquoi le contexte peut aider à les rendre plus facilement exploitables, en les formatant, en les filtrant, etc. Ainsi, par le biais du contexte, il est possible d'estimer les informations qui seront intéressantes au cours de l'interaction, aussi bien du point de vue de l'utilisateur, que du système. A partir de ces données, il alors possible d'extraire de nouvelles connaissances (informations contextuelles, services). L'ensemble des données géré par le biais du contexte pourra être transmis à l'utilisateur, généralement à sa demande, dans un but purement informationnel (par exemple un historique des actions passées).

2. *Le contexte est modélisé pour mieux interpréter le comportement de l'utilisateur, dans le but d'y répondre de manière adaptée.*
   Cette réponse peut être une mise en alerte du système et de l'utilisateur, dans le cas d'un système de surveillance vidéo par exemple. Le système pourra également agir sur la scène virtuelle au sein de laquelle l'utilisateur évolue, que cela soit au niveau de la représentation virtuelle de ce dernier (représentation intelligente de l'utilisateur, censure visuelle du comportement de l'utilisateur, adaptation du comportement visuel de l'utilisateur en fonction d'observateurs, etc.) ou au niveau de l'environnement interactif (activation de ressources interactives, apport d'une aide visuelle, etc.). La plupart du temps, le but est d'enrichir l'interaction avec l'utilisateur et souvent de la rendre implicite. Le système pourra alors fournir les services dont l'utilisateur a besoin ou qui sont les plus appropriés à son comportement, lui transmettre intelligemment des informations données (affichage en fonction de ses préférences, de son profil, par exemple).

3. *Un modèle de contexte permettra de mettre en place un ensemble de mécanismes d'adaptation au niveau du système, de manière à fiabiliser et optimiser son fonctionnement global.*
   Tous les aspects d'un système adaptatif peuvent être adaptés en fonction du contexte modélisé : adaptation de l'observation de la scène (sélection pertinentes des données à observer, etc.) ; adaptation d'un processus donné ; adaptation de la gestion des connaissances ; adaptation des interfaces (affichage en fonction de la plateforme matérielle utilisée, etc.) ; adaptation de l'architecture logicielle et matérielle (reconfiguration, démarrage ou stop, ajout ou suppression, changements des interconnections, etc.). Ces mécanismes d'adaptation auront de nombreux avantages au niveau du fonctionnement du système : diminution du nombre d'erreurs d'interprétation d'informations, augmentation de la précision des recherches, diminution des temps de traitement, sélection utile des ressources matérielles et

logicielles, anticipation à court terme des réactions (processus et ressources) du système vis-à-vis des comportements utilisateur à venir…

4. *le modèle de contexte sera utilisé pour façonner un environnement, au sein duquel l'interaction a lieu.*
   C'est notamment le cas dans le domaine de l'informatique ubiquitaire. Le modèle de contexte servira à construire des environnements intelligents, des *smart environments*, c'est-à-dire des environnements physiques importants, qui interagissent activement et de manière invisible avec l'utilisateur. Ces *smart environments* permettront de mettre en évidence, par exemple, les ressources matérielles, telles qu'un périphérique, qui amélioreront l'interaction de l'utilisateur avec le système.

2.2. Le contexte au sein d'un scénario

Les différents travaux exposés au cours de cette section étudient la **modélisation du contexte au sein d'un scénario**.
Ce scénario peut concerner évidemment l'utilisateur, mais également d'autres entités participant à l'interaction, telles qu'un processus informatique ou un périphérique du système. Cette remarque va de paire avec le préambule du premier chapitre. L'évolution du scénario de l'application influe sur les scénarios définis par la logique système. Modéliser le contexte dans le scénario de l'application permettra de décliner ce modèle au sein des différents scénarios suivis par les modules composant le système.

Dans ses travaux, Schmidt représente tout d'abord **l'espace problème autour de la notion de contexte** [Schmidt & al. 99a], puis élabore un modèle implémentable articulé autour d'**artefacts sensibles au contexte** (*context aware artefacts*) [Schmidt 02]. Son espace problème est un espace en 3 dimensions (voir Fig. 2.4.), au sein duquel le contexte peut être positionné vis-à-vis de **l'environnement** (physique et social), de **l'utilisateur** et du **système** (états des périphériques, modèle physiologique de l'utilisateur, modèle cognitif rattaché à l'utilisateur) et de **l'activité** (comportements, tâches). La notion d'artefacts sensibles au contexte est centrée sur celle d'**entité** et permet le développement de systèmes sensibles au contexte. Pour l'auteur, une entité peut être un environnement physique, un objet, un utilisateur, un périphérique, une application, un autre contexte, ou une combinaison de plusieurs de ces éléments. Chaque entité est sensible au contexte, c'est-à-dire qu'elle est rattachée à un ou plusieurs contextes. De plus, des liaisons particulières peuvent s'établir entre entités. Par exemple, une entité peut être composée de plusieurs autres entités. En ce cas, le contexte de l'entité résultante est défini par la fusion de ceux des entités la composant. L'écriture d'un scénario se fera donc en mettant en place différents contextes, au sein desquels évoluent diverses entités bien identifiées.

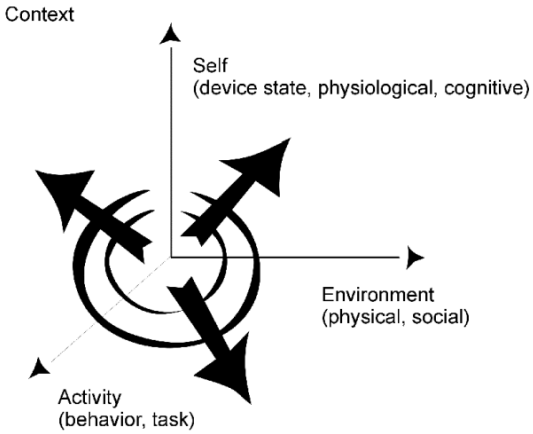


Figure 2.4. : Représentation du contexte au sein d'un espace en 3D, tiré de [Schmidt 02]

Les travaux de Crowley [Crowley & al. 02, Crowley & al. 06] représentent une contribution majeure dans le cadre de la modélisation du contexte d'interaction au sein d'un scénario, mettant en jeu le système et un ou plusieurs utilisateurs. Cette modélisation est issue d'une **ontologie**, développée par l'auteur, définissant et structurant un ensemble de concepts, tels qu'une **entité**, une **situation** et un **contexte**, entre autres. Cette ontologie est développée suivant 2 approches : **une approche descendante** qui définit la notion de contexte du point de vue de l'utilisateur, autour des notions d'humain, d'utilisateur, d'activité, d'objectifs, d'actions, de tâches, etc. ; et **une approche ascendante** qui la définit du point de vue du système, autour des notions de variables observables, d'entités, de relations, etc. De plus, elle décrit ce que suppose l'introduction d'un modèle de contexte au sein d'un système, du point de vue de son architecture. Cette ontologie peut servir de cahier des charges pour élaborer un système sensible au contexte.

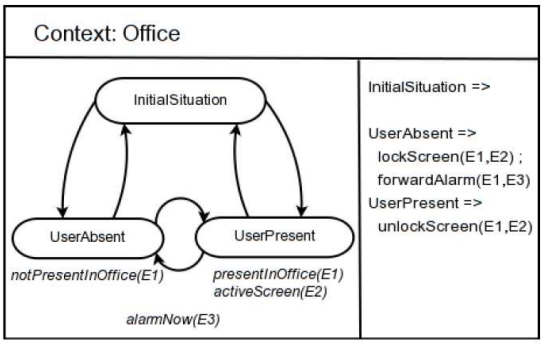


Figure 2.5. : Exemple de modèle de contexte, tiré de [Crowley & al. 02]

La Figure 2.5. donne un exemple de modélisation du contexte, mettant en œuvre un scénario bien défini. Un modèle de contexte est un **réseau structuré de situations**, chacune caractérisant un état particulier du monde. Ce réseau est non linéaire et permet la mise en place d'actions spécifiques, de la part du système, relativement aux comportements interprétés de l'utilisateur. Le système fournit un certain nombre de **services**, adaptés aux comportements de l'utilisateur, lorsqu'une situation est activée.

La définition d'une **situation** s'accompagne de celle des **entités** la composant. Une entité peut être une personne, un objet, un processus compris dans le système, etc. et est caractérisée par un ensemble de propriétés.
Un ou plusieurs **rôles** sont attribués aux entités. Un rôle constitue un ensemble d'actions pour accomplir une tâche donnée. Au sein d'une situation particulière, seules les entités capables de remplir cette tâche sont considérées. Ces rôles sont conditionnés par un ensemble de **tests d'acceptation**, effectués sur les propriétés des entités concernés. Ainsi, l'auteur donne l'exemple d'une entité 'siège', qui jouera donc son rôle si ses propriétés 'dimensions' le permettent. Ces tests sont divisés en deux parties : tout d'abord est effectuée une opération de filtrage (réduction du nombre d'entités candidates, tri des entités, contraintes sur leurs propriétés, etc.), puis d'attribution d'un rôle pour les entités pertinentes. Empruntée au théâtre, le rôle est complété par l'**acte** : un acte est un rôle attribué à une entité qui peut spontanément agir pour changer de situation.
La définition d'une situation est complétée lorsque les **relations entre les entités** la composant sont définies. Celles-ci sont également fonction de la tâche à effectuer par les entités et sont décrites sous la forme de prédicats testant les propriétés d'une ou plusieurs entités. Par le biais de ces relations, il sera par exemple possible de savoir si une entité est devant une autre, si elle possède les mêmes propriétés qu'une autre, etc. Les tests d'acceptation, décrits précédemment, peuvent permettre l'attribution ou la vérification de relations entre plusieurs entités.
Toutes les situations d'un contexte partagent les mêmes entités, rôles et relations. Cette modélisation évolue par le biais de trois types d'**événements**. Tout d'abord, lorsqu'une entité change de rôle (*role events*), le système agit alors pour redonner ce rôle à cette entité. Un changement de relations entre plusieurs entités implique un changement de situation (*relation events*). Enfin, un changement de contexte (*context events*) implique une reconfiguration des processus, particulièrement liés à l'observation de la scène (appelés processus perceptuels), au sein du système.
Ce modèle de contexte a été utilisé dans le cadre du développement de plusieurs systèmes [Brdiczka & al. 06ab, Brdiczka & al. 07, Zaidenberg & al. 09], ainsi que dans le cadre du développement d'interfaces plastiques (voir dans le Chapitre 1., la Section 2.2.3.).

Le lecteur intéressé pourra se référer à d'autres états de l'art. [Strang & Linnhoff-Popien 04] propose une classification des différentes représentations du contexte existantes. Les modèles de contextes peuvent être représentés sous la forme de paires type (clé, valeur), de langages à balises type HTML, de modèles graphiques type UML ou réseau de Pétri, de modèles orientés objets, de modèles basés sur des descriptions logiques, d'ontologies, etc. Chaque classe est analysée, les travaux utilisant une représentation donnée étant exposés et des outils pour les implémenter étant présentés. [Bolchini & al. 07] propose une comparaison de plusieurs modèles de contexte existant, suivant plusieurs critères. Il fournit un *framework* complet pour analyser exhaustivement un modèle suivant ces critères. Le concepteur en choisit un, en fonction de ces critères et de ses besoins.

2.3. Informations contextuelles bas niveau

Dans le cadre de nos travaux, les contextes sont modélisés à partir de **données bas niveau**, qu'il est nécessaire d'identifier précisément. Ces données sont mesurées par le biais de différents **capteurs** observant la scène. Les travaux suivant nous montrent à quel point un

modèle de contexte, et son degré d'abstraction, est dépendant du type de systèmes et d'applications développés.

Dans [Bremond & Thonnat 96], dans le cadre d'un processus d'interprétation d'activités au sein d'une scène observée, **plusieurs représentations sont utilisées pour structurer certains types d'informations contextuelles bien distinctes en un modèle de contexte**. Tout d'abord, le raisonnement sur l'espace délimité par la scène observée s'effectue par le biais d'une décomposition des images 2D en un arbre hiérarchisé de partitions polygonales. Ces zones polygonales peuvent rattachées à un ensemble d'informations contextuelles, telles que la matrice de calibrage associée à une caméra, pour raisonner à partir d'un espace en 3D. Pour la reconnaissance d'objets, un élément du contexte est défini par l'ensemble des informations nécessaires pour décrire les irrégularités optiques. Ainsi, le processus peut évaluer un degré de confiance relativement à la détection d'un objet particulier. Pour l'analyse et la reconnaissance de comportements d'objets particuliers, un élément du contexte met en relation les informations comportementales liées aux objets (contraintes liées aux comportements, comportements pouvant être adoptés par l'objet, etc.) avec celles liés à la zone spatiale que ces objets occupent (comportements pouvant être observés dans telle ou telle zone, etc.). Ces travaux nous montrent à quel point il est nécessaire, en fonction du type d'applications développées, de bien **identifier les données contextuelles** et de bien **structurer** ces dernières pour obtenir un **modèle de contexte pertinent**.

[Chen & Kotz 00] s'intéressent aux systèmes mobiles et proposent un état de l'art concernant **le type de données à collecter**, dans le cadre de ce type de systèmes. Ayant fait le constat que la plupart des systèmes n'utilisent que des informations contextuelles liées à la localisation de l'utilisateur, ce travail discute les différents modèles de contextes, établis à partir de ces informations, et les différentes façons de les représenter. Les informations utilisées sont à la fois sémantiques (id, nom) et géographiques (coordonnées GPS). De plus, ce type de représentations entraîne la définition au sein du système de structures de données particulières pour exprimer et échanger des informations : couple id sémantique et valeur associée, tags, modèles basés objets, modèles logiques, structures prenant en compte le temps, etc.

2.4. Propriétés d'un modèle de contexte

Un modèle de contexte doit présenter plusieurs **propriétés** importantes, pour être efficace et pertinent au sein de l'architecture logicielle d'un système interactif [Strang & Linnhoff-Popien 04, Bolchini & al. 07] :

1. *le modèle de contexte doit pouvoir être facile et rapide à manipuler et à gérer par le système.*
   Il doit pouvoir être modifié facilement et rapidement, dans le but par exemple d'ajouter une contrainte qui jouera sur sa sélection lorsque le système devra choisir parmi plusieurs contextes concurrents. De plus, les modèles de contextes sont largement utilisés dans le cadre de systèmes distribués, tels que les systèmes ubiquitaires. Un modèle doit donc permettre sa gestion de manière distribuée (création, suppression, modification, déploiement, etc.). Enfin, un modèle doit pouvoir être mis à jour dynamiquement au cours du temps, sans avoir à reconstruire celui-ci dans sa globalité.

2. *Un modèle est caractérisé par différents degrés d'abstraction, de précision et de granularité.*
   Ces degrés doivent pouvoir évoluer dynamiquement, selon les informations qui sont et seront nécessaires au cours de l'interaction. De plus, selon son degré de généricité, ce modèle doit être compréhensible par des processus aux traitements différents. Cela suppose que le modèle doit être spécialisable pour un processus donné, héritant alors d'un modèle de contexte plus haut niveau.

3. *Un modèle de contexte efficace doit permette la résolution d'ambigüité et d'incertitudes.*
   Il doit donc proposer des solutions en cas d'observations non complètes ou aux multiples significations potentielles.

4. *Un modèle peut être incomplet.*
   A contrario, seule une partie de ce modèle pourra être intéressante pour une situation d'interaction donnée. Un bon modèle de contexte doit donc permettre sa gestion et utilisation de manière partielle et incomplète.

5. *Un modèle de contexte comporte des données contextuelles hétérogènes.*
   Selon les systèmes développés, ces informations peuvent être importantes et présenter une grande diversité. Le système, ou tout du moins le module chargé de gérer ces connaissances contextuelles, doit pouvoir faire face à cette hétérogénéité de données. Cette problématique est à mettre en relation avec toutes les précédentes. Un modèle de données hétérogènes peut s'avérer difficile à gérer et exploiter facilement et rapidement ; les différents degrés d'abstraction, de précision et de granularité doivent permettre l'intégration de données hétérogènes ; et l'hétérogénéité est une plus-value nécessaire pour résoudre les ambigüités de sens, les incertitudes et pour faciliter le raisonnement à partir d'une modèle de contexte incomplet.

6. *Un modèle de contexte existant doit pouvoir être validé en l'implémentant dans plusieurs systèmes existants.*
   L'implémentation d'un modèle de contexte sur plusieurs systèmes existants démontre son efficacité et sa complétude.

7. *Un modèle peut être représenté par le biais de plusieurs types de formalismes.*
   Chaque formalisme a ses avantages et inconvénients. Les caractéristiques d'un modèle dépendent donc de celles du formalisme choisi : caractère intuitif de la représentation, possibilité de traitement automatique, degré de sémantique, flexibilité de la représentation qui doit pouvoir permettre la description de plusieurs types de contextes (contexte général ou dédié à une à une tâche donnée), capacité à décrire les informations contextuelles à différents degrés de détails et de qualité, capacité à représenter un grand nombre d'informations, etc. Le concepteur devra donc choisir un formalisme en fonction de ses besoins relativement au type de systèmes développés.

3. Caractérisation des différents contextes

Comme nous avons pu le constater dans la Section 1.4., **il n'existe pas qu'un seul contexte**. Il y a de fait autant de contextes que de **participants** à une situation d'interaction. Le terme 'participant' est ici utilisé au sens large, englobant l'utilisateur, le système, les différents processus mis en jeu au cours de l'interaction, etc. La Section 3.1. donne la **classification des contextes** que nous avons établie.
Puis, nous allons énumérer les différentes **informations bas niveau**, et un peu moins bas niveau, qui pourront être utilisées pour la **caractérisation des différents contextes** énumérés précédemment (Section 3.2.). Ces informations pourront être utilisées pour la construction des différents contextes et/ou être utilisées directement comme informations contextuelles par le reste du système. Nous ne prétendons pas énumérer exhaustivement toutes les informations qui peuvent permettre la caractérisation d'un contexte. Cependant, nous essayerons de montrer l'importance et la diversité d'informations qui peuvent être utilisées.
Par la suite, nous discuterons, par le biais d'exemples, comment le concepteur peut savoir s'il a besoin de telles ou telles informations, en fonction de ses besoins (Section 3.3.).

3.1. Les différents contextes

Nous distinguons dans ces travaux **6 contextes** différents, compris dans le modèle du contexte d'interaction au sein duquel l'utilisateur évolue. Ces contextes nous permettent de caractériser chaque élément impliqué dans le processus d'interaction scénarisé, de manière à exécuter les mécanismes adaptatifs adéquats pour améliorer ce processus, tant du point de vue de l'utilisateur que du système. Nous rappelons ici que nous ne considérons qu'un **utilisateur unique**.
Ces contextes peuvent bien sûr se combiner les uns avec les autres, pouvant même dépendre les uns des autres. Il est bien évident que tous ces contextes ne servent pas directement à caractériser la gestuelle de l'utilisateur, et l'activité de manière plus générale. Cependant, certaines informations contextuelles peuvent s'avérer nécessaires au bon fonctionnement du système, d'un processus, d'un algorithme… et sont donc indirectement indispensables au cours de l'interactivité.

Les six contextes (voir Fig. 2.6.) que nous distinguons sont :

- le contexte **« Utilisateur »** (Section 3.1.1.)
- le contexte **« Système/Application »** (Section 3.1.2.)
- le contexte **« Scène réelle »** (Section 3.1.3.)
- le contexte **« Interface »** (Section 3.1.4.)
- le contexte **« Observation »** (Section 3.1.5.)
- le contexte **« Temps »** (Section 3.1.6.)

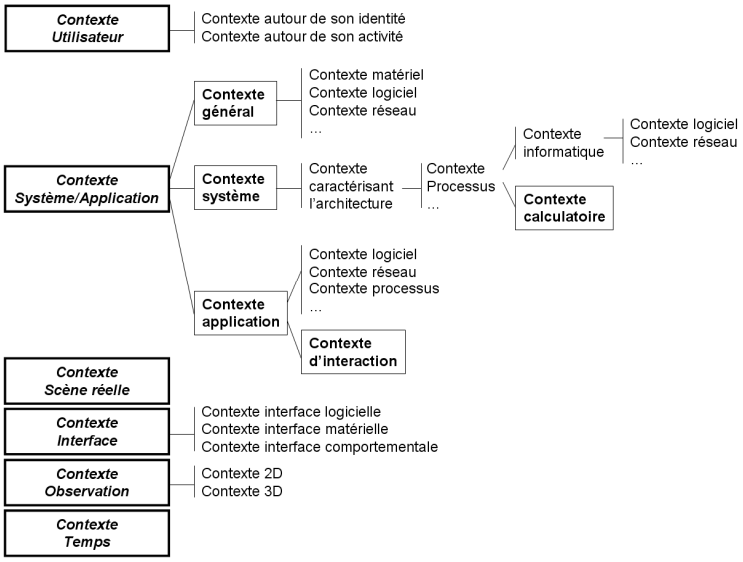


Figure 2.6. : Les 6 contextes compris dans le modèle de contexte

3.1.1. Contexte « Utilisateur »

Le **contexte « Utilisateur »**, regroupe l'ensemble des informations relatives à l'utilisateur, qui pourront contribuer à expliquer les gestuelles qu'il adoptera au cours de l'interactivité. Ce contexte regroupe un ensemble de connaissances statiques qui définissent qui est l'utilisateur, tandis que les connaissances dynamiques compris au sein de celui-ci permettent de caractériser son activité. Il est bien évident que ce contexte peut regrouper un ensemble de contextes « Utilisateur », dans le cas où le système interagirait avec plusieurs utilisateurs.

3.1.2. Contexte « Système/Application »

Le **contexte « Système/Application »**, ou contexte « Système » si nous considérons le système comme un ensemble englobant l'application, est complexe et comporte de nombreux degrés de granularité et de précision. Il est défini par la **logique concepteur** (logiques système et application) et doit **caractériser l'architecture du système** (et donc les différents aspects de l'interactivité entre l'utilisateur et le système).
Nous considérons ici que le contexte « Système/Application » ne comprend aucune information concernant l'interface avec l'utilisateur. En effet, pour des raisons que nous exposerons par la suite, nous distinguons le contexte « Interface » du contexte « Système/Application ».

Le contexte « Système/Application » regroupe ainsi plusieurs sous-contextes, qu'il n'est pas possible d'énumérer exhaustivement, ces derniers dépendant fortement du type de système développé. Nous pouvons cependant distinguer **3 classes de sous-contextes** :

(1) les sous-contextes **généraux** qui caractérisent de manière globale le système et l'application ;
(2) les sous-contextes décrits par la **logique système** ;

(3) les sous-contextes décrits par la **logique application**.

Les contextes d'ordres plus généraux décrivent les différentes caractéristiques générales du système et de l'application.
Ainsi, les contextes **matériels** et **logiciels** décriront les différents profils matériels et logiciels du système : quelle plateforme ? quel système d'exploitation ? quelles bibliothèques ? etc. Il pourra également s'agir, par exemple, d'un contexte **réseau**, fournissant ainsi un ensemble d'informations contextuelles sur le réseau auquel est rattaché le système.
Typiquement, ce sont des contextes qui ne sont pas utilisés directement au cours de l'interprétation de l'activité mais qui peuvent être utilisés par le système au cours de son fonctionnement.

Le contexte système est défini par la **logique système** et donc par l'architecte du système. C'est ce contexte qui caractérise principalement **l'architecture du système**.
Une architecture est divisée en plusieurs niveaux sémantiques, chaque niveau comprenant un ensemble de processus interconnectés. A l'image de la structure de l'architecture, il y aura autant de sous-contextes systèmes que de couches architecturelles sémantiques, chaque couche comprenant alors autant de sous-contextes qu'il y a de processus (ou de modules de traitements).
Le sous-contexte d'un processus donné, ou d'un module donné, d'un traitement donné, d'un algorithme donné, décrit les contextes **informatiques** (logiciel, réseau, etc.) et **calculatoires** dans lequel il s'exécute à un instant donné. Grâce au contexte informatique, par exemple, un processus pourra choisir tel ou tel traitement selon les bibliothèques logicielles utilisées et la disponibilité du réseau à ce moment là. Quant au contexte calculatoire, celui-ci est particulièrement intéressant car il s'agit de l'ensemble des paramètres qui sont utilisés par un processus. Ainsi, la modification de ce contexte permet d'orienter les traitements et ainsi d'atteindre un état spécifique et attendu du système, à un instant donné.

La **logique application** (le développeur), quant à elle, définit **le contexte application**. A l'instar du contexte relatif à un processus compris dans l'architecture du système, ce contexte peut regrouper un ensemble de sous-contextes caractérisant les profils logiciels et réseaux, utilisés par l'application. De plus, plusieurs processus peuvent être spécifiques à l'application et sont alors associés à un contexte informatique et calculatoire précis.
Enfin, et c'est sa contribution principale, le contexte application définit, au cours du temps, les **contextes d'interaction** au sein duquel l'utilisateur évolue, mettant ainsi en relief les **situations d'interaction** auxquelles il doit faire face, ainsi que les **objectifs** et les **tâches** à effectuer. C'est véritablement ce contexte d'interaction qui permettra l'identification de l'activité observée et la mise en place d'une réponse du système adaptée à celle-ci.

Une remarque pour conclure. Le contexte « Système/Application » représente un ensemble d'informations particulières, plus locales, au sein du système et, à l'instar des connaissances globales comprises au sein du système, elles sont gérées par un gestionnaire spécifique à chaque niveau sémantique de l'architecture.

3.1.3. Contexte « Scène réelle »

Ce que nous appelons **« Scène réelle »** constitue l'environnement physique observé, au sein duquel l'utilisateur interagit physiquement avec le système. Il est à noter que l'utilisateur, le système et tout objet servant d'interfaces avec le système au cours de l'interaction ne sont pas considérés ici comme faisant partie de la scène réelle.

Comme le contexte « Utilisateur », le **contexte « Scène réelle »** peut être caractérisé par un ensemble de connaissances statiques et dynamiques, introduites au sein du système au cours de son élaboration et/ou acquises par le biais de capteurs dédiés au cours de l'interactivité. D'une application à une autre, une connaissance peut être considérée comme étant statique ou dynamique.
Ces connaissances peuvent être utilisées pour la modélisation de la scène, la caractérisation des gestuelles utilisateur et, de manière plus générale, l'interprétation de l'activité.

3.1.4. Contexte « Interface »

Nous choisissons de distinguer le **contexte « Interface »** des autres contextes (en particulier du contexte « Système/Application »), car ce contexte englobe les moyens d'interagir offerts par le système à l'utilisateur. Il est donc très spécifique et peut se révéler essentiel au cours de l'interprétation de l'activité, particulièrement celle relative à l'utilisateur.

Comme établi dans le premier chapitre (Section 1.4.), l'interface est divisée en trois parties : l'**interface logicielle**, l'**interface matérielle** et l'**interface comportementale**. Le contexte « Interface » peut donc être logiquement divisé en trois sous-contextes distincts. Il est à noter que ces contextes ne donnent pas d'informations relativement au contenu de l'interface, mais bien au support que cette dernière offre à l'utilisateur.
Le contexte lié à l'**interface logicielle** décrit les différentes caractéristiques de la scène virtuelle, au sein de laquelle l'utilisateur évolue. Il comporte, par exemple, les informations concernant la visualisation de la scène 3D (paramètres liés à la caméra 3D) et pourra donc être utilisé lors de l'adaptation visuelle de cette scène à plusieurs périphériques de restitution.
Le contexte lié à l'**interface matérielle** sera principalement utilisé par les différents processus du système. C'est par exemple dans ce contexte que seront décrites les matrices de calibrage des caméras, matrices utilisées par de nombreux algorithmes de traitements d'images.
Le contexte lié à l'**interface comportementale** servira principalement à connaître les différentes techniques d'interaction et modalités mises à disposition à l'utilisateur au cours de l'interactivité, ou d'une partie de celle-ci.

3.1.5. Contexte « Observation »

Le **contexte « Observation »** existe de par le fait que la scène est observée, dans le but d'interpréter ce qu'il s'y passe. Cette observation est la plupart du temps visuelle mais peut se faire par d'autres canaux complètement différents, dans le cas de systèmes multimodaux. Ce contexte est dédié à la caractérisation pertinente de la scène, de son contenu, et particulièrement à celle de l'activité de l'utilisateur. Autrement dit, le contenu observable, permettant une caractérisation et une interprétation pertinentes des gestuelles de l'utilisateur et de l'activité au sein de la scène de manière plus générale, sera défini dans ce contexte.
Le contexte « Observation » s'articule autour des caractéristiques de la **scène virtuelle**, telle qu'elle est observée par le système. Comme nous le verrons dans le prochain chapitre (Section

2.), la scène virtuelle est modélisée en fonction de la **scène réelle** capturée et d'une **scène 3D**, établie à partir du scénario de l'application. Le contexte « Observation » définit véritablement ce qui est perçue par le système, au niveau de ces deux scènes.

Dans ces travaux, nous restreignons l'observation de la scène à une observation purement visuelle. C'est pourquoi notre contexte regroupe uniquement un contexte lié à une image vidéo en 2D (scène réelle, contexte 2D) et un contexte lié à un environnement en 3D (scène 3D, contexte 3D). Ces 2 contextes sont caractérisés par un ensemble de connaissances, généralement extraites par le biais d'algorithmes de traitement et de synthèse d'images.

3.1.6. Contexte « Temps »

Le **contexte « Temps »** illustre la **dimension temporelle** de toutes informations. Il ne s'emploie que très rarement tout seul et englobe la plupart du temps d'autres informations contextuelles. Ainsi, il sera possible de caractériser une information, à un instant donné, au cours d'un intervalle de temps, au sein d'un historique passé ou au sein d'une prédiction (autrement dit, d'un scénario), en l'englobant dans le contexte « Temps ». Le contexte « Temps » peut donc permettre, par exemple, d'établir un contexte temporel autour des gestuelles de l'utilisateur, des réponses du système, des traitements d'un processus, de situations et contextes d'interaction. Il pourra également permettre de savoir si une tâche a été répétée plusieurs fois, si elle a été exécutée lentement ou rapidement, quelles sont les situations précédentes, quelles sont les gestuelles antérieures de l'utilisateur, quelles sont les réactions antérieures du système, quelles ont été les événements observés antérieurement, etc.

3.2. Informations utilisées pour construire un contexte

Un modèle de contexte peut avoir différents degrés d'abstraction, de précision et de granularité, degrés pouvant évoluer au cours de l'interaction. **Il est construit à partir de données et de connaissances spécifiques.**
La construction d'un contexte est pyramidale, c'est-à-dire qu'à partir d'informations bas-niveau, de nouvelles informations sont construites, puis de nouvelles à partir de ces dernières, etc. et ce, de plus en plus haut niveau, jusqu'au modèle de contexte final. Les informations peuvent donc se combiner, et à différents niveaux, jusqu'à devenir exploitables par le système. De plus, elles doivent se combiner de manière à ce qu'elles deviennent compréhensibles au niveau sémantique auquel elles sont destinées.
Au cours de l'interaction, ces informations sont observées et vérifiées en temps réel, de manière à détecter rapidement les changements de contextes d'interaction.

Une information, utilisée par un système sensible au contexte, peut être introduite au sein de ce système de **deux manières** :

(1) Soit l'information est **introduite avant l'interactivité**, par le concepteur, l'utilisateur ou une tierce personne.
(2) Soit elle est **acquise au cours de l'interaction**, par le biais d'un capteur physique, interface matérielle avec la scène réelle, ou d'un capteur virtuel, dédié à l'observation de la scène virtuelle où évolue l'utilisateur.

Une information représente une connaissance, plus ou moins importante et dédiée pour un processus donné. Cette connaissance peut être statique, c'est-à-dire étant vraie tout au long de l'interactivité, ou dynamique, c'est-à-dire évoluant au cours de l'interactivité. Selon les applications développées, une connaissance peut être statique dans un cas et dynamique dans un autre.

Nous allons énumérer ici les différentes sources d'informations, qui pourront permettre l'extraction de connaissances, dans le but de construire un contexte particulier, suivant notre classification présentée précédemment.

3.2.1. Contexte « Utilisateur »

Le contexte « Utilisateur » est construit à partir d'un ensemble de connaissances, aussi bien statiques que dynamiques.

Les connaissances statiques nous indiquent **qui est exactement l'utilisateur**. Il s'agira d'un profil, ou d'un type de profil, de son identité, de son âge ou de sa classe d'âge, de sa culture, de ses handicaps, de ses préférences, sa situation géographique, de son niveau en informatique, de son niveau dans un type d'applications ou de jeux, de son identifiant au sein d'un réseau, de son activité prévue dans son agenda, etc.

Les connaissances dynamiques relatives à l'utilisateur, quant à elles, permettent de **caractériser son activité**, ce qu'il fait.
Il sera question de toutes les caractéristiques cinématiques (position, orientation, vitesse, vitesse angulaire, accélération, accélération angulaire) qui concerne l'utilisateur ou tout du moins une ou plusieurs parties de son corps, définies de manière absolue ou de manière relative. Une pression, une force, un moment, etc. pourront également caractériser le contexte « Utilisateur ».
Particulièrement dans le cas de systèmes mobiles, toute information concernant la localisation de l'utilisateur, comme ses coordonnées GPS, pourra être intéressante pour construire le modèle de contexte.

Plusieurs informations relatives au contexte « Utilisateur » pourront être **combinées avec d'autres informations issues d'autres contextes**, permettant ainsi de savoir quel périphérique l'utilisateur utilise (contexte « Interface »), quel objet de l'environnement il garde dans sa main (contexte « Scène réelle »), depuis combien de temps il possède cet objet (contexte « Temps »), etc.

Enfin, plusieurs informations, moins bas niveau et souvent **construites à partir d'autres informations**, pourront caractériser le contexte « Utilisateur ». Il s'agira de la gestuelle (mouvement, geste, action et comportement) qu'il adoptera, qu'elle soit implicite ou explicite ; de son état interne, sa fatigue par exemple ; de ses objectifs et de ses besoins ; de ses compétences, etc.

3.2.2. Contexte « Système/Application »

Le contexte « Système/Application » étant une structuration de différents contextes, les informations le caractérisant sont nombreuses et diversifiées.

Les différents contextes **matériels** du système pourront être caractérisés par les différents profils de ses composants, des plateformes matérielles utilisées, la disponibilité de ses ressources matérielles à un instant donné, les caractéristiques des processeurs le composant, ses capacités de mémoire et de stockage, etc.
De la même manière, les contextes **logiciels**, que ce soit d'ordre général, vis-à-vis de la logique système ou de la logique application, seront caractérisés par les différents profils des logiciels utilisés, les systèmes d'exploitation opérant sur le système, les différentes ressources logicielles disponibles à un instant donné, les différents processus disponibles à un instant donné, etc.
Quant aux contextes **réseaux**, ces derniers regrouperont des informations sur la disponibilité du réseau ou d'un de ses éléments à un instant donné, sur l'identité et la localisation d'un élément sur le réseau, sur son débit, etc. Ils seront construits à partir de l'observation des réseaux rattachés au système.

Le contexte **calculatoire** d'un processus, que ce soit au niveau du système ou de l'application, comprendra l'ensemble structuré des paramètres que le dit processus utilisera pour effectuer ses traitements. Ces paramètres seront déterminés à partir d'expérimentations, d'une vérité terrain, d'une documentation, etc.
Un processus suit, dans ses traitements, un scénario établi par la logique système. C'est pourquoi les contextes d'un processus, extraits de ce scénario, évoluent au cours du temps. Ainsi, connaissant ce scénario, il est possible d'orienter les traitements opérés par le processus par le biais de la modification de ses contextes, de manière à atteindre un état attendu par le système.

Enfin, l'interactivité entre le système et l'utilisateur est dirigée par le scénario de l'application. Les réponses du système sont donc orchestrées par ce scénario, vis-à-vis des gestuelles reconnues de l'utilisateur. C'est de ce scénario (logique application) que sont extraits les différents **contextes d'interaction** (contexte application) dans lesquels l'utilisateur va évoluer au cours du temps.

3.2.3. Contexte « Scène réelle »

Le contexte « Scène réelle » peut également être caractérisé par un ensemble de connaissances statiques et dynamiques. Ces connaissances sont acquises par le système par le biais de **capteurs**. L'ensemble des données qu'il est possible de capter de nos jours est décrit au cours de la Section 4.3.1.
Les connaissances comprises au sein du contexte « Scène réelle » peuvent donner des indices concernant la modélisation de la scène, la caractérisation des gestuelles utilisateur et l'interprétation de l'activité générale au sein de la scène.

Les **caractéristiques statiques** comprennent le type de scène réelle observée (intérieur, extérieur, autoroute, etc.), les dimensions de l'environnement physique, la liste des objets statiques, n'étant pas interfaces, compris dans la scène réelle, la description des structures spatiales caractérisant l'environnement physique et les objets statiques compris au sein de celui-ci, les occultations, les réflexions de lumière, etc.

Les connaissances statiques sont généralement introduites au sein du système au moment de sa conception, ou sont construites par l'analyse de données captées, telles que la température, la luminosité, le moment de la journée, etc.

Quant aux **connaissances dynamiques**, il sera question des caractéristiques à un instant donné des objets mobiles, n'étant pas interfaces, compris au sein de la scène réelle, de l'éclairement, du bruit, de la description des structures spatiales caractérisant l'environnement physique et les objets mobiles, n'étant pas interfaces, compris au sein de celui-ci, des occultations, des réflexions de lumière, de la température, de la pression sur le sol ou sur un mur, du champ magnétique, etc. (voir Section 4.3.1.).
Les connaissances dynamiques sont acquises au cours du temps par le biais de capteurs. Les connaissances relatives à la position d'un objet mobile nécessitent, d'un point de vue uniquement visuel, des traitements d'images spécifiques, tels que des algorithmes de suivi par exemple.

3.2.4. Contexte « Interface »

Au niveau du contexte de l'**interface logicielle**, il pourra être intéressant de savoir comment la scène 3D est simulée, à quel niveau de précision elle est simulée, par quelle technique, quelles sont les caractéristiques du cône de vision restituée par une caméra 3D, etc. Ces connaissances sont définies par le scénario de l'application (logique application).

Le contexte de l'**interface matérielle** sera, quant à lui, caractérisé par la nature et la disponibilité des périphériques, les paramètres relatifs à une caméra (matrices caméra), la disposition spatiale des capteurs, les caractéristiques des images captées (type, taille, date, etc.), etc. C'est dans ce contexte que les connaissances statiques et dynamiques, liées aux objets de la scène réelle servant d'interfaces, seront regroupées : nom, type d'objets, dimensions, poids, formes, couleurs, date de fabrication, etc.
La plupart de ces connaissances sont introduites au sein du système au moment de sa conception ou sont acquises lors de phases de calibrages, antérieures à l'interactivité. Elles sont définies par la logique concepteur.

Enfin, le contexte relatif à l'**interface comportementale** pourra être construit sachant, par exemple, quelles techniques d'interaction sont utilisées, quelles modalités, etc. Ce contexte est construit par le biais de la logique concepteur (définition des gestuelles par la logique application, des multimodalités par la logique système, etc.).

3.2.5. Contexte « Observation »

Ce contexte regroupe un **contexte 2D**, lié à l'image vidéo captée par une caméra (textures vidéo extraites ; zones 2D ; silhouettes ou contours de manière plus générale ; mesures de couleur, etc.) et un **contexte 3D**, relatif à la scène 3D telle qu'elle est modélisée par le système (ressources interactives présentes au sein de la scène 3D ; positions et rotations, absolues ou relatives, des objets ; vitesses et accélérations, absolues ou relatives, des objets ; zones 3D ; caractéristiques des structures 3D représentant objets, etc.). Ces données sont acquises par le biais d'algorithmes de traitement (segmentation, calcul de flux optiques, etc.) et de synthèse d'images (simulation physique de la scène, etc.).

L'ensemble de ces caractéristiques permettent l'extraction de nouvelles connaissances, pouvant ou non être utilisées comme informations contextuelles, telles que les structures spatiales au sein de la scène, la caractérisation des gestuelles de l'utilisateur, la caractérisation d'une des parties du corps de l'utilisateur, etc.

3.2.6. Contexte « Temps »

Le contexte « Temps » est caractérisé par une mesure de temps, quelle qu'elle soit : une heure, une date, un instant, un intervalle de temps. Les mesures de temps peuvent être obtenues par le biais d'une horloge interne, d'un agenda, etc. ou peuvent être définies arbitrairement par la logique concepteur. A partir de données bas niveau, il est également possible de construire des informations plus haut niveau.
Associé à d'autres informations contextuelles, il permet de construire un contexte temporel autour de celles-ci.

3.3. Quelles informations pour quel contexte ?

Les sections précédentes montrent à quel point la notion de contexte est rendue complexe par la diversité des points de vue. De plus, il existe un nombre quasi illimité d'informations, diversifiées et importantes, qu'il est possible de capter pour construire un modèle de contexte.

En fonction des besoins et objectifs du concepteur, ce dernier devra déterminer de quels types de contextes il peut avoir besoin, à quel niveau de l'architecture du système, et de quelles informations il peut disposer pour les construire. Il est bien évident que, selon les applications développées, certains contextes sont plus importants et prioritaires que d'autres. A ce titre, les contextes qui permettent de répondre aux questions qui ? où ? quand ? sont souvent appelés, dans la littérature, les contextes primaires, démontrant leur importance en priorité vis-à-vis des autres contextes.

Pour chaque contexte dont il a besoin (à un instant donné, à un certain niveau du système, etc.), le concepteur devra alors déterminer les différentes informations, qui seront les bases de la construction du modèle. Là encore, en fonction du système qu'il élabore, il devra déterminer si une information qu'il peut capter est véritablement utile et bénéfique pour la construction du contexte. Toutes les informations ne sont pas nécessaires et certaines informations peuvent être redondantes ou dépendantes avec d'autres. De plus, il devra définir le degré de précision avec lequel il veut capter ces informations, déterminant ainsi celui du contexte qu'il construira par la suite.
[Bremond & Thonnat 96] est un bon exemple de démarche, pour identifier les différentes sources d'informations contextuelles, en fonction des besoins du concepteur. Dans ces travaux, un système de surveillance vidéo est élaboré, le but étant d'interpréter l'activité au sein de la scène observée. Les auteurs cherchent à optimiser et à fiabiliser le processus d'interprétation de la scène. Ce processus est alors divisé en 4 étapes :

(1) La première étape concerne l'ensemble des traitements d'images, qui permettront de détecter et de suivre les régions en mouvement ;

(2) La seconde étape à pour objectif d'associer aux zones en mouvement détectés des objets d'intérêt, actifs et mobiles ;
(3) La troisième étape s'occupe de l'analyse comportementale de ces objets d'intérêt, interprétant les comportements de ces objets, en fonction des événements qui ont lieu autour de ceux-ci ;
(4) Enfin la dernière étape, qui peut rentrer en jeu à n'importe quel moment du processus d'interprétation, regroupe l'ensemble des raisonnements spatiaux, permettant de déterminer les relations dans l'espace entre les objets d'intérêt et la scène.

Ce travail est particulièrement intéressant dans sa démarche de conception, car pour chaque étape, les sources d'informations, qui pourront optimiser cette dernière, sont clairement et précisément identifiées, permettant alors sa contextualisation. Ces sources d'informations sont la scène (contexte « Scène réelle »), le temps (contexte « Temps »), l'image en elle-même vis-à-vis de son acquisition (contexte « Interface », interface matérielle), et les requêtes d'un superviseur.

Pour conclure cette section, nous ajouterons que le concepteur doit véritablement se demander s'il a besoin d'une information captée donnée. Certes, ils existent de nombreux contextes et de nombreuses informations permettant de les construire. Mais, certaines informations, même si elles permettent la construction efficace d'un contexte, ne sont pas forcément nécessaires, le contexte construit n'étant pas toujours exploitable dans les systèmes actuels. [Chen & Kotz 00] soulève ce problème dans le cas des systèmes mobiles. Il expose effectivement qu'un grand nombre d'informations sont disponibles, pour construire un modèle de contexte complet, telles que la localisation de l'utilisateur, le temps, la proximité d'autres personnes vis-à-vis de l'utilisateur, l'activité de l'utilisateur, etc. Mais, dans la plupart des applications développées actuellement, seule la localisation de l'utilisateur est utilisée pour construire le modèle de contexte permettant l'adaptation de ces dernières.

4. Gestion du contexte au sein d'un système interactif

Un système interactif traditionnel met en place, au cours de l'interactivité, des comportements mécaniques et automatisées en réponse aux commandes de l'utilisateur. Les comportements d'un tel système ne dépendent pas du profil de l'utilisateur, des caractéristiques de l'environnement, etc. Ce n'est pas le cas pour un **système sensible au contexte d'interaction** au sein duquel l'utilisateur évolue.
Un contexte, lorsqu'il est actif, fournit au système un ensemble de données qui peuvent influencer aussi bien les mécanismes mis en jeu lors de la réponse du système à l'utilisateur (**logique application**), que ceux relatifs au fonctionnement propre du système et à la gestion des connaissances au sein de celui-ci (**logique système**).
Le concepteur d'un système interactif sensible au contexte doit faire face à plusieurs difficultés. Tout d'abord, il doit déterminer **quelles informations contextuelles** sont nécessaires et compréhensibles **pour quel processus** au sein de l'architecture. Le système doit donc être capable :

(1) de gérer l'importante diversité des informations contextuelles comprises dans le modèle de contexte ;
(2) de manipuler ces données, en totalité ou partiellement ;

(3) d'attribuer certaines informations contextuelles spécifiques aux processus cibles du système.

Les informations contextuelles génèrent alors un ensemble de mécanismes (aussi bien au niveau de l'utilisateur qu'au niveau du système en lui-même) que le système doit pouvoir mettre en œuvre efficacement. De plus, l'architecture du système doit pouvoir faire face facilement et rapidement, d'une part, aux changements de contextes au cours de l'interactivité, d'autre part, aux changements de scénarios. Enfin, comme nous l'avons vu, les systèmes sensibles au contexte peuvent être des systèmes distribués. C'est le cas des systèmes ubiquitaires par exemple. En ce cas, le concepteur devra ajouter aux difficultés décrites précédemment, celles propres à de tels types de systèmes.

Nous présentons dans cette section différents travaux traitant du problème de la gestion de contexte au sein d'un système interactif.
La Section 4.1. met en évidence la nécessité de **couches intermédiaires logicielles**, par le biais de la présentation de différentes architectures logicielles de systèmes sensibles au contexte.
La Section 4.2. s'arrête sur ces couches, appelés parfois **serveurs de contexte**, qui permettent la gestion du contexte, tout en assurant la bonne modularité de l'architecture. Il existe deux types de serveurs : les serveurs **centralisés** et les serveurs **distribués**.
La Section 4.3. présente **plusieurs problématiques** qui devront être abordées et traitées par le concepteur au cours de l'élaboration du système sensible au contexte.

4.1. Gestion du contexte par le biais de serveurs dédiés

Nous allons mettre en évidence dans cette section le besoin au sein de l'architecture du système, de **couches logicielles intermédiaires**, type middleware, appelées parfois des **serveurs de contexte**. Ces serveurs sont dédiés à la gestion du contexte indépendamment du fonctionnement global du système.

Dans ses travaux, Schmidt propose l'architecture du système ***TEA*** (Technology for Enabled Awareness) [Schmidt & al. 99a, Schmidt 02] (voir Fig. 2.7.). Cette architecture comprend 4 couches principales et 2 couches optionnelles, nécessaires dans le cas d'un système distribué :

(1) La couche **'capteur'**, *sensor*, englobe l'ensemble des capteurs physiques, mesurant les stimuli extérieurs de l'environnement, et l'ensemble des capteurs logiques, captant les informations provenant du système en lui-même (l'heure par exemple).
(2) La couche **'file d'attente'**, *cue*, filtre et transforme, en temps réel, les données brutes issues des capteurs en un ensemble de données compréhensibles par le système. Grâce à cette couche, les capteurs peuvent être remplacés, sans changer l'ensemble des traitements au sein du système.
(3) La couche **'contexte'**, *context*, est responsable de la construction du modèle de contexte à partir des données en sortie de la couche 'file d'attente'. Cette construction est basée sur de simples règles logiques.
(4) Enfin la dernière couche **'application'**, *application*, inclut les informations contextuelles, au sein des mécanismes compris dans l'application.

Dans ces travaux, plusieurs systèmes (principalement des systèmes mobiles) et applications ont été développés, se basant sur la construction de contextes simples et non ambigus.

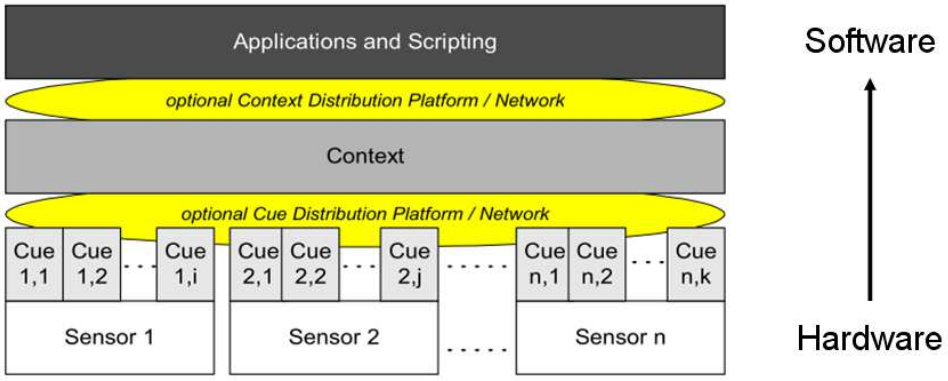


Figure 2.7. : Architecture du système ***TEA***, tiré de [Schmidt & al. 99a, Schmidt 02]

Typiquement, comme le montre l'architecture du système ***TEA***, la gestion du contexte requiert l'ajout de couches logicielles intermédiaires, type middleware, au sein de l'architecture du système. Ainsi, la modularité de l'architecture est assurée, ainsi que sa robustesse, d'une part, face aux changements d'applications et d'autre part, lors de l'utilisation d'un autre modèle de contexte. Un autre exemple d'architecture présentant une telle couche intermédiaire est donné par [Bolchini & al. 07]. Dans cette architecture (voir Fig. 2.8.) est ajoutée une couche nommée *context-based data tailoring*, dont le but est de rendre exploitable pour l'application, l'ensemble des données fournies par les différents contextes potentiels, au sein desquels l'utilisateur peut évoluer. Ces données pouvant être diversifiées, ce module se charge de traduire les données issues de contextes identifiés pour le système.

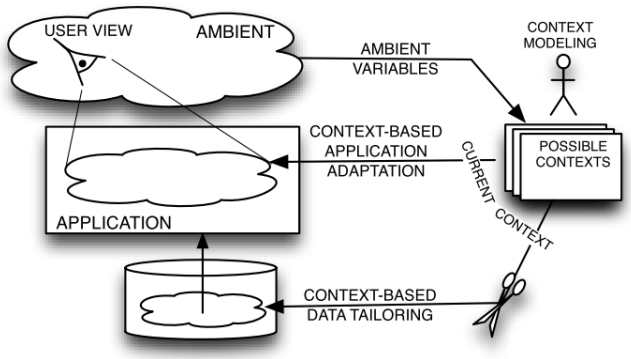


Figure 2.8. : Architecture d'un système sensible au contexte, tiré de [Bolchini & al. 07]

Face au nombre important de capteurs et de modèles de contextes différents, ces types de couches intermédiaires logicielles, appelées parfois **serveurs de contexte** (*context server*) sont absolument nécessaires, permettant la gestion du contexte indépendant de l'application et du fonctionnement global du système. Ces serveurs doivent donc permettre le stockage, le partage et l'archivage des données contextuelles ; la gestion d'un modèle de contexte évoluant

au cours de l'interactivité et des données contextuelles diversifiées ; et le filtrage, la traduction et la transmission des données contextuelles aux processus cibles du système. Certains serveurs peuvent également fournir un ensemble de services en adéquation avec les données contextuelles considérées.

4.2. Context server

[Chen & Kotz 00] proposent un état de l'art concernant ces serveurs de contexte (majoritairement dans le cas de systèmes mobiles). Ils distinguent deux types d'infrastructures : les infrastructures **centralisées** et les infrastructures **distribuées**.

Les serveurs centralisés, plus faciles à mettre en œuvre, collectent les données brutes issues des capteurs, les traitent et les interprètent, avant de transmettre un ensemble de données contextuelles haut niveau, compréhensibles par le reste du système. Ce type de serveurs permet d'assurer une gestion plus facile, rapide et séquentielle des données contextuelles. D'un autre côté, de tels systèmes sont difficiles à faire évoluer en taille et en complexité.

Les serveurs distribués, quant à eux, permettent la gestion à distance des informations contextuelles. Autrement dit, les informations contextuelles ne se trouvent pas à un endroit donné du système. Ces systèmes sont plus difficiles à réaliser et sont de plus en plus complexes à développer et à maintenir, au fur et à mesure qu'ils deviennent importants, en termes de taille, de capteurs utilisés et de services fournis à l'utilisateur.

[Salber & al. 99] proposent une SDK (*Software Development Toolkit*), trousse à outils logiciels, appelée la ***Context Toolkit***, permettant le développement de systèmes sensibles au contexte (mobiles, diffus et ubiquitaires), dont le serveur de contextes peut être centralisé ou distribué. Cet outil, s'inspirant des outils utilisés pour le développement d'interfaces graphiques, s'articule autour de la notion de *context widget*, composant logiciel autonome et indépendant, fournissant à l'application un accès aux informations contextuelles. Les *widgets* se situent entre les capteurs du système et des interpréteurs de données, chargés de collecter plusieurs données et de les fusionner en de nouvelles informations, ou des serveurs de contexte. Ces serveurs sont chargés de regrouper et de stocker les informations contextuelles, et de transmettre ces informations à l'application. La *toolkit* rend transparent, pour le concepteur, la potentielle distribution de l'architecture et modère l'ensemble des communications entre les différents modules de traitements.

4.3. Problématiques liées à la gestion du contexte au sein d'un système interactif

Les **problématiques** liées à la gestion de modèles de contexte au sein d'un système sont évidemment nombreuses, le domaine d'étude étant récent. Il est possible de trouver dans la littérature plusieurs travaux présentant ces différentes problématiques, comme [Pascoe & al. 99]. Nous essayons d'énumérer dans cette section les principales problématiques auxquelles le concepteur du système doit faire face, ainsi que différents travaux qui y ont répondu.

La prochaine sous-section traite tout d'abord des problèmes liés aux **capteurs du système**, chargés d'observer la scène. Puis, suivant les différentes étapes du traitement des données captées, les sous-sections suivantes s'attaqueront aux problèmes liés à **la collection et à**

l'interprétation des données captées (Section 4.3.2.) ; à **la construction du contexte d'interaction** en fonction de ces données (Section 4.3.3.) ; aux **différents raisonnements effectués à partir des informations contextuelles** (Section 4.3.4.) ; et à **la mise en place de l'adaptativité** au sein du système en fonction des informations contextuelles considérées (Section 4.3.5.). Nous conclurons cette section par des remarques plus générales (Section 4.3.6.).

4.3.1. Problématique liée aux capteurs

La première problématique, à laquelle devra faire face l'architecte du système, est construite autour des **capteurs** utilisés par le système. Le but des capteurs est d'**observer la scène au sein de laquelle l'activité prend place**. Ils ont un rôle majeur quant à la modélisation du **contexte d'interaction courant**. Ces capteurs peuvent être **matériels**, physiquement présents au sein de l'environnement réel, ou **virtuels**, dans le cas d'une modélisation virtuelle de la scène observée. Généralement, les capteurs sont séparés du reste du système, par un ensemble de composants logiciels qui se chargent de collecter les données brutes et de les traduire en un format compréhensible par l'application, à qui ils les transmettent.

En fonction du système et de l'application développés, la première question à se poser concerne **la nature des données à mesurer**. Quelles données seront pertinentes au cours de l'interaction et dans quelle mesure vont-elles faciliter la construction du modèle de contexte ? Cette question est loin d'être triviale car elle est dépendante de nombreux éléments, comme les objectifs que devra atteindre l'utilisateur par exemple, et le concepteur se retrouvera également face à un choix, quasi illimité et évoluant littéralement de jour en jour, de capteurs ou d'outils pour les développer.

Le concepteur devra également s'intéresser aux traitements informatiques nécessaires après la captation des données brutes par le capteur. En effet, certaines captations peuvent s'appuyer sur des algorithmes de traitements d'images gourmands et longs, en amont de tout autre traitement et ralentissant ainsi le système. Par exemple, l'usage de trames ***Bayer*** pour une image donnée nécessitera des algorithmes d'interpolation plus ou moins lents pour reconstituer l'image visuelle à interpréter.

Selon [Schmidt 02], et selon un point de vue particulièrement ubiquitaire, un capteur matériel pourra être jugé selon plusieurs critères :

1. *Un capteur doit être esthétique et doit pouvoir s'intégrer facilement et rapidement au sein d'un environnement ou d'un périphérique.*
   Il doit pouvoir se fondre dans l'environnement, être l'environnement et non pas être un ajout de matériel altérant celui-ci. De plus, l'utilisateur ne doit pas avoir à s'adapter à celui-ci. Si l'utilisateur doit adopter un comportement particulier pour manipuler ce capteur, alors ce comportement doit être naturel, démontrant la facilité d'utilisation du capteur. Il est à noter que ces remarques sont également vraies dans le cadre d'un capteur virtuel.

2. *Un capteur doit être rapidement et efficacement opérationnel.*
   Un capteur est opérationnel une fois allumé et calibré et nécessite de l'énergie pour fonctionner. Les temps d'initialisation et de calibrage doivent être minimisés, la

calibrage doit être simplifiée voire automatique et la consommation d'énergie doit être basse. De plus, et la remarque suivante s'applique également aux capteurs virtuels, un capteur doit être autonome et ne nécessiter que peu de maintenance de la part du concepteur. Enfin, un capteur, qu'il soit matériel ou virtuel, doit pouvoir être remplacé sans avoir à modifier l'ensemble de l'architecture du système.

3. *Un capteur doit être précis, robuste et fiable.*
   Mis à part pour la robustesse, ces remarques sont également vraies pour les capteurs virtuels.

4. *Un capteur matériel, si l'interaction le requiert, doit être facilement manipulable par l'utilisateur.*
   Il doit donc être portable et ses dimensions et poids doivent être acceptables. De plus, par sa forme et ses dimensions, il ne doit pas encombrer l'utilisateur au cours de son interaction avec le système.

5. *L'utilisateur doit accepter l'usage de ce capteur.*
   Il ne doit pas porter atteinte à sa sécurité ou à sa vie privée, par exemple.

6. *Les coûts d'achat et d'installation de ce capteur doivent être acceptables.*
   Ces coûts correspondent, dans le cas d'un capteur virtuel, aux coûts de développement informatique et de maintenance.

7. *Un capteur doit pouvoir être facilement manipulable et combinable avec d'autres capteurs.*
   Un bon capteur, matériel ou virtuel, doit pouvoir se combiner facilement et rapidement avec d'autres capteurs, qu'ils soient de même nature ou de natures différentes. De plus, il doit pouvoir être déplacé et positionné facilement et rapidement.

Pour conclure, [Schmidt 02] proposent un état de l'art sur les différentes technologies matérielles existantes actuellement pour capter les données brutes, issues de l'environnement et qui serviront à la construction du modèle de contexte. [Chen & Kotz 00] ajoute à cette liste les technologies matérielles permettant de mesurer un intervalle de temps ou de donner l'heure, la date, etc. Ce travail présente également les technologies permettant de capter l'inclinaison, les vibrations ou encore la proximité d'autres entités vivantes. La Figure 2.9. énumère ces technologies. Nous distinguons dans cette énumération les capteurs matériels et virtuels. De plus, nous rajoutons les technologies permettant de capter l'état courant du système (états courants de certains processus, de l'interface, du réseau, etc.).

	Capteurs matériels	Capteurs virtuels
Image visuelle	V	V
Lumière	V	V
Son	V	X
Position et orientation, inclinaison	V	V
Mouvement et accélération	V	V
Détection de mouvement	V	V
Interaction (tactile ou non) de l'utilisateur	V	V
Localisation géographique	V	X
Proximité d'objets ou de personnes vivantes	V	V
Etats du système (processus, réseau, interface, etc.)	X	V
Biocapteurs	V	X
Température, humidité, pression	V	X
Poids	V	X
Odeur, goût	V	X
Gaz (CO_2 par exemple)	V	X
Vibration	V	X
Champ magnétique	V	X
Capteurs zero-power	V	X

Figure 2.9. : Technologie de captation, inspiré de [Schmidt 02, Chen & Kotz 00]

4.3.2. Problématique liée à la collection des données

La deuxième problématique notable regroupe l'ensemble des difficultés auxquelles le concepteur doit faire face lors du développement du module de **collection de données**. Ce module récupère l'ensemble des données brutes issues des capteurs et les traite de manière à les rendre compréhensibles par le système, notamment par le module suivant dédié à la construction du modèle de contexte à proprement parler. Le module de collection de données peut également être appelé un **« interpréteur de données »**.

Le module de collection de données devra pouvoir gérer aussi bien un très petit nombre qu'un nombre très important de données. De plus, ces informations sont parfois très diversifiées et ce module doit pouvoir faire face à cette hétérogénéité. Certains travaux s'adressent également au problème de la qualité des données, aussi bien en entrée qu'en sortie du module de collection de données.
Toujours est-il que le problème principal concerne la **fusion** des données, de manière à les formater et les traduire, pour le reste du système. Symboliquement, la fusion de données permet de rendre plus compréhensible et plus naturel le message de l'utilisateur. Cette fusion permet la construction d'informations contextuelles de plus haut niveau à partir d'un ensemble de données de plus bas niveau. [Wu & al. 02] s'adresse au problème en proposant une architecture logicielle dédiée à la fusion de données pour comprendre le contexte d'interaction courant. Les capteurs peuvent être distribués et reconfigurés au cours de l'interactivité. La fusion de données est particulièrement abordée dans le domaine de la conception de systèmes multimodaux, où un ensemble de données issues de canaux hétérogènes doit être traité [Dumas & al. 09].

4.3.3. Problématique liée à la construction du modèle de contexte

Une des problématiques majeures concerne la **construction du modèle de contexte** à partir des données collectées. Un contexte peut être plus ou moins bas niveau, un contexte bas niveau étant construit à partir des données brutes issues des capteurs, et un contexte haut niveau utilisant les différentes connaissances relatives à l'activité courante de l'utilisateur et des gestuelles qu'il a adoptées et qu'il est supposé adopter. Pour un contexte bas niveau, les données issues des capteurs ont des caractéristiques similaires, d'une situation à une autre et pour un même contexte. Pour un contexte haut niveau, il sera possible par exemple de s'appuyer sur l'agenda de l'utilisateur, sur l'observation purement visuelle de la scène ou sur des techniques d'Intelligence Artificielle qui utiliseront les données bas niveau [Schmidt & al. 99b].

La construction du modèle de contexte peut être automatique, au contraire complètement supervisée, ou à la fois automatique et supervisée. De plus, cette construction peut se faire de manière centrale ou distribuée. Le modèle de contexte peut être construit avant l'interactivité, et peut être mis à jour dynamiquement au cours de l'interactivité.
Enfin, il est également possible d'implémenter des mécanismes pour apprendre de nouveaux contextes au cours de l'interactivité. Cet apprentissage des « bonnes situations » ou des « bons contextes » permet l'interprétation fiable et rapide des gestuelles de l'utilisateur et de l'activité de manière plus générale. Les modèles de contexte restent difficiles à acquérir et à apprendre automatiquement à partir des données collectées, cela d'autant plus si ce sont des modèles complexes. [Brdiczka & al. 07] propose une méthode pour apprendre et construire ces modèles de situations à partir des feedbacks de l'utilisateur. Ces modèles sont tirés de [Crowley & al. 02, Crowley & al. 06]. Un modèle de contexte initial est introduit au sein du système de manière supervisée avant l'interactivité à proprement parler. Puis le modèle est modifié au cours de l'interactivité, prenant en compte le profil et les préférences de l'utilisateur, dans le but de lui offrir un ensemble de services adéquats. Au sein d'un modèle, une situation peut être ajoutée, supprimée, modifiée, décomposée, ainsi que les services associés à celle-ci. Des chaînes de Markov cachées sont utilisées pour apprendre les situations au cours de la supervision et de l'adaptation du système aux nouvelles situations. Cette méthode souffre du fait que les rôles des entités participant à une situation, ainsi que les relations entre les entités, ne sont pas acquis et que le modèle de situation appris dépend fortement du graphe initial, introduit au sein du système avant l'interactivité. [Brdiczka & al. 06b] propose un *framework* complet pour l'acquisition automatique d'un modèle de situations à partir des données collectées. Différents scénarios, et donc différents contextes, sont joués et sont appris à partir de séquences vidéo. Cet apprentissage pourra alors permettre la reconnaissance de scénarios, observés réellement. L'approche est appliquée dans le domaine de la surveillance vidéo.

4.3.4. Problématique liée aux raisonnements à partir des données contextuelles

La problématique suivante est relative à l'ensemble des **raisonnements effectués à partir des données contextuelles**, soit collectées, soit directement fournies par le contexte d'interaction courant. L'objectif est de transmettre au système l'ensemble des informations contextuelles, issues du contexte, pertinentes pour mettre en place **l'adaptativité** du système vis-à-vis de l'utilisateur. C'est pourquoi les informations contextuelles doivent être interprétées, pour pouvoir les utiliser de manière significative et pertinente.

Ces raisonnements impliquent le filtrage des données, la sélection des données les plus pertinentes en fonction du contexte, la mesure de la qualité des données considérées, captées ou construites, la mesure de la qualité du contexte courant, etc.

Un point important, soulevé dans [Schmidt 02], est que le modèle de contexte n'est pas forcément une connaissance certaine. **Ce modèle doit être analysé et interprété**, avant d'être utilisé. En effet, un contexte décrit un ensemble de caractéristiques qui peuvent être observées dans la situation d'interaction courante. Cette description possède un certain degré de précision et de granularité. Un contexte mal décrit, ou décrit de manière incomplète, pourra être source d'erreurs. Par exemple, [Schmidt 02] donne l'exemple du contexte « l'utilisateur dort », dont la description est la suivante : « Il fait sombre, c'est silencieux, l'environnement est intérieur, l'heure correspond à une heure de la nuit, l'utilisateur est en position horizontale, sa position absolue est stable et il n'effectue que certains mouvements spécifiques pouvant être facilement décrits ». Dans ce cas précis, l'interprétation du contexte courant n'aboutira pas si l'utilisateur dort la lumière allumée par exemple.

Cette problématique regroupe également les problèmes liés à l'**ambigüité de sens** que peut présenter une donnée contextuelle, ainsi qu'à son incohérence vis-à-vis du contexte courant. De plus, les informations contextuelles peuvent se révéler incomplètes et/ou imprécises. Les raisonnements devront donc pouvoir prendre en compte la gestion de données contextuelles incertaines. Dans [Ranganathan & al. 04], le contexte d'interaction est représenté par le biais de prédicats, eux-mêmes organisées par une ontologie particulière. Cette ontologie regroupe et structure un ensemble de contextes d'interaction types. Chaque prédicat est associé à une valeur de confiance, comprise entre 0 et 1, calculée par le biais de règles probabilistes ou de logiques floues. Ces valeurs permettent le raisonnement à partir de données incertaines. L'activité de l'utilisateur est structurée par le biais de réseaux bayésiens, dont chaque nœud correspond à un prédicat et chaque arc décrivant une relation causale entre deux nœuds. Dans le cadre de ces travaux, le système ***Gaia*** est développé, infrastructure pervasive, middleware et distribuée, permettant le raisonnement à partir de données contextuelles incertaines. Le lecteur intéressé pourra trouver, dans ces travaux, d'autres références traitant du problème du raisonnement à partir de données incertaines.

Enfin, un des derniers problèmes, fortement lié à la problématique du raisonnement à partir des données contextuelles, concerne la **gestion des connaissances** au sein du système. En effet, ces connaissances, selon le système développé, peuvent être nombreuses et hétérogènes et doivent pouvoir être introduites au sein du système avant l'interactivité, et construites ou mises à jour au cours de l'interactivité. Il sera donc parfois nécessaire de constituer donc un module (ou des modules selon les niveaux de sémantiques du système) spécifiquement dédié à la gestion des connaissances.
Par exemple, il pourra être possible, comme [Zaidenberg & al. 09], d'ajouter à l'architecture une base de données de connaissances (voir Fig. 2.10.). Cette base de données, dite ubiquitaire et constituant la mémoire et le savoir du système, est partagée par l'ensemble des processus, à la demande de ces derniers. Cette base de données est divisée en quatre parties : une partie 'historique', regroupant les cycles de vie des modules du système, les événements survenus au cours de l'interactivité et les actions prises par le système ; une partie 'infrastructure', rassemblant les propriétés de l'environnement ; une partie 'utilisateur' décrivant les différentes caractéristiques des différents utilisateurs ; et une partie 'services', fournissant sur demande les services offerts à l'utilisateur, en fonction de ses comportements, et mis en œuvre par le système.

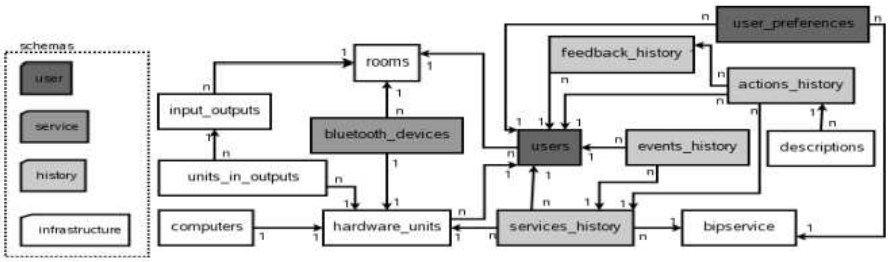


Figure 2.10. : Modélisation d'une base de connaissances, inspiré de [Zaidenberg & al. 09]

4.3.5. Problématique liée à la mise en place de l'adaptativité

La **mise en place des mesures adaptatives** au sein du système est la dernière étape de la réponse du système vis-à-vis de l'activité. Cette étape a lieu après la captation et la collection des données, la construction du contexte et enfin le raisonnement à partir de celui-ci. Nous donnons ici deux exemples de mesures adaptatives s'effectuant en fonction du contexte d'interaction courant, donnant un aperçu des différentes dimensions de la problématique présentée dans cette section.

Le premier exemple concerne la réponse établie à l'utilisateur par un système multimodal [Dumas & al. 09]. Cette réponse peut se faire par le biais de plusieurs canaux hétérogènes. Le processus prenant en compte la réponse du système en entrée et la restituant sur plusieurs canaux à l'utilisateur en sortie s'appelle la fission multimodale. Dans un système sensible au contexte, cette fission s'effectue en accord avec le contexte d'interaction courant au sein duquel l'utilisateur évolue. La réponse est transmise à l'utilisateur par un « Gestionnaire de Dialogue », *Dialog Manager*, via le mécanisme de fission. Le « Gestionnaire de Contexte », *Context Manager*, s'occupe de collecter les données et de construire le contexte d'interaction courant. Ce module transmet alors aux autres unités de traitements, les informations contextuelles pertinentes, pour que ces dernières puissent adapter leurs comportements. Une des dimensions de la problématique présentée dans cette section concerne l'**adéquation de la réponse du système en fonction des informations contextuelles jugées pertinentes**.

Le deuxième exemple illustre le fait que le système doit être pensé précisément avant d'être développé véritablement. Pour cela, le concepteur devra s'appuyer sur de nombreux exemples d'applications et de contextes d'interaction. Cet exemple montre l'**interdépendance de certaines parties de l'architecture vis-à-vis du contexte d'interaction**. [Crowley & al. 02] décrit un mécanisme d'adaptation, fonction du contexte d'interaction courant, au niveau de l'architecture du système, dédié à améliorer la perception du système vis-à-vis des actions de l'utilisateur. Le modèle de contexte utilisé a été décrit dans la Section 2.2. Les différents éléments du modèle sont utilisés pour caractériser les processus mis en jeu lors de l'observation de l'activité de l'utilisateur. Ces processus ont pour objectif d'observer et de détecter les rôles des entités participant à la situation d'interaction courante, ainsi que les relations entre ces dernières. Pour chaque nouveau contexte ou pour chaque modification du contexte courant, l'architecture logicielle du système est reconfigurée, de manière à adapter le système à la nouvelle situation d'interaction observée. Autrement dit, l'organisation, appelée

fédération, des processus perceptuels du système reflète la situation et le contexte courants. Tant que l'utilisateur se trouve dans la même situation d'interaction, la fédération de processus reste inchangée. Ces processus d'observation doivent s'auto-réguler (capacité à gérer leurs différentes caractéristiques, relations, priorités, etc.), s'auto-décrire (description symbolique de leurs capacités et états) et s'auto-évaluer (évaluation de la qualité de leurs données en sortie). Pour plus de détails concernant cette fédération de processus, le lecteur intéressé pourra se référer également à [Crowley & Reignier 03].

4.3.6. Autres problématiques

Le concepteur de systèmes sensibles au contexte devra également se heurter à des problématiques plus générales.

Il devra par exemple veiller à la **modularité de l'architecture** du système. En effet, l'architecture du système ne doit pas être repensée pour chaque type d'application, et donc pour chaque nouveau scénario.
De plus, et ce problème est important, le concepteur devra particulièrement **étudier les communications**, et notamment **les flux de données**, au sein du système. En effet, l'importance, le nombre et la diversité des données pourront s'avérer être des problèmes de conception considérable. De plus, les transmissions de données seront dépendantes des profils matériels envisagés des différents composants du système.
Il sera également parfois nécessaire de **changer de modèles de contexte**. Le système doit donc être capable de formater et de traduire de manière cohérente, les informations issues de contextes différents (au sens modélisation), de manière à ne pas avoir à repenser complètement son architecture logicielle à chaque fois.
Enfin, un système sensible au contexte devra être capable de **gérer**, facilement et rapidement, **l'ajout ou la suppression de nouveaux capteurs**, **de nouvelles informations contextuelles** et **de nouveaux services** à offrir à l'utilisateur en fonction des comportements de ce dernier.

5. Positionnement de notre approche

Au sein du système développé dans le cadre de cette thèse, l'estimation du contexte d'une situation d'interaction donnée nous permet de fiabiliser et d'améliorer le processus d'interprétation de l'activité.

Nous observons, par le biais de notre système, les interactions entre plusieurs **participants** (l'utilisateur, le système ainsi que ses différents processus qui entrent en jeu au cours de l'interaction), évoluant au sein d'une **scène virtuelle**. Cette scène virtuelle est la fusion de la modélisation de la **scène réelle** capturée (une salle de classe, dans notre cas) avec celle d'une **scène 3D**, établie à partir du scénario de l'application. L'**activité** observée est constituée d'un ensemble d'**interactions**, entre l'utilisateur et le système. C'est pourquoi, il est question de **situation d'interaction** et de **contexte d'interaction**.
Une **situation d'interaction** décrit un cadre scénarisée entre deux états stables de l'univers, au sein de laquelle l'activité, mettant en jeu les acteurs évoluant dans la scène, est observable. Les différentes interactions entre les acteurs sont donc structurées au sein de la situation d'interaction. Les objectifs que doivent remplir les différents acteurs, en particulier l'utilisateur, sont définis par les scénarios mis en place par la logique concepteur.

A l'instar de [Dey 01], nous pensons que le **processus de contextualisation** permet d'expliquer une situation d'interaction donnée. Dans notre cas, il doit permettre l'amélioration du processus d'interprétation de l'activité, particulièrement les gestuelles utilisateur. L'existence du contexte à trois niveaux vis-à-vis de la situation nous semble profitable, puisque elle exprime la granularité de cette notion, ainsi que celle de la situation, nous offrant alors de larges possibilités quant à la description de ces dernières. Autrement dit, une situation et un contexte d'interaction peuvent être décrits sur plusieurs niveaux d'abstraction.

La bonne interprétation de l'activité se fera en partie par la **confrontation du contexte d'interaction observé**, construit à partir de données bas niveau telles qu'un ensemble d'images vidéo, **avec le contexte d'interaction attendu** par le scénario et extrait de ce dernier. La distance qui peut exister entre ces deux contextes regroupent les distances entre les différents sous-contextes (« Utilisateur », « Système/Application », etc.) observés et attendus, confrontés à tous les niveaux du système.
Il sera donc question, d'une part, de la **construction de connaissances** à partir de données bas niveau, illustrant ainsi le contexte d'interaction observé, comme il est question dans [Brémond 97], et, d'autre part, de l'**extraction de connaissances** à partir de la modélisation du contexte attendu, compris dans le scénario, sujet traité dans [Crowley & al. 06]. Nous souhaitons adopter, dans ce travail, des démarches descriptives similaires. Ceci implique que les différents contextes dans lesquels l'utilisateur doit évoluer soient prévus et travaillés au cours de l'élaboration du scénario de l'application.
La modélisation du contexte d'interaction nous permettra bien sûr de supporter le processus d'interprétation de l'activité observée, dans le but d'y répondre **de manière adaptée** et ainsi d'**améliorer l'interactivité entre l'utilisateur et le système**. Mais, le modèle de contexte nous permettra également d'**optimiser la gestion des connaissances** (filtrage, sélection des connaissances pertinentes, construction de nouvelles informations de plus haut niveau) et de mettre en place des **mécanismes d'adaptation** vis-à-vis du système, pour fiabiliser le fonctionnement global de celui-ci. Il nous faut noter la forte interdépendance entre l'adaptation du système et les informations contextuelles comprises dans le modèle. En effet, ces informations seront de puissants paramètres de l'adaptation, mais l'adaptation du système concernera également l'ajustement de ces informations.

Dans ces travaux, notre contribution se situe au niveau de la proposition d'un modèle. Notre contribution se situe également au niveau de la conception d'une architecture logicielle capable de gérer un ensemble de données contextuelles spécifiques issues de ce modèle. Notre modèle doit donc présenter plusieurs propriétés. En revanche, nous ne proposons pas de contributions au niveau de la couche sémantique de l'architecture correspondant à la logique concepteur. Autrement dit, nous ne nous occupons pas, dans ces travaux, de l'élaboration, de l'implémentation et de la gestion du modèle de contexte au plus haut niveau de l'architecture. Pour nous, le scénario, et donc les contextes d'interaction dans lesquels l'utilisateur évolue, sont déjà pensés et implémentés.
Suivant la classification énoncée dans la Section 3.1., notre modèle de contexte doit comporter les **6 types de contextes**. Plus particulièrement, le contexte « Utilisateur » nous donnera des informations sur la gestuelle observée de l'utilisateur ; le contexte « Système/Application », par le biais des contextes calculatoires des processus et du contexte application, nous donnera des moyens d'orienter les différents modules compris au sein du système et nous indiquera le contexte d'interaction attendu au sein duquel l'utilisateur doit évoluer ; et le contexte « Observation » nous délivrera les informations 2D et 3D, qui seront le point de départ de l'interprétation de la gestuelle de l'utilisateur par le système. Nous nous

efforcerons de détailler l'ensemble des contextes compris dans notre modèle (Chapitre 5.), une fois l'architecture du système bien établie (Chapitre 3.).
L'ensemble des données que nous allons traiter dans notre système est moyennement important et hétérogène. Il s'agira à première vue d'un ensemble de paramètres et de données image, qu'il est facile de gérer. Cependant, à partir de ces données bas niveau, de nouvelles connaissances seront construites (une gestuelle à partir de données image par exemple), qui peuvent s'avérer plus difficiles à gérer, suivant les représentations et formalismes adoptés. De plus, plusieurs degrés de granularité du modèle de contexte seront abordés (au niveau de l'utilisateur, d'une gestuelle, d'un processus informatique, etc.). En revanche, nous nous appliquerons pour le moment à veiller à ce que les ambigüités de sens et les incertitudes soient simples à résoudre et que le modèle de contexte ne présente pas de données incomplètes, dans un souci de validation de cette première étape de modélisation. Dans le cadre de ces travaux, nous ne développons qu'un système particulier mais d'autres systèmes sont étudiés actuellement au sein de l'équipe ***ImagIN***, utilisant le même modèle de contexte. Pour finir, l'implémentation informatique du modèle n'est pas étudiée dans le cadre de cette thèse. C'est pourquoi, nous n'adoptons pas de formalismes particuliers pour cette tâche. Tout formalisme capable de décrire le modèle que nous proposons est *a priori* envisageable.

Concernant la conception du système sensible au contexte, notre approche est majoritairement **centralisée**. Le système commercial de capture de gestuelles[3], augmenté de nos contributions (Chapitre 4.) implique une distribution, peu importante, des données et des commandes. Il sera donc nécessaire de ne pas oublier cette distribution lors de la gestion des données contextuelles au sein de ce système et de l'adaptation de ce dernier.
D'une part, pour chaque étape scénarisée du jeu, le scénario fournit, par le biais d'un module dédié, un **contexte d'interaction attendu** sous la forme de données contextuelles particulières. D'autre part, à partir des données bas niveau captées par le système, au niveau de l'observation de la scène, les données contextuelles correspondant au **contexte d'interaction observé** sont construites. Les données contextuelles observées sont alors confrontées aux données contextuelles attendues. Encore une fois, notre travail concerne la gestion de ces données contextuelles mais pas nécessairement la gestion du modèle de contexte extrait du scénario. Nous n'aurons donc pas, *a priori*, la nécessité de construire et de gérer le modèle informatique de contexte. Ces modèles sont élaborés au cours de la conception.
Nous allons potentiellement nous heurter à des problèmes de **collection de données** issues des capteurs, collection à partir de laquelle le contexte d'interaction observés sera construit.
Nos **mécanismes de raisonnement** doivent alors nous permettre de comparer la distance qu'il peut exister entre le contexte d'interaction observé et le contexte d'interaction attendu (et donc de la gestuelle utilisateur observée avec celle attendue).
A partir de ces raisonnements, nous pouvons donc mettre en place les **mécanismes d'adaptation**, fournis par le scénario, à tous les niveaux du système. Tout d'abord, dans un souci de validation de notre proposition, les mécanismes d'adaptation sont des règles et des conditions simples et relativement basiques. En revanche, il nous faudra certainement raisonner à partir de mécanismes haut niveau, délivrés par le scénario, pour en établir de plus complexes et plus spécialisés (à un processus donné par exemple).

[3] Nous rappelons ici le cadre scientifique particulier de cette thèse (thèse en convention ***CIFRE*** avec la société XD Productions). Cette société développe un système de capture de mouvements que nous utilisons dans ces travaux de thèse. Nous présenterons ce système plus en détails dans les chapitres suivants.

Nous partons du principe que les contextes d'interaction, construits ou extraits du scénario sont complets mais les travaux présentés dans le cadre de l'interprétation de données incertaines sont tout à fait envisageables.
Enfin, nous n'échappons pas, pour chaque nouveau scénario mis en œuvre, à la redéfinition du modèle de contexte.

6. Contribution à la modélisation du contexte

Nous proposons dans cette section une modélisation du contexte. Cette modélisation se base, comme le présentera la Section 6.1., sur la caractérisation des **acteurs** de l'interaction, au cours du temps, par un **vecteur d'états**. Nous montrerons alors, au cours des Sections 6.2. et 6.3., que ce vecteur d'états est modifié au fil des **situations** et des **contextes d'interactions**, cadres scénarisés, que nous définirons, où évoluent les acteurs de l'interaction en tant que participants. Puis nous proposerons un **modèle de scénario**, basé sur l'organisation structurelle de situations et de contextes d'interaction (Section 6.4.). Pour conclure, en nous replaçant dans le cadre de cette thèse, nous étudierons la construction de notre modèle de contexte par le biais d**es différentes sources d'informations** dont nous disposons, lors de la conception du système et de l'application et au cours de l'interaction (Section 6.5.).

6.1. L'univers sous la forme d'un vecteur d'états

Nous choisissons, selon la classification des contextes décrites précédemment, de diviser l'**univers** en **5 entités**, appelées **acteurs de l'interaction** : l'utilisateur, le système/l'application, la scène réelle, l'interface, et l'observation. Le contexte temps n'est pas représenté ici car les informations temporelles seront considérées différemment dans notre modèle, comme nous le justifierons par la suite.
La classification des contextes, que nous avons exposée précédemment, nous a permis de répertorier et de classer les différentes entités mises en jeu au cours de l'interaction. Mis à part pour le temps, les acteurs au sein de notre modèle sont donc les entités **de plus haut niveau**.

Les acteurs de l'interaction sont composés de plusieurs **entités** de plus bas niveau, elles-mêmes composées d'entités, etc., qui pourront évoluer au cours du scénario. L'utilisateur sera composé de bras, de jambes, etc., le système de processus, de diverses ressources logicielles et matérielles, de variables, etc. et ainsi de suite pour tous les acteurs de l'interaction.
Nous supposons que l'univers est modélisable par un **vecteur d'états**, ou mot d'états (voir Fig. 2.11.). Ce n-uplet (collection ordonnée de n états) constitue la base théorique de notre contribution, au niveau de la modélisation du contexte.
Ce vecteur d'états, défini à un instant donné, peut être décomposé en 5 sous-vecteurs, chacun correspondant à un acteur de l'interaction. A leur tour, le vecteur d'état, caractérisant un acteur donné, peut être divisé en plusieurs sous vecteurs, correspondant aux entités composant l'acteur. Sous vecteurs qui peuvent être à leur tour divisés… et ainsi de suite jusqu'aux entités de plus bas niveau.
Un vecteur d'états regroupe donc l'ensemble **des états**, données numériques ou attributs sémantiques, **qui caractérisent une entité**, à un instant donné, sans prendre en compte ses objectifs. Tous les états, de toutes les entités, de tous les acteurs ne sont *a priori* pas tous énumérables, de manière exhaustive.

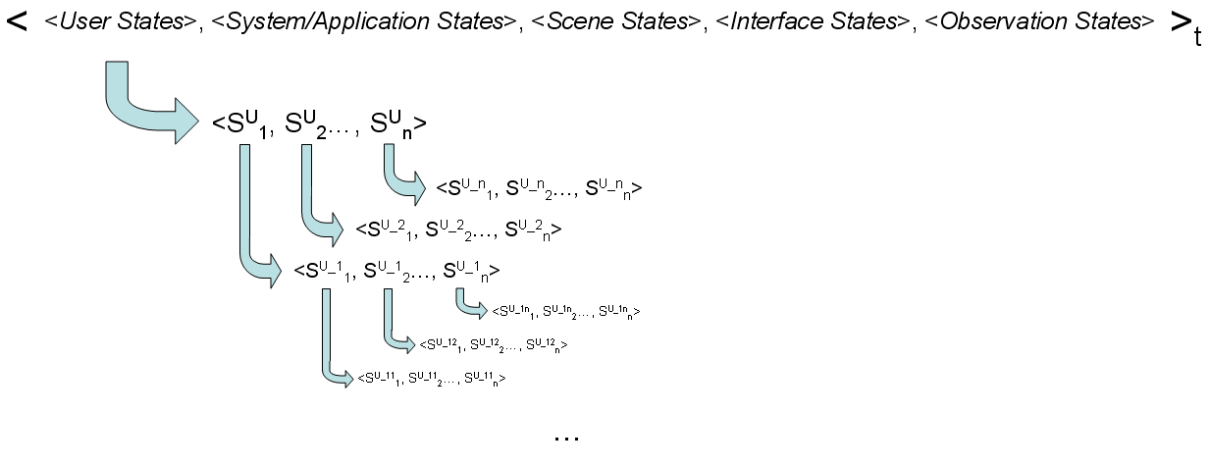


Figure 2.11. : Modélisation de l'univers sous la forme d'un vecteur d'états

6.2. Situation d'interaction

Une **situation d'interaction** (voir Fig. 2.12.) décrit une étape scénarisée entre **deux états stables de l'univers**, identifiés explicitement par le concepteur. Elle prend en compte différents acteurs de l'interaction, renommés alors les « **participants** » de la situation. De plus, par le biais d'un processus de sélection et de filtrage, toutes les entités, composant un participant, ne sont pas prises en compte au cours de la situation. Il en est de même pour les états d'un participant et des entités associées.

Au cours de la situation, chaque participant est guidé par ses règles comportementales et ses objectifs propres. Les états du participant évoluent donc en réponse aux comportements adoptés par ce dernier. Mais ces états peuvent également évoluer sous l'effet des autres participants, qui peuvent les utiliser en tant que ressources, paramètres ou outils, au cours de leurs propres comportements.

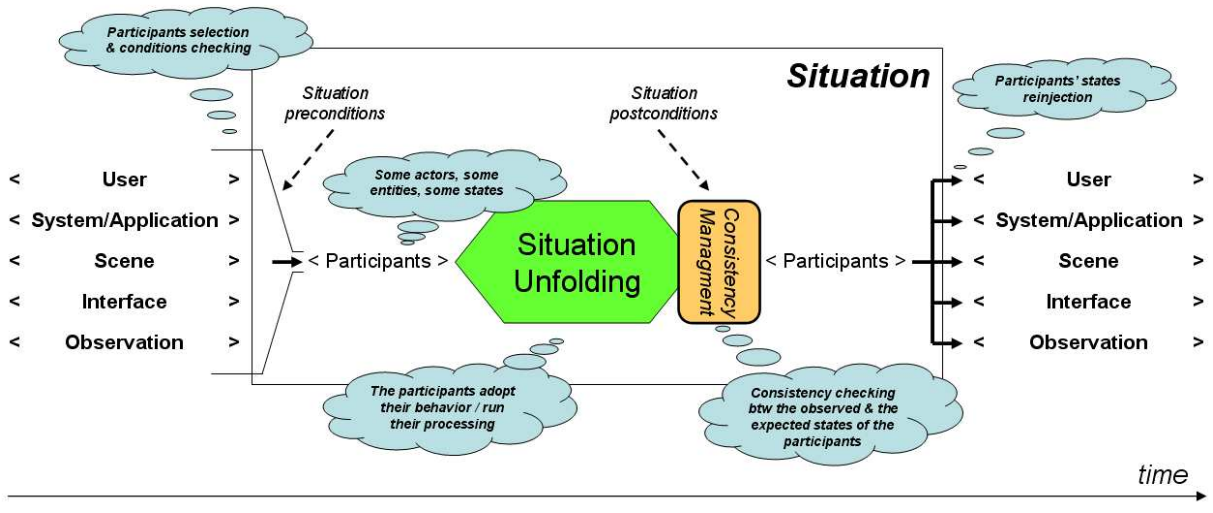


Figure 2.12. : Situation d'interaction

Une situation d'interaction peut être considérée à la fois d'un point de vue **statique et dynamique**.
D'une part, la situation en elle-même n'impose aucun comportement de la part des participants, chacun suivant son propre scénario. C'est pourquoi la description d'une situation est générique et illustre son statisme. Elle regroupe l'ensemble des événements qui peuvent avoir lieu au sein de celle-ci.
D'autre part, son exécution est dynamique et dépend de l'interactivité scénarisée entre le système et l'utilisateur.

Une situation est constituée :

- **de pré-conditions :** ces pré-conditions concernent les participants à la situation d'interaction, ainsi que certains de leurs états. Seuls les participants remplissant ces pré-conditions sont acceptés par la situation d'interaction. Les pré-conditions peuvent être organisées en plusieurs **sets**, chaque set traduisant un **début de situation**. Une situation est activée et commencée si l'ensemble des pré-conditions d'un même set ont été remplies.
- **de post-conditions :** les post-conditions sont appliquées sur certains états spécifiques des participants à la situation d'interaction. La complétion de ces post-conditions indique que la situation d'interaction s'est déroulée et qu'elle est alors terminée. Les différents états, sélectionnés et modifiés par la situation, sont réinjectés au sein de l'univers. La situation modifie donc l'état de l'univers par son exécution. A l'instar des pré-conditions, les post-conditions peuvent être organisées en différents **sets**, chaque set correspondant à une **fin de situation**.
- **d'un déroulement :** ce déroulement représente plus un **cadre libre**, une « boite noire » au sein du scénario, où les participants exécutent des comportements particuliers en accord avec leurs objectifs et scénarios propres. Dans la mesure où les comportements des participants ne sont pas tous modélisables à tout moment, le déroulement d'une situation est non prévisible. Ce déroulement a lieu une fois que toutes les pré-conditions de la situation ont été remplies.
- **d'un processus de gestion de cohérence :** ce processus illustre l'incertitude relative au déroulement de la situation. Il gère la cohérence entre le vecteur d'états résultant des comportements des participants au cours du déroulement de la situation, et le vecteur d'états tel qu'il devrait être pour remplir les post-conditions de la situation d'interaction. C'est ce processus qui détermine si la situation est terminée ou si les participants n'ont pas encore rempli leurs objectifs. C'est un processus, bien sûr, dépendant des objectifs des participants et de la suite du scénario et qui, selon les cas, pourra être souple, acceptant un état de l'univers imparfait, après le déroulement de la situation ; ou plus stricte, n'acceptant qu'un état de l'univers complet et remplissant parfaitement les post-conditions.

Une situation d'interaction est définie selon un degré de **granularité** bien défini. En effet, une situation peut mettre en jeu d'autres éléments d'interaction (situations et contextes) plus bas niveau, précisément définis et structurés, ou peut être englobée par un élément de plus haut niveau (voir Figure 2.13.). Nous parlons de **situation atomique** lorsque la situation

d'interaction se situe au niveau le plus bas et n'englobe pas d'éléments d'interaction de plus niveau ; et de **situation composée** lorsque cette situation englobe d'autres éléments de plus bas niveau.

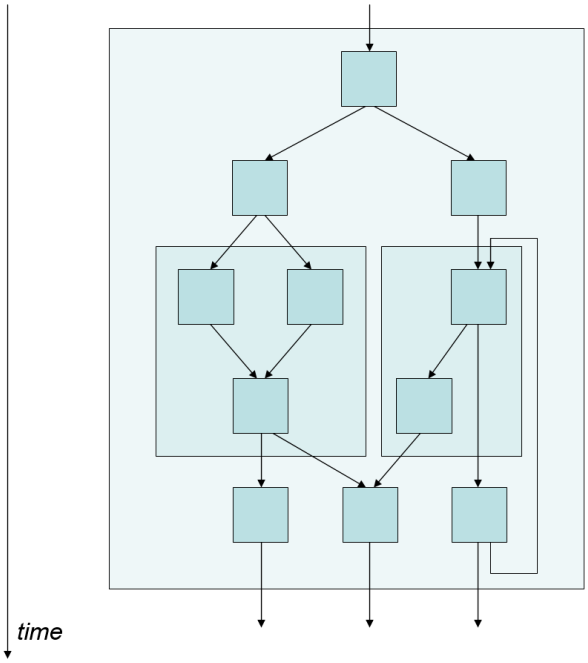


Figure 2.13. : Situation composée de plusieurs situations, sur 3 niveaux de description

6.3. Contexte d'interaction

Dans notre modèle, un **contexte d'interaction** (voir Fig. 2.14.) est similaire à une situation d'interaction, dont la modélisation a été décrite précédemment. Un contexte d'interaction est également un **cadre scénarisée**, entre deux états stables de l'univers, et au sein duquel différent **participants** identifiés évoluent.

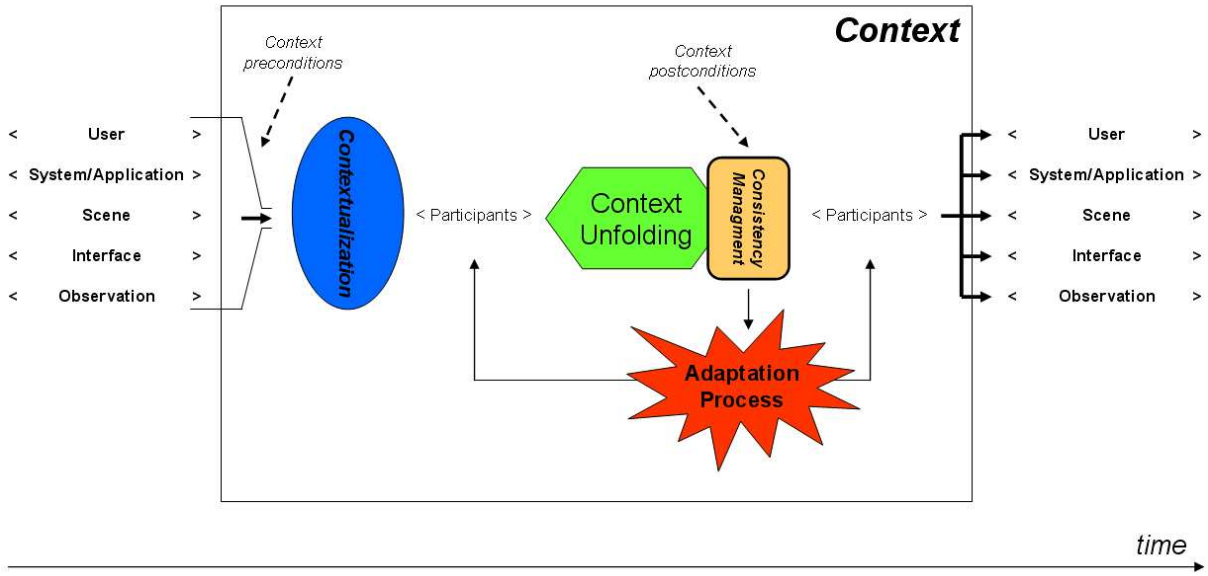


Figure 2.14. : Contexte d'interaction

Un contexte d'interaction est constitué :

- **de pré-conditions :** A l'instar de la situation d'interaction, un contexte d'interaction est activé si un ensemble de pré-conditions sont remplies. Ces pré-conditions permettent, d'une part, la sélection des différents participants qui évolueront au sein du contexte et, d'autre part, la vérification des valeurs prises par ces participants, certaines de leurs entités et certains de leurs états. Ces pré-conditions peuvent être une condition nulle (seule la présence du participant, de l'entité ou de l'état est vérifiée), sur une valeur précise, sur un intervalle de valeurs, etc. Les pré-conditions peuvent être organisées en plusieurs sets, chaque set représentant un début potentiel de contexte.
- **de post-conditions :** Les post-conditions définies à ce niveau donnent une signification particulière au contexte d'interaction, puisqu'elles traduisent les objectifs que doivent atteindre les différents participants au cours de celui-ci. C'est à ce niveau que sera défini, entre autres, les gestuelles attendues de la part de l'utilisateur. Les post-conditions peuvent également être organisées en sets, chaque set traduisant une fin de scénario.
- **d'un déroulement :** Le déroulement est le cadre libre du contexte d'interaction, au cours duquel les différents participants adoptent différents comportements, en accord avec leurs objectifs et scénarios propres.
- **d'un processus de gestion de cohérence :** Ce processus a le même rôle que celui d'une situation d'interaction. Il interprète la cohérence du vecteur d'états résultant du déroulement du contexte, vis-à-vis d'un set de post-conditions. Ce processus peut être régi par un ensemble de règles simples, élaborés dans le but de résoudre les ambigüités qui peuvent exister si les objectifs atteints après le déroulement du contexte sont sensiblement différents des objectifs attendus, matérialisés par les post-conditions. Une fois la gestion de cohérence terminée, les différents participants, que le contexte d'interaction a modifiés, sont réinjectés au sein de l'univers.

- **d'un processus de contextualisation :** ce processus est spécifique au contexte d'interaction. Il se situe entre la vérification des pré-conditions et le déroulement du contexte d'interaction. Par le biais de ce processus, une nouvelle signification est apportée aux participants pris en compte par le contexte d'interaction. Les participants sont alors dits être **contextualisés**. La contextualisation entraine une focalisation et une orientation plus importantes des différents participants, vers les objectifs à atteindre pour faire évoluer le scénario.
  C'est par le biais de ce processus que certains aspects des participants sélectionnés vont être filtrés et mis en relief, leur fournissant ainsi des indications sur le comportement à adopter au cours du déroulement du contexte d'interaction. Ces aspects mis en relief constituent alors un ensemble d'informations contextuelles, disponibles au sein du système, au cours du déroulement du contexte. Comme nous l'avons expliqué précédemment, ces informations contextuelles peuvent être classées en 6 contextes : le contexte **« Utilisateur »**, le contexte **« Système/Application »**, le contexte **« Scène réelle »**, le contexte **« Interface »**, le contexte **« Observation »** et le contexte **« Temps »**.
  Cette contextualisation consiste en un formatage du mot d'états avant que le contexte ne se déroule et peut prendre la forme : d'une assignation de valeur, d'un tri des états selon leur importance vis-à-vis du contexte, d'une priorisation d'un ou de plusieurs états, d'une re-sélection, d'une mise en valeur des états pertinents et d'une non considération des autres et de conditions supplémentaires, etc.
  Au cours du déroulement du contexte, les participants focalisent et adaptent leurs comportements selon les différentes informations contextuelles. Sans informations contextuelles, un participant adopte des comportements plus généraux, pouvant présenter des ambigüités de sens (un comportement, pour une même situation, peut être différent d'un contexte à un autre). Mais, un participant, ayant reconnu le contexte dans lequel il évolue, adopte des comportements plus ciblés et directs, dont l'interprétation serait plus fiable et rapide. Pour les participants, connaître le contexte, i.e. prendre en compte les différentes informations contextuelles, n'implique pas automatiquement l'adoption du 'bon' comportement. Mais ils peuvent percevoir plusieurs indices supplémentaires sur le véritable sens du contexte et sur la pertinence de certains comportements qu'ils peuvent adopter.
- **d'un processus d'adaptation :** ce processus est également propre au contexte d'interaction. Il se déroule en parallèle par rapport au processus de gestion de cohérence. En fonction des résultats de la gestion de cohérence, le processus d'adaptation met en place plusieurs mécanismes adaptatifs, dans le but de clarifier le scénario pour les participants et de les orienter vers différents objectifs à atteindre.
  Si la gestion de cohérence entraine la fin d'un contexte d'interaction, le processus d'adaptation ajuste le vecteur d'états, dans le but d'orienter et de conforter les participants vis-à-vis de l'évolution du scénario. D'un autre côté, si le contexte d'interaction n'est pas terminé, les mesures adaptatives permettent de clarifier le contexte d'interaction pour les participants, pour que ces derniers puissent atteindre les objectifs qu'il définit. Cette clarification se fait également par l'ajustement du vecteur d'états.
  Ce que nous appelons 'ajustement' du vecteur d'états peut prendre la forme d'une assignation de valeur, d'un tri des états selon leur importance vis-à-vis du contexte, d'une priorisation d'un ou de plusieurs états, d'une re-sélection, d'une activation ou désactivation d'un comportement donné (un affichage par exemple), d'une mise en

valeur des états pertinents et d'une non considération des autres et de conditions supplémentaires, etc.
Les mécanismes adaptatifs sont paramétrés par les différentes informations contextuelles présentes au niveau du contexte. Le processus d'adaptation peut mettre en place un ensemble de mesures en accord avec un set particulier de post-conditions, mais il peut également activer plusieurs mécanismes généraux, quel que soit l'ensemble de post-conditions vérifié.

A l'instar de la situation d'interaction, le contexte d'interaction est défini selon un **niveau de granularité** bien particulier. Le contexte d'interaction définit ainsi un nouveau cadre scénaristique, plus ciblé et plus adapté, englobant un ou plusieurs éléments d'interaction (situation et contexte). Ainsi, au niveau du déroulement d'un contexte, se situe un scénario de plus bas niveau, composé de situations et de contextes d'interaction. Le déroulement du contexte implique le déroulement de ce scénario de plus bas niveau. A l'issue du contexte d'interaction, si le gestionnaire de cohérence ne fait pas aboutir ce dernier, le scénario de plus bas niveau est recommencé. Pour finir, un contexte d'interaction peut également être englobé au sein d'un élément d'interaction de plus haut niveau.
Chaque situation et chaque contexte, par le biais de pré-conditions, sélectionnent un certain nombre de participants. Ces participants, et leurs différents états, contextualisent implicitement tous les éléments de plus bas niveau englobés dans le déroulement de l'élément qui les ont sélectionnés. En effet, les éléments de plus bas niveau ne pourront considérer que ces différents participants, leur univers se limitant à eux. Ainsi, le déroulement de ces éléments de plus bas niveau sera, implicitement et automatiquement, orienté et cadré. C'est pourquoi, du point de vue d'un ensemble d'éléments d'interaction, il est possible de qualifier de « contexte » tout élément englobant ces derniers.
Le contexte d'interaction, à la différence de la situation, permettra une contextualisation explicite, plus centrée, plus orientée et plus focalisée, des éléments de plus bas niveau. Le déroulement des éléments contextualisés de plus bas niveau est donc influencé de manière beaucoup plus importante. Une bonne modélisation d'un contexte d'interaction influera directement sur le déroulement des éléments le composant, **sans modifier ces derniers** ainsi que **la nature des comportements des participants mis en jeu**. En effet, le déroulement d'un élément d'interaction n'est pas contrôlé par le système. Les participants à cet élément ne sont pas contraints d'adopter un comportement particulier. Au sein d'un contexte d'interaction, un élément d'interaction de plus niveau n'est pas remodelé et les comportements de ses différents participants ne sont pas altérés.

Notre système se sert des propriétés du contexte pour, d'une part, **orienter indirectement les comportements adoptés par les différents participants** (immersion des participants dans le contexte d'interaction attendu / modélisation du contexte attendu pour donner un objectif) et pour, d'autre part, **interpréter ces comportements** (reconnaissance du contexte observé dans lequel les participants évoluent / modélisation du contexte observé pour connaître un objectif). Le système met alors en œuvre des mécanismes adaptatifs plus ou moins directs et stricts pour orienter la gestuelle adoptée par l'utilisateur ou le comportement d'un processus mis en jeu au cours de l'interactivité.
Au cours de l'interactivité, le système cherche donc tout d'abord à modéliser un contexte d'interaction, en accord avec les objectifs du scénario (scénario de l'application, scénario suivi par un processus, etc.). Ce peut être un contexte 3D pour l'utilisateur, un contexte calculatoire pour un processus, etc. Ainsi, en orientant leurs comportements de manière plus ou moins directe, le système place les différents participants dans un nouveau cadre de

perception et d'analyse, définissant alors leurs objectifs et vis-à-vis duquel ils agissent en adéquation. Deuxièmement, le système analyse et interprète les comportements des différents participants en construisant, à partir des différentes observations effectuées, le contexte d'interaction dans lequel ils évoluent. Le système peut donc confronter le contexte d'interaction qu'il attend, défini par le scénario, avec le contexte d'interaction observé, construit à partir des comportements des participants. S'en suit un processus d'interprétation, en fonction de la distance pouvant exister entre le contexte attendu et le contexte observé. Cette interprétation, moins ambigüe, est plus simple, plus fiable car plus ciblée.

6.4. Représentation du scénario de l'application

L'état de l'univers évolue par le biais du déroulement de situations et de contextes d'interaction. Nous proposons de représenter un **scénario** par le biais de **situations** et de **contextes d'interaction**. La représentation d'un scénario à l'aide de situations contextualisées est actuellement un sujet d'études au sein du ***L3i*** de l'université de La Rochelle [Rempulski & al. 08].

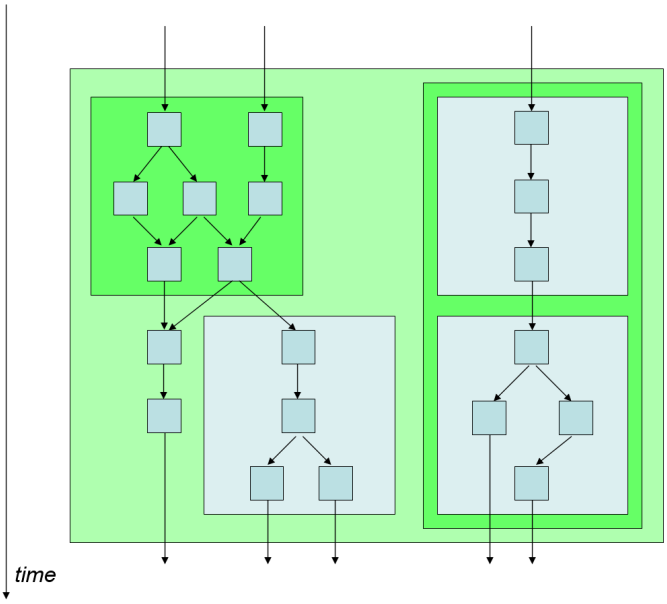


Figure 2.15. : Scénario sous la forme de situations et de contextes d'interaction

Le scénario peut en effet être décrit comme **une séquence organisée de situations et de contextes d'interaction, représentés sur plusieurs niveaux sémantiques** (voir Fig. 2.15.). Au sein d'un élément d'interaction pourra être modélisé un sous-scénario de situations et de contextes d'interaction. Ainsi, il est possible de définir un degré de granularité quant à la description du scénario de l'application.
Il est évidemment impossible de représenter l'ensemble des situations et contextes d'interaction dans lesquelles les acteurs de l'interaction peuvent évoluer. C'est pourquoi les différents éléments d'interaction modélisés définissent des instants clés, moments après lesquels le scénario évolue. Les comportements des participants n'étant ni définies, ni

attendues, en amont ou en aval de ces situations et contextes, tout comportement reste possible et n'engendrera aucun traitement spécifique de la part du système.

De par la triple nature du contexte (Section 1.2.3.), toutes les combinaisons d'organisation sont possibles. Le contexte, ou la situation, peut **englober contextes et situations**, ou peut **être compris(e) dans un contexte ou une situation de plus haut niveau**. La troisième nature du contexte est exprimée par le fait qu'un élément d'interaction, en sélectionnant un ensemble de participants, **contextualise implicitement** les éléments de plus bas niveau qu'il englobe. Un contexte d'interaction ajoute une **contextualisation explicite**, par le biais d'un processus particulier (voir section précédente).

Les situations et les contextes se déroulent comme il l'a été expliqué précédemment. Au cours de ce déroulement, tout participant adopte le comportement adéquat, en accord avec le scénario personnel qu'il suit. L'utilisateur adoptera une gestuelle pertinente vis-à-vis du scénario qu'il suit personnellement. Ces comportements sont observables au cours des phases de déroulement des éléments d'interaction, cadre libre dans lequel les participants peuvent évoluer librement.

L'auteur du scénario choisira d'exprimer une étape de ce dernier par le biais d'une situation ou d'un contexte d'interaction. Ce choix se fera, d'une part, en accord avec ses besoins en terme d'adaptation du système vis-à-vis des comportements des participants. Seul un contexte d'interaction pourra permettre cette adaptativité du système. D'autre part, ce choix se fera en fonction des informations qu'il désire présentes à un niveau de granularité donné. La situation permet de mettre en place un cadre informatif plus large et plus vague, au niveau des éléments d'interaction de plus bas niveau qu'elle englobe, qu'un contexte d'interaction.

Si plusieurs situations ou contextes sont éligibles au même instant au sein du scénario, la situation ou le contexte qui s'activera alors sera la première ou le premier dont les pré-conditions seront remplies. Il est également possible que plusieurs situations ou contextes s'exécutent en parallèle. Le concepteur devra alors veiller à ce qu'il n'y ait pas de conflits au niveau de la modification d'un participant, ou d'une entité ou d'un état en particulier, compris dans plus d'une situation et/ou plus d'un contexte.

Comme nous le verrons au cours de la Section 6.5.6., la définition d'un agenda s'accompagne de celle d'un **agenda**. Cet agenda permettra la temporisation structurée du scénario, tout en mettant en valeur des informations contextuelles particulières, telles que des durées d'événements, des instants, etc.

6.5. Construction de nos différents contextes

Le contexte d'interaction est construit à partir des états des différents acteurs de l'interaction : contexte **« Utilisateur »**, contexte **« Système/Application »**, contexte **« Scène réelle »**, contexte **« Interface »** et contexte **« Observation »**. Comme nous avons pu le voir précédemment, nous distinguons également un 6ème contexte, le contexte **« Temps »**, traduisant la dimension temporelle du contexte d'interaction. Nous nous intéressons ici à **la construction de ces 6 contextes**. La modélisation du contexte est fortement dépendante de l'application développée. Nous décrivons ici, de manière plus ou moins détaillée et non finale, ces 6 contextes.

6.5.1. Contexte « Utilisateur »

Le **contexte « Utilisateur »** comprendra principalement l'ensemble des informations, concernant l'utilisateur, permettant la caractérisation et l'interprétation de sa gestuelle.
Comme expliqué dans le prochain chapitre, sa morphologie sera acquise, de manière supervisée, lors d'une étape de calibrage, avant l'interactivité à proprement parler.

De plus, durant l'interactivité, la capture de mouvements nous permet d'avoir les poses (translations et rotations) de chacune des parties du corps de l'utilisateur.

L'animation de l'avatar au sein de la scène 3D, dans laquelle les lois physiques réelles sont en vigueur, nous permettent de connaître les vitesses et accélérations (linéaires et angulaires) des différentes parties du corps de l'utilisateur. Indirectement, via cette animation, il sera également possible de définir, en fonction de l'élément d'interaction courant, la possible connexion existante entre une partie du corps de l'utilisateur et le contenu de la scène 3D.

Le scénario de l'application est également une source d'informations, nous fournissant les objectifs globaux de l'utilisateur, ainsi que ses objectifs locaux, définis pas à pas, au cours de l'interactivité.

Enfin, ce contexte pourra comprendre éventuellement une description du profil de l'utilisateur, du point de vue de ses préférences de jeu. Ce profil pourra être acquis lors d'une phase de configuration avant ou au cours de l'interactivité, par l'utilisateur ou un superviseur tiers, ou avant même l'exécution de l'application par le développeur.

6.5.2. Contexte « Système/Application »

Un certain nombre de contextes matériels et logiciels seront compris dans le contexte **« Système/Applications »**, nous permettant d'assurer une robustesse des différents processus du système face aux changements de matériels ou de versions de bibliothèques logicielles. Par exemple, les caméras utilisées dans le système de capture de mouvements peuvent être remplacées, mais ce remplacement ne doit pas entraîner la modification des algorithmes traitant les flux vidéo. Ou encore, certains algorithmes pourront adoptés différents comportements, selon la version de la bibliothèque logicielle utilisée. Ces contextes pourront être établis à partir des différentes documentations, du matériel et du logiciel.

Le **contexte système** regroupe l'ensemble des contextes calculatoires des différents processus mis en jeu au cours de l'interactivité : capture de la gestuelle de l'utilisateur, observation des scènes réelle et 3D, caractérisation et interprétation des comportements de l'utilisateur. Ces contextes calculatoires comprennent par exemple les différents paramètres nécessaires aux processus pour assurer leurs traitements. Ces contextes sont déterminés sur la base de diverses expérimentations effectuées au cours du développement du système. Ils sont organisés par le biais de scénarios, définis lors de la conception du système par la logique système.

Enfin, le **contexte application** nous permettra de connaître, au cours d'un élément d'interaction donné, le contexte d'interaction associé attendu. Ces contextes d'interaction sont

organisés par le scénario de l'application. Chaque contexte d'interaction extrait de la logique application sera associé à un ensemble de contextes compréhensibles par les différents processus composant le système, défini donc au niveau de la logique système.
Il sera possible, entre autres, d'extraire de chaque contexte d'interaction la gestuelle de l'utilisateur attendue par le scénario ainsi que la réponse adaptée du système. Cette réponse adaptée sera déclinée à travers l'ensemble des couches sémantiques de l'architecture, et entrainera le comportement adapté de l'ensemble des processus du système. C'est pourquoi, à partir de la logique concepteur, le contexte d'interaction attendu permet au système de mettre en place un ensemble de mécanismes adaptatifs aussi bien du point de vue de l'utilisateur (mise à jour adaptée de la scène 3D) que du système en lui-même (adaptation des différents processus et gestionnaires). Les scénarios (logiques application et système) sont précisément établis au moment de la conception du système et de l'application.

6.5.3. Contexte « Scène réelle »

Le **contexte « Scène réelle »** est assez simple, puisque la scène observée est une salle de classe vide, relativement basique. C'est donc une salle intérieure, dont certains emplacements peuvent comporter du mouvement, et dont l'éclairage peut relativement varier. Ces connaissances seront particulièrement utilisées au cours du processus de capture.

6.5.4. Contexte « Interface »

Le **contexte « Interface »** comprend les informations qui caractérisent les points de vue réels et 3D (matrices caméras, cône de vision, etc.). Les paramètres relatifs à l'interface matérielle sont acquis au cours d'une face de calibrage, avant l'interactivité. Il en est de même pour les paramètres relatifs à l'interface logicielle, paramètres qui peuvent également être décrits dans le scénario de l'application (un mouvement de caméra 3D décrit dans le scénario, par exemple).

De plus, il peut regrouper un ensemble de techniques d'interaction spécialisées en accord avec l'application. En effet, prenant en compte l'interface utilisée, l'éventail des techniques d'interaction que peut adopter l'utilisateur est infini. Cependant, les objectifs définis par le scénario de l'application poussent forcément l'utilisateur à adopter une gestuelle connue, limitée et donc descriptible. Il est donc possible, pour un objectif donné d'énumérer un ensemble de techniques d'interaction pertinentes.

6.5.5. Contexte « Observation »

Le **contexte « Observation »** est fortement dépendant de l'interface de capture utilisée. Dans le cadre de cette thèse, la capture de la gestuelle utilisateur est basée sur l'observation de la scène réelle, uniquement par le biais de la vision. Le contexte « Observation » comprendra alors l'ensemble des informations 2D qui permettront la capture, la caractérisation et l'interprétation des gestuelles de l'utilisateur (silhouettes, zones 2D, etc.). Nous arrêterons plus précisément dans les prochains chapitres sur l'acquisition de ces informations 2D.

De plus, la gestuelle de l'utilisateur est interprétée à partir de l'observation de la scène 3D. Le contexte « Observation » comprendra donc tout ce qui est relatif au contenu de la scène et qui permettra l'interprétation de la gestuelle utilisateur (zones 3D, connections spatiales entre l'avatar et le reste de la scène, événements générés par les ressources interactives, caractéristiques liées à la représentation de l'avatar, etc.). C'est le scénario de l'application qui permettra la description de la scène et donc la construction de ce contexte.

6.5.6. Contexte « Temps »

Le dernier contexte, le **contexte « Temps »**, associe une temporalité à toute données citées précédemment, dans la mesure où cette temporalité s'avère pertinente lors de l'interprétation de la gestuelle de l'utilisateur : interactions passées, contextes d'interaction passées, gestuelles adoptées par l'utilisateur à un instant précis, etc. Ce contexte regroupe l'ensemble des informations temporelles qui peuvent permettre d'effectuer une prédiction à court terme, et donc une interprétation, de la gestuelle de l'utilisateur.

Ce contexte est un peu particulier puisqu'il ne peut pas être construit à partir du vecteur d'états modélisant l'univers. En effet, celui ne comporte que les états des acteurs de l'interaction, auxquels sont associés les contextes cités précédemment et qui sont définis à un instant donné. La question se pose donc sur la définition d'une source d'informations permettant d'établir le contexte « Temps ».

Nous avons vu qu'entre chaque situation et chaque contexte, l'univers est représenté par un vecteur d'états stable, défini à un instant donné. Cet instant, qui est une mesure de temps, est fixé par le concepteur du scénario, en fonction de l'application, du type de systèmes développés, du processus développé, etc. Il peut s'agir d'une heure de la journée, d'un jour dans le mois, du début d'un niveau de jeu, du début d'une conversation, ou de tout autre mesure de temps arbitraire ou non.
Toujours est-il qu'il est donc possible de découper le scénario en plusieurs intervalles de temps et sur plusieurs niveaux selon le degré de granularité du scénario. Nous élaborons donc alors un **agenda**, mettant en jeu des situations et des contextes d'interaction spécifiques, dans l'intervalle de temps défini par ce dernier (voir un exemple Fig. 2.16.).

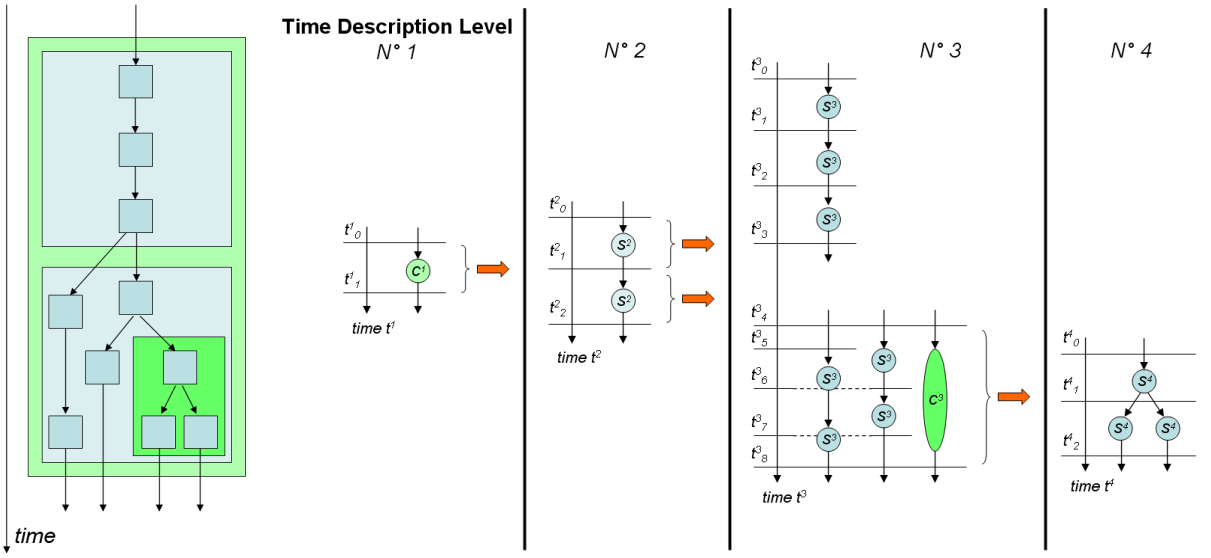


Figure 2.16. : Exemple de scénario et de son agenda associé

C'est cet agenda qui nous permet de construire le contexte « Temps ». En effet, à partir de cet agenda, il est non seulement possible de connaître l'instant auquel une situation ou un contexte donné se déroule, traduisant alors le présent. Mais il est également possible de s'y rapporter pour connaître un historique, une trace passé de l'interactivité, relativement à un participant. De la même façon, le scénario étant planifié, le système peut s'y référer pour anticiper un comportement, une réaction, un besoin, etc.

7. Conclusion : Evaluation de notre contribution

Nous allons présenter dans cette section une analyse de notre modèle de contexte, proposé précédemment. Ce modèle est actuellement en train d'être étudié sous différents aspects, au sein de l'équipe ***ImagIN*** : utilisation d'un tel modèle pour la résolution de quiproquos, mise en place de ce modèle dans un système dont l'adaptativité se situe au niveau du scénario, etc.
Nous rappelons ici que, bien que nous proposions un modèle de contexte, sa mise en œuvre et les mécanismes qui le régissent au sein d'un système ne font potentiellement pas l'objet de développement dans nos travaux de thèse. Nous nous intéresserons en revanche à la gestion des informations contextuelles à un niveau sémantique plus bas au sein du système.

Notre modèle semble cohérent et s'applique à un grand nombre de cas d'étude théoriques. Il est simple à comprendre et il est aisé d'imaginer pour le concepteur les mécanismes de mise en œuvre et de gestion de ce modèle. Les difficultés résident plus dans l'augmentation et la diversité des données à gérer. En effet, dans le cas d'un système relativement simple, mettant en jeu des processus bien définis et identifiés, les mécanismes restent simples. En revanche, plus le système se complexifie, plus ce modèle est difficile à créer et manipuler.
Comme nous l'avons fait au cours de la Section 6.5., il est important d'énumérer les différentes informations contextuelles qui seront nécessaires au cours de l'interaction, ainsi que les sources d'informations qui permettent de les déterminer. Cependant, ce travail reste fastidieux et relativement impossible à exécuter exhaustivement. En effet, le nombre

d'informations disponibles et leur diversité augmentent, au fur et à mesure que le système se complexifie.
Ce modèle ne propose pas de solutions vis-à-vis de l'hétérogénéité possible des données. En effet, nous parlons d'entités très diversifiées, qui peuvent aller d'un processus informatique à un comportement utilisateur. Une des suites de ces travaux serait donc d'étudier les structures de données possibles, pouvant être supportées informatiquement.

Le travail de définition du modèle sera important et sera fortement dépendant de l'application et du système développés. Plus le concepteur sera fin et prévoyant dans sa conception, plus le système pourra faire face à des ambigüités de sens, des incertitudes, des informations incomplètes, etc. Il devra détecter les événements clés au sein du scénario et les concrétiser par le biais d'une situation ou d'un contexte d'interaction.
Le grand atout de notre modèle, aussi bien au niveau de la situation que du contexte, est qu'il propose un cadre scénarisé complètement libre, qu'est le déroulement. Cette étape permet une grande flexibilité quant à la description du scénario. Cependant, cette souplesse peut constituer un important problème dans la mesure où le déroulement d'une situation ou d'un contexte peut résulter en un vecteur d'états complètement inattendu, incohérent et/ou incomplet. Cet aspect au niveau du modèle de contexte est étudié au sein de l'équipe ***ImagIN***, dans le cadre de travaux de vérification formelle [Rempulski & al. 08, Champagnat 11].
Notre modèle peut être défini à plusieurs niveaux de granularité et d'abstraction, en accord avec le scénario de l'application. La démarche du concepteur sera tout d'abord d'élaborer des situations et des contextes simples, aux participants bien identifiés et aux mécanismes simples et faciles à mettre en œuvre. Puis, en progressant doucement et prudemment, le concepteur pourra ajouter des informations et des mécanismes au fur et à mesure pour complexifier le système. Il aura en permanence à faire des compromis entre la précision du processus de contextualisation du vecteur d'états caractérisant les participants, la complexité de gestion de cohérence et le nombre et la complexité des mécanismes d'adaptation.

Nous pensons que notre modèle peut être utilisé dans le cadre de plusieurs types de systèmes. Il peut être supporté par différents formalismes, plus ou moins rigoureux, tels que le langage UML ou les réseaux de Petri.

L'agenda nous semble une plus value importante de notre modèle. Il permet la mise en œuvre du contexte temporelle dans tout type de systèmes, aussi bien du système mobile qui nécessite une mesure de l'heure, voire d'un agenda, que du système ubiquitaire englobant une salle de travail par exemple.
De plus, indirectement, il permet de connaître à quel niveau de granularité le concepteur doit s'arrêter dans la description du scénario. Autrement dit, l'agenda nous permet d'identifier une situation atomique d'une situation composée.
Nous nous plaçons constamment dans une démarche de composants réutilisables. Une situation sera décomposée en situations de plus en plus atomiques et de moins en moins composées. Cette décomposition s'arrêtera lorsque la situation sera atomique, c'est-à-dire qu'elle sera parfaitement réutilisable dans tout type de situations et de contextes. Une re-décomposition de celle-ci ne sera pas justifiée. L'agenda peut permettre d'identifier quand cette décomposition doit s'arrêter.
En effet, à l'instar du scénario, l'agenda va se décrire sur plusieurs niveaux et donc sur plusieurs échelles de temps. Ainsi, lorsqu'une situation correspond à une mesure de temps, que le concepteur n'a pas besoin de rediviser, alors cette situation est atomique.

Pour conclure, nous terminerons sur quelques mots concernant le scénario et sa représentation par le biais de situations et de contextes d'interaction. Notre représentation est proche de celle proposée dans [Crowley & al. 06, Brdiczka & al. 06b, Zaidenberg & al. 06, Brdiczka & al. 07, Zaidenberg & al. 09].

Notre représentation semble pertinente et permet la description du scénario sur plusieurs niveaux. A l'image du modèle de contexte, cette représentation est simple et cohérente. Mais bien que simple, la modélisation d'un scénario ne doit pas être prise à la légère. En effet, le scénario reste un ensemble important de données, lourd à manipuler et difficile à contrôler. Cependant, pour un scénario bien défini, cette représentation permet un contrôle puissant du récit narratif et illustre bien son évolution.

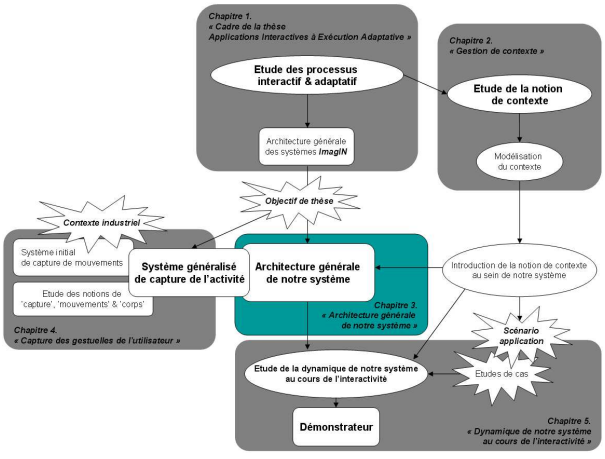
Une des perspectives de ce travail serait de concevoir un logiciel auteur de création et d'édition de scénarios représentés par le biais de situations et de contextes d'interaction. Ces situations et contextes seront alors des éléments informatiques que le concepteur pourra créer et éditer à loisir. Il aura la possibilité de structurer ces situations et contextes dans le temps et suivant plusieurs niveaux.
Un élément d'interaction bien défini sur plusieurs niveaux sémantiques pourra également être utilisé comme un élément à part entière, un macroélément éprouvé et contrôlé. La réutilisation de ces macroéléments automatisera les traitements et les décisions relatives à ces derniers. En effet, ces traitements et ces décisions seront automatiquement mis en œuvre, en réutilisant un macroélément déjà éprouvée au cours de la conception du scénario.

Chapitre 3. Architecture générale de notre système

Many scientists, when they found they couldn't do a problem finally began to study why not
- unknown -

Fringeons !!!
- GH -

Résumé



Les deux chapitres précédents nous ont permis d'étudier l'ensemble des aspects que doit traduire l'architecture de notre système. En accord avec nos différentes hypothèses et nos objectifs de thèse, nous proposons au cours de ce chapitre **l'architecture finale de notre système**.

Notre architecture est divisée en **4 couches sémantiques spécifiques** : support opérationnel, gestion de la scène virtuelle, analyse du dialogue interactif et logique concepteur. Nous détaillons dans ce chapitre les différents modules de traitement composant chaque couche. Ces modules constituent une étape bien particulière de l'interaction se déroulant entre l'utilisateur et le système. Nous détaillons nos contributions vis-à-vis de notre architecture.

Plan du chapitre

L'objectif de cette thèse étant de concevoir un système interactif complet, sur lequel s'exécutent de manière adaptée des applications scénarisées, nous proposons au cours de ce chapitre **le modèle d'architecture de notre système**. Nous définissons l'architecture d'un système interactif comme étant le cadre de conception et d'implémentation, sur différents niveaux d'abstraction, de ses différents modules logiciels de traitements et des connections existant entre ces derniers (voir Annexe A.). L'architecture proposée est à mettre en relation avec celle d'un **système adaptatif**, **développé dans le cadre des travaux de l'équipe *ImagIN*** et présenté au cours du premier chapitre de cette thèse (voir Fig. 3.1.).

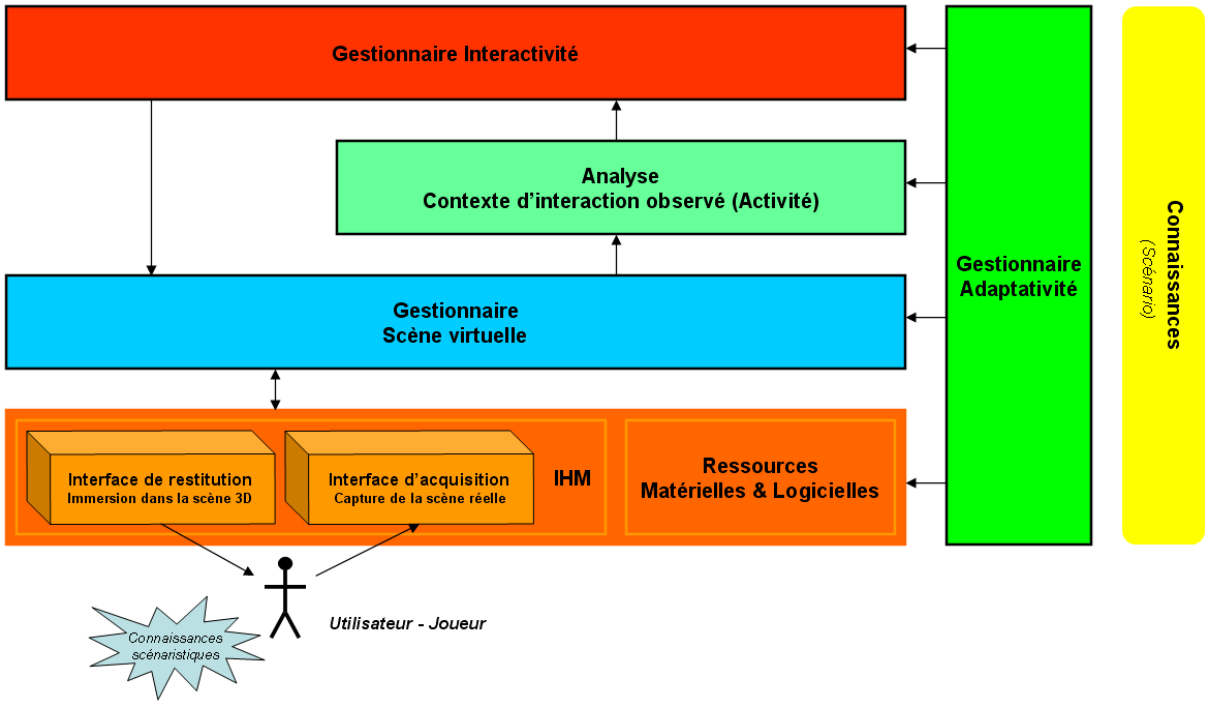


Figure 3.1. : Architecture d'un système adaptatif, dans le cadre des travaux de l'équipe ***ImagIN***

Les règles de conception que nous avons établies au cours du premier chapitre sont globales et peuvent s'appliquer dans le but d'élaborer un système particulier dans le cadre du contexte de recherche, défini par l'équipe ***ImagIN***. En plus de ces règles, **les objectifs de ce travail de thèse** nous permettent d'établir **une description précise et spécialisée de notre système**, ainsi que du processus interactif prenant place entre ce dernier et l'utilisateur.
Dans notre système, **l'utilisateur** interagit avec une **application scénarisée** par le biais de sa **gestuelle**. En fonction du **scénario** qui lui est proposé, l'utilisateur adopte une gestuelle explicite (ou implicite, i.e. réalisée de manière non consciente) **pour interagir avec une scène 3D**, qui lui est visuellement restituée de manière **immersive**. En plus de cette gestuelle peuvent être enregistrés **divers événements** dont la source n'est pas l'utilisateur (un spectateur par exemple). L'ensemble de **l'activité** au sein de **la scène réelle** est alors capturée en **temps réel**, par le biais d'un système de capture de gestuelles, le ***Cyberdôme***, augmenté de nos contributions (voir Chapitre 4.).
Les modélisations de la **scène 3D** et de la **scène réelle** capturée permettent alors la mise en place et la gestion d'une **scène virtuelle**, **système d'informations virtuelles traduisant l'état du système à un instant donné**. L'activité retranscrite au sein de cette scène virtuelle est **observée** par un processus dédié du système. Les différentes observations permettent ainsi

la caractérisation du contexte d'interaction au sein duquel l'activité prend place. Ce **contexte d'interaction observé** est alors confronté au **contexte d'interaction attendu** extrait du scénario, permettant donc **l'interprétation de l'activité** au sein de la scène.
Par la suite, le système met en place **une réponse interactive et adaptée** aux gestuelles de l'utilisateur et événements ayant eu lieu au sein de la scène réelle. Elle résulte en une modification de la scène 3D restituée à l'utilisateur de manière immersive et **une adaptation des comportements des différents processus** composant le système, et ce à tous les niveaux sémantiques de l'architecture du système.

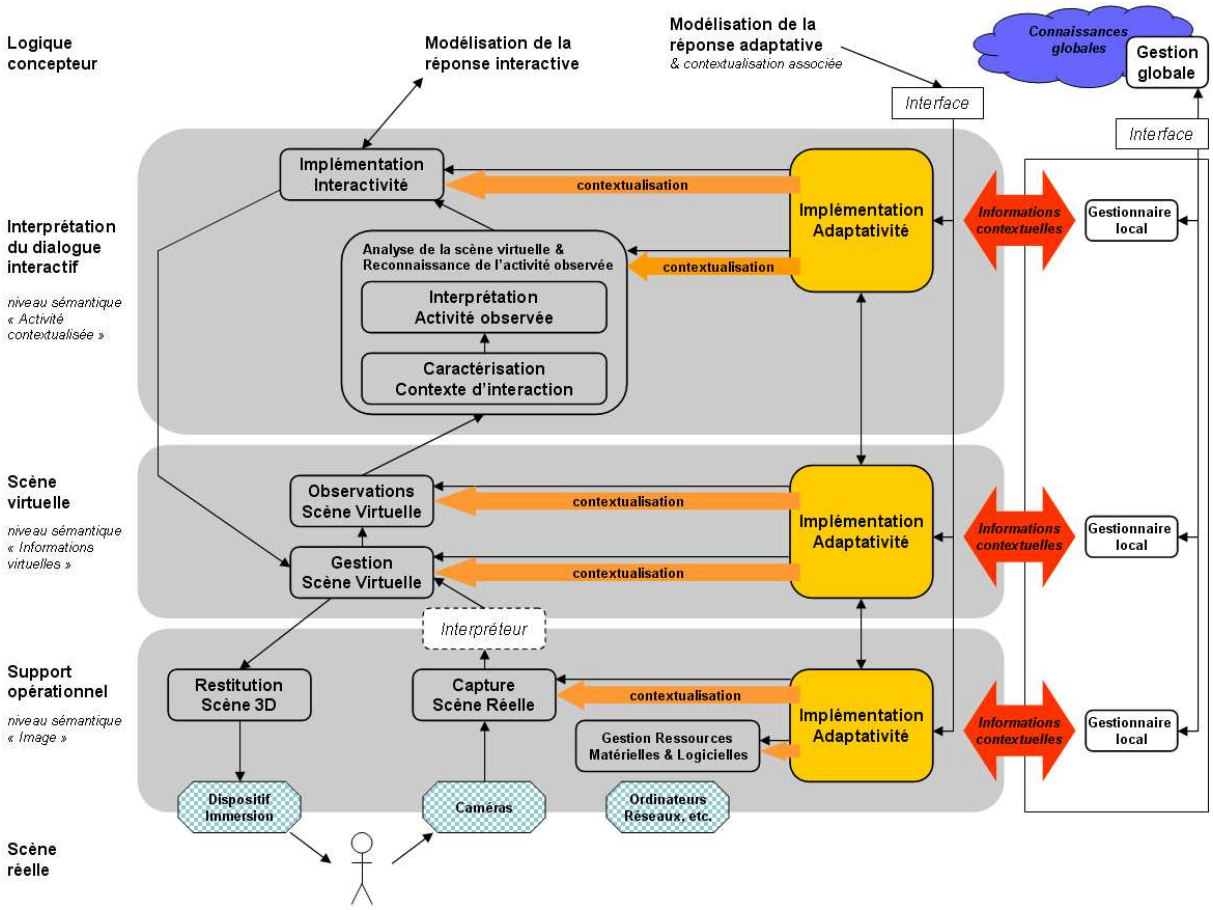


Figure 3.2. : Architecture du système conçu dans le cadre de ces travaux de thèse

Il est ainsi possible de proposer **une architecture de système**, répondant à l'ensemble des hypothèses de travail énoncées précédemment (voir Fig. 3.2.). Cette architecture est donc la spécialisation de l'architecture proposée par la Figure 3.1., vis-à-vis de ces travaux de thèse.
L'architecture de notre système a été établie sur **4 couches**, chacune correspondant à un niveau sémantique donné. Ces couches, en plus de permettre l'agencement des modules de traitements composant le système, illustrent **les différentes étapes de l'interactivité** prenant place avec l'utilisateur. Chaque niveau sémantique décrit la nature et la granularité des connaissances traitées au niveau de la couche de l'architecture, dans le cadre de nos travaux de thèse. Ainsi, **la première couche** traitera des données de type **« image »**, les traitements se focaliseront alors sur l'ensemble des **« informations virtuelles »** au niveau de **la seconde couche** puis **la 3ème couche** de l'architecture manipulera des données type **« activité**

contextualisée ». Le niveau sémantique devient plus important au fur et à mesure que l'interaction se déroule de l'utilisateur au système et diminue lorsqu'elle se dirige du système à l'utilisateur. Nous ne décrirons pas, dans ce chapitre, le niveau sémantique de **la dernière couche de l'architecture**, correspondant à **la logique concepteur**, dans la mesure où cette couche ne constitue pas à proprement parler une problématique de nos travaux de thèse.
La description précise et complète de l'architecture de notre système nous permet d'assurer une modularité de cette dernière face aux changements d'interfaces avec l'utilisateur et d'applications (donc de scénarios).

Le premier niveau, le niveau le plus bas sémantiquement (niveau **« image »**), décrit **le support opérationnel**. Ce niveau se situe immédiatement après l'utilisateur (ou la scène réelle) et comprend l'ensemble des modules visant à interagir directement avec ce dernier. Ces modules concernent **les interfaces d'acquisition** (capture de l'activité au sein de la scène réelle) et **de restitution** (immersion au sein d'une scène 3D) avec l'utilisateur. **Les autres ressources** se situant à ce niveau sont celles définies par les ordinateurs supportant le système, le réseau, etc. Nous présenterons cette couche de l'architecture au cours de la première section de ce chapitre.

Le second niveau (Section 2.) est lié à **la gestion de la scène virtuelle** (niveau sémantique moyen, type **« informations virtuelles »**). La scène virtuelle regroupe **les modélisations de la scène réelle capturée et de la scène 3D**, mise en œuvre par le biais du scénario de l'application, avec laquelle l'utilisateur interagit par le biais de l'immersion. C'est au sein de cette scène virtuelle que **l'utilisateur évolue** et que **l'activité observée dans les scènes réelle et 3D est retranscrite**.
La scène virtuelle est gérée par un module dédié du système. A partir de cette scène sont extraites différentes connaissances qui permettront par la suite la caractérisation du contexte d'interaction au sein duquel l'utilisateur évolue à un instant donné.
Enfin, la scène virtuelle traduit l'état du système à un instant donné de l'interactivité avec l'utilisateur. Sa mise à jour constitue la réponse du système vis-à-vis de l'activité observée.

La troisième couche de l'architecture, d'un haut niveau sémantique (**« activité contextualisée »**), est relative à **l'analyse du dialogue interactif ayant pris place entre l'utilisateur et le système**. Elle comprend d'une part **la caractérisation et l'interprétation de l'activité** se déroulant au sein de la scène virtuelle. Ces deux processus sont supportés par les connaissances apportées par le scénario de l'application. D'autre part, cette couche comprend le module dédié à **la mise en place de la réponse visuelle**, qui découle de l'analyse de la scène observée et en accord avec l'évolution du scénario, et qui est restituée à l'utilisateur à plus bas niveau, par le biais de l'immersion. Cette couche de l'architecture sera présentée par la Section 3.

Le dernier niveau sémantique, le plus haut, englobe l'ensemble de **la logique concepteur** (logique architecte et logique développeur). C'est à ce niveau que l**es réponses interactives et adaptatives** du système vis-à-vis de la scène virtuelle, observée et interprétées, sont modélisées. Cette modélisation se fait en accord avec l'évolution du scénario de l'application, dont la description est basée **situation** et **contexte** (exposée au cours du Chapitre 2.). Ce niveau comprend également un ensemble de connaissances qui sont propagées et déclinées au sein de l'architecture en accord avec les différents niveaux sémantiques. Nous détaillerons de manière plus approfondie ce niveau de l'architecture au cours de la Section 4.

La mise en place des réponses interactives et adaptatives est supportée par **un ensemble de connaissances**, **globales et contextuelles**, **se déclinant sur tous les niveaux sémantiques du système**. Ces connaissances, introduites lors de la phase de conception du système et/ou construites au cours de l'interactivité, sont propagées au sein du système, depuis la couche logique concepteur, en accord avec l'évolution des différents scénarios (application et processus) mis en jeu au cours de l'interactivité. De globales, ces connaissances deviennent donc locales et contextuelles, en accord avec le niveau sémantique des différentes couches de l'architecture. Les connaissances sont gérées au sein de notre système par des gestionnaires dédiés, de manière globale ou locale (serveurs de contexte). La Section 5. traite de la gestion des connaissances au sein de notre système.

Enfin, **la gestion de l'adaptativité** (Section 6.) au sein du système est également mise en place à tous les niveaux de l'architecture et permet au système de répondre e**n adéquation avec ce qui est observé et interprété dans la scène virtuelle**. A partir de la modélisation de la réponse adaptative du système (supportée par le scénario de l'application) au niveau de la logique concepteur, les différentes mesures adaptatives, qui en découlent, sont propagées et déclinées à travers les différentes couches de l'architecture. Chaque couche de l'architecture comprend donc un module dédié, chargé de la mise en œuvre, au cours de l'interactivité, des mesures adaptatives en fonction du niveau sémantique de la couche. L'adaptativité, face à une situation d'interaction observée, est précédée de **la contextualisation** de cette dernière, dans le but de mieux l'expliquer et la désambigüiser, du point de vue du système. Les contextualisations, suivis de la mise en place des différentes mesures adaptatives, sont supportées par **les différentes connaissances contextuelles**, présentes au niveau sémantique de la couche considérée.

1. Support opérationnel

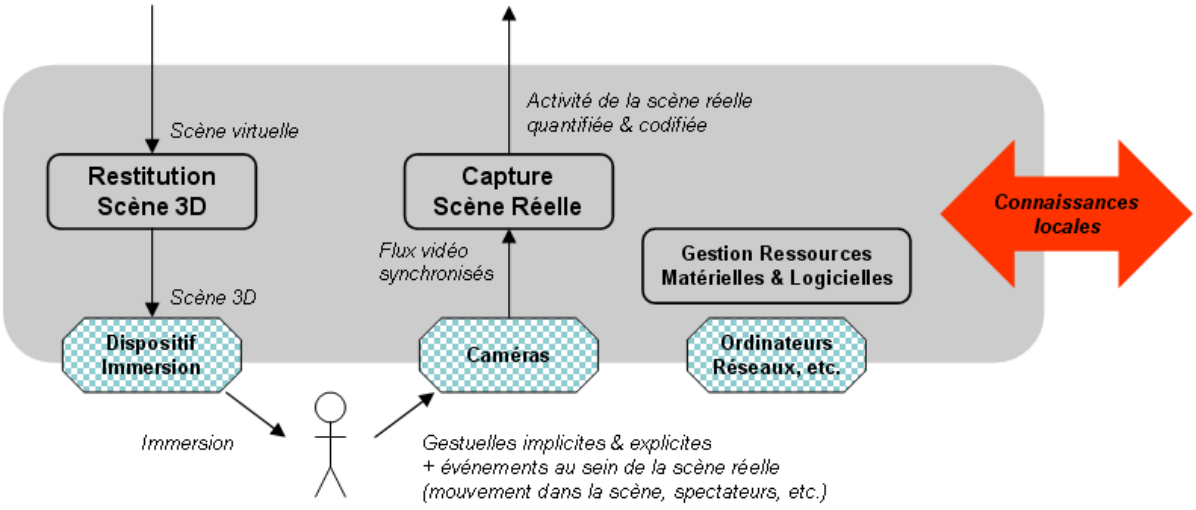


Figure 3.3. : Support opérationnel

La couche **support opérationnel** (voir Fig. 3.3.) comprend les différents modules **d'acquisition** (Section 1.1.) de l'activité (gestuelles et événements divers, dont l'utilisateur n'est pas la source) observée au sein de la scène réelle, et de **restitution** (Section 1.2.) à

l'utilisateur de la réponse visuelle du système. Enfin, elle comprend les modules responsables de **la gestion des ressources matérielles et logicielles** (Section 1.3.).

Le niveau sémantique de cette couche correspond à celui de **l'image**. En effet, les modules de cette couche de l'architecture manipulent les différentes données extraites directement, ou construites à partir, de l'image. Nous rappelons ici que nous caractérisons uniquement le niveau sémantique des données que nous utilisons pour répondre aux problématiques de la thèse.

1.1. Acquisition de l'activité au sein de la scène réelle

Un ensemble de caméras constitue **l'interface matérielle d'acquisition**, dont le but est de capturer de manière globale ce qui se passe au sein de la scène. Ces caméras sont dédiées à observer non seulement **la gestuelle explicite et implicite de l'utilisateur**, mais également **les événements non résultant d'une gestuelle volontaire de l'utilisateur**, qui pourraient survenir au sein de la scène réelle (spectateurs, dynamisme de la scène, etc.). Les différents flux vidéo synchronisés sont alors traités par un système commercial de capture de gestuelles, le ***Cyberdôme*** de la société XD Productions, augmenté de nos contributions, que nous présenterons en détail dans le prochain chapitre. Cette interface caractérise l'ensemble des événements se déroulant au sein de la scène réelle, tout particulièrement les gestuelles de l'utilisateur. Ces gestuelles sont alors reproductibles en temps réel, permettant ainsi l'animation d'un avatar, personnage virtuel évoluant au sein d'un univers 3D.

1.2. Restitution de la réponse visuelle par le biais de l'immersion

La couche support opérationnel comprend également **un dispositif d'immersion** de l'utilisateur dans une **scène 3D**. Ce dispositif, supporté par un vidéo projecteur, se charge de restituer **la réponse**, **visuelle et interactive**, du système à l'utilisateur. Grâce à ce dispositif, l'utilisateur peut interpréter la réponse du système et adopter la gestuelle adaptée à cette dernière.

1.3. Gestion des ressources matérielles et logicielles

La couche « support opérationnel » comprend également **l'ensemble des dispositifs et ressources** supportant et assurant le fonctionnement du système au cours de l'interactivité : ordinateurs, réseau, etc. Les différentes ressources matérielles et logicielles sont gérées à ce niveau.

2. Scène virtuelle

La couche **« scène virtuelle »** (voir Fig. 3.4.) de l'architecture est dédiée à **la modélisation**, **la gestion et l'observation**, **temps réel**, **d'une scène virtuelle**. Le niveau sémantique de cette couche correspond donc à celui de **l'information virtuelle**.

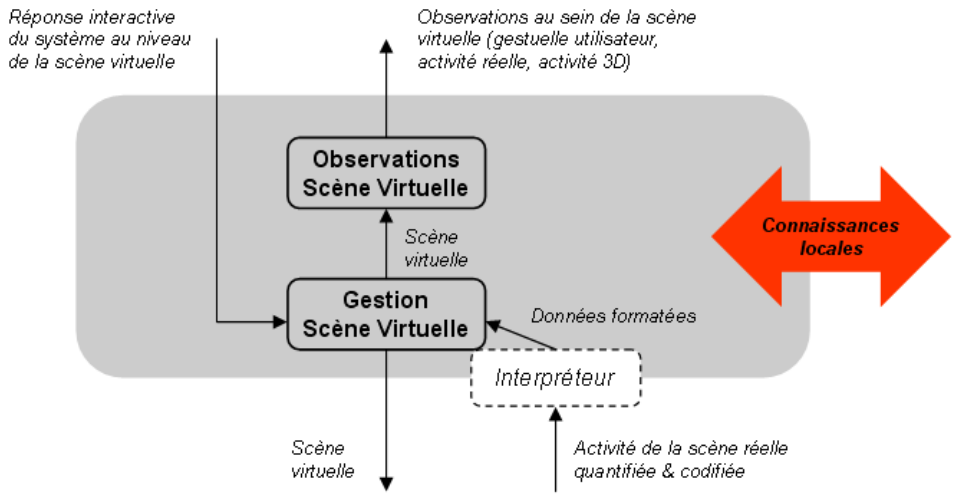


Figure 3.4. : Scène virtuelle

La scène virtuelle a une double nature. Elle est fonction à la fois de la **scène réelle**, capturée et modélisée au niveau de la couche « support opérationnel », et d'une **scène 3D**, modélisée par le scénario de l'application et avec laquelle l'utilisateur interagit par le biais d'un avatar. L'activité au sein de la scène virtuelle retranscrit donc l'activité au sein de la scène réelle et de la scène 3D. Les Sections 2.1., 2.2. et 2.3. définissent respectivement ce que nous appelons la scène réelle, la scène 3D et la scène virtuelle.

Après l'avoir modélisée, un processus dédié s'occupe de **gérer cette scène virtuelle**. De celle-ci sont extraites **différentes connaissances**, permettant par la suite la caractérisation du contexte d'interaction au sein duquel l'utilisateur évolue. Cette extraction est la conséquence de **l'observation de la scène virtuelle** par un module dédié. Nous présentons ces différents modules au cours de la Section 2.4.

2.1. La scène réelle

La scène réelle est la scène véritable, au sein de laquelle l'utilisateur évolue physiquement. Le dynamisme observable au sein de cette scène réelle peut également être indépendant de l'utilisateur, pouvant ainsi être généré, par exemple, par un changement d'illumination ou un éventuel spectateur.

L'activité au sein de la scène réelle peut donc être générée par les gestuelles qu'adoptent l'utilisateur et par les changements d'états de cette scène, dont l'utilisateur n'est pas la source, générés par des spectateurs et/ou par une évolution physique de la scène réelle (changement d'illumination, de la configuration spatiale, etc.). L'activité est capturée par le ***Cyberdôme***, augmenté de nos contributions, que nous présentons au cours du Chapitre 4.

2.2. La scène 3D

La scène 3D, ou environnement/univers 3D, décrit l'image visuelle de l'état du système, plus particulièrement de celui de la scène virtuelle.

Cette image visuelle est restituée à l'utilisateur, par le biais du dispositif d'immersion. En accord avec sa compréhension de la scène 3D, l'utilisateur interagit alors avec celle-ci, en adoptant des gestuelles adaptées, dans le but d'atteindre ses objectifs.
L'utilisateur est représenté au sein de cette scène 3D par un avatar, animé par ses gestuelles capturées. Cet avatar peut interagir avec les éléments 3D interactifs, composant la scène 3D.

La scène 3D, et son évolution, sont élaborées par les concepteurs lors de l'écriture du scénario de l'application. Elle est mise à jour par l'évolution du scénario de l'application et des réponses interactives du système qui en découlent, et par l'activité capturée au sein de la scène réelle. Avant sa restitution à l'utilisateur, la scène 3D est extraite de la scène virtuelle par le processus dédié compris dans la couche « support opérationnel » du système.

2.3. La scène virtuelle

La scène virtuelle rassemble les modélisations de la scène réelle capturée et de la scène 3D. D'une part, la scène virtuelle reflète l'ensemble de l'activité et des événements en résultant, au sein des scènes réelle et 3D. D'autre part, elle permet la traduction de l'interprétation du système de cette activité, car modélisant la réponse de ce dernier à celle-ci. Ainsi, cette scène virtuelle matérialise **l'état du système à un instant donné** de l'interactivité avec l'utilisateur.
Les observations effectuées par le système au niveau de la scène virtuelle sont exploitées à plus haut niveau par les modules d'analyse du contexte d'interaction au sein duquel l'utilisateur évolue. C'est pourquoi le niveau sémantique de la couche, dédiée à la gestion et à l'observation de la scène virtuelle, correspond à celui des **« informations virtuelles »**, qualifiant ainsi les données que nous gérons et utilisons au sein de cette couche et des couches de niveau sémantique supérieur.

La scène virtuelle retranscrit l'activité capturée au sein des scènes réelle et 3D. Par 'activité', nous entendons :

- **l'activité au sein de la scène réelle**. Cette activité comprend tout d'abord celle de **l'utilisateur**, concrétisée par les gestuelles que celui-ci adopte, en accord avec les objectifs qu'il s'est définis vis-à-vis du scénario de l'application. Ces gestuelles peuvent être explicites ou peuvent être la source d'une interaction non volontaire de la part de l'utilisateur (gestuelles implicites). Elles permettent l'animation de l'avatar, représentant l'utilisateur au sein de la scène 3D. Deuxièmement, cette activité peut être indépendante de l'utilisateur et générée **par d'autres éléments variables et mobiles** au sein de la scène réelle.
- **L'activité au sein de la scène 3D**. Cette activité est générée par l'avatar représentant l'utilisateur au sein de cette scène, animé par ses gestuelles réelles. Cet avatar interagit avec la scène 3D et les éléments 3D interactifs la composant. Les événements résultant de cette activité sont gérés au niveau de la gestion de la scène virtuelle.

La scène virtuelle est modélisée et mise à jour à partir de l'activité capturée au sein des scènes réelle et 3D, et des réponses scénarisées du système vis-à-vis de l'activité observée et interprétée. Les modifications au sein de la scène virtuelle peuvent se traduire soit explicitement, soir de manière invisible, vis-à-vis de sa restitution visuelle à l'utilisateur.

2.4. Modules de la couche « Scène virtuelle »

La couche **« scène virtuelle »** comprend tout d'abord un module chargé de **la modélisation et de la gestion de la scène virtuelle**. La scène virtuelle est donc considérée comme un important système d'informations, mis à jour en permanence en fonction de l'activité capturée et des réponses du système à celle-ci. La mise à jour de la scène virtuelle implique la mise à jour de la scène 3D, tandis que la mise à jour des scènes réelle et 3D implique la mise à jour de la scène virtuelle. Quoiqu'il en soit, la mise à jour de la scène virtuelle est scénarisée et établie lors de la phase de conception du système.

La couche de l'architecture comprend également un module chargé **d'observer et de collecter les différents événements survenant au sein de la scène virtuelle**. Ces événements permettent la construction de nouvelles connaissances, qui seront utilisées par la suite, à un niveau sémantique supérieur, pour caractériser le contexte d'interaction observé, au sein duquel l'utilisateur évolue.

Enfin**, un interpréteur de données** est nécessaire entre le gestionnaire de la scène virtuelle et le système de capture de la scène réelle, ou, autrement dit, **entre les couches « scène virtuelle » et « support opérationnel »**. Ce module supplémentaire permet le formatage des données entre les deux modules, ou les deux couches, assurant ainsi la mise en correspondance et la cohérence entre les informations réelles capturées en accord avec la scène virtuelle. De plus, il permet d'assurer une indépendance relative du système vis-à-vis du système de capture, assurant ainsi la modularité du système à ce niveau.

3. Interprétation du dialogue interactif

C'est au sein de cette couche que le dialogue interactif entre l'utilisateur et le système est analysé et mis en œuvre (voir Fig. 3.5.). Un module dédié se charge tout d'abord **de caractériser le contexte d'interaction au sein duquel l'activité prend place** (Section 3.1.). Un second module **interprète** par la suite **l'activité observée au sein de la scène virtuelle** (Section 3.2.). Cette analyse est alors soumise à la logique concepteur qui, sous le support du scénario de l'application, établit **la réponse du système**. A partir de cette réponse décrite à un haut niveau sémantique, un troisième module se charge de **mettre en place la réponse interactive du système**, au niveau de la scène virtuelle (Section 3.3.).
L'ensemble des raisonnements s'effectuent donc autour des données de type **« activité contextualisée »**, qualifiant ainsi le niveau sémantique de la couche de l'architecture.

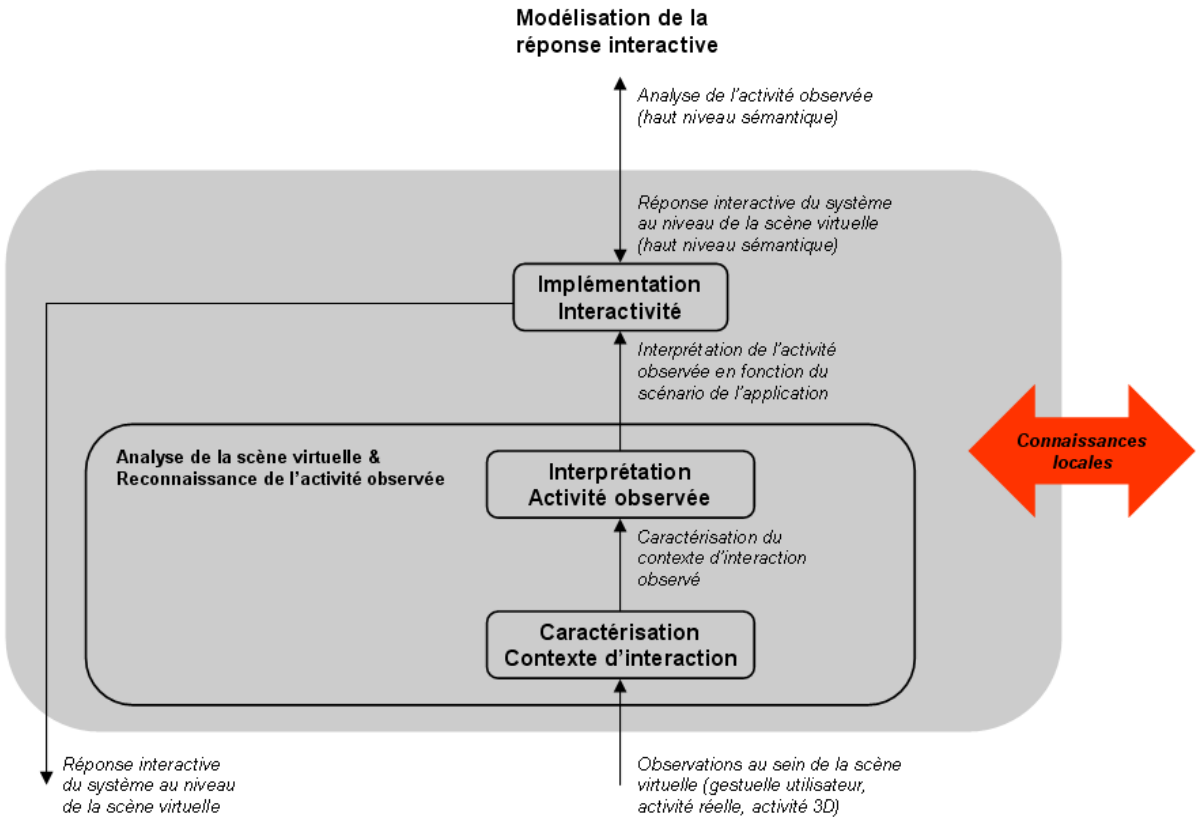


Figure 3.5. : Interprétation du dialogue interactif

3.1. Caractérisation du contexte d'interaction

Le module de caractérisation se charge de traiter et de formater les connaissances extraites de la scène virtuelle, à partir de la couche « scène virtuelle ». Ces connaissances sont donc relatives aux gestuelles adoptées par l'utilisateur et capturées par notre système de capture, aux événements indépendants de l'utilisateur et ayant lieu au sein de la scène réelle (spectateurs, changement d'illumination, etc.), et aux événements ayant lieu au sein de la scène 3D (déclenchements de réactions du système par le biais d'interactions 3D, pose du modèle 3D représentant l'utilisateur au sein de la scène 3D, etc.).

Ces connaissances collectées permettent de caractériser **le contexte d'interaction** au sein duquel l'activité est observée, et dans lequel évolue l'utilisateur. Au cours de cette caractérisation, les connaissances de plus bas niveau peuvent être fusionnées et permettre la construction de plus haut niveau.

Le processus de caractérisation du contexte d'interaction observé est supporté par **le scénario de l'application**. En effet, en fonction du scénario, le module de caractérisation sait quoi caractériser, quand et comment. En plus de rendre le processus de caractérisation plus efficace, les connaissances extraites du scénario permettent une optimisation de celui-ci.

3.2. Interprétation de l'activité observée

Le contexte d'interaction caractérisé, par le biais du module dédié présenté précédemment, traduit **l'activité observé au sein de la scène virtuelle**. Le système comprend donc un module chargé **d'interpréter ce contexte d'interaction**, dans le but **de reconnaître l'activité observée au sein de la scène virtuelle**.

A l'instar du processus de caractérisation, le processus d'interprétation est supporté par les connaissances apportées par **le scénario de l'application**. Par le biais de ce dernier, il est effectivement possible **de confronter les contextes d'interaction observés**, **avec ceux attendus par le scénario** de l'application au cours de l'interactivité.
Le résultat de cette interprétation est alors transmis, via le module responsable de la mise en place de l'interactivité, à **la logique concepteur**, couche de l'architecture de plus haut niveau sémantique. Ce résultat est alors étudié, à ce niveau, vis-à-vis du scénario de l'application et des objectifs attendus par celui-ci. Cette étude entraine alors **l'évolution du scénario** et donc **la mise en place des réponses du système**.

3.3. Mise en place de la réponse interactive

Le dernier module de la couche « interprétation du dialogue interactif » est dédié à **la mise en place de la réponse interactive du système**.

Tout d'abord, ce module se charge de soumettre, à la couche « logique concepteur », **l'analyse de l'activité observée au sein de la scène virtuelle**. Au sein de la couche « logique concepteur », un ensemble de modules dédiés confrontent alors cette analyse vis-à-vis du scénario de l'application.
La réponse du système, **à tous les niveaux de l'architecture**, est alors établie en fonction de **l'évolution du scénario**. Cette réponse est décrite à un haut niveau sémantique. Elle se concrétise notamment au niveau de **la scène virtuelle** (sa description, son évolution au cours de l'interaction, etc.) et est **adaptée à l'activité observée** au sein de cette dernière. En effet, ce sont les modifications au niveau de **la scène 3D**, extraite à partir de la scène virtuelle, que l'utilisateur observe et à partir desquelles il comprend la réponse du système vis-à-vis de sa gestuelle. Cette réponse le guidera alors **pour adopter sa prochaine gestuelle**.
La réponse du système, établie au niveau de la logique concepteur, est enfin transmise à la couche « Interprétation du dialogue interactif », via **le module de mise en œuvre de l'interaction**. Etant décrite à un haut niveau sémantique, le module se charge de traduire cette réponse, de manière à ce qu'elle soit applicable au niveau de la scène virtuelle. Par la suite, cette réponse sera restituée à l'utilisateur à plus bas niveau, par l'intermédiaire de l'immersion de ce dernier au sein de la scène 3D.

4. Interface avec la logique concepteur

La conception d'un système, tel que celui que nous proposons dans ces travaux, implique la collaboration de plusieurs types de concepteurs (voir Annexe A). Nous avons distingué principalement **l'architecte**, responsable de l'élaboration du **système**, et le **développeur**, responsable de **l'application** à proprement parler. La **logique concepteur**, incluant la **logique architecte** et **la logique développeur**, est la couche **de plus haut niveau sémantique** au sein du système au sein duquel nous intégrons nos travaux (voir Fig. 3.6.).

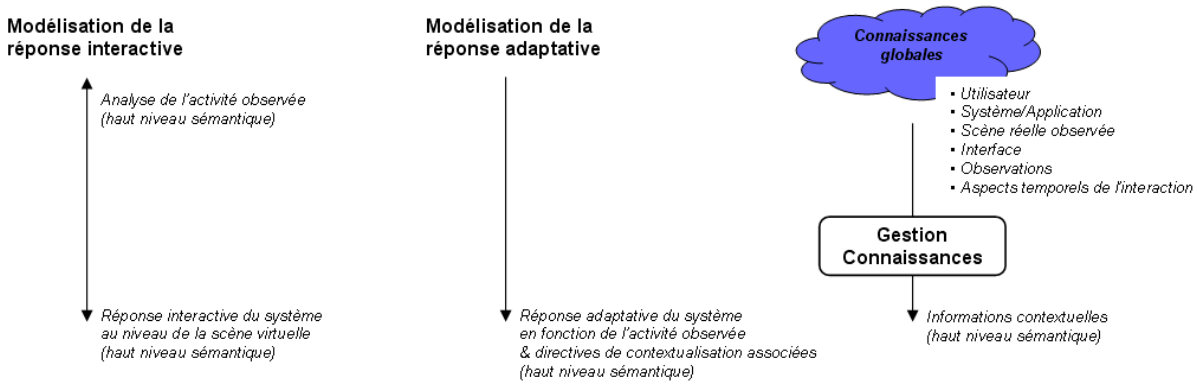


Figure 3.6. : Logique concepteur

La section suivante expose **la mise en place des réponses interactives et adaptatives** au niveau de la logique concepteur, vis-à-vis de **l'observation**, **la caractérisation** et **l'interprétation de l'activité au sein de la scène virtuelle**. La Section 4.2. traite de l'ensemble de **connaissances**, décrites à un haut niveau sémantique et comprises au sein de la couche « logique concepteur », qui sont propagées à tous les niveaux de l'architecture de notre système.

4.1. Mise en place des réponses interactives et adaptatives

Au sein de notre système, l'analyse de l'activité observée, au sein de la scène virtuelle, est transmise à la logique concepteur par le biais du module responsable de la mise en place du dialogue interactif. Ainsi, au niveau de la logique concepteur, la réponse du système, fonction de cette **analyse** et définie par le **scénario** de l'application, est établie. Cette réponse est double, à la fois **interactive** et **adaptative**.

La réponse du système est tout d'abord **interactive** et se traduit par une mise à jour de la scène virtuelle. Cette mise à jour se répercutera sur la scène 3D, au sein de laquelle l'utilisateur est immergé.
En plus d'être une mise à jour visuelle de la scène 3D, la réponse visuelle, restituée à l'utilisateur, peut également être une sollicitation directe (impérative) de ce dernier, par le système. Le système, de par son initiative, interpelle volontairement l'utilisateur et lui suggère, voire lui impose, une gestuelle particulière, de manière à faire évoluer le scénario de l'application, dans une direction donnée particulière. Quel que soit la nature de la réponse interactive du système, l'utilisateur adoptera la gestuelle qu'il jugera la plus adaptée à la réponse visuelle du système.

D'autre part, la réponse du système traduit **l'adaptativité** du système, se concrétisant par un ensemble de mesures adaptatives, dont l'objectif est d'adapter son propre fonctionnement au contexte d'interaction observée, et donc à l'activité au sein de la scène, particulièrement celle relatives aux gestuelles adoptées par l'utilisateur.
Ces mesures adaptatives sont transmises aux différents modules de l'architecture et déclinées en accord avec le niveau sémantique de la couche cible. A noter qu'une mesure adaptative peut concerner directement la réponse interactive du système, au niveau de la scène virtuelle.

La réponse adaptative du système est accompagnée de la propagation d'**informations contextuelles**, extraites des connaissances globales présentes au sein de la logique concepteur. Ces informations sont alors d'une part **distribuées** aux modules de l'architecture (voir Section 5.) et d'autre part associées aux processus de **contextualisation**, qui paramètrent la gestion de l'adaptativité au sein du système (voir Section 6.). Elles sont également propagées au sein de l'architecture, déclinées en fonction du niveau sémantique des différentes couches de cette dernière.

4.2. Propagation des connaissances haut niveau au sein de l'architecture

En plus des modules dédiés à la modélisation des réponses du système, la logique concepteur comprend l'ensemble des **connaissances globales,** dont le système dispose au cours du dialogue interactif. Ces connaissances sont décrites à de très haut niveaux sémantiques et, en fonction du module de traitement auquel elles sont dédiées, sont déclinées et caractérisées en fonction du niveau sémantique cible.
Les connaissances globales, conformément à nos propos de la Section 3.1. du Chapitre 2., sont relatives à l'utilisateur, au système et à l'application, à la scène réelle observée, à l'interface, aux observations et aux aspects temporels de l'interactivité. Nous aborderons plus en détail la problématique de la gestion de ces connaissances au cours de la section suivante.

Parmi les connaissances globales se trouve le **scénario** de l'application à proprement parler, et les différentes connaissances qu'il rassemble. C'est par le biais de ce scénario, et de son évolution, que la réponse à l'utilisateur est établie.
A l'instar de l'application, chaque module effectue ses traitements en suivant un scénario bien particulier, défini au moment de la phase de développement. Au cours de l'interactivité, le scénario de l'application évolue, définissant **la réponse du système**, traduite par les différents modules de l'architecture. Ainsi le déroulement du scénario de l'application devient le point de départ du déroulement des différents autres scénarios.
Nous modélisons le scénario de l'application selon notre approche basée **situation** et **contexte**, présentée au cours de la Section 6.4. du Chapitre 2..

5. Gestion des connaissances

Un ensemble de **connaissances** supportent le fonctionnement du système, particulièrement **la mise en place des réponses interactives et adaptatives, vis-à-vis de l'activité observée** au sein de la scène virtuelle (voir Fig. 3.7.).

La nature des différentes connaissances, comprises au sein de notre système, est très diverse d'une connaissance à une autre. La Section 5.1. traite des différences que peuvent présenter, entre elles, les différentes connaissances que nous gérons au sein de notre système.
Des modules sont nécessaires, dédiés à la gestion de ces connaissances, de manière globale ou au niveau de toutes les couches de l'architecture. La Section 5.2. détaille la gestion des connaissances au sein de notre système.

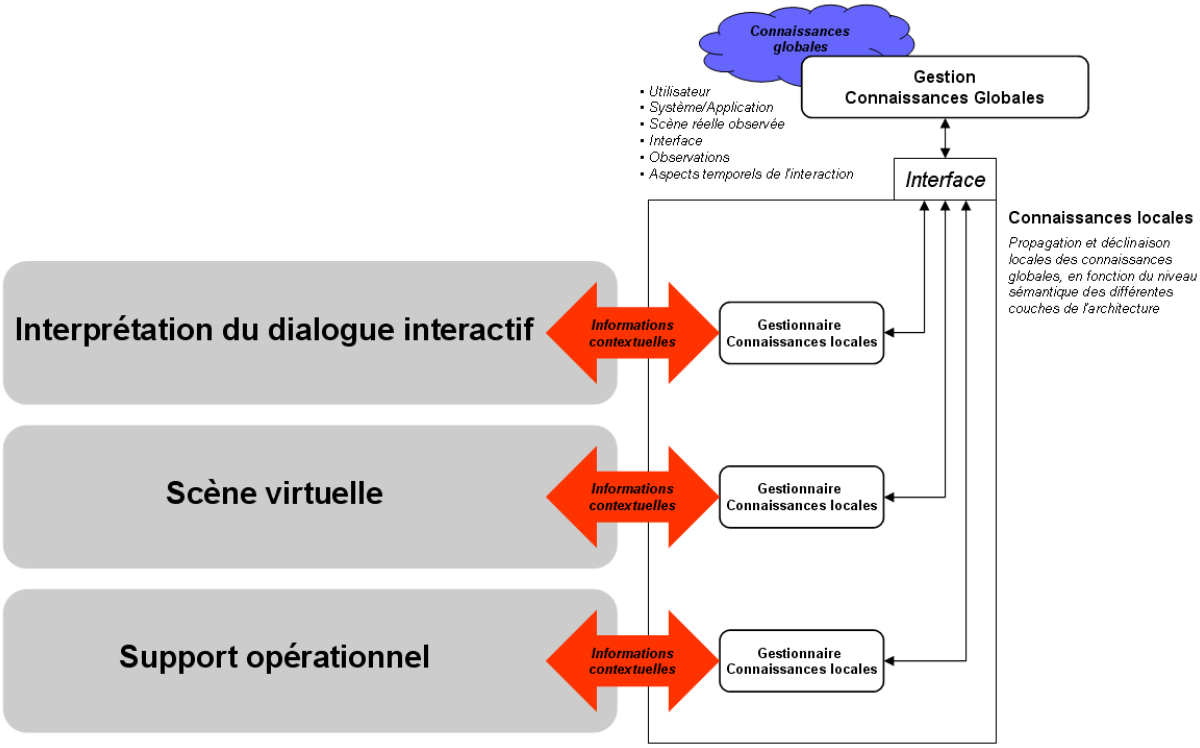


Figure 3.7. : Gestion des connaissances

5.1. Diversité des connaissances

Notre système comprend un ensemble important de connaissances **hétérogènes** et **différentes d'un niveau sémantique à un autre** de l'architecture, d'un module de traitements à un autre, etc. Elles sont identifiées au moment de la conception du système et sont décrites initialement à un très haut niveau sémantique, correspondant à celui de **la logique concepteur** (voir la Section 4.).

De plus, elles peuvent être **statiques** ou **dynamiques**. Les connaissances statiques sont invariables au cours de l'interactivité. En revanche, les connaissances dynamiques se construisent et évoluent au cours du dialogue interactif.

Enfin, une connaissance peut être **globale** ou **locale**. Nous parlerons d'une **connaissance contextuelle** pour mentionner une connaissance locale.
Il est tout à fait envisageable qu'une connaissance soit attendue et utilisable par l'ensemble des modules du système. Cette connaissance est alors **globale**. Typiquement, il s'agira des mesures temporelles, dédiées à la temporisation et à la synchronisation des différents processus mis en œuvre au cours de l'interactivité.
Une **connaissance contextuelle** n'est n'utilisable qu'au niveau d'une couche particulière de l'architecture du système, voire que par un module donné de traitements. Cette connaissance traduit donc un certain niveau sémantique et n'est interprétable que par un ou plusieurs modules dédiés, correspondant à ce niveau sémantique cible.

5.2. Gestion au sein de notre système

Les connaissances du système sont décrites **initialement au niveau de la logique concepteur**. Elles sont donc définies à un haut niveau sémantique. Ainsi établie au niveau de la logique concepteur, une connaissance donnée ne peut pas être utilisée telle quelle par le ou les modules cibles.
C'est pourquoi une connaissance décrite au niveau de la logique concepteur doit être **propagée** et **déclinée** au sein de l'architecture, par le biais de **gestionnaires de connaissances dédiés**.

Lors du déroulement du scénario de l'application, un **gestionnaire** dédié, présent au niveau de la logique concepteur, se charge de **propager** et de **décliner les différentes connaissances aux modules du système**, **en fonction du niveau sémantique** dans lequel se situent ces derniers. Il opère de concert avec un ensemble de **gestionnaires dédiés aux différents niveaux sémantiques** de l'architecture du système, gérant ainsi les connaissances **au niveau local**. Une interface entre le gestionnaire de connaissances globales et les gestionnaires de connaissances contextuelles permet le dialogue et la redistribution des différentes informations.
L'utilisation (sélection des connaissances pertinentes, filtrage, etc.) et la modification (construction, évolution) des différentes connaissances sont décidées et pilotées par les différents **scénarios** (application et modules) mis en jeu au cours de l'interactivité.

Au niveau d'une couche de l'architecture, un gestionnaire manipule un ensemble de **connaissances contextuelles**. A ce titre, ce gestionnaire peut être considéré comme un **serveur de contexte**. Les informations contextuelles utilisées par les différents modules de traitement du système, quel que soit le niveau sémantique, sont à associer aux **contextes** décrits au cours de la Section 6.5. du Chapitre 2. Ces différents contextes **(« Utilisateur »**, **« Système/Application »**, **« Scène réelle »**, **« Interface »**, **« Observations »** et **« Temps »**) sont rappelés par la Figure 3.8., dans le cadre de nos travaux.

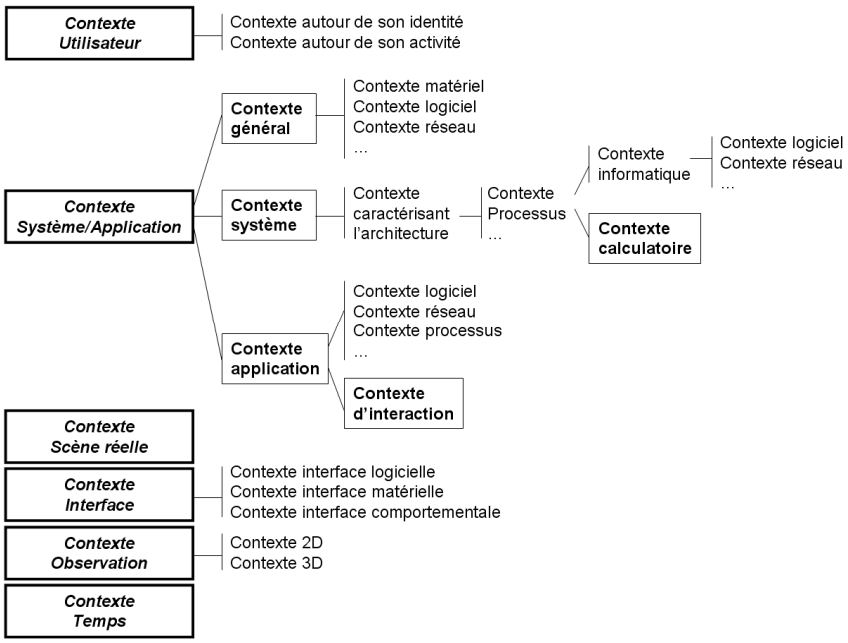


Figure 3.8. : Informations contextuelles comprises au sein de notre système (Chapitre 2., Section 6.5.)

6. Gestion de l'adaptativité

L'objectif de notre système est de **s'adapter à la situation d'interaction observée**, via la scène virtuelle modélisée. Une fois la situation d'interaction interprétée, le système met en place sa réponse **adaptative**, qui, en plus de modifier la scène virtuelle en adéquation avec l'activité observée, oriente et optimise l'ensemble de son fonctionnement global.

Dans nos travaux, l'adaptativité se traduit de **deux façons**. Tout d'abord, elle prend la forme d'**une réponse visuelle** à proprement parler, restituée à l'utilisateur. Ainsi, la réponse interactive du système se verra modifiée, de manière à être adaptée à la gestuelle adoptée par l'utilisateur et au contexte d'interaction dans lequel il évolue. D'autre part, la réponse adaptative se concrétise par l'**adaptation** que le système doit opérer **vis-à-vis de son fonctionnement interne propre**, **et ce à tous les niveaux de son architecture**.

La réponse adaptative du système se concrétise par la mise en œuvre de mécanismes adaptatifs scénarisés, à tous les niveaux sémantiques de l'architecture, et pour tous les modules de traitements du système. Nous détaillons, au cours de la première section, **l'interface et les modules chargés de mettre en place l'adaptativité** du système, en fonction de l'activité observée au sein de la scène virtuelle.
Ces mesures adaptatives sont corrélées à des **processus de contextualisation**, dans le but d'expliquer et de désambigüiser les différentes situations observées, du point de vue du système. Ces processus sont supportés par les différentes connaissances contextuelles propagées à partir de la logique concepteur. La Section 6.2. traite de ces contextualisations.
La Figure 3.9. illustre la mise en place de ces mécanismes adaptatifs et de ces processus de contextualisation au sein de notre système.

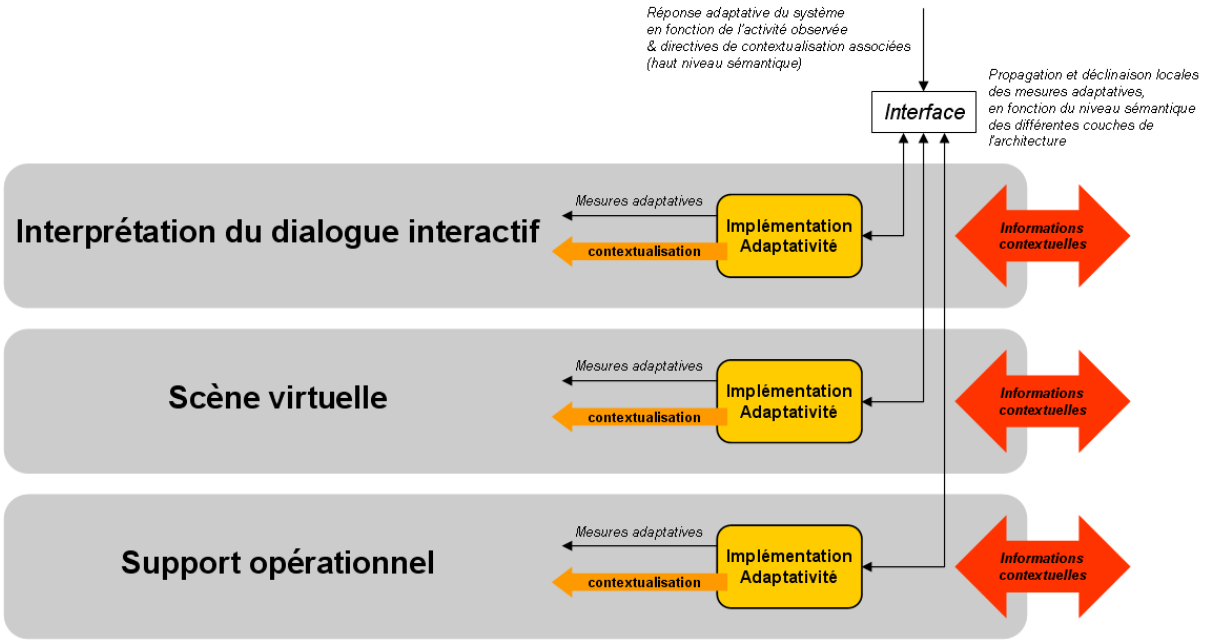


Figure 3.9. : Gestion de l'adaptativité

6.1. Gestion au sein de notre système

Une situation d'interaction prend place au sein de la scène réelle et résulte en une activité modélisée au niveau de la scène virtuelle. La situation d'interaction se décline donc en différentes situations, se déroulant au niveau des différentes couches de l'architecture et des différents modules composant ces couches.

Une fois l'activité observée et interprétée, la réponse adaptative du système est établie et mise en place. C'est au plus haut niveau de l'architecture, au niveau de **la logique concepteur**, que cette réponse adaptée (comprenant les directives associées de contextualisation) est définie. Cette réponse est fonction de l'**analyse de la scène virtuelle** et du **scénario de l'application**.

Parce que la situation d'interaction observée se décline à travers l'architecture du système (couches sémantiques et modules de traitements), l'adaptation du système doit s'effectuer et se propager à tous les niveaux sémantiques de l'architecture et pour l'ensemble des modules de traitements. La réponse adaptative est donc fonction **du niveau sémantique** de la couche cible et de la nature des traitements exécutés par un module donné. Ainsi, le comportement du système est adapté **aux différentes situations observées** au sein de la couche sémantique considérée, et pour un module de traitement donné.

La propagation des mesures adaptatives au sein du système se fait par l'intermédiaire de **modules dédiés**, communiquant avec une **interface simple**, se situant au niveau de la logique concepteur. Ces modules permettent d'assurer une cohérence des différentes mesures adaptatives avec le niveau sémantique de la couche cible. De plus, ils permettent de faire face à l'hétérogénéité des données, assurant ainsi de meilleures performances.
Au niveau de la couche dédiée à l'**interprétation du dialogue interactif**, le module de mise en place de l'adaptativité transmet, aux modules d'analyse de la scène virtuelle, les différentes informations contextuelles qui leur permettront d'identifier l'activité observée et caractérisée : contextes d'interaction attendus ; seuils sur les distances existantes entre les contextes d'interaction observés et attendus ; caractérisations focalisées fonctions du scénario ; orientation des fusions d'informations et des constructions d'informations de plus haut niveau... De plus, ce module transmet à celui chargé de mettre en place la réponse interactive les modifications que ce dernier doit apporter à celle-ci, de manière à la rendre adaptée à la gestuelle adoptée par l'utilisateur et au contexte d'interaction au sein duquel il évolue.
Au niveau de la couche **scène virtuelle**, le module de mise en place de l'adaptativité se charge, d'une part, de mieux gérer la scène virtuelle (gestion intelligente des différentes ressources interactives ; mise en évidence visuelle d'éléments pouvant influer la gestuelle adoptée par l'utilisateur ; affichage pertinent et intelligent vis-à-vis de l'utilisateur…) et, d'autre part, de guider les différents observateurs au sein de celle-ci (gestion intelligente des observateurs, orientation des observateurs en fonction du scénario…).
Au niveau de la couche de plus bas niveau (**support opérationnel**), le module responsable de la mise en place de l'adaptativité permet d'améliorer et de guider la capture de la scène réelle (gestion intelligente de l'acquisition vidéo ; orientation et optimisation de la capture de mouvements ; traitements intelligents et orientés des événements ayant lieu au sein de la scène réelle), optimisant et anticipant également la gestion des différentes ressources logicielles et matérielles disponibles via les machines utilisées, le réseau, etc.
Enfin, comme nous l'expliquerons au cours de la section suivante, les communications entre les modules chargés de mettre en œuvre l'adaptativité du système et les différents gestionnaires de connaissances sont à **double sens**. Les informations contextuelles permettent

de mettre en place les différents processus de contextualisation, paramétrant ainsi les mécanismes adaptatifs. Mais ces mécanismes adaptatifs peuvent opérer directement sur ces informations contextuelles, de manière à rendre plus efficace la contextualisation des différentes situations observées.

6.2. Processus de contextualisation

Le module, chargé de mettre en place et de décliner la réponse adaptative du système, en accord avec le niveau sémantique d'une couche de l'architecture et des modules la composant, s'occupent également d'exécuter **un processus de contextualisation**. Ce processus a pour objectif de mieux **expliquer** et **désambigüiser les différentes situations observées**, au niveau d'une couche de l'architecture et/ou d'un module de traitements. Il prend en compte et manipule **les informations contextuelles** mises à disposition au niveau de la couche sémantique considérée.

Au sein de notre système sont donc mis en œuvre un ensemble de processus de contextualisation, dont l'objectif est de paramétrer l'adaptativité du système. Les éléments pertinents d'une situation donnée sont mis en évidence par le processus de contextualisation, permettant ainsi une meilleure explication et désambigüisation de cette dernière et donc une adaptation adéquate du système vis-à-vis de celle-ci.
D'un autre côté, l'adaptativité du système, face à une situation donnée, peut prendre la forme d'une adaptation du processus de contextualisation (ajustement des informations contextuelles et de leur utilisation), anticipant alors les situations à venir.
Autrement dit, le processus de contextualisation permet de mieux expliquer (dans le sens observer et analyser) une situation, pour mieux s'adapter à cette dernière. Mais l'explication de la situation permet de mieux adapter le processus de contextualisation, pour les situations à venir. Cette interdépendance entre la contextualisation et l'adaptativité illustre la pertinence de notre décision quant à la prise en charge du processus de contextualisation par le module chargé de mettre en œuvre les mécanismes adaptatifs.

7. Conclusion : Contributions

En accord avec nos propos précédents, il nous est possible de situer nos **contributions** au niveau de l'architecture développée.

1. Support opérationnel

Une grande partie des contributions de ce travail de thèse se situe au niveau du système de capture de la scène réelle (couche « support opérationnel »).
Tout d'abord, le système commercial de capture de gestuelles, adopté dans ces travaux, comprend plusieurs contraintes que nous avons cherché à alléger au maximum : total contrôle de l'environnement de capture (espace fermé et uni, éclairage puissant empêchant toute immersion, etc.) et processus nécessitant le port de marqueurs de la part de l'utilisateur. Notre premier objectif fut donc de trouver des solutions pour palier ces problèmes tout en améliorant le processus de capture.
D'autre part, le système choisi est initialement conçu pour uniquement capturer les mouvements de l'utilisateur, dans le but d'animer un personnage virtuel. Il nous a donc fallu

étendre le système de manière à capturer non pas juste les mouvements de l'utilisateur mais l'ensemble des indices, implicites et explicites, nécessaire à la caractérisation dans le temps de ses gestuelles (mouvements, gestes, actions et comportements, voir Chapitre 4.). Enfin, nos contributions ont permis également à notre système d'être plus sensibles à différents éléments pouvant prendre place au sein de la scène réelle et dont l'utilisateur n'est pas responsable. Ces événements, changements d'éclairage, présence de spectateurs, etc. génèrent un dynamisme de scène que le système doit être capable de traiter.
Le prochain chapitre traite exhaustivement du thème de la capture de mouvements et du système utilisé dans le cadre de ces travaux de thèse.

2. Gestion de la scène virtuelle

La gestion de la scène virtuelle, ainsi que la mise en place des observateurs au sein de celle-ci, font également partie de nos contributions. L'ensemble des développements concernant ces deux modules a été effectué au cours de ces travaux de thèse. Pour illustrer ces travaux, un jeu basique, au scénario simple, a été développé. Sa conception de A à Z nous a permis de mettre en place les différents mécanismes logiciels permettant la gestion de la scène virtuelle et des observateurs (se référer au Chapitre 5.).
Concernant les observations, nous avons cherché à extraire les différentes caractéristiques permettant la caractérisation du contexte d'interaction englobant l'activité observée, vis-à-vis du scénario de l'application : codification quantitative et qualitative des gestuelles utilisateur, relations spatiotemporelles entre les gestuelles utilisateur et le contenu de la scène 3D, événements interactifs déclenchés par les gestuelles utilisateur, événements déclenchés par des modifications de l'état de la scène réelle, dont l'utilisateur n'est pas responsable. Certains indices, observés au sein de la scène réelle, ne sont pas modélisés visuellement au sein de la scène 3D (comme la présence de spectateurs par exemple). Cependant, ils constituent également des informations importantes, nécessaires à la caractérisation du contexte d'interaction auquel le système doit s'adapter.
Le système de capture, adopté dans ces travaux, utilise sa propre scène virtuelle, au sein de laquelle l'utilisateur, par ses mouvements, anime une représentation virtuelle. Il nous a donc fallu développer un interpréteur de données, assurant ainsi une correspondance efficace entre la scène virtuelle mise en œuvre par le système de capture et celle conçu à partir du scénario de l'application. Une fois élaborée, cette correspondance reste la même, quel que soit le scénario de l'application. En revanche, elle devra être redéterminée si un nouveau système de capture est utilisé. Ainsi, grâce à cet interpréteur, la modularité du système global est assurée vis-à-vis du système de capture.

3. Interprétation du dialogue interactif

Nos contributions, concernant la couche relative à l'interprétation du dialogue interactif entre l'utilisateur et le système, sont exprimées dans le cadre de l'application qui a été développée.
Encore une fois, le scénario de cette application est simple. Ce qui implique par conséquent une simplicité et limitation relatives des gestuelles adoptées par l'utilisateur, des contextes d'interaction au sein duquel ce dernier évolue et des réponses interactives du système vis-à-vis de ces gestuelles. De plus, nous sommes partis du scénario de l'application pour déterminer les différents indices, extraits des scènes réelle et 3D, caractérisant la gestuelle utilisateur et le contexte d'interaction, avant de définir les mécanismes pour les interpréter. Notre méthodologie est donc très dépendante du scénario de l'application et un seul scénario

ne permet certainement pas d'avoir une approche générale et globale quant à la définition des gestuelles, contextes et mécanismes d'interprétation. Cependant, nous pensons que notre approche constitue un bon point de départ pour de futurs travaux.
Dans les prochains chapitres, nous définissons notre *framework* de caractérisation de la gestuelle utilisateur et du contexte d'interaction, définissant ainsi les différents indices, extraits des scènes réelle et 3D le permettant. Cependant, pour illustrer ces travaux de thèse et à l'image du scénario de l'application, nous nous limitons à des indices simples. D'autres indices, plus complexes et haut niveau, sont évidemment tout à fait envisageables.
Il en est de même pour les différents mécanismes permettant l'interprétation de la gestuelle utilisateur et du contexte d'interaction dans lequel ce dernier évolue. Ces mécanismes simples, générant de la part du système des décisions tranchées et peu sujettes aux compromis, peuvent bien sûr devenir plus complexes. Cette complexification ira d'ailleurs de paire avec celle des indices caractérisant les différents éléments à interpréter. Nous proposons dans ces travaux une interprétation basée sur la comparaison au cours du temps des contextes attendus par le scénario avec ceux construits et observés au cours de l'interactivité.
Pour finir, un module interface avec la logique concepteur, dédié à la mise en place de la réponse interactive du système vis-à-vis de l'activité au sein de la scène virtuelle, a été développé. Ce développement a suivi la logique de programmation de l'application, dans le sens où celle-ci et le module ont été développés ensemble et par le biais des mêmes outils informatiques.

4. Logique concepteur

Nous ne contribuons pas véritablement, dans ces travaux de thèse, à l'élaboration de la couche relative à la logique concepteur. En effet, la sémantique de cette couche est très élevée et nous ne sommes pas responsables ici du développement de cette dernière, ainsi que des différents modules la composant et des communications partant et arrivant de celle-ci. Ces différentes problématiques sont toutefois l'objet de plusieurs études au sein de l'équipe ***ImagIN***.
Nous avons cependant proposé, au cours du chapitre précédent, un modèle de scénario, basée sur les notions de situations et de contextes d'interaction. Dans le cas de notre scénario, nous déterminons les différents contextes d'interaction, qui sont, à l'image du scénario, simples et limités. Seul le scénario de l'application est modélisé par le biais de notre approche. En revanche, nous ne nous occupons pas dans ces travaux de l'implémentation et de la gestion, au sein de la couche logique concepteur, de ce modèle, pas plus celles des objets (situations & contextes d'interaction) qui le composent.
Au sein de la couche de l'architecture correspondant à la logique concepteur, l'analyse de la scène virtuelle observée entraine, à partir du scénario, la modélisation de la réponse du système à l'utilisateur (interactive et adaptée). Tout ce qui concerne les mécanismes de décision au niveau de la couche logique concepteur, vis-à-vis de la scène observée et du scénario de l'application, n'est pas étudié dans ces travaux de thèse. Pas plus que les mécanismes de modélisation de la réponse du système à partir de ces décisions. En fonction d'une scène observée, nous disposons en effet directement d'une nouvelle description de la scène virtuelle, facilement implémentable et automatiquement déterminée au cours de l'interactivité, et d'un ensemble d'informations contextuelles et de mesures adaptatives immédiatement utilisables, quelque soit le niveau sémantique de la couche ou du module cible.

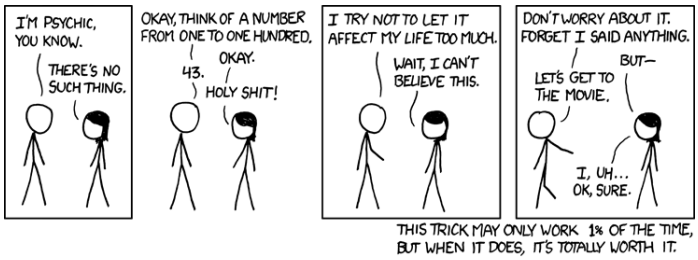
5. Gestion des connaissances

Nous énumérons, dans le cadre de ces travaux, les différentes connaissances que nous manipulons, qu'elles soient globales ou contextuelles, et de manière aussi exhaustive que possible.
Nous ne nous occupons pas dans ces travaux de la gestion, au niveau de la couche logique concepteur, des connaissances globales, pas plus que la déclinaison automatique de celles-ci en informations contextuelles, en fonction des contextes d'interaction extraits du scénario. Ces processus sont pour nous transparents, et nous avons directement et immédiatement à notre disposition les différentes connaissances dont nous avons besoin, en accord avec le déroulement de l'application et le niveau sémantique de la couche ou du module cible.
En revanche, il sera question, dans ces travaux, de la gestion et du traitement des différentes connaissances contextuelles au cours de l'interactivité, en accord avec les niveaux sémantiques des différentes couches de l'architecture. Nous regrouperons ces différentes informations contextuelles au sein des contextes définis au cours du Chapitre 2.

6. Gestion de l'adaptativité

Enfin, nos dernières contributions sont relatives à l'implémentation de l'adaptativité au sein du système.
Comme nous l'avons mentionné précédemment, nous ne nous occupons pas des mesures adaptatives décrites et implémentées au niveau de la couche logique concepteur. La gestion de ces dernières, ainsi que leur sélection en fonction de la scène observée et leur déclinaison à travers les différentes couches de l'architecture nous sont transparentes. A l'instar des informations contextuelles, pour une couche sémantique donnée, nous avons à notre disposition une description adéquate des différentes mesures adaptatives que déclenche l'observation d'un contexte particulier. Nous avons donc développé dans ces travaux les différents modules chargés de la mise en œuvre des différentes mesures adaptatives, relativement aux différentes couches de l'architecture du système.
Les différents mécanismes adaptatifs, établis à partir du scénario de l'application, sont relativement simples. Ils ont en effet été plus définis pour illustrer la pertinence de notre approche, plutôt que pour concevoir un système interactif complexe. Un des buts premiers de la mise en place de ces mécanismes est d'améliorer le fonctionnement des différents processus mis en jeu au cours de l'interaction, particulièrement le processus de capture de la scène réelle. Ces mécanismes sont du type de ceux que nous avons énumérés au cours de la section précédente. Nous les énumérons précisément au cours du chapitre concernant le démonstrateur développé au cours de ces travaux, ainsi que les différentes informations contextuelles qu'ils utilisent (Chapitre 5.).

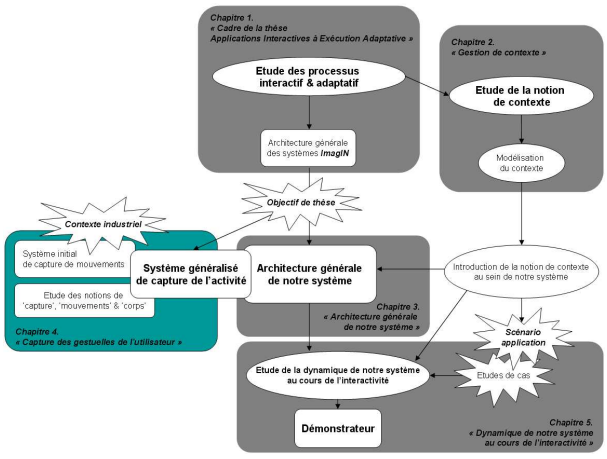
Chapitre 4. Capture des gestuelles de l'utilisateur



L'homme se dégrade à mesure que l'ouvrier se perfectionne
- Bergson -

A chaque fois qu'on parle d'ontologie, une fée s'éteint
- RR -

Résumé



Le point de départ du processus interactif est **la capture de l'activité au sein de la scène réelle**. Cette capture permet la modélisation des indices caractéristiques de l'activité au sein de la scène virtuelle.
Le processus de capture de l'activité est **la généralisation** et **l'extension du processus de capture des mouvements de l'utilisateur**, mis en œuvre par le ***Cyberdôme***. Le ***Cyberdôme*** est le point de départ autour duquel s'articulent ces travaux de thèse et constituent le système initial adopté en accord avec le contexte industriel de cette thèse.

Notre objectif fut donc de passer d'un processus de capture des mouvements du corps de l'utilisateur à celui de **l'activité de manière plus globale au sein de la scène réelle**, activité englobant particulièrement celle relative aux gestuelles de l'utilisateur.
L'expression « capture des mouvements du corps » présente plusieurs significations et ambigüités. Nous définissons tout d'abord les différentes facettes de cette expression. Ces définitions nous permettent de nous positionner et de préciser les aspects du processus de capture, que nous implémentons dans le cadre de ces travaux.
Nous présentons ensuite un état de l'art non exhaustif des différents systèmes de capture actuels, industriels et académiques, permettant ainsi de souligner les différents aspects d'un processus de capture, ainsi que les problématiques que ce processus englobe.
Nous présentons par la suite le ***Cyberdôme***, développé par la société XD Productions. Nous exposons **nos solutions pour alléger les deux contraintes qu'il présente** (contrôle du fond de la scène et nécessité de marqueurs colorés).

Plan du chapitre

Le chapitre précédent nous a permis d'établir l'architecture du système interactif que nous avons conçu dans le cadre de ces travaux de thèse. Nous avons vu qu'un système était requis pour **capturer la scène réelle**, au sein de laquelle **l'activité** prend place. Une des composantes majeures de cette activité est l'ensemble des **gestuelles** qu'adopte l'utilisateur au cours de l'interactivité. Ce chapitre traite de la capture des **gestuelles utilisateur** dans le cadre de notre système interactif.
Dans le cadre de cette thèse en convention ***CIFRE***, notre système de capture a été élaboré à partir d'un système commercial de capture de gestuelles, le ***Cyberdôme*** de la société XD Productions. **Nos contributions**, que nous présentons dans ce chapitre, ont permis d'**alléger les contraintes** que pouvait présenter ce système (contrôle de la scène réelle et nécessité de marqueurs pour l'utilisateur), tout en l'**étendant à une capture de l'activité globale** au sein de la scène (gestuelles utilisateur et événements dont l'utilisateur n'est pas la source). Notre système de capture obéit aux paradigmes de la manipulation directe et de l'interaction gestuelle, présentés au cours du Chapitre 1. (Section 1.2.). Il constitue une interface transparente qui permet à l'utilisateur d'adopter une gestuelle naturelle, tout au long de l'interactivité.

La première section de ce chapitre propose une définition des termes **'capture'**, **'mouvement'** et **'corps'**, suivi d'une caractérisation des différentes problématiques s'articulant autour de ces notions. Nous verrons qu'il est important de préciser la signification de ces différents concepts. En effet, la **'capture'** englobe 3 aspects, que nous présenterons, et peut être effectuée par le biais de 3 processus bien particuliers. Le terme de **'gestuelle'**, préféré à celui de **'mouvement'** s'exprime, en accord avec les besoins concepteur, par le biais de mouvements, de gestes, d'actions et de comportements, composantes donc de cette gestuelle que nous définirons. Enfin, selon les applications, le processus de capture de gestuelles est dédié à un **'corps'** particulier, du regard de l'utilisateur à son corps dans sa globalité, en passant par ses doigts, ses expressions, etc.

La deuxième section sera plus technique et **présente plusieurs systèmes permettant la 'capture' des 'gestuelles' du 'corps'**. Nous nous arrêterons plus particulièrement sur les systèmes **non invasifs** qui permettent à l'utilisateur d'évoluer librement, sans être contraint par une quelconque interface matérielle. Ces systèmes constituent des interfaces à transparence maximale, et l'élaboration d'un tel système est un de nos objectifs. C'est pourquoi les systèmes non invasifs nous intéressent.
Cette section n'a pas pour objectif d'établir un état de l'art exhaustif autour de la capture de gestuelles. En effet, en accord avec le contexte industriel de cette thèse, nous travaillons avec un système commercial de capture déjà développé, dont les grandes lignes directrices ne peuvent pas être modifiées. L'objectif de cet état de l'art est plutôt de souligner **les différents aspects du processus de capture**, ainsi que **les problématiques rencontrées suivant les approches envisagées**. Ces aspects et problématiques seront illustrés par plusieurs projets industriels ou relatifs au domaine public.

Enfin, la dernière section présentera le ***Cyberdôme***, système commercial de capture de gestuelles, développé au sein de la société XD Productions. Ce système, utilisé dans cette thèse, a été acquis par l'équipe ***ImagIN***. Ce système multi-caméras permet l'animation d'un modèle 3D par le biais des gestuelles de l'utilisateur. Nous détaillerons les différentes facettes de ce système au cours de la section.
Nous présenterons ensuite nos **contributions** vis-à-vis de ce système. Dans le cadre de ces travaux de thèse, nous proposons des solutions pour alléger les contraintes que peut présenter

le ***Cyberdôme*** (total contrôle de la scène réelle et nécessité de porter des marqueurs pour l'utilisateur). De plus, nos travaux ont permis d'étendre le processus de capture, mis en place par le système, à une activité plus large au sein de la scène réelle (capture des gestuelles de l'utilisateur et d'événements relatifs à des modifications visuelles de la scène et dont l'utilisateur n'est pas responsable).

1. 'Capture' des 'mouvements' du 'corps'

Il nous semble important de bien préciser les différents aspects qu'englobent l'expression « capture des mouvements du corps ». En effet, les termes **'capture'**, **'mouvement'** et **'corps'** peuvent être interprétés de différentes manières, en fonction des objectifs du concepteur, du système élaboré et des applications développées.
Nous allons étudier dans cette section les différentes facettes que présentent les termes **'capture'** (Section 1.1.), **'mouvement'** (Section 1.2.) et **'corps'** (Section 1.3.), afin d'identifier précisément les différentes problématiques s'articulant autour de ces notions.

1.1. 'Capture'

Qu'est-ce que 'capturer' une gestuelle? Capturer, c'est s'emparer d'une chose, l'intercepter, l'arrêter, la saisir, la photographier dans le temps. Nous parlerons de la capture d'une personne, d'un animal, d'une « capture écran », etc. Nous pensons que la notion de 'capture' comprend **trois facettes**, **la détection**, **l'acquisition** et **le suivi** au cours du temps d'une gestuelle. Nous décrierons ces trois facettes au cours de la section suivante. Il ne s'agit pas d'une définition précise de cette notion mais plutôt d'une présentation des différents concepts, très interdépendants les uns avec les autres, qu'elle englobe entièrement.

Nous pensons que **trois types de processus** sont développés dans le but de présenter les trois facettes qu'englobe la notion de **'capture'**. Ces trois processus, tous qualifiables de « capture de gestuelles » (ou de l'activité dans certains cas), mis en jeu par le biais de systèmes très diversifiés, sont présentés au cours de la Section 1.2. **Le premier processus** correspond à l'image commune qu'il est possible de se faire de la capture de gestuelles. Il permet la caractérisation précise des gestuelles d'éléments spécifiques composant un sujet d'intérêt. **Le second processus** s'arrête plus sur la détection d'une gestuelle (d'une activité de manière générale), qu'elle qu'en soit la source et sans forcément la caractériser. Enfin, **le dernier processus** ne cherche pas non plus à caractériser précisément une gestuelle mais la photographie à un instant donné, en saisit l'apparence visuelle.

1.1.1. La 'capture', un processus de détection, d'acquisition et de suivi

Tout d'abord, capturer consiste à capter, à **détecter la gestuelle**. Il est d'ailleurs important de souligner ici que la traduction française officielle de *motion capture* n'est pas la capture de gestuelles, mais la captation de gestuelles, c'est-à-dire l'opération de capter, par le biais de capteurs, la gestuelle. Cet aspect de la capture implique une observation préalable de la scène. Cette observation peut reposer sur un ensemble de connaissances quant à la nature de la scène observée, l'identité et les propriétés de l'objet mobile, etc. En se basant sur ces connaissances, l'observation de la scène est alors généralement suivie d'une phase de segmentation et de caractérisation de l'objet d'intérêt.

La deuxième facette qu'englobe la notion de capture est alors **l'acquisition de la gestuelle**, à un instant donné. La gestuelle est interceptée, dans le sens enregistrée et stockée, sous une représentation informatique plus ou moins variée. Selon le type de systèmes et d'applications développés, cet enregistrement pourra être utilisé, par exemple, pour attribuer la gestuelle à un personnage virtuel.
Pour être enregistrée, la gestuelle sera d'abord codifiée, caractérisée. Pour que cette caractérisation soit significative par la suite, il faut qu'elle traduise exactement quel type de gestuelles le système a capturé. C'est pourquoi cette facette implique la connaissance du type de **'mouvement'** que le système doit capturer (voir Section 1.2.).

Enfin, la dernière facette de la capture est relative au **suivi de la gestuelle au cours du temps**. La capture devient donc un processus temporel, qui s'attache à la caractérisation de la gestuelle sur un intervalle de temps donné, et de manière cohérente d'un instant à un autre (**acquisition** de la gestuelle sur un intervalle de temps donné). Comme précédemment, cela suppose que le type de **'mouvement'** soit connu, pour le caractériser de manière représentative.

1.1.2. 3 processus de 'capture'

Lorsqu'ils pensent à la 'capture' de gestuelles, certains imaginent immédiatement un acteur de cinéma, bardé de capteurs aussi bien sur ses vêtements que sur sa peau, jouant une scène donnée, pour qu'il soit alors possible de restituer visuellement ses gestuelles à travers un personnage 3D (un avatar). Et c'est bien sûr à cette vision qu'est principalement attribuée la notion de capture. Le processus de capture est ainsi u**n processus de caractérisation de la dynamique de la gestuelle d'une entité d'intérêt** (un homme, un animal, un objet). Ce processus inclut aussi bien **la détection**, **l'acquisition** et **le suivi** de la gestuelle. Il est alors possible de réattribuer cette dernière, de l'analyser, etc.

Mais, une activité peut correspondre à un changement visuel de la scène observée, ou d'une partie de celle-ci, à un instant donné. Le dynamisme au niveau d'une scène observée peut être du à l'utilisateur (une gestuelle de sa part dans le haut de l'image, par exemple), ou à un facteur complètement extérieur celui-ci (un changement d'éclairage, par exemple). L'activité est alors détectée en comparant, selon différentes approches, une image courante avec une image ou un ensemble d'images passées. L'***EyeToy*** de Sony[1], par exemple, ne cherche pas à caractériser et suivre la gestuelle effectuée par le joueur, mais détecte les changements visuels, au cours du temps, au sein de la zone observée, en comparant l'apparence de celle-ci dans le passé avec son apparence dans le présent. Peu importe que la gestuelle utilisateur soit effectuée par sa main ou son pied, ce que veut savoir le système, c'est s'il existe une modification visuelle de la zone observée, modification caractérisant l'activité au sein de la scène réelle et qui entrainera une réaction scénarisée de sa part.
Relativement aux concepts englobés dans la notion de 'capture', **ce processus de perception**, sans forcément s'intéresser à la source de la gestuelle utilisateur et sans chercher à la caractériser précisément, peut également être qualifié de capture (de l'activité dans ce cas).

Enfin, pour présenter un dernier type de processus de capture de gestuelles, partons d'un exemple. Le 4 novembre 2008, sur la chaîne ***CNN***, le présentateur Wolf Blitzer couvre les

[1] http://www.us.playstation.com/PS2/Games/EyeToy_Play/ogs/

élections présidentielles avec sa collègue, Jessica Yellin. Il est à New York, elle à Chicago, et pourtant ils dialoguent sur le même plateau de télévision, lui s'adressant au soit disant « hologramme » de la présentatrice[2] (voir Fig. 4.1.).
Malgré le terme employé par la célèbre chaîne de télévision, il ne s'agit pas d'un hologramme. Les présentateurs ne se trouvent effectivement pas dans la même ville, lui étant sur son plateau de télévision, elle se trouvant dans un studio entourée de caméras haute définition (entre 35 et 44, selon les sources). Cette dernière évolue devant le fameux « fond vert » (ou plutôt bleu, à en juger par la couleur du halo l'entourant) et est segmentée (extraite des images vidéo), en temps réel. Il y a donc autant d'images de la présentatrice extraites que de caméras, chacune correspondant à un point de vue particulier. En fonction du mouvement de la caméra sur le plateau de télévision, le bon point de vue, et donc la bonne image de la présentatrice, est sélectionné et projeté sur l'image retransmise finale, vue par les téléspectateurs. Le présentateur, quant à lui, ne voit rien, si ce n'est un plateau vide. Il sait juste dans quelle direction regarder pour rester 'en face' de la présentatrice.
Il est donc clair qu'il ne s'agit pas ici d'un hologramme. En revanche, respectant ce qui a été dit précédemment, ne s'agit-il pas d'une forme de capture de gestuelles ? La gestuelle n'est pas du tout caractérisée mais bien photographiée au cours du temps, dénotant une **détection**, une **acquisition** et un **suivi** de celle-ci. Nous pensons que **ce processus de reproduction visuelle de la gestuelle**, sans la caractériser et sans en connaître la source, peut être qualifié de 'capture'.

Figure 4.1. : Un processus de capture de mouvements ?

Ces exemples nous montrent que le type de processus élaboré est fortement dépendant de l'application développée et du besoin que le système doit satisfaire. De plus, ils mettent en évidence que pour chaque processus, les 3 facettes qu'englobe la notion de 'capture' sont mises en avant suivant différents niveaux les unes par rapport aux autres.

1.2. 'Mouvement'

Le terme **'mouvement'** est vague et regroupe un important ensemble de concepts. Jusqu'à maintenant, dans ces travaux de thèse, nous lui avons préféré le terme de **'gestuelle'**, sans définir précisément cette notion.

[2] http://www.youtube.com/watch?v=thOxW19vsTg

Nous définissons la **'gestuelle'** comme étant la logique dans laquelle s'inscrivent les **'mouvements'**, les **'gestes'**, les **'actions'** et les **'comportements'** que peut adopter l'utilisateur. Nous allons, au cours de cette section, définir et caractériser chacun de ces termes. La Figure 4.2. illustre la notion de gestuelle, relativement aux concepts que nous allons définir par la suite.
Pour établir ces définitions, nous nous plaçons dans le contexte de notre thèse, à savoir une interaction de l'utilisateur par le biais uniquement de ses gestuelles. La gestuelle est une spécialisation particulière du comportement de l'utilisateur, dans notre contexte scientifique. Nos définitions seraient à retravailler (l''action', le 'comportement') si nous prenions en compte plusieurs autres modalités, telles que la voix par exemple, un comportement décrivant alors une structure organisée de mouvements, gestes, actions multimodales et d'autres composantes relatives aux modalités envisagées.
Il à noter qu'au sein de l'équipe ***ImagIN***, plusieurs travaux sont en cours pour définir une ontologie au sein de laquelle ces différentes notions s'articuleraient.

Comme nous l'avons mentionné précédemment, la gestuelle de l'utilisateur est composée de plusieurs composantes que nous allons définir tour à tour dans les prochaines sections. Ainsi, la Section 1.2.1. définit la notion de **'mouvement'** ; la Section 1.2.2. celle de **'geste'** ; la Section 1.2.3. d' **'action'** ; et enfin la dernière section définit la notion de **'comportement'**.

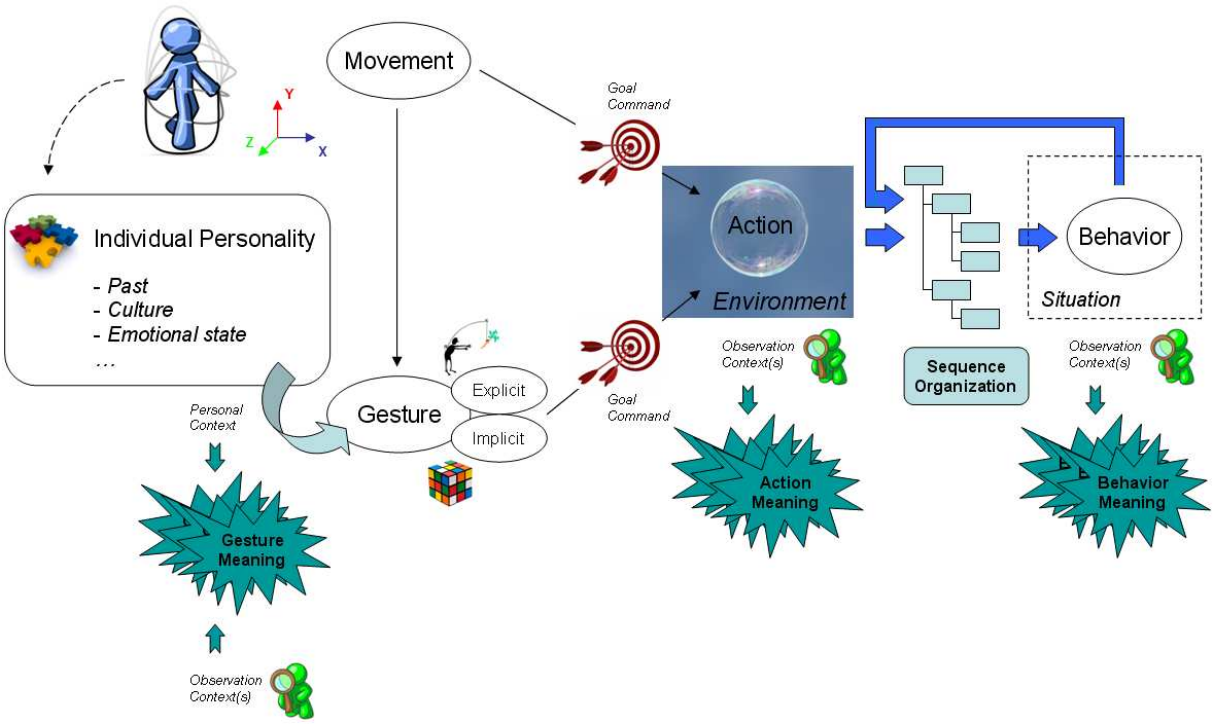


Figure 4.2. : Illustration de la 'gestuelle'

1.2.1. Le mouvement au sein d'une gestuelle

Le mouvement est **le déplacement** d'un sujet d'intérêt, ou d'un élément le composant, **au cours du temps**, **par rapport à un repère de référence**, fixe ou mobile, comprenant une origine et 2 ou 3 axes, si nous parlons respectivement d'une image 2D ou d'un monde en 3D.

Selon le processus de capture développé (voir section précédente), le concepteur s'intéressera à la dynamique du mouvement ou aux modifications visuelles (dans l'image 2D par exemple) que celle-ci entraine. Le mouvement représente l'unité de la gestuelle la plus bas niveau. Il est le support de toute autre unité composant la gestuelle.
Un mouvement peut être **composé** et donc se référer à une séquence structurée de mouvements de plus bas niveau. Un mouvement composé peut être divisé (il sera alors question de segmentation de mouvement) par rapport au temps (découpage de la séquence en fonction du temps) ou par rapport aux mouvements de plus bas niveau (découpage de la séquence en fonction des dynamiques des mouvements de plus bas niveau).
Enfin, un mouvement peut être **explicite**, c'est-à-dire être intentionnellement effectué par l'utilisateur, suivant un objectif bien précis. Ou bien, il peut être **implicite**, c'est-à-dire non effectué de manière consciente. Les réflexes, par exemple, sont à classer dans cette dernière catégorie.

Par exemple, la pose de la main de l'utilisateur est définie comme l'ensemble des translations et rotations, auxquelles cette main obéit, par rapport à un point origine et respectivement suivant et autour des axes du repère. L'utilisateur bougeant sa main au cours de l'interaction, la pose de la main évolue au cours du temps. La Figure 4.3. illustre l'exemple d'une analyse du mouvement effectué par un joueur de golf.

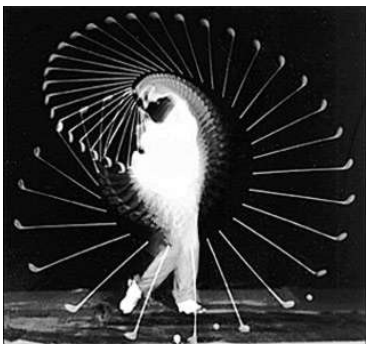


Figure 4.3. : L'analyse du mouvement d'un joueur de golf

1.2.2. Le geste au sein d'une gestuelle

Le geste est défini comme **un mouvement véhiculant un sens**, une signification. Le geste permet de différencier une machine d'un humain : les mécanismes d'horlogerie d'une montre seront considérés comme des mouvements alors que l'humain n'effectue volontairement et consciemment que des gestes (voir Fig. 4.4.).

Figure 4.4. : Le mouvement d'un mécanisme d'horlogerie et le geste d'un humain

Le geste est teinté non seulement par **les connaissances personnelles** de la personne effectuant ce dernier, mais également par **sa personnalité individuelle** : culture, passé, état émotionnel courant, etc. La Figure 4.5. (à gauche) montre un exemple de geste teinté par la culture de l'individu : au Japon, le 'Moi' passe par la désignation de son nez par son doigt.

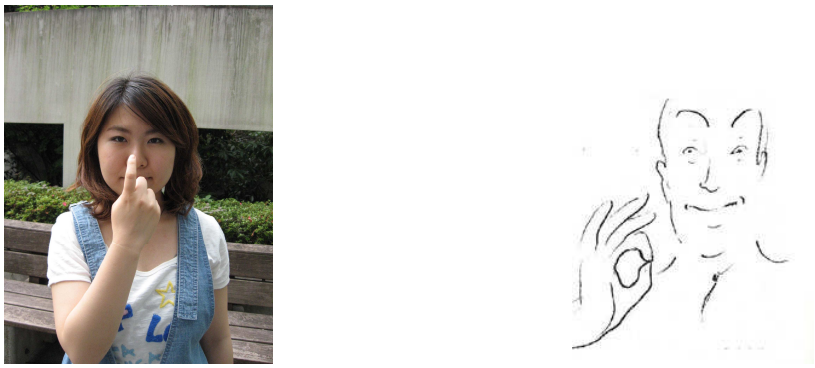


Figure 4.5. : A gauche, le 'Moi' au Japon – A droite, un geste différent suivant le contexte d'observation

La signification d'un geste change donc **d'un contexte personnel à un autre**, mais également **d'un contexte d'observation à un autre** ! Le geste illustré par la Figure 4.5. (à droite) sera considéré comme un assentiment ou signifiera que tout va bien, dans la plupart des cultures, alors qu'un brésilien le considérera comme un geste grossier. Le geste s'accompagne donc de la notion d'observateur.

A l'image des mouvements, un geste pourra être **explicite**, effectué consciemment, car piloté par une envie, une motivation, une intention ; ou **implicite**, effectué de manière non consciente, mais qui, une fois déchiffré par l'observateur, caractérisera un état interne spécifique de la personne observée (signes de fatigue, état d'inattention, etc.).

Comme nous l'avons souligné dans le premier chapitre de cette thèse, la meilleure interaction avec un système est celle pilotée par une gestuelle naturelle de l'utilisateur. La question se pose donc de savoir ce qui rend un geste **'naturel'** pour une personne.
Selon [Morris 78], un geste, c'est-à-dire le fait d'effectuer un mouvement en accord avec son sens, peut être assimilé par une personne de trois manières différentes :

(1) Selon l'auteur, un geste peut tout d'abord être inscrit dans nos gênes, dès notre naissance. Il parle alors de geste **inné**.
(2) Le geste **acquis** est assimilé inconsciemment par la personne. Ce type de geste est sujet à discussion car certains considèrent qu'un geste non inné a du être appris par l'exemple et donc non découvert de manière naturelle.
(3) Enfin, le dernier geste est le geste **appris**, assimilé par l'observation réfléchie ou l'enseignement d'un tiers.

Comme [Baudel 95], nous constatons que selon cette classification, une gestuelle naturelle s'appuiera d'avantage sur des gestes innés et que ce degré de naturel diminuera au fur et à mesure que des gestes acquis, puis des gestes appris, seront considérés. Cependant, comme l'auteur, nous pensons qu'il est trop simpliste d'adopter cette classification lorsqu'il s'agit d'une gestuelle naturelle employée dans le cadre d'une interaction avec un système. Pour nous, une gestuelle naturelle s'appuiera sur des gestes dont le sens est complètement assimilé dans la logique cognitive d'une personne. Pour une personne, un geste sera naturel si elle peut le reconnaître et l'adopter immédiatement et de manière non ambigüe, même si elle ne l'a jamais effectué auparavant.

[Cadoz 94] définit **les 3 fonctions**, à la fois complémentaires et interdépendantes, **du geste** d'une personne **vis-à-vis de son environnement**. Cet environnement englobe la personne effectuant le geste mais peut englober également d'autres personnes. En ce sens, il devient le destinataire du geste.
Tout d'abord, la première fonction du geste est appelée **la fonction sémiotique** du geste. Elle traduit le fait qu'un geste émet et communique des informations sémiotiques significatives à l'environnement. Ces informations sont fonction des connaissances, du profil, des objectifs, etc. de la personne effectuant le geste. Le geste d' « au revoir » ou la désignation d'un objet ou d'un endroit sur la demande d'une autre personne illustrent la fonction sémiotique du geste.
La fonction ergotique illustre la capacité d'un geste à pouvoir directement manipuler et modifier l'environnement. L'auteur donne l'exemple d'un potier qui, dans son métier, utilise grandement la fonction ergotique du geste.
Enfin, le geste permet l'exploration tactile de l'environnement. La personne effectuant ce geste en extrait alors des connaissances spécifiques, en accord avec ses connaissances sur l'environnement qu'elle a déjà acquises. Il s'agit ici de **la fonction épistémique** du geste. En touchant une surface, nous en extrayons un certain nombre de connaissances concernant sa structure, sa composition, la matière dont elle est composée, etc.
[Cadoz 94] soulignent également, par plusieurs exemples (celui d'un chef d'orchestre notamment), que ces 3 fonctions peuvent être augmentées si le geste est instrumentalisé.
Dans le domaine de l'Interaction Homme-Machine, la fonction ergotique du geste a été tout d'abord majoritairement exploitée, par le biais des périphériques standards que sont le clavier et la souris. Puis avec l'avènement des périphériques non standards et de nouveaux paradigmes d'interaction, comme la réalité virtuelle, la fonction épistémique du geste a elle-aussi été de plus en plus considérée. Aujourd'hui, avec le développement de systèmes diffus et ubiquitaires, invisibles pour l'utilisateur, c'est de plus en plus la fonction sémiotique du geste qui est étudiée.

Toutes ces notions sont extrêmement importantes pour nos travaux puisque, dans le cadre de notre système, nous cherchons à interpréter les gestes explicites et implicites de l'utilisateur, tels qu'ils ont été définis précédemment. De plus, l'utilisateur interagit virtuellement avec un environnement en 3D, sans interaction tactile directe (pas de sensation de toucher, pas de retour de force, interaction purement virtuelle). Nous orienterons donc nos travaux vers **l'étude des fonctions sémiotique et ergotique du geste**.
Comme nous le verrons dans la section suivante, nous définissons l'action comme étant un geste dont la fonction est, au moins, ergotique.

1.2.3. L'action au sein d'une gestuelle

Nous définissons **l'action** comme **étant un mouvement** ou **un geste** (fonction ergotique), généralement **volontaire**, dont le but est d'**agir** (un acte) **sur l'état de l'environnement**, qui s'en trouve alors modifié.
L'environnement inclut aussi bien le milieu englobant la personne effectuant l'action que d'autres personnes. Cet environnement définit un cadre dans lequel peuvent être captés des stimuli spécifiques en accord avec les règles (physiques par exemple) le régissant.

Une action véhicule un but, une commande. Une fois l'acte fini, le but est atteint, la commande est exécutée. Une action est généralement caractérisée sémantiquement par un verbe à l'infinitif. Comme le geste, l'action peut véhiculer un objectif différent, **d'un observateur à un autre**.
Ces actions sont facilitées par l'emploi d'**affordances**[3], introduites par [Gibson 77]. En effet, celles-ci doivent inviter immédiatement l'utilisateur à effectuer une action bien particulière, en accord avec l'élément informatique que l'affordance représente.

Une réaction est un acte retour, répondant aux stimuli générés par l'environnement extérieur, précédemment modifié par une action antérieure. La Figure 4.6. donne quelques exemples d'actions et de réactions.

Figure 4.6. : Quelques exemples d'actions et de réactions

1.2.4. Le comportement au sein d'une gestuelle

Le comportement est la dernière 'unité' interprétée, de plus haut niveau, qui compose la gestuelle de l'utilisateur. Un comportement peut donc être décomposé en **une séquence structurée** de **mouvements**, de **gestes**, d'**actions** (et réactions) et d'autres **comportements**.

Un individu adopte un comportement spécifique face à un ensemble de stimuli générés par l'environnement extérieur dans lequel il se trouve. De plus, il adopte ce comportement en fonction des différentes connaissances qu'il possède, en rapport avec son état interne, avec la scène, avec sa tâche, etc. Autrement dit, c'est l'ensemble des mouvements, gestes, actions et comportements qu'il adoptera **face une situation donnée**, dans un contexte donné. Ce comportement est observable et interprétable par autrui (notion d'observateur). Son sens peut évidemment être différent d'un observateur à un autre.

[3] Eléments dont la représentation suggère leur usage, sans apprentissage au préalable

Un comportement est adopté, il est alors question de manière de se comporter. La Figure 4.7. illustre quelques exemples de comportements de groupe, animal, agressif, etc.

Figure 4.7. : Quelques exemples de comportements

Nous cherchons à caractériser et à interpréter les comportements qu'adopte un unique utilisateur vis-à-vis des réactions d'un système. Un comportement **explicite** est adopté intentionnellement par l'utilisateur, tandis que ce dernier peut interagir de manière non consciente avec le système, en adoptant des comportements **implicites**. Ces comportements sont caractérisables par le biais d'indices que le système doit détecter et interpréter. Ces indices sont fonction de l'application et des objectifs qu'elle définit et devront être déterminés par le concepteur au cours de l'élaboration du système et de l'application.
Dans le cadre de nos travaux, une gestuelle est une spécialisation du comportement de l'utilisateur, dans la mesure où nous ne considérons qu'une famille de modalités (le langage du corps) pour interagir avec le système.

1.3. 'Corps'

Lors de l'élaboration d'un système de capture de gestuelles du corps de l'utilisateur, le concepteur doit déterminer comment il veut capturer ('capture'), quel type de gestuelles il veut capturer ('mouvement') et effectuées **par quelle partie du corps** ('corps') de l'utilisateur. Cette section traite de ce dernier aspect.
Cette section est courte et de nombreux exemples de travaux seront donnés par la suite, illustrant chaque type de système.

Nous distinguons 6 types de **'corps'**, chacun correspondant à des problématiques scientifiques bien particulières. A l'heure actuelle, peu de systèmes sont conçus pour capturer les gestuelles de plusieurs types de 'corps', chacun impliquant des approches souvent complètement différentes. C'est pourquoi nous les distinguons dans cette section. Nous présentons ici notre classification :

- Les traits du visage, les expressions
- Les yeux, le regard
- La tête
- Les doigts
- Les mains
- Le corps dans sa globalité ou bien un des membres spécifiques le composant (jambe, bras, épaule, etc.)

2. Systèmes de capture de gestuelles

Nous allons présenter dans cette section divers **systèmes de capture de gestuelle du corps**, en accord avec nos définitions précédentes.
Cette section n'a pas l'objectif de présenter un état de l'art exhaustif, relativement à la capture de gestuelles du corps. En effet, dans la mesure où nous utilisons dans nos travaux, un système commercial, les aspects principaux du processus de capture sont déjà imposés par ce dernier.
Cette section cherche plutôt à présenter et à illustrer ce qu'est la capture de gestuelles par le biais de divers exemples. Ainsi, nous allons mettre en évidence les différentes formes de capture, ainsi que les différentes problématiques rencontrées lors de l'élaboration d'un système donné. Toutefois, le lecteur intéressé par un état de l'art sur le sujet, ainsi que le survol des différentes problématiques relatives à celui-ci, pourra se référer à [Moeslund & Granum 01, Sminchisescu 02, Moeslund & al. 06].

La Section 2.1. présente les raisons pour lesquelles un système de capture de gestuelles du corps est conçu, tandis que la Section 2.2. expose les avantages et les inconvénients d'un tel processus.
La Section 2.3. proposera une classification, accompagnée d'une présentation succincte, des différents systèmes de capture existants aujourd'hui. Nous nous arrêterons plus précisément sur la capture de gestuelles **non invasive**, qui est un objectif vers lequel nous cherchons à nous diriger dans le cadre de ces travaux.

2.1. Pourquoi développer un système de capture de gestuelles ?

Un système de capture de gestuelles peut exister dans le but d'accomplir diverses tâches, allant de l'animation d'un personnage virtuel à la reconstruction visuelle du sujet filmé, en passant par l'analyse de sa gestuelle. De plus, un système de capture de gestuelles est souvent conçu pour qu'il puisse accomplir plusieurs de ces tâches. Nous distinguons dans cette section 3 raisons principales pour lesquelles un système de capture de gestuelles du corps est développé : **pour animer**, **pour reconstruire** et **pour comprendre**.

1. *Pour animer.*
Majoritairement, un système de capture de gestuelles est utilisé pour animer un personnage 3D (un avatar), **en lui attribuant les gestuelles du sujet filmé**. L'avatar reproduit donc fidèlement, au sein d'un environnement 3D, les gestuelles du sujet filmé. L'assignation (il sera parfois question de *mapping*) des gestuelles du sujet filmé peut se faire en temps réel ou *a posteriori*. L'enregistrement d'une gestuelle du sujet filmé peut alors être retravaillé lors d'une phase de finalisation, en travaillant les courbes d'animation par exemple.
Généralement, ce type de système nécessite une représentation virtuelle du sujet filmé intermédiaire, animé également par les gestuelles capturées et se trouvant en amont de l'avatar. Cette représentation intermédiaire permet de faire le lien entre le sujet physique et l'avatar, en associant chaque partie du corps du sujet à un élément spécifique composant l'avatar. Cette liaison est définie par l'animateur lors d'une étape préalable de calibrage. La représentation intermédiaire du sujet filmé est

particulièrement intéressante et utile lorsque la morphologie de l'avatar est différente de celle du sujet filmé.

Un avatar peut être animé dans le but de communiquer. Ainsi, en 2007, la ***SNCF*** utilisa des avatars pour communiquer en langage des signes le texte écrit sur les écrans dans les gares[4] (voir Fig. 4.8., à gauche). A la place d'un avatar, un robot peut être animé par le biais des gestuelles d'une personne (voir Fig. 4.8., à droite). Ce type d'animation est de plus en plus usité pour que la machine animée effectue des tâches pénibles, voire impossibles, pour l'homme[5], ou encore dans le domaine médical. Enfin, et c'est l'usage le plus connu, la capture de gestuelles permet l'animation de personnages virtuels dans des films (quelques exemples : ***Avatar***, ***Le Seigneur des Anneaux***, ***Iron Man***, ***Polar Express***, ***Final Fantasy: The Spirit Within***, ***Hulk***, ***Happy Feet***, etc.), des émissions TV, des publicités ou encore des jeux vidéo (quelques exemples : ***Guitar Hero: Metallica***, ***Devil May Cry 3***, ***Battlefield Bad Company***, ***Assassins Creed 2***, ***Dance Groove Online***, ***Heavenly Sword***, ***Deadly Creatures***, ***Headbang Hero*** [Martins & al. 09], etc.).

Généralement, pour ce type d'animation, d'importants systèmes de capture de gestuelles sont utilisés pour obtenir une meilleure animation. L'acteur est bardé de capteurs sur l'ensemble de son corps et est entouré de caméras. Ainsi, la capture est libérée des problèmes typiques respectivement liés aux occultations (le fait qu'une partie du corps de l'acteur cache une autre partie) et au dynamisme potentiel de l'environnement de capture. De plus, la contrainte temps réel n'est pas nécessaire, ce qui permet de traiter les courbes d'animation en *post processing* (enregistrement des données de capture et traitement de celles-ci *a posteriori*).

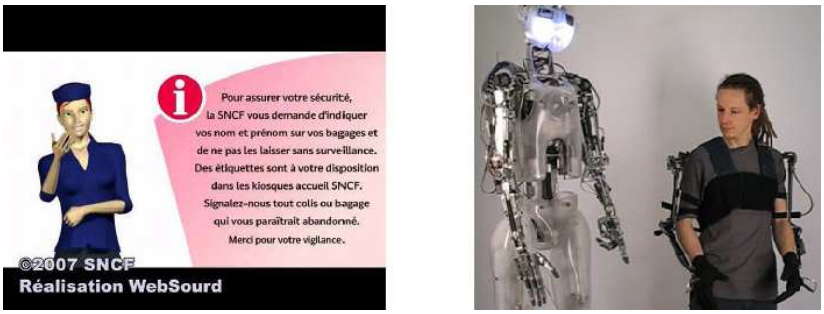


Figure 4.8. : La capture de mouvements du corps pour animer

2. *Pour reconstruire.*

La reconstruction d'un sujet filmé implique **la reproduction visuelle réaliste**, en temps réel ou différé, **du sujet d'intérêt capturé**. Cette reconstruction peut être plus ou moins élaborée, de la sélection d'une image en fonction du point de vue (voir l'exemple de ***CNN*** plus haut) à l'animation, la déformation et l'habillage visuel (en utilisant les textures vidéo) de la représentation virtuelle du sujet d'intérêt utilisée par le processus de capture. Quoiqu'il en soit, la reconstruction du sujet capturé est effectuée en accord avec les caractéristiques 2D & 3D extraites de la scène observée.

Dans ce type d'approche, le nombre de caméras, leur caractéristiques vidéo (résolution, *frame rate*, etc.) et leur disposition, sont des facteurs déterminants

[4] http://www.websourd.org/
[5] Un autre exemple : http://www.youtube.com/watch?v=v1AJ_OBJUpY

concernant la complétude et la précision de la reconstruction du sujet filmé. Un nombre de caméras important, disposées autour de l'utilisateur (i.e. englobant tous les points de vue possibles), permettra d'obtenir une information complète voire redondante. De plus, des caractéristiques caméras déterminées avec précision permettront la reconstruction fidèle et de qualité du sujet d'intérêt, au cours du temps. Enfin, une reconstruction nécessite un haut niveau de contrôle au niveau de l'environnement de capture. Ainsi, les caractéristiques 2D & 3D peuvent être facilement et rapidement extraites des flux vidéo (d'un point de vue algorithmique).
Actuellement, la reconstruction temps réel (pour les méthodes les plus élaborées) d'une personne capturée reste un problème ouvert et ne cesse d'offrir des challenges scientifiques intéressants dans le domaine du traitement et de la synthèse d'images.

3. *Pour comprendre.*
Ce dernier usage de la capture de gestuelles constitue un sujet scientifique exhaustivement étudié de nos jours. Il s'appuie sur le fait qu'une gestuelle, en plus de sa dynamique dans l'espace, apporte un ensemble d'informations sémantiques sur sa signification.
La capture permet ainsi **l'analyse et l'interprétation de la gestuelle** du sujet filmé. La dynamique de la gestuelle adoptée par le sujet d'intérêt est alors étudiée et est généralement accompagnée de la reconnaissance de celle-ci au sein d'un éventail de gestuelles répertoriées et caractérisées.
La capture peut donc permettre l'étude d'une gestuelle, sportive par exemple [Urtasun & al. 05], sa traduction (langage des signes) en un langage compréhensible par l'homme, ou son utilisation pour interagir avec une application, dans le cadre d'une interface avancée (traduction de la gestuelle selon le vocabulaire compréhensible par l'application) [Maes & al. 94, Baudel 95, Bobick & al. 99, Billon & al. 08, le Life Wall de Panasonic[6], Sixth Sense Technology de Mistry[7]].

2.2. Avantages/Intérêts & Inconvénients/Limites

Au cours de cette section, nous allons tout d'abord exposer les avantages et intérêts que peut présenter un système de capture de gestuelles, que ce soit au niveau de l'utilisateur, de l'interaction ou du système en lui-même (Section 2.2.1.). Puis, la Section 2.2.2. énumérera alors les inconvénients et limites d'un tel système.

2.2.1. Avantages & Intérêts

1. *Une interaction naturelle ne nécessitant pas, ou peu, d'apprentissage*
Tout d'abord, une interaction par le biais d'un système de capture est généralement plus **naturelle** pour l'utilisateur. Les concepteurs du système et de l'application cherchent effectivement de plus en plus à ce que l'utilisateur puisse interagir facilement et rapidement, sans apprentissage au préalable de la part de l'utilisateur. L'utilisateur n'est plus nécessairement un expert et peut être un homme, une femme, un personne jeune, une personne corpulente, etc.

[6] http://www.youtube.com/watch?v=pekz2XH69CY
[7] http://www.ted.com/talks/lang/eng/pranav_mistry_the_thrilling_potential_of_sixthsense_technology.html

Pour ce faire, les concepteurs disposent de plusieurs moyens. Ils peuvent s'appuyer sur la classification de Morris en élaborant les interactions basées sur une gestuelle à caractère de plus en plus inné et de moins en moins appris [Morris 78]. Comme nous l'avons mentionné précédemment, cette distinction est trop simple et trop restrictive. En effet, si nous considérons que les gestuelles innées sont les gestuelles, par définition, les plus naturelles qui soient, certaines gestuelles acquises ou apprises peuvent pourtant s'avérer tout à fait naturelles pour l'utilisateur. Par exemple, les gestuelles du type « mouvement de coup droit au tennis», ou « lancer au bowling » sont des gestuelles acquises ou apprises, et pourtant tout à fait intuitives et naturelles pour n'importe qui.
C'est pourquoi nous pensons que l'interaction doit être élaborée de telle manière à ce qu'elle celle-ci soit une véritable métaphore d'une gestuelle physique dans le monde réel, connue par l'utilisateur. [Baudel 95] parle de cette métaphore comme d'une « bijection, une correspondance directe entre le modèle mental que l'utilisateur a de l'application et le modèle choisi pour l'interaction ». Plus la traduction entre le sens véhiculé par la gestuelle de l'utilisateur et le vocabulaire compréhensible par le système est faible, plus cette bijection est atteinte et donc plus l'interaction est naturelle.
Le naturel de l'interaction et la rapidité d'apprentissage seront accrue par l'exploitation efficace d'affordances [Gibson 77]. L'interface devient simple, la jouabilité[8] (*gameplay*) plus évidente, s'il est question d'un jeu vidéo. Grâce à la représentation adéquate des objets d'intérêt et à une courte suite d'essais et d'erreurs de la part de l'utilisateur, ce dernier apprend rapidement à manipuler de manière cohérente l'application, améliorant progressivement ses gestuelles pour atteindre un objectif donné.

2. *Une interaction concise*
Nous reprenons ici une idée de [Baudel 95]. Comme nous l'avons défini précédemment, un geste apporte un sens au mouvement. Le geste est donc porteur d'une dynamique, mesurable et quantifiable, et d'une signification, ensemble d'informations sémantiques qu'il est beaucoup plus difficile de mesurer. Dès la naissance, nous apprenons à faire véhiculer un sens bien particulier dans nos gestes, de manière à communiquer **avec concision**. En développant son application, le concepteur attribue à une gestuelle, effectuée par l'utilisateur, le sens qu'il désire, en fonction de la tâche qu'il veut faire exécuter par ce dernier. C'est pourquoi si l'utilisateur doit adopter une gestuelle naturelle pour interagir, alors il interagira avec concision. Autrement dit, plus une interaction sera naturelle, plus elle sera concise.

3. *Une interaction directe et facilement combinable avec d'autres paradigmes d'interaction*
Interagir par le biais d'un système de capture de gestuelles est la meilleure façon d'obéir au paradigme de **la manipulation directe** (voir la Section 1.2. du Chapitre 1.), cela d'autant plus si la gestuelle adoptée est naturelle.
De plus, en choisissant ce type d'interaction, le travail de conception d'établissant autour des techniques d'interaction, qui vont permettre à l'utilisateur d'interagir avec le système, est grandement facilité et optimisé. Si la gestuelle envisagée est naturelle, la détermination de ces techniques d'interaction est **immédiate**. Mais ce type d'interaction invite également à imaginer de nouvelles techniques d'interaction innovantes. Enfin, cette interaction intègre également les opérations standards couramment utilisées de nos jours

[8] La jouabilité d'un jeu regroupe les possibilités d'action du joueur, les règles guidant ses actions définissant la manière de jouer, l'impact de ces actions sur l'environnement virtuel, l'ergonomie des commandes, la facilité à interagir avec le système, etc.

avec le clavier et la souris : manipulation directe (désignation, entrées caractères et numérotation,…), interaction parallèle de type bimanuelle, etc.
D'autres paradigmes d'interaction peuvent très facilement **complétés** cette forme d'interaction, comme, par exemple, en ajoutant la voix aux possibilités d'interaction gestuelles (interaction parallèle et multimodalité).

4. *Une interface transparente*
Un avantage majeur de ce type d'interaction est qu'elle permet à l'utilisateur **d'oublier cognitivement l'interface avec le système**. En effet, celle-ci disparaît (il sera question de **la transparence** de l'interface) de la prise en compte cognitive de l'utilisateur, permettant à ce dernier de se focaliser uniquement sur ses objectifs. Le corps de l'utilisateur devient le périphérique, interaction idéale en matière d'ergonomie. Une capture de gestuelles non instrumentalisée, ou encore non invasive, et une interaction aussi naturelle que possible définit une transparence maximale de l'interface. Et plus l'interface sera transparente, plus l'interaction sera directe.

5. *Une interaction modélisable en 3D*
Un avantage non négligeable, en tout cas dans le cadre de nos travaux, de ce type d'interaction est qu'une gestuelle de l'utilisateur peut potentiellement être représentée dans un **environnement en 3 dimensions**. Ainsi, toutes les applications nécessitant un environnement 3D sont envisageables et tout à fait pertinentes (jeux vidéo, modélisation 3D, etc.).
Généralement, l'utilisateur est représenté par un avatar. L'avatar est alors animé par les gestuelles de l'utilisateur. Ainsi, ces gestuelles doivent être codifiées et traduites par le système. Autrement dit, il existe une phase de mise en correspondance, de *mapping*, entre les gestuelles de l'utilisateur et ceux de l'avatar. Plus les gestuelles de l'utilisateur seront reproduites fidèlement, moins elles seront traduites et plus ce *mapping* sera direct. Un *mapping* indirect (donc une traduction non évident *a priori* de la gestuelle utilisateur) permet à l'utilisateur de 'tricher' sur ses gestuelles. La ***Wiimote*** de Nintendo est un exemple de *mapping* indirect. Les gestuelles du joueur sont captées par des accéléromètres, ce n'est pas la position de la main qui est détectée mais bien ses accélérations linéaires suivant 3 axes orthogonaux. C'est l'application en elle-même (particulièrement les simulations sportives) qui invite le joueur à effectuer la vraie gestuelle (à rapprocher finalement de la notion d'affordance de Gibson). Mais ce dernier n'est absolument pas obligé de la faire et peut se servir de ses connaissances sur le contrôleur pour ne pas faire la 'vraie' gestuelle. Dans le cas d'un *mapping* direct (la gestuelle est fidèlement reproduite), l'utilisateur effectue la véritable gestuelle, il ne peut pas tricher sur sa gestuelle. Si un élément 3D est hors de portée de l'utilisateur, celui-ci doit physiquement se déplacer dans l'espace réel pour pouvoir l'atteindre, comme il le ferait dans la vie de tous les jours.
Quoiqu'il en soit, interagir par le biais de ses gestuelles donne à l'utilisateur le sentiment **de manipuler et d'influer directement sur les objets de la tâche**, d'agir véritablement, de s'engager et de s'impliquer personnellement et physiquement dans le scénario. L'interaction lui donne le sentiment de contrôler véritablement l'application. Ce sont ces aspects qui sont particulièrement mis en avant dans les applications impliquant un contrôle de l'utilisateur sur une scène 3D (jeux vidéo, modélisation 3D, etc.).

2.2.2. Inconvénients & Limites

1. *De nombreuses sources d'imprécisions et d'erreurs*
   Tout d'abord, **la fidèle reconnaissance** des gestuelles de l'utilisateur reste un problème scientifique majeur de nos jours. Le concepteur ne peut plus se contenter d'adopter des solutions matérielles adaptées pour résoudre cette problématique. Il doit dorénavant les combiner avec des solutions logicielles et algorithmiques avancées.
   De plus, si nous nous référons à notre contexte de thèse, **la contrainte temps réel** est une contrainte dure, qu'il est de plus en plus difficile de satisfaire au fur et à mesure que les traitements s'ajoutent et se complexifient. Or, encore une fois, la capture et l'interprétation des gestuelles de l'utilisateur n'est pas un problème trivial et nécessitent souvent des approches complexes et difficiles. Le travail d'optimisation est donc constant et le concepteur doit en permanence faire un compromis entre la robustesse, la précision et la rapidité du processus de capture.
   Enfin, pour une gestuelle capturée, les ambigüités de sens, que l'interprétation de cette dernière peut engendrer, ne sont pas rares. Pourquoi un 'bonjour' de la main ne pourrait-il pas être un 'au revoir' ? Nous proposons dans ces travaux de résoudre ce problème en modélisant le contexte d'interaction au sein duquel l'utilisateur effectue une gestuelle, fiabilisant ainsi le processus d'interprétation. D'autres solutions existent, comme l'exploitation de modalités supplémentaires, telles que la voix par exemple.

2. *Interagir par le biais du geste n'est pas forcément le plus adapté*
   Comme l'explique très bien [Dragicevic 04], la question de la pertinence d'une telle interaction vis-à-vis de l'application peut se poser. En effet, à partir du moment où il existe une forte traduction des gestuelles de l'utilisateur en un langage compréhensible par le système (mapping indirect ou référence à des situations non connues de l'utilisateur), il devient ardu pour le concepteur d'élaborer l'ensemble des techniques pour interagir avec le système. Cette gestuelle doit apparaître aussi naturelle et instinctive que possible pour l'utilisateur. De plus, certaines opérations restent difficiles à retranscrire au moyen d'une gestuelle naturelle. Les opérations à distance ainsi que les opérations fines, nécessitant une grande précision de la part de l'utilisateur, ne sont pas possibles dans la plupart des cas. Une saisie textuelle ou numérique peut également vite s'avérer laborieuse. Enfin, dans les deux cas, le concepteur a devant lui un choix quasi illimité de gestuelles permettant l'interaction avec le système. Ce travail de conception est souvent très difficile.
   Lorsqu'interagir par le biais d'une gestuelle s'avère délicat, le concepteur a toutefois plusieurs possibilités pour faciliter cette interaction. Tout d'abord, il peut élaborer une gestuelle d'interaction, composées d'actions incrémentales, amenant progressivement l'utilisateur vers son objectif. De plus, il doit lui donner le droit à l'erreur : les actions de l'utilisateur doivent être réversibles, l'environnement devant pouvoir revenir à un état initial automatiquement et rapidement. Enfin, la réponse du système doit être immédiatement perceptible et doit mettre en relief l'effet de la gestuelle effectuée par l'utilisateur (meilleurs visualisation et réalisme, retour de force, etc.).

3. *Une interaction potentiellement fatigante*
   L'utilisateur, en interagissant avec le système via des gestuelles, peut ressentir, à la longue, une certaine fatigue physique. Une des solutions à ce problème est de proposer une interaction par le biais d'une gestuelle simple et rapide.
   Il est à noter que dans le cadre de nos travaux, la fatigue n'est pas forcément un problème à résoudre. En effet, dans un contexte d'*exergames* (*exercise* + *games*), la fatigue est un facteur que les concepteurs ne cherchent pas à éviter d'une manière ou d'une autre car faisant partie intégralement partie du jeu. Dans le cadre de nos travaux, nous plaçant

également dans un contexte de simulation sportive, la fatigue physique du joueur, si elle est détectée, peut nous permettre de mettre en place des mécanismes d'adaptation, dans le but, toujours, d'améliorer l'interactivité.

4. *Des problèmes d'ordre plus général*
   Les systèmes de capture, particulièrement ceux qui permettent l'animation d'un personnage virtuel par le biais des gestuelles du corps de l'utilisateur (films, jeux vidéo), sont très onéreux. De plus, certains systèmes sont encombrants et nécessitent un équipement qui peut s'avérer dangereux (des câbles par exemple). Un problème de sécurité peut donc se poser. Enfin, l'interaction par le biais de gestuelles n'est pas accessible à tout le monde.

2.3. Les différents systèmes de capture de mouvements du corps

Nous allons présenter au cours de cette section une classification des différents systèmes actuels de capture de gestuelles du corps. Nous verrons alors que certaines distinctions sont contestables et discutables et les différents systèmes présentés permettront d'illustrer les différentes problématiques rencontrées lors de l'adoption ou de l'élaboration de ceux-ci. Nous développerons plus particulièrement le dernier type de systèmes, qui concerne la capture non invasive, vers laquelle nous tendons à nous diriger dans le cadre de ces travaux de thèse.

Nous énumérons ici les différents types de systèmes que nous distinguons dans ces travaux :

- Les systèmes de capture électromécanique (Section 2.3.1.)
- Les systèmes capture magnétique (Section 2.3.2.)
- Les systèmes capture inertielle (Section 2.3.3.)
- Les systèmes de capture optique (Section 2.3.4.)
- Les systèmes de capture vestimentaire (Section 2.3.5.)
- Les systèmes basés contrôleurs actifs (Section 2.3.6.)
- Les systèmes tactiles (Section 2.3.7.)
- Les systèmes de capture non invasive (Section 2.3.8.)

2.3.1. Systèmes de capture électromécanique

Les systèmes de capture électromécanique furent les premiers systèmes de capture de gestuelles. Le sujet d'intérêt doit porter un exosquelette ou une combinaison (intégrale ou partielle), parsemé de potentiomètres et de résistances, autour des éléments à capturer le composant.
Par exemple, la société Sonalog[9] vend un tel système (le ***Gypsy MIDI***) pour contrôler des sons ***MIDI***[10] (voir Fig. 4.9., à gauche). Les ***Data Gloves*** (voir Fig. 4.9., à droite) sont aussi un exemple de capture électromécanique.

Ces systèmes permettent une capture précise et rapide, dans tout type d'environnement. En revanche, ce sont des systèmes encombrant et contraignant, limitant les gestuelles de l'utilisateur. De plus, ils impliquent une adaptation manuelle de l'exosquelette au corps à

[9] http://www.sonalog.com/
[10] http://www.youtube.com/watch?v=S8mTd1GzLQw

capturer et nécessitent parfois des connexions filaires, pouvant alors poser des problèmes de sécurité. Enfin, l'exosquelette peut également s'avérer fragile et le système a tendance à se décalibrer dans le temps (le système est calibré lors d'une phase initiale).

Figure 4.9. : Utilisation du ***Gypsy MIDI***, société Sonalog (à gauche) – ***Data Gloves*** (à droite)

2.3.2. Systèmes de capture magnétique

Une capture magnétique requiert que le sujet revête des capteurs (des bobines électromagnétiques) et qu'il soit immergé dans un champ électromagnétique partant d'un générateur, qui constitue l'origine du monde virtuelle. Les bobines perturbent alors le champ et ces perturbations sont alors détectées par une antenne réceptrice. Les positions et rotations des capteurs peuvent alors être calculées.
La société française Novamotion[11] utilise ce type de systèmes pour l'animation de personnage 3D. La société Ascension[12], quant à elle, l'utilise pour des applications médicales et militaires temps réel. Enfin, [Saponas & al. 09] proposent une alternative intéressante à ce type de systèmes, capturant les gestuelles des doigts de l'utilisateur par le biais des muscles de ses bras, lui permettant ainsi de garder les mains libres[13]. En dernière remarque, nous mentionnerons que la société XD Productions utilisait également ce type de systèmes (voir Fig. 4.10.).

[11] http://www.novamotion.com/
[12] http://www.ascension-tech.com/index.php
[13] http://www.youtube.com/watch?v=6_7BzUED39A

Figure 4.10. : Exemple de capture magnétique chez XD Productions

Ce type de systèmes assure une capture précise et rapide et permet l'animation temps réel d'un personnage au sein d'un environnement 3D. Cela dit, ces systèmes ont des inconvénients similaires à ceux que présentent les systèmes électromécaniques (voir section précédente). De plus, l'espace de capture est en général restreint et son calibrage est souvent fastidieux. Enfin, le champ magnétique dans lequel évolue le sujet capturé peut être perturbé par tout élément générant un champ électromagnétique, comme un câble électrique passant sous l'espace de capture par exemple.

2.3.3. Systèmes de capture inertielle

Les systèmes de capture inertielle sont actuellement des systèmes couramment utilisés, que cela soit pour des applications industrielles, des simulations pour l'analyse de gestuelles, des jeux vidéo, etc. Les gestuelles sont captées par des centrales inertielles, insérées dans une combinaison intégrale que doit porter l'utilisateur.
La société Xsens, par exemple, est connue pour utiliser ce type de systèmes[14] (voir Fig. 4.11.).

[14] http://www.xsens.com/en/general/mvn

Figure 4.11. : Exemple de capture inertielle, société Xsens

Ces systèmes permettent la résolution d'une partie des problèmes des captures électromécanique et magnétique. Ils sont précis et rapide, complètement insensible à l'environnement de capture, qui peut être très important, et non filaire. Cependant, ces systèmes ne permettent pas de se libérer d'une phase de calibrage vis-à-vis de la morphologie du sujet capturé et de l'espace de capture (tendance également à se décalibrer dans le temps). Enfin, les combinaisons de capture peuvent présenter une certaine fragilité et, même si ces dernières sont plus confortables, le sujet capturé peut toutefois ressentir une gêne dans ses gestuelles.

2.3.4. Systèmes de capture optique

De nos jours, c'est la capture de gestuelles la plus populaire. Le sujet capturé est bardé de marqueurs, actifs ou passifs, qui seront détectés visuellement et en deux dimensions sur plusieurs points de vue synchronisés, permettant alors le calcul de leurs positions dans un espace en 3D. Cette remarque implique donc une phase de calibrage préalable de l'espace de capture.

Les marqueurs actifs génèrent électroniquement un signal lumineux. Ce signal est détecté visuellement sur chaque image vidéo. Dans les derniers systèmes commerciaux, des signaux lumineux de couleur rouge sont détectés par des caméras infrarouges. NaturalPoint[15] est un exemple de société utilisant des marqueurs actifs au cours d'une capture optique[16] (système ***OptiTrack***, voir Fig. 4.12., à gauche).
Les marqueurs actifs sont plus fiables que les marqueurs passifs et permettent une capture rapide et robuste. Ces systèmes sont bien sûr non filaires et permettent une capture au sein de grands espaces (en fonction du nombre de caméras, de leurs caractéristiques, des marqueurs, etc.). En revanche, les systèmes optiques sont des systèmes complexes. La qualité de la capture dépend du nombre et des caractéristiques des caméras, du nombre et de la puissance des marqueurs, de l'espace de capture, etc. De plus, la capture étant assurée visuellement, elle devient sensible aux occultations (le fait qu'une partie du corps du sujet capturé cache un marqueur pour un point de vue donné). Il n'est donc pas toujours possible, si le marqueur est caché sur un nombre de point de vue important, de calculer sa position 3D avec précision.

[15] http://www.naturalpoint.com/optitrack/
[16] http://www.youtube.com/watch?v=22t_75DV720

C'est pourquoi, ce type de capture requiert généralement une phase de *post processing*, durant laquelle les positions des marqueurs sont interpolées et les courbes d'animation retravaillées. Enfin, les systèmes optiques commerciaux sont extrêmement onéreux.

Figure 4.12. : Système ***OptiTrack*** (marqueurs actifs), à gauche – Système ***Vicon***[17] (marqueurs passifs), à droite

Figure 4.13. : Marqueurs passifs pour la capture des mouvements du visage, société Mova

Figure 4.14. : Marqueurs passifs pour la capture des mouvements du visage, société Image Metrics

Les marqueurs passifs délivrent un signal lumineux par le biais d'une substance réfléchissante particulière, couvrant un support matériel que porte le sujet capturé (une balle de mousse attaché par un scratch généralement) ou déposée à même la peau (typiquement, pour la capture de type reconstruction visuelle du visage, comme la société Mova[18], voir Fig. 4.13.).

[17] De http://spacecraft.ssl.umd.edu/SSL.photos/SSLevent.photos/2009/090323.Vicon/090323.index.html
[18] http://www.mova.com/

Ce signal est généré par le biais d'un éclairage particulier porté sur le marqueur : une succession de flashs, une lumière directionnelle générée par des LEDs, etc. Les sociétés Vicon[19] (voir Fig. 4.12., à droite) et Image Metrics[20] sont des sociétés utilisant ce type de procédé (voir Fig. 4.14.).

S'ajoutant aux avantages que peut présenter les systèmes à marqueurs actifs, le principal avantage de ces systèmes est qu'ils permettent une reconfiguration des marqueurs rapides et pratiques. Cela dit, ils présentent également les mêmes inconvénients que les systèmes à marqueurs actifs. L'utilisation de marqueurs passifs nécessite également l'emploi d'algorithmes de traitement d'images complexes et difficiles (résolution des occultations, problèmes liés à la proximité des marqueurs les uns par rapport aux autres, aux gestuelles rapides, etc.).

2.3.5. Systèmes de capture vestimentaire

Dans le cas d'une capture vestimentaire, le sujet capturé porte le système complet sur soi. C'est un système tout-en-un, composé des capteurs disposés sur le sujet capturé, d'un ordinateur portable dédié aux différents calculs relatifs à la capture et potentiellement un système d'immersion (généralement un masque immersif que doit porter le sujet).
Ces systèmes ne sont généralement produits commercialement que pour des applications militaires ou dans d'autres domaines spécialisés. Le projet *Anywhere Augmentation*[21] est le cadre du développement de tels systèmes, intégrant une immersion de l'utilisateur dans tout type d'environnement. Des applications de réalité augmentée s'exécutent sur ces systèmes. [Di Verdi & Höllerer 07] propose le ***GroundCam***, un autre exemple intéressant de système vestimentaire (voir Fig. 4.15.). Ce système assure le suivi des gestuelles d'une personne évoluant aussi bien en intérieur qu'en extérieur, par le biais d'une approche se basant sur le traitement de flux optiques, d'un filtre de Kalman et de données GPS. Enfin, [Vlasic & al. 07] est aussi un exemple de systèmes vestimentaires, focalisant sur le côté pratique d'une telle forme de capture de gestuelles[22].

Figure 4.15. : ***GroundCam*** [Di Verdi & Höllerer 07], exemple de systèmes de capture vestimentaires

[19] http://www.vicon.com/index.html
[20] http://www.image-metrics.com/
[21] http://ilab.cs.ucsb.edu/projects/anywhereAugmentation/
[22] http://www.youtube.com/watch?v=V0yT8mwg9nc

Les systèmes vestimentaires sont en général peu coûteux, insensible à tout type d'environnement, et pouvant opérer une capture dans un espace illimité. Cependant, ces systèmes, qui restent souvent à l'état de prototype, sont fragiles, encombrants et contraignants, et ne fournissent pas forcément une capture robuste et précise.

2.3.6. Systèmes basés contrôleurs

Les systèmes basés contrôleurs nécessitent de la part de l'utilisateur la manipulation de contrôleurs, actifs ou passifs. L'industrie visée est généralement celle du jeu vidéo.

Les contrôleurs actifs génèrent électroniquement un signal, détecté par le système. La souris & le clavier, une manette, un joystick, la ***Wiimote***, ou tout autre contrôleur de jeux vidéo sont des exemples de contrôleurs actifs. Les signaux générés par ces contrôleurs (ils peuvent être purement visuels) permettent le calcul d'une position, d'une accélération, d'une force, d'une pression, etc. et sont détectés par le biais de capteurs dédiés. Il existe généralement un *mapping* indirect entre la gestuelle de l'utilisateur et l'application, traduisant ainsi cette gestuelle en un langage compréhensible par le système.
Les travaux de Chung Lee sont de bons exemples de développement de systèmes de capture à partir d'une ***Wiimote***[23]. [Zhang & al. 09] proposent également l'intéressant développement d'un système de capture, sur lequel s'exécutent des jeux vidéo, basé sur l'emploi de plusieurs contrôleurs hétérogènes en simultané.

Il existe des systèmes basés contrôleurs actifs, qui nécessitent l'emploi de contrôleurs plus importants que le simple accessoire. A dire vrai, ce sont de véritables systèmes englobant physiquement l'utilisateur. Ces systèmes se situent entre le périphérique et le *smart environment*. Généralement, l'utilisateur doit adopter une gestuelle naturelle pour interagir avec de tels systèmes, dont le fonctionnement est alors connu *a priori*. Ces systèmes sont bien sûr onéreux et nécessitent un espace important. [Mori & al. 08] présente un tel système de capture.

Les contrôleurs passifs sont des objets ne générant pas de signal mais qui sont reconnus par le système en tant qu'interfaces matérielles avec l'utilisateur. Ce type de contrôleurs est particulièrement adopté dans des applications de réalité mixte (voir Chap. 1., Section 1.2.2.3.). Ces contrôleurs peuvent posséder des propriétés d'objets utilisés couramment par l'utilisateur, ou être plus abstraits, leur utilisation devant alors être apprise.
La société Cam-Trax Technologies développe le logiciel ***CamSpace***[24], permettant à un ou plusieurs utilisateurs d'interagir avec plusieurs jeux (réalité augmentée ou virtuelle) par le biais d'objets de la vie courante ou fabriqués pour l'occasion (une bouteille en plastique, un volant en carton, etc.).

2.3.7. Systèmes tactiles

[23] http://johnnylee.net/projects/wii/
[24] http://www.camspace.com/

Les systèmes tactiles permettent une capture des gestuelles des parties du corps qui rentrent en contact avec le système. Nous sommes bien conscients que dire qu'un système tactile est un système de capture de gestuelles est contestable. Nous ne les mentionnons donc que brièvement dans cette section.
Il existe plusieurs types de systèmes, se différenciant dans leur capacité à détecter le toucher de l'utilisateur : systèmes résistifs, capacitifs, basés onde de surface, infrarouges, basés vision, etc.
Les travaux de Han sont des exemples connus de systèmes tactiles commerciaux[25]. La Figure 4.16. présente également des systèmes tactiles connus.

Figure 4.16. : Quelques exemples de systèmes tactiles : ***iPhone***, ***Nintendo DS***, ***Surface***, Mur tactile (Han)

2.3.8. Systèmes de capture non invasive

Les systèmes de capture non invasive permettent la capture des gestuelles de l'utilisateur sans que ce dernier ne doive porter ou ne doive manipuler un quelconque équipement. Ces systèmes sont basés vision, i.e. permettant la capture de gestuelles en se basant uniquement sur l'observation de la scène. Cette observation est généralement purement visuelle mais de nouvelles modalités, telles que, par exemple, le son, la voix de l'utilisateur, sont de plus en plus considérées.
Il est à noter qu'un système de capture non-invasive peut s'étendre à la capture de l'activité (gestuelles utilisateur + autres événements dont l'utilisateur n'est pas la source) prenant place au sein de la scène.

Une capture de gestuelles non invasive est l'objectif de nombreux travaux de recherche, aussi bien publics que privés. Il est à noter que chaque approche est différente en fonction des moyens techniques et financiers des concepteurs, ainsi que des objectifs de ces derniers. Il leur faudra déterminer précisément au préalable quelle 'capture' ils veulent effectuer, pour quel 'mouvement' et de quel 'corps'.
Les sociétés Aguru Images Technology[26] et Dimensional Imaging[27] cherche à capturer (reproduction visuelle au cours du temps) les expressions du visage (voir Fig. 4.17.).

[25] http://cs.nyu.edu/~jhan/
[26] http://www.aguruimages.com/home.htm
[27] http://www.di3d.com/

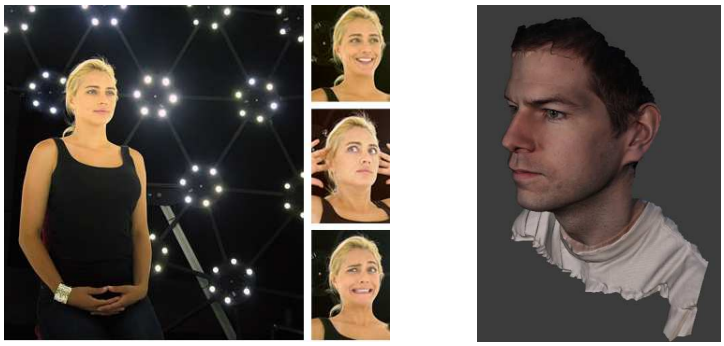


Figure 4.17. : Aguru Images Technology (à gauche) & Dimensional Imaging (à droite)

La société Seeing Machines[28] a développé depuis quelques années le système ***faceLAB***, dont le but est de capturer (dans le sens de suivi) les gestuelles du regard. C'est un système efficace qui est utilisé par de nombreux laboratoires et autres sociétés. En plus du regard, ce système peut tracker quelques traits du visage, comme les commissures de lèvres, les narines, etc.
[Vacchetti 04] propose une méthode robuste et temps réel pour capturer (dans le sens de suivi) les gestuelles de la tête de l'utilisateur. Son approche se base sur la corrélation de points d'intérêt particuliers (les 'coins' de Harris) entre l'image courante, l'image précédente et, éventuellement, une *keyframe*, enregistrée automatiquement ou au cours d'une phase d'initialisation.
[Ye & al. 03] présente un nouveau paradigme d'interaction basé sur les ***VICs*** (*Visual Interface Cues*). Ce paradigme est centré autour des *VIC-based interface components* (***VICons***). Ces éléments observent la scène de manière autonome et permanente, comprennent une représentation visuelle, et réagissent en fonction des événements qui ont lieu dans leur zone d'observation. Les auteurs donnent l'exemple d'un utilisateur appuyant sur un bouton virtuel type ***VICon*** : l'élément détecte tout d'abord la gestuelle puis détermine si c'est bien un doigt qui interagit avec lui. Deux types d'interaction, par le biais du doigt de l'utilisateur, sont présentés. Mais ce paradigme est intéressant car plusieurs types de ***VICons*** peuvent être développés pour capter les gestuelles d'autres parties du corps.
De nombreuses études cherchent à capturer les gestuelles du corps global de l'utilisateur, de manière non invasive. La société Organic Motion[29] vend de tels systèmes et [Michoud & al. 07, Michoud 09] propose également un système de capture des mouvements du corps de l'utilisateur (voir Fig. 4.18.).

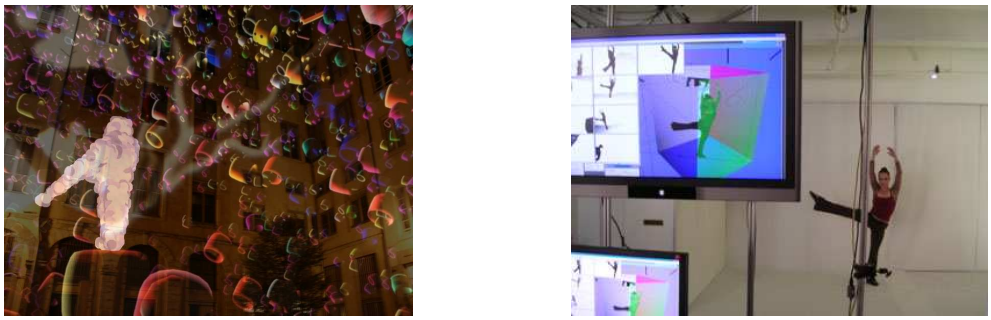


Figure 4.18. : Systèmes non invasifs : [Michoud & al. 07, Michoud 09] (à gauche) & Organic Motion (à droite)

[28] http://www.seeingmachines.com/
[29] http://www.organicmotion.com/

Enfin, au niveau des consoles de jeux vidéo, plusieurs sociétés vendent des plateformes permettant la capture des gestuelles du joueur, sans que celui-ci n'ait à manipuler un contrôleur, ou à porter une combinaison particulière. L'***Eyetoy*** ou le ***Playsation Eye*** de Sony, ou encore plus récemment ***Kinect*** de Microsoft, sont des exemples de telles consoles (voir Fig. 4.19.).

Figure 4.19. : De gauche à droite : l'***Eyetoy***, le ***Playstation Eye*** et ***Kinect***

L'avantage majeur de ce type de systèmes réside bien sûr dans le fait que ces derniers sont non invasifs. L'interface entre le système et l'utilisateur affiche une transparence maximale, permettant à l'utilisateur de s'engager véritablement dans le scénario de l'application, sans aucune limitation ni contrainte au niveau de sa gestuelle. De plus, les systèmes non invasifs, même commerciaux, sont bien moins chers que les autres systèmes de capture, puisque ne nécessitant que peu de matériel.
En revanche, les inconvénients découlent également du fait que ces systèmes soient non invasifs. Comme nous l'avons mentionné précédemment, la capture d'une gestuelle en se basant uniquement sur l'observation de la scène est un problème scientifique difficile. Le processus de capture doit à la fois être robuste, précis et rapide, cela d'autant plus s'il doit obéir à la contrainte temps réel. En effet, ce type de processus doit faire face au dynamisme potentiel de la scène observée, aux occultations, aux ambigüités et aux incertitudes que peuvent présenter les observations, etc. Ce sont donc des systèmes complexes, nécessitant autant une étude au niveau du matériel (nombre, type et disposition des caméras, etc.) qu'au niveau du logiciel (quelles observations pour quel processus de capture ? etc.).

Dans la suite de cette section, nous expliquerons le processus de capture non invasive plus en détails. Nous ne cherchons pas à établir un état de l'art des différentes méthodes existantes, puisque nous utilisons un système de capture de gestuelles déjà existant. Cela dit, ce système est invasif et l'allégement de cette contrainte fait partie de nos objectifs.
De nombreuses connaissances pilotent le processus de capture de gestuelles. Nous allons les énumérer au cours de la Section 2.3.8.1. La Section 2.3.8.2. présentera les différentes étapes du processus.
Le lecteur intéressé pourra trouver dans la littérature de nombreux états de l'art sur le sujet, comme [Bray 01].

2.3.8.1. Connaissances & hypothèses pilotant le processus de capture

De nombreuses **connaissances** et **hypothèses** sont utilisées au cours du processus de capture de gestuelles. Lors de l'élaboration du système de capture, le concepteur doit prendre en compte ces connaissances et hypothèses, qui orienteront le développement du processus, ou certains aspects de celui-ci. En fonction de certaines connaissances ou hypothèses, des traitements supplémentaires seront nécessaires (comme par exemple, la distinction du sujet d'intérêt avec un fond dynamique). D'autres permettront d'en construire de nouvelles, qui influenceront alors le processus. Enfin, certaines connaissances seront utilisées au cours du processus en lui-même, supportant la capture des gestuelles du sujet d'intérêt.
Nous allons ici énumérer les différentes connaissances et hypothèses, sous la forme de questions, qui pilotent les concepteurs dans leur travail. Nous complétons ainsi celles énumérées par [Moeslund & al. 06].
Nous rappelons ici que nous parlons de systèmes **non invasifs**. De plus, nous faisons l'hypothèse que les gestuelles d'un unique sujet d'intérêt sont capturées. Enfin, nous nous arrêtons à l'observation purement visuelle de la scène.

1. *Sujet capturé*

- Le sujet d'intérêt est-il connu ? Est-ce un humain, un animal, un objet ?
- Quel 'corps' doit être capturé ?
- Le sujet d'intérêt est-il en face des périphériques observant la scène ?

Cette connaissance peut permettre d'élaborer un processus de capture à partir de la détection du visage du sujet capturé, par exemple.

- Le sujet d'intérêt reste-t-il dans l'environnement de capture ?
- Si le sujet d'intérêt est un humain, a-t-il une apparence particulière ?

Nous parlons ici particulièrement des vêtements de la personne capturée. Des vêtements particuliers (collant à la peau par exemple), le port de couleurs particulières, etc. peuvent permettre l'initialisation d'un modèle d'apparence, qui supportera alors le processus de capture au cours du temps. La génération d'un modèle d'apparence requiert généralement une phase de calibrage avant la capture des gestuelles de l'utilisateur.

2. *Gestuelle*

- Quelle 'mouvement' ?
- Quelles sont les caractéristiques de la gestuelle observée ? Est-ce que ce sont des gestuelles lentes, rapides ? La gestuelle est-elle connue, suit-elle une modélisation connue (*motion pattern*) ?

Certains systèmes cherchent à capturer et à reconnaître un type particulier de gestuelle, par exemple, la marche ou une gestuelle cyclique similaire, une gestuelle sportive, etc. Lors d'une phase d'apprentissage, un modèle, numérique et sémantique, est créé à partir d'observations enregistrés de ces gestuelles. Ce modèle sert alors de support lors de la capture d'une gestuelle du sujet d'intérêt.

- La pose initiale du sujet d'intérêt est-elle connue ?

Cette pose représente une connaissance absolue, à partir de laquelle le processus de capture peut être déclenché. Par exemple, [Michoud 09] présente une méthode de construction du modèle du sujet capturé, lors d'une phase d'initialisation, à partir d'une pose initiale précise.

- La gestuelle observée va-t-elle générer des occultations ?

Selon le cas, le processus de capture devra être robuste face aux occultations qui seront générées par la gestuelle du sujet capturé. Les occultations peuvent être résolues par le biais d'une solution logicielle (robustesse de l'algorithme de capture) ou matérielle (disposition des caméras permettant d'englober l'ensemble des points de vue).

3. *Environnement de capture – Scène réelle*

- Quelles sont les caractéristiques de l'environnement observé (scène extérieure ou intérieure, relief, etc.) ?

Le processus de capture dépendra bien évidemment du type de scène observée, notamment au niveau de l'extraction des informations permettant de caractériser le sujet capturé au sein de la scène réelle.

- Quelles sont les caractéristiques du fond (*background*) ? Est un fond dynamique (éclairage, personnes mobiles dans le fond, etc.) ou statique ? Est-il de couleur uniforme ?

Selon les processus de capture, il sera nécessaire de différencier le sujet capturé du fond de la scène (étape de segmentation). L'approche envisagée pour cette différenciation dépendra bien sûr des caractéristiques du fond.

4. *Caméras observant l'environnement de capture*

- Quel est le nombre de caméras ? Comment sont-elles disposées vis-à-vis de l'environnement de capture ?

Ces connaissances sont importantes. Le nombre de caméras, ainsi que leur disposition, influent grandement sur le processus de capture.
En effet, les approches sont différentes selon si le système comprend 1 caméra, 2 caméras ou plus de 2 caméras. Plus le nombre de caméras sera important, plus l'information sera redondante et permettra d'atteindre une précision de capture importante.
De plus, un nombre important de caméra et une disposition homogène de ces dernières au sein de l'environnement de capture permettra de résoudre les incertitudes et ambigüités que génèrent les différentes observations extraites de l'environnement : occultations, silhouettes ambigües, etc.

- Quelles sont les caractéristiques des caméras ?

Ces caractéristiques comprennent aussi bien de la focale de la caméra, que les dimensions des images délivrées, en passant par leur position et leur inclinaison. Ces paramètres peuvent être introduits au sein du système par le concepteur et/ou calculés lors d'une phase de calibrage, antérieure à la capture à proprement parler.

- Les caméras sont-elles statiques ou mobiles ?

5. *Processus de capture*

- Quel 'capture' ?
- Le processus est-il temps réel ou est-il exécuté à partir de séquences enregistrées ?

La contrainte temps réel est une contrainte dure pour le développeur du processus de capture. En fonction de l'application, il devra établir un compromis entre la robustesse du processus, sa précision et sa rapidité. Encore une fois, ce type de processus est complexe et la contrainte temps réel implique que tous les traitements soient effectués au minimum en 40ms (ce qui correspond à 25 images/seconde).

- Le processus nécessite-t-il un modèle du sujet d'intérêt (approche *model-based*) ?

Certains processus de capture nécessitent **un modèle géométrique du sujet d'intérêt**, ce qui peut l'alourdir au niveau des traitements mais qui correspond certainement plus à notre processus de vision humain. Ce modèle peut être introduit au sein du système *a priori* ou construit au cours de l'interactivité, il peut être hiérarchisé ou non, en 2D ou en 3D. Il peut également être la conséquence d'une phase de calibrage (apparence, formes, dimensions, structure cinématique), antérieure à la capture des gestuelles du sujet d'intérêt. Nous allons présenter ici quelques aspects de ce modèle, mais le lecteur intéressé pourra se référer aux travaux suivants pour plus de détails : [Delamarre 03, Moeslund & al. 06, Sminchisescu 02].

Ce modèle représente une division du corps du sujet d'intérêt en **segments**. Ces segments sont particulièrement observés et sont instrumentalisés dans le cas d'une capture invasive. Un segment du modèle peut être **identifié par une valeur sémantique** (un 'bras', une 'jambe, etc.) ou numérique (un indice).
Chaque segment est représenté par une **primitive**, pouvant aller d'une simple ligne 2D uniforme à une forme 3D complexe (un *mesh*) pouvant être texturé. Ces primitives peuvent être **rigides** (qui ne se déforment pas au cours du temps) ou **déformables** (qui s'adaptent au cours du processus aux observations courantes).
Ces primitives peuvent alors être organisées (il sera question de **hiérarchie**) et **articulées** les unes par rapport aux autres, par le biais de contraintes, cinématiques par exemple. Les articulations d'un modèle peuvent tout à fait être un ensemble de segments également.

Ce modèle du sujet capturé évolue dans un **environnement virtuel**, en deux ou trois dimensions, selon les approches envisagées concernant le processus de capture. Cet environnement comprend donc un repère orthonormé fixe, selon lequel la pose (postions et rotations suivant les axes du repère) du segment représenté est définie au cours du temps.
La pose d'un segment peut être définie soit de manière **absolue**, c'est-à-dire par rapport au repère de référence de l'environnement virtuel, soit de manière **relative**, vis-à-vis d'un autre segment. Il est en effet courant, lorsque le modèle est structuré sous la forme d'une hiérarchie

de segments, que ces segments soient positionnés les uns par rapport aux autres, chacun étant associé à un repère particulier. Le premier élément de la hiérarchie, qui est, lui, positionné de manière absolue, est alors appelé la 'racine' du modèle. Ainsi, dans cette hiérarchie, le segment 'fils' est positionné par rapport au segment 'père' (qui peut avoir plusieurs 'fils').

Le modèle du sujet d'intérêt est **animé** par les gestuelles de ce dernier, au cours de la capture. Ce modèle évolue donc au cours du temps dans l'environnement virtuel. Les propriétés de cet environnement sont variables : il est possible d'y simuler les lois physiques qui régissent le monde réel (gravité, collision, élasticité, etc.) et le modèle peut interagir avec les éléments virtuels qui composent cet environnement.
Toujours est-il que le mouvement qu'effectue le modèle au sein de cet environnement est **codifiable**, ou **caractérisable**. Cette caractérisation est bien sûr fonction des besoins de l'application, des objectifs des concepteurs, etc. Il existe un grand nombre de formats connus, pour permettre l'import/export d'animations entre systèmes différents (système de capture ou logiciels d'animation). Typiquement, ces formats standardisés décrivent la composition et la structuration du modèle du sujet d'intérêt, ainsi que la caractérisation des gestuelles de ce dernier au sein d'un environnement virtuel au cours du temps. Pour ne citer qu'eux, les formats standards connus sont : ***H-Anim***[30], ***ASF/AMC***[31], ***BVH***[32], ***C3D***[33], ***MPEG-4***[34]…

- Quel est le point de départ du processus de capture ? Quelles sont les observations images qui permettront de capturer la gestuelle du sujet d'intérêt ?

Typiquement, ces observations sont **extraites des images vidéo** et sont **confrontées aux connaissances & hypothèses** déjà introduites ou construites au sein du système. Ces observations peuvent être les silhouettes du sujet capturé, des contours, des intensités, des textures, des zones aux caractéristiques similaires, etc. Elles permettent de construire **l'image observée du sujet capturé**, qui permettra l'estimation du positionnement du modèle (dans le cas d'une approche *model-based*) ou la caractérisation de la gestuelle du sujet.

2.3.8.2. Les différentes étapes du processus de capture

Nous décrivons, dans cette section, les différentes étapes du processus de capture non invasive des gestuelles de l'utilisateur.
Nous reprenons et faisons ici suite à [Moeslund & Granum 01, Moeslund & al. 06], travaux que nous complétons ici. Contrairement à ces travaux qui décrivent le processus de capture en 4 phases, nous divisons ce dernier en **5 étapes** : **l'initialisation du processus**, **le suivi**, **l'estimation du positionnement du modèle** (optionnel), l**a caractérisation de la gestuelle observé**e et, enfin, **l'interprétation de cette gestuelle** (optionnel).

1. *L'initialisation du processus de capture*
   Cette étape concerne **l'acquisition et la construction de toutes les connaissances qui seront nécessaires au système au cours du processus de capture** (de l'interactivité).

[30] http://h-anim.org/
[31] http://www.darwin3d.com/gamedev/acclaim.zip
[32] http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html
[33] http://www.c3d.org/
[34] http://www.m4if.org/mpeg4/

Ces connaissances peuvent être introduites par le concepteur ou peuvent être générées par des étapes dites de calibrages.
Dans le cas d'un système de capture non invasif, au sein duquel le sujet capturé est représenté par un modèle évoluant dans un espace 3D, il est par exemple nécessaire de calculer les matrices caméras, permettant de passer d'une image 2D à un espace 3D et réciproquement, et d'initialiser, voire de construire, le modèle (forme, dimensions, apparence, structure cinématique, pose initiale, etc.).
Plus que de l'initialisation, certains systèmes nécessitent une phase d'apprentissage, vis-à-vis de la gestuelle qu'ils sont sensés capturer et éventuellement reconnaître. Le système est alors entrainé à distinguer les caractéristiques d'une ou de plusieurs gestuelles particulières, par le biais de séquences enregistrées considérées comme des vérités terrain. Par la suite, au cours de la capture, s'il reconnaît ces caractéristiques, le système peut donc reconnaître le ou les gestuelles qu'il observe.

2. *Le suivi du sujet capturé*
   L'objectif de cette étape est **de construire au cours du temps l'image du sujet d'intérêt à partir des informations extraites de l'observation de la scène**. Pour une image vidéo donnée (il sera question de *frame*), le sujet est tout d'abord segmenté (découpé) du reste de l'image vidéo puis caractérisé par certaines données, 2D et/ou 3D, bien particulières (silhouettes, couleurs, arêtes, textures, volume englobant, etc.). Ces informations définissent **l'image du sujet observé**. Puis, pour chaque nouvelle *frame*, l'image du sujet capturé sera mise en correspondance avec la ou les précédentes. Une relation temporelle est donc calculée entre une information caractérisant le sujet d'intérêt à la *frame* courante et cette même information à la ou aux *frames* précédentes.
   L'étape de segmentation cherche à séparer le sujet d'intérêt du reste de l'image. Typiquement, il est donc question, au cours de cette étape, de supprimer le *background* (le fond) du *foreground* (le sujet).

3. *L'estimation du positionnement du modèle du sujet capturé (optionnel)*
   Cette troisième étape est bien sûr nécessaire si le système capture les mouvements du sujet par le biais d'un modèle de ce dernier. Il s'agit ici **d'estimer la pose du modèle au cours du temps, vis-à-vis de la gestuelle du sujet d'intérêt**. La pose du modèle doit donc caractérisée celle adoptée par le sujet à une *frame* donnée.
   Pour chaque *frame*, **l'image du modèle** est calculée, correspondant à une configuration du modèle. Pour pouvoir être comparables, l'image du modèle doit être similaire à celle du sujet capturé (informations de même nature, représentées dans le même repère, en 2D ou en 3D). Par la suite, l'image du modèle est ajustée jusqu'au moment où la distance (ou l'erreur) entre celle-ci et celle du sujet observé est **minimale**. A partir de là, le processus de capture pour la *frame* courante est terminé et la nouvelle pose du modèle est déterminée.
   Plusieurs méthodes permettent le calcul de l'image du modèle. Celle-ci peut simplement correspondre à la pose du modèle à la *frame* précédente, ou d'une *frame* particulière, appelée *keyframe*. L'image du modèle peut être aussi calculée à partir de plusieurs *frames* passées et/ou de *keyframes*. Dans le cas de l'étude d'une gestuelle par le biais d'un *motion pattern*, ce dernier dirigera le processus du calcul de l'image du modèle. Ou encore, une ou plusieurs images du modèle peuvent être calculées par le biais d'un processus de prédiction, se basant sur les *frames* passées, sur des *keyframes*,

sur des relations temporelles particulières, etc. Chaque image correspond alors à une hypothèse supplémentaire, concernant le positionnement du modèle à une *frame* donnée. Enfin, si le système a été entrainé à reconnaître une ou plusieurs gestuelles particulières, c'est l'image même du sujet observé qui permettra le calcul de l'image du modèle.
Les méthodes existantes de capture de gestuelles diffèrent de par les algorithmes utilisés pour le calcul de l'image du modèle, la modification de cette image vis-à-vis de celle du sujet capturé, le calcul de l'erreur entre ces deux images, la minimisation de cette dernière, etc.

4. *La caractérisation de la gestuelle du sujet capturé*
   Généralement, le processus de capture est suivi par **l'animation d'un personnage virtuel**, **l'interprétation de la scène observée**, ou encore **l'établissement de la réponse du système en accord avec la gestuelle**. Dans la mesure où la gestuelle capturée doit être compréhensible par les processus suivant, cette dernière doit être caractérisée, c'est-à-dire codifiée en un langage et un format adéquat.
   La gestuelle du sujet d'intérêt peut être caractérisée sur plusieurs niveaux d'abstraction, des simples poses de chaque élément d'un modèle au cours du temps à la description sémantique des mouvements, gestes, actions et comportements (voir Section 1.2.).

5. *L'interprétation de la gestuelle du sujet capturé (optionnel)*
   Cette étape n'est pas comprise dans le processus de capture à proprement parler, mais le complète la plupart du temps. Il s'agit ici de donner u**ne identité sémantique à la gestuelle du sujet capturé**, ou tout du moins à ces mouvements, gestes, actions et comportements. Toutes les connaissances, sur la scène observée, le sujet d'intérêt, son mouvement, etc. sont exploitées de manière à interpréter et reconnaître ce qui est observé par le système. Il pourra également être question de « reconnaissance d'activités », d' « interprétation de scène », d' « interprétation de gestuelles », etc.

3. Le Cyberdôme

Dans le cadre de nos travaux de thèse, les gestuelles de l'utilisateur sont capturées par un système commercial, le ***Cyberdôme***, développé au sein de la société XD Productions.

Le ***Laboratoire Informatique, Image, Interaction*** (***L3i***) de l'Université de La Rochelle a acquis un ***Cyberdôme*** au cours de l'année 2005, que l'équipe ***ImagIN*** utilise dans le cadre de ses travaux de recherche. Cette thèse a été effectuée à partir du mois de décembre 2006, dans le cadre d'une convention ***CIFRE***, établie entre la société XD Productions et l'équipe ***ImagIN***.

Au cours de cette section, nous allons présenter le ***Cyberdôme***, développé au sein de la société XD Productions, ainsi que les modifications que nous avons apportées au système dans le cadre de ces travaux de thèse. Le ***Cyberdôme*** fut le point de départ de ces travaux de thèse.
Nous allons tout d'abord énumérer les différentes utilisations que la société fait de ce système (Section 3.1.). Puis la Section 3.2. abordera les différents aspects matériels et logiciels du ***Cyberdôme***. La Section 3.3. présentera le processus initial de capture mis en jeu par ce

système, permettant le positionnement en temps réel d'un modèle en 3D, en accord avec les gestuelles de l'utilisateur. La Section 3.4. exposera une évaluation de différents aspects du système. Enfin, au cours de la Section 3.5., nous aborderons les modifications que nous avons apportées au système, dans le cadre de ces travaux de thèse, dans le but d'alléger au maximum les contraintes que présente ce dernier et d'étendre le processus de capture à une activité plus globale au sein de la scène réelle. Nos résultats seront présentés et les différents aspects du système seront évalués.

3.1. Objectifs du système

Le ***Cyberdôme*** est un système commercial de **capture des gestuelles d'un unique utilisateur**. **L'ensemble du corps** de l'utilisateur est considéré. Le système a été développé pour **animer** et **reconstruire**. Les processus de capture mis en jeu sont donc ceux qui permettent **le suivi temporel** et **l'enregistrement des gestuelles de l'utilisateur**, ainsi que **la reconstruction visuelle de son apparence au cours du temps**. La gestuelle capturée n'est donc composée que des mouvements de l'utilisateur, la signification de ces derniers n'étant pas prise en compte.

Ce système a donc été tout d'abord développé dans le but d'animer, en temps réel, un personnage, ou caractère, en 3D, évoluant au sein d'un environnement virtuel potentiellement interactif. L'animation est utilisée non seulement dans le cadre de films et d'émissions télévisées, mais également dans de petits jeux interactifs, développés par la société (voir Fig. 4.20.).
Le système en lui-même permet de calculer les poses 3D, repérées de manière absolue, de chacune des parties du corps de la personne capturée. Ces poses sont alors transmises en temps réel à un logiciel tiers dédié, tel que ***Motion Builder***[35] de la société Autodesk.

[35] http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13581855

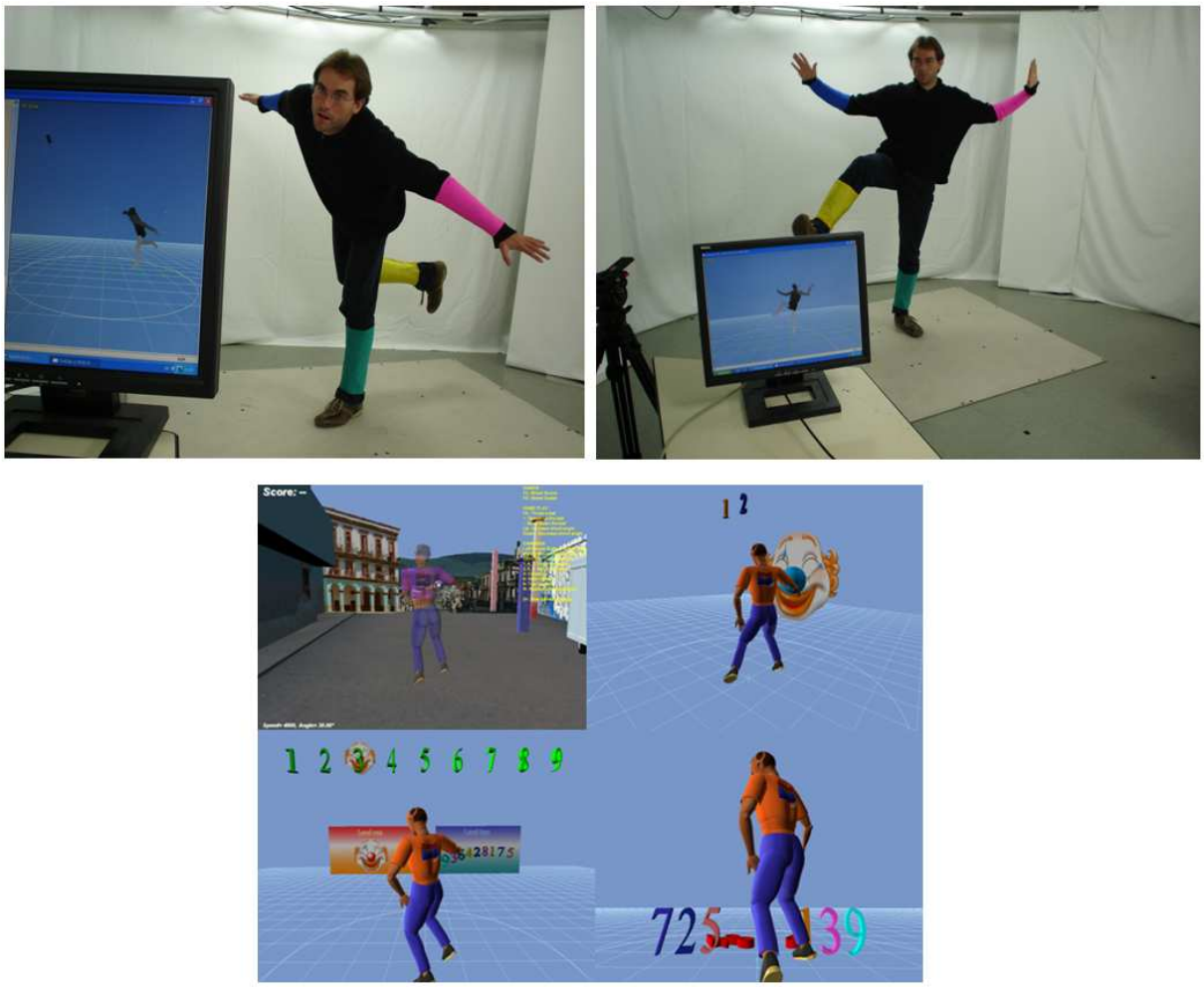


Figure 4.20. : Animation d'un personnage 3D par le biais du ***Cyberdôme***

Bien que dédié principalement à l'animation d'un personnage virtuel, ce système a pour objectif de permettre également la reproduction visuelle et en temps réel de l'apparence de l'utilisateur. En effet, les sujets de recherche actuels concernent la reconstruction temps réel du sujet filmé par des techniques de déformation de modèle 3D et de *texturing* vidéo. Dans la mesure où ces travaux sont en cours et ne concernent pas ces travaux de thèse, nous ne parlerons à partir de maintenant que de l'aspect « suivi du mouvement » du processus de capture mis en œuvre par le ***Cyberdôme***.

3.2. Aspects matériels et logiciels

Cette section présentera tout d'abord les aspects matériels du ***Cyberdôme*** (Section 3.2.1.), puis les aspects logiciels de ce système (Section 3.2.2.).

3.2.1. Aspects matériels

Chaque système vendu par la société présente une configuration matérielle différente. Dans le cadre de ces travaux de thèse, nous ne présenterons que la configuration matérielle du

Cyberdôme acquis par le ***L3i***. La Figure 4.21. regroupe un ensemble de photographies du système.

Figure 4.21. : Le ***Cyberdôme*** : aspects matériels

Le système définit, par le biais d'une armature, un espace circulaire de 4m de diamètre. Cette armature est pourvue d'une toile blanche, réfléchissant la lumière de manière diffuse. L'utilisateur, qui se trouve au centre du cercle, est donc enfermé au sein de cet espace, entouré de cette toile blanche.
Le demi-cercle se trouvant en face de l'utilisateur est dédié à son immersion au sein d'un environnement 3D. Le demi-cercle se trouvant derrière lui constitue donc le fond (*background*). Ce *background* est donc totalement connu et contrôlé (toile blanche et éclairage puissant).

L'utilisateur est filmé par 4 caméras *firewire* fixes Sony, se situant sur le demi-cercle en face de lui. Elles peuvent délivrer des images de dimensions 1024x768 pixels à un *framerate* de 25 fps (*frames* par seconde). Les données caméras brutes doivent être traitées par un filtre ***Bayer***.

L'utilisateur interagit avec l'application par le biais d'une immersion de ce dernier au sein d'un environnement 3D défini par le scénario. Deux vidéoprojecteurs se trouvent au dessus de l'utilisateur, projetant la scène 3D sur la toile du système en face de l'utilisateur, et donc sous les caméras. L'utilisateur réagit en fonction de l'état de la scène 3D projetée devant lui.

Enfin, le processus de capture mis en place au sein du ***Cyberdôme*** suppose que l'utilisateur porte des marqueurs, qui sont 4 bandes de tissu de différentes couleurs, autour des bras et des jambes.

Au niveau des ressources informatiques, le processus de capture s'effectue de manière distribuée sur 5 PCs multicoeurs, reliés en réseau. Chaque caméra est connectée à un PC

(*firewire*), sur lequel le processus de segmentation est exécuté. Les résultats de ce processus sont alors envoyés au dernier PC central, où le reste du processus de capture est effectué. Les cartes graphiques utilisées sont standards (***NVidia 8800 GTS***) mais permettent toutefois la programmation et l'exécution sur GPU (*Graphics Processing Unit*).

3.2.2. Aspects logiciels

Le logiciel ***CoolCap*** est développé par le département R&D de la société XD Productions. Ce logiciel commercial permet d'établir le processus de capture de gestuelles au sein du ***Cyberdôme***. La Figure 4.22. donne quelques exemples de capture d'écran du logiciel.

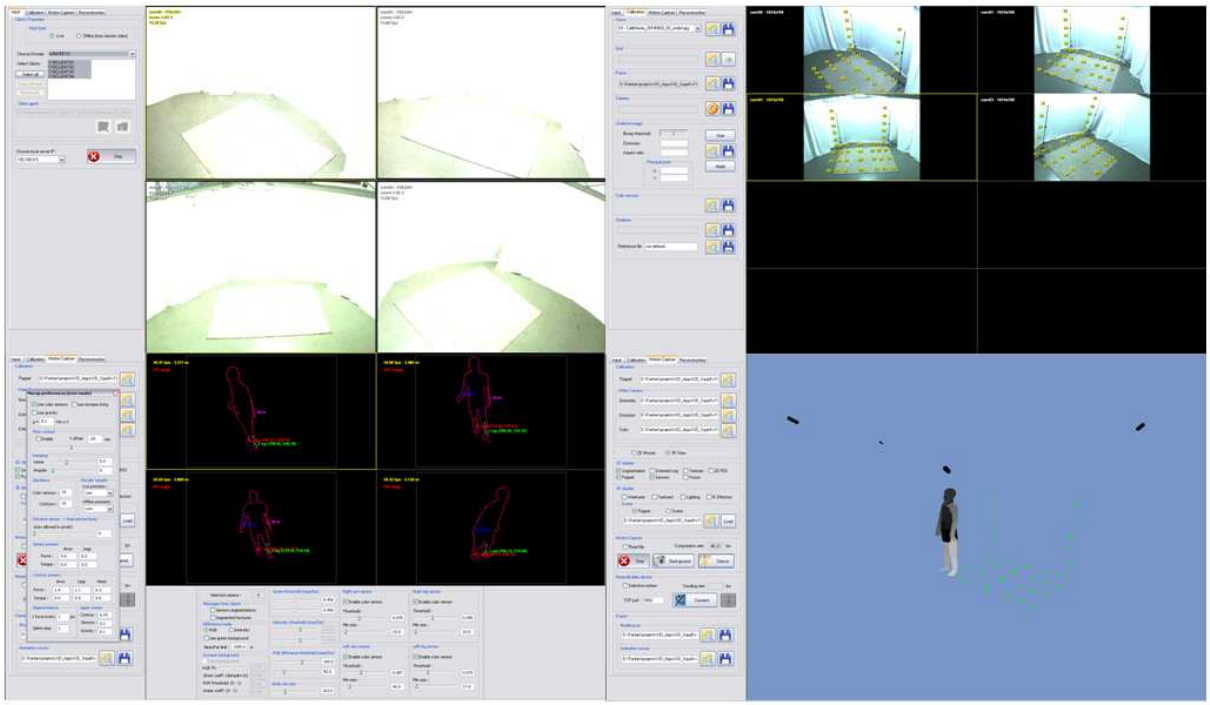


Figure 4.22. : Captures d'écran du logiciel ***CoolCap***

Ce logiciel a été développé sous ***Windows***, en C++, sous l'environnement ***Microsoft Visual Studio***. L'interface graphique a été développée par le biais de la bibliothèque MFC (***Microsoft Foundation Class***).
Parmi toutes les bibliothèques utilisées pour le bon fonctionnement du logiciel, deux sont particulièrement importantes. Tout d'abord, la librairie ***Microsoft DirectX*** permet à la fois la gestion des flux vidéo (***DirectShow***) et l'affichage 3D (***Direct3D***). De plus, nous utilisons l'accélération matérielle en programmant directement sur le GPU (*Graphics Processing Unit*) des cartes graphiques utilisées. Cette programmation se fait en ***HLSL*** (*High Level Shader Language*), langage intégré dans ***Direct3D***. L'accélération matérielle est utilisée aussi bien pour l'affichage 3D que dans un cadre GPGPU (***General-Purpose computation on Graphics Processing Units***). La deuxième bibliothèque importante est ***NVidia PhysX***. Cette bibliothèque permet la simulation des lois physiques qui régissent notre monde virtuel (collision, gravité, fluide, etc.). Cette bibliothèque nous permet bien sûr de simuler un environnement physique véritable au sein de notre scène virtuelle. Mais elle est également un

acteur majeur dans le processus de capteur de gestuelles (création des forces, dont nous parlerons dans une prochaine section).
Pour finir, nous utilisons la librairie ***CMU 1394 Digital Camera Driver***[36], développée au sein de l'Université Carnegie Mellon, pour acquérir les flux vidéo issus des caméras.

3.3. Processus de capture

Le processus de capture que nous présentons ici a été développé dans le cadre des travaux de thèse de Delamarre [Delamarre 03]. Le lecteur intéressé pourra s'y référer pour avoir plus de détails sur le processus de capture mis en œuvre au sein du ***Cyberdôme***. La Figure 4.23. illustre le processus développé dans [Delamarre 03].

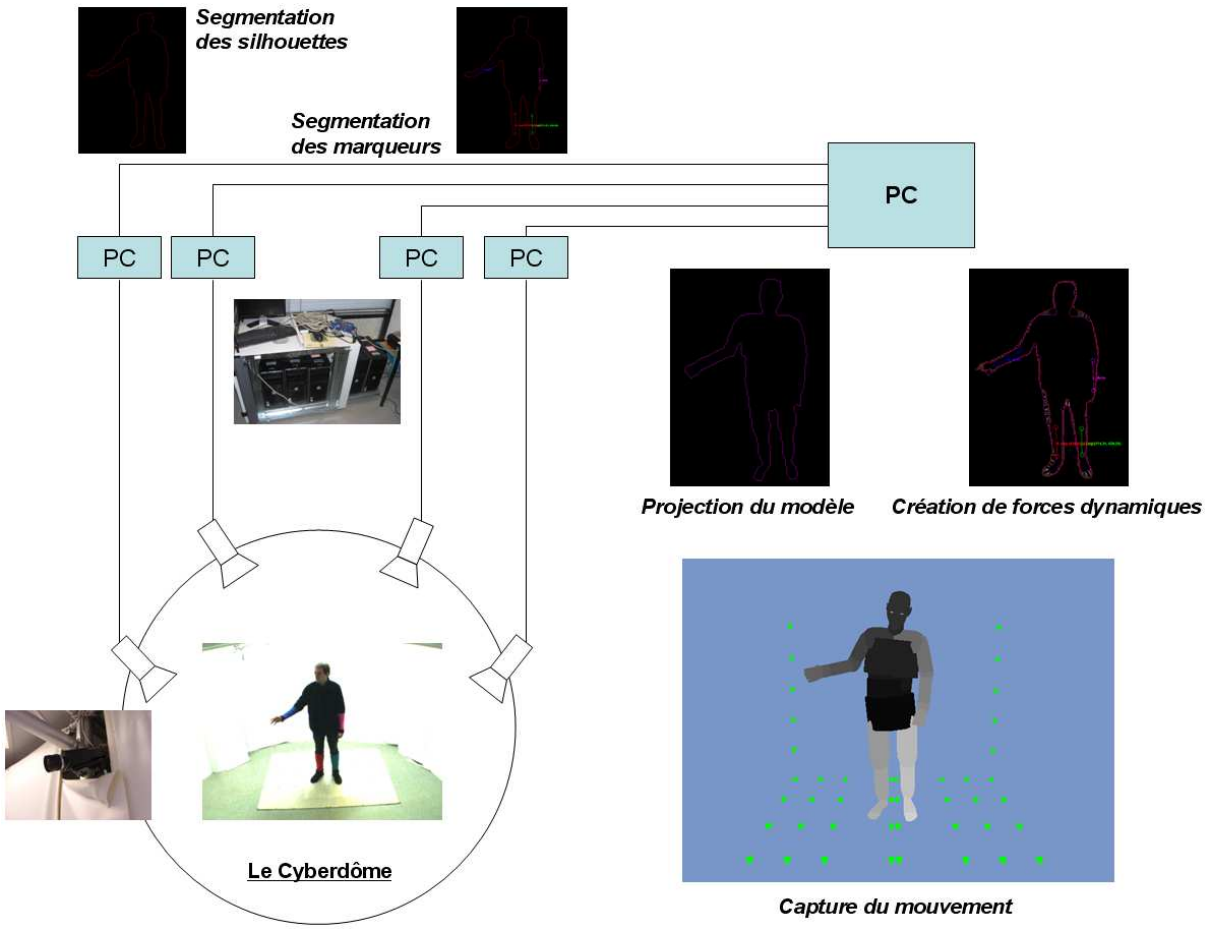


Figure 4.23. : Processus de capture [Delamarre 03]

Cette section est divisée en deux parties. La première partie (Section 3.3.1.) s'intéressera aux connaissances et hypothèses qui seront utilisées au cours du processus de capture de gestuelles, mis en œuvre par le ***Cyberdôme***. Pour cela nous référerons à la Section 2.3.8.1. Puis nous détaillerons les différentes étapes de ce processus (Section 3.3.2.), en accord avec les différentes phases décrites au cours de la Section 2.3.8.2.

[36] http://www.cs.cmu.edu/~iwan/1394/index.html

3.3.1. Connaissances & hypothèses

A l'instar de la Section 2.3.8.1., nous énumérons ici les connaissances & hypothèses relatives à l'utilisateur, à sa gestuelle, à l'environnement de capture, aux caméras et enfin au processus en lui-même, dans le contexte établi par le ***Cyberdôme***.

1. *L'utilisateur*
   A priori, le ***Cyberdôme*** est capable de capturer les gestuelles de tout type de sujet. En revanche, chaque sujet implique une description de celui-ci pour en établir un modèle (le processus de capture est *model-based*). Nous partons du principe que nous capturons une personne humaine, avec deux bras, deux jambes, une tête, etc.
   Le système capture les gestuelles de l'ensemble du corps de l'utilisateur. En revanche, le système ne capture par les gestuelles du visage (expressions), du regard et des doigts, celui-ci ne permettant pas une capture avec ce degré de granularité.
   En accord avec notre configuration matérielle, l'utilisateur se trouve *a priori* en face des caméras, dans la mesure où il regarde la scène 3D restituée en face de lui. Cela dit, rien ne l'empêche de se tourner de manière à ne plus faire face aux caméras.
   La remarque est la même concernant sa présence au sein de l'espace de capture. Dans la mesure où l'utilisateur interagit *a priori* avec l'application, il se trouve dans l'espace de capture. En revanche, il peut très bien s'en aller à n'importe quel moment.
   Quant à son apparence, l'utilisateur évolue devant un fond uni blanc, que constitue la toile entourant le système. Il doit donc s'habiller de couleur différente, voire de couleur sombre idéalement. En effet, une forte différence de couleur avec le blanc permettra une meilleure segmentation de l'utilisateur, et fiabilisera donc le processus de capture. Il est à noter que dans la mesure où le système est initialement invasif, l'utilisateur doit porter les 4 marqueurs décrits au cours de la Section 3.2.1. Nous expliquerons l'utilité de ces marqueurs dans les sections prochaines.

2. *La gestuelle*
   Vis-à-vis de notre définition de la notion de gestuelle (Section 1.2.), ce sont les mouvements de l'utilisateur qui sont capturés.
   Aucun mouvement particulier n'est plus traité qu'un autre. L'utilisateur peut effectuer n'importe quel mouvement, qu'il soit lent, rapide, etc.
   Typiquement, le système a en mémoire une configuration de modèle associée à une position particulière de l'utilisateur (face aux caméras, jambes légèrement écartées et bras écartés en croix). Cela dit, une autre position du modèle tout à fait arbitraire peut être enregistrée (une seule à la fois). Cette configuration n'est pas obligatoire mais permet d'assurer tout de même une réinitialisation du processus de capture à tout instant, sous le contrôle d'un superviseur.
   Les mouvements de l'utilisateur peuvent bien sûr générer des occultations, dans la mesure où les caméras se trouvent toutes en face de l'utilisateur (bras derrière le dos par exemple). Cependant, la configuration des caméras en demi-cercle permet tout de même de pallier ce genre de problèmes. Si les occultations sont trop importantes, il est toujours possible de modifier différemment la configuration des caméras.

3. *L'environnement de capture*
   L'environnement de capture est un espace circulaire et fermé, délimité par une toile blanche unie.
   L'éclairage est contrôlé et puissant, pour atténuer au maximum les ombres que génère l'utilisateur. La présence d'ombres cause en effet une mauvaise segmentation de l'utilisateur, diminuant l'exactitude et la précision du processus de capture.

4. *Les caméras*
   4 caméras fixes sont disposées en demi-cercle en face de l'utilisateur.
   Leurs caractéristiques sont soit introduites par le concepteur (résolution, taille des images, *frame rate*, etc.), soit calculées lors d'une phase de calibrage (focale, pose, etc.).

5. *Le processus*
   Le processus de capture mis en œuvre au sein du ***Cyberdôme*** permet le suivi et la codification temps réel des mouvements (en accord avec notre définition au sein d'une gestuelle, voir Section 1.2.) de l'utilisateur.
   Comme nous l'avons mentionné précédemment, le processus est temps réel. Mais il peut également s'appliquer à des séquences de mouvements enregistrées.
   L'approche est basée modèle. Nous détaillerons ce modèle dans une prochaine section.
   Le point de départ de la capture est l'ensemble des silhouettes de l'utilisateur, extraites de chaque point de vue. Ce sont également les textures vidéo comprises dans ces silhouettes, par le biais desquelles les marqueurs que doit porter l'utilisateur sont détectés (grâce à leurs couleurs).

3.3.2. Etapes

En accord avec les différentes phases du processus de capture, décrites au cours de la Section 2.3.8.2., nous allons décrire les différentes étapes du processus de capture de gestuelles du ***Cyberdôme*** : **initialisation du processus** (Section 3.3.2.2.) ; **suivi** (Section 3.3.2.3.) ; **estimation du positionnement du modèle** (Section 3.3.2.4.) ; **caractérisation de la gestuelle observée** (Section 3.3.2.5.). Le ***Cyberdôme***, tel qu'il a été conçu initialement, n'est dédié qu'à **l'animation** et **la reconstruction**. Le processus de capture mis en œuvre par ce système ne comprend donc pas de dernière étape relative à l'interprétation de la gestuelle de l'utilisateur.
De manière à être clair dans nos explications, il nous est cependant nécessaire d'ouvrir tout d'abord une brève parenthèse sur le modèle 3D utilisé au cours du processus de capture de gestuelles (Section 3.3.2.1.).

3.3.2.1. Modèle

Le modèle (voir Fig. 4.24.), utilisé par le processus de capture, est divisé en **26 parties** ou **primitives**, chacune associée à une partie du corps de l'utilisateur (la tête, l'avant bras, le bassin, etc.). Chaque primitive est d'ailleurs identifiée **sémantiquement** par le nom de la partie du corps de l'utilisateur qu'elle représente. Chaque primitive est une **forme simple** (sphère, ellipsoïde ou parallélépipède rectangle, appelé encore *box*), chacune étant

caractérisée par un ensemble de données telles que la longueur, la largeur, la hauteur, le rayon, etc. La forme d'une primitive est bien sûr choisie vis-à-vis de la géométrie de la partie du corps de l'utilisateur à laquelle la primitive est associée. Ces primitives sont restituées visuellement sous la forme de *meshes* (ensemble géométrique en 3D, composé d'un nombre variable de triangles). Dans le cadre de ces travaux de thèse, ces primitives sont **rigides**, ne se déformant donc pas au cours du processus de capture.

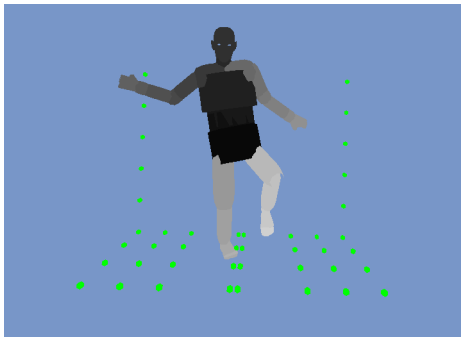


Figure 4.24. : Modèle 3D utilisé au cours du processus de capture

Le modèle est **hiérarchisé**, chaque primitive étant structurée et positionnée **à partir de la primitive racine** (*root*) que constitue le bassin. Chaque primitive fille est donc liée à une primitive mère (sauf la racine) par un **joint**, contrainte cinématique symbolisant une articulation réelle entre deux parties du corps de l'utilisateur. Le nombre de degrés de liberté d'un joint dépend de l'articulation qu'il représente. Au maximum égal à 6 (3 translations et 3 rotations), certains degrés de liberté peuvent être supprimés, ou seulement limités dans leur portée (au niveau des rotations permises notamment).

Le modèle évolue au sein d'un univers virtuel, au sein duquel les lois physiques réelles sont en vigueur. Cet univers est interactif, le modèle pouvant donc interagir avec les éléments interactifs le composant. Le modèle est animé par les gestuelles de l'utilisateur (estimation des poses de chaque primitive au cours du processus de capture). Chaque primitive du modèle étant identifiée sémantiquement, il est possible de codifier le mouvement capturé, pour une partie du corps de l'utilisateur particulière, vis-à-vis d'un repère 3D fixe de référence, associé à la scène virtuelle.

3.3.2.2. Initialisation du processus

Le processus de capture du ***Cyberdôme*** nécessite **3 étapes de calibrage**, effectuées avant la capture des gestuelles à proprement parler. Ces 3 étapes permettent la construction des connaissances nécessaires au processus pour son bon fonctionnement.

La première étape de calibrage, appelée **le calibrage géométrique**, consiste à estimer, d'une part, **les distorsions des lentilles des caméras** et, d'autre part, **les paramètres internes et externes de ces dernières**.
La lentille d'une caméra présente une distorsion géométrique, qui augmente du centre de la lentille aux bords de celle-ci. Ceci se traduit visuellement par le fait qu'une ligne droite filmée

n'est pas 'droite', mais courbée, cela d'autant plus si elle se trouve au bord de l'image vidéo. La première étape consiste donc à estimer cette distorsion, par le biais d'une mire (ensemble de droites parallèles), de manière à corriger celle-ci au cours du processus de capture, et ce pour chaque flux vidéo (voir Fig. 4.25.).

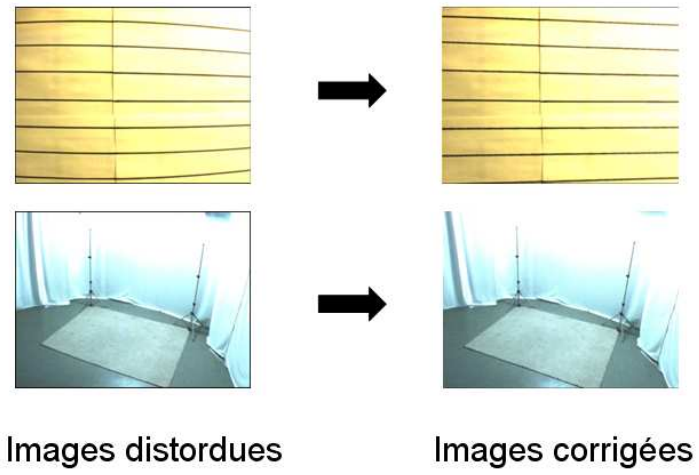


Figure 4.25. : Correction de la distorsion liée aux lentilles des caméras

Le suivi de la gestuelle se fait par le biais d'un modèle 3D, évoluant au sein d'une scène virtuelle. Il est alors nécessaire de mettre en correspondance cet environnement virtuel avec les images vidéo en 2D. Cette mise en correspondance se fait par le biais de matrices caméras, regroupant à la fois les paramètres internes des caméras (focale, coordonnées du centre optique, etc.) et leurs paramètres externes (poses des caméras au sein de l'environnement de capture). Ces matrices sont estimées au cours de cette première étape de calibrage (repérage manuelle de points connus au sein de l'environnement de capture sur chaque image vidéo, comme le montre la Figure 4.26.).

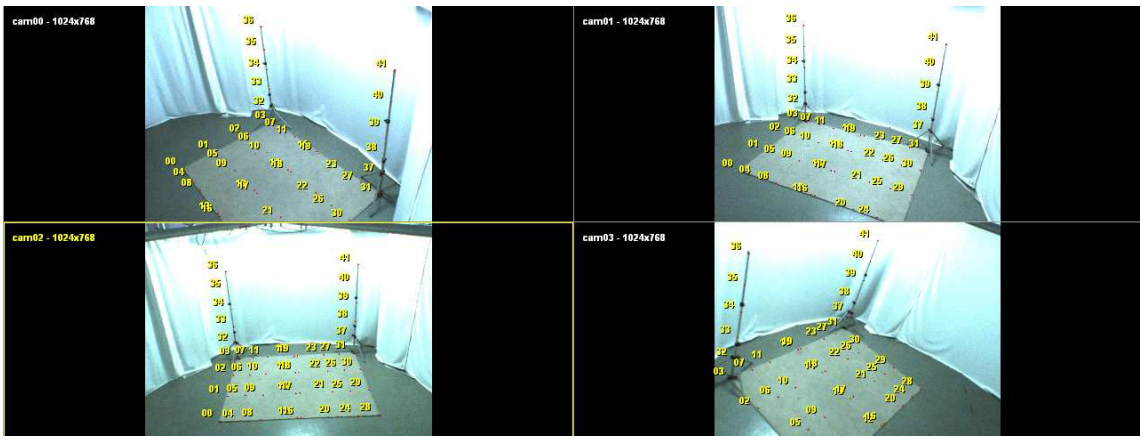


Figure 4.26. : Calcul des matrices caméras

Le principe de la capture consiste à mettre en correspondance, pour chaque point de vue (4 caméras), la silhouette du sujet extraite des images vidéo, avec la silhouette du modèle projetée. Plus ces silhouettes sont similaires, plus elles sont mises en correspondance rapidement et facilement. Pour cela, il faut donc que les dimensions des différentes primitives du modèle soient similaires à celles des parties du corps de l'utilisateur associées. C'est au cours de **cette deuxième étape de calibrage** que les **différentes dimensions des primitives sont ajustées** interactivement, **pour que les morphologies du modèle et de l'utilisateur soient identiques** (voir Fig. 4.27.).

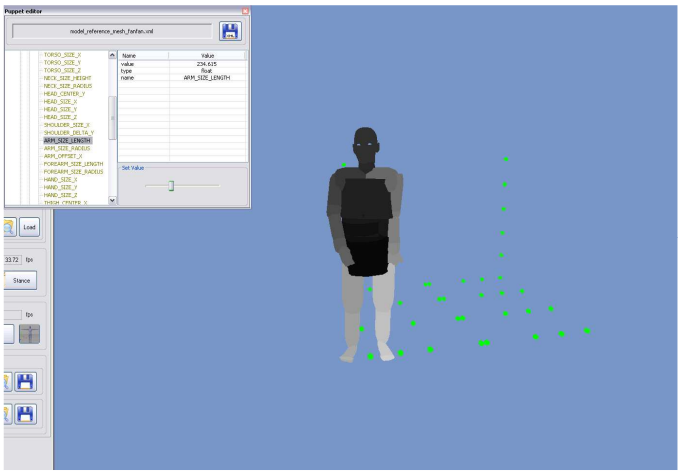


Figure 4.27. : Ajustement manuel et interactif des dimensions des différentes primitives

Le processus de capture est invasif : l'utilisateur doit porter 4 marqueurs de couleur, un pour chaque bras et un pour chaque jambe. Les couleurs de ces marqueurs sont différentes les unes avec les autres et se détachent généralement très nettement des vêtements de l'utilisateur et du fond de l'environnement de capture. Les marqueurs colorés sont utilisés pour distinguer les bras des jambes de l'utilisateur et pour savoir de quel côté du plan médian, ou sagittal, un membre se situe (à droite ou à gauche). Ces distinctions sont immédiates et permettent de relier directement et sans ambiguïtés les membres de l'utilisateur avec les membres du modèle.
Le système sait quel marqueur coloré correspond à quel membre de l'utilisateur (connaissance introduite par le concepteur). De plus, l'éclairage étant contrôlé, la couleur des marqueurs ne varient que très faiblement au cours du processus de capture. **La 3ème étape de calibrage** (voir Fig. 4.28.) consiste donc à détecter manuellement ces marqueurs sur chaque point de vue, de manière à estimer et enregistrer des intervalles particuliers de valeurs autour de leurs attributs couleur (teinte, saturation, lumière).

Figure 4.28. : Calibrage couleur

Ne consistant pas à proprement parler en une étape supplémentaire de calibrage**, une dernière opération** rapide doit être effectuée avant l'exécution du processus de capture. Celle-ci consiste, pour chaque point de vue, à **enregistrer une image de l'environnement de capture**, **sans l'utilisateur au sein de celui-ci**. Ce fond enregistré est utilisé au cours du processus d'extraction des silhouettes de l'utilisateur (image de l'utilisateur), point de départ du processus de capture mis en œuvre au sein du ***Cyberdôme***.

3.3.2.3. Suivi

Au cours du processus de capture, la phase de suivi est relativement simple. Elle consiste à **extraire**, pour chaque point de vue et pour chaque *frame* (4 flux vidéo synchronisés), **la silhouette de l'utilisateur**, **ainsi que la texture vidéo délimitée par celle-ci**. L'ensemble des silhouettes de l'utilisateur et des textures vidéo comprises au sein de celles-ci, pour chaque *frame*, constitue **l'image de l'utilisateur**.
D'une *frame* à une autre, aucune relation temporelle entre les silhouettes et les textures vidéo n'est calculée ou utilisée. En revanche, l'extraction des silhouettes de l'utilisateur, ainsi que les textures vidéo comprises au sein de celles-ci, nécessite **une étape de différenciation du *background* avec le *foreground***.

Tout d'abord, pour chaque point de vue et pour chaque *frame*, un algorithme simple de soustraction d'images (fond enregistré – image vidéo courante = image différence) permet l'extraction de l'utilisateur du reste de la scène réelle (le fond enregistré ne comprend pas l'utilisateur). De manière à supprimer les quelques artefacts qui peuvent être générés par cette soustraction, un seuillage sur les valeurs RGB des pixels de l'image différence est appliquée. Ainsi, pour chaque point de vue et pour chaque *frame*, l'utilisateur est distingué du fond grâce aux images différences.
Par la suite, pour chaque image différence, une chaîne 8-connexité de ***Freeman*** [Feldman 92] code la silhouette à proprement parler de l'utilisateur. Cette chaîne est calculée par le biais d'un algorithme particulier, qui caractérise, rapidement et de manière continue, le contour de l'utilisateur dans l'image différence. A partir d'un pixel initial choisi arbitrairement, cet

algorithme parcourt dans le sens des aiguilles d'une montre et code le contour à partir pixel après pixel, jusqu'à retrouver le pixel initial. Le contour est alors décrit par une séquence ordonnée de vecteurs. Un vecteur traduit pour un pixel donné la direction relative, codé de 0 à 7, du pixel suivant sur le contour (voir Fig. 4.29.).

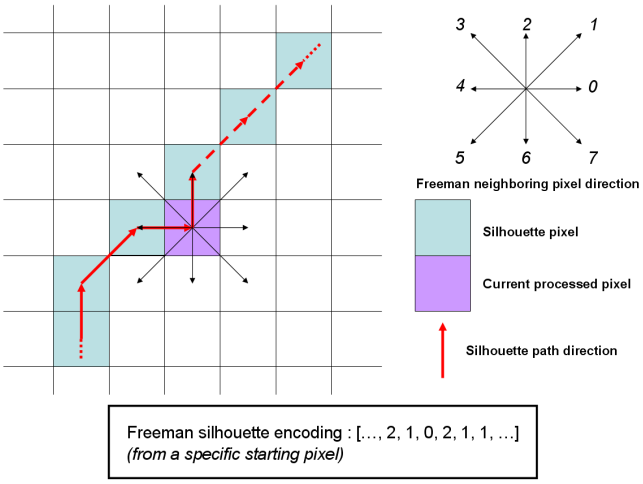


Figure 4.29. : Codification de la silhouette 2D par une chaîne 8-connexité de ***Freeman***

Ainsi, pour chaque point de vue et pour chaque *frame*, la silhouette de l'utilisateur est représenté sous la forme d'une suite de chiffres allant de 0 à 7. Ce format permet une représentation efficace, rapidement et facilement transmissible via réseau, et permettant la reconstruction fidèle de la silhouette de l'utilisateur (voir Fig. 4.30.).

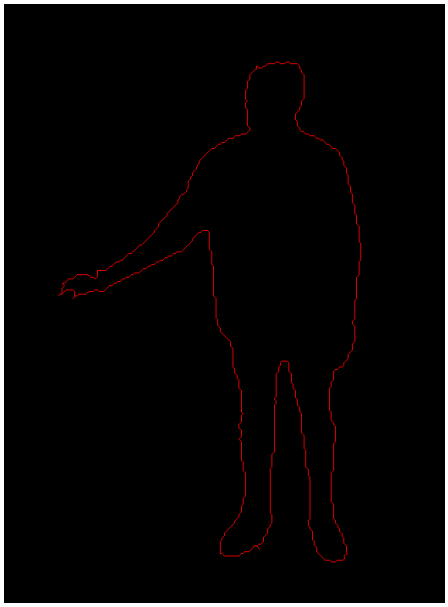


Figure 4.30. : Silhouette de l'utilisateur segmentée

Aux silhouettes de l'utilisateur sont rajoutées les textures vidéo comprises au sein de ces silhouettes, facilement extraites par le biais de ces dernières. Ces textures vidéo, grâce à la deuxième phase de calibrage concernant les marqueurs de couleur, sont la source de la détection des ces derniers. Cette détection se traduit alors par un ensemble de régions 2D, pour chaque point de vue, correspondant aux marqueurs. Ces régions sont caractérisées par un centre, une hauteur, une largeur, un axe, etc. (voir Fig. 4.31.).

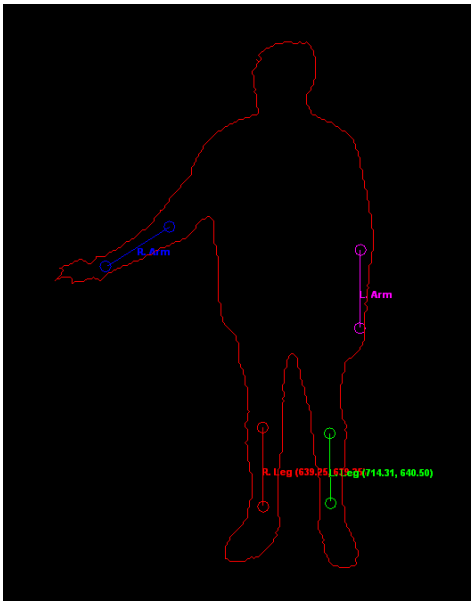


Figure 4.31. : Silhouette et marqueurs

3.3.2.4. Estimation du positionnement du modèle

Nous allons présenter ici la méthode utilisée au cours du processus de capture **pour estimer la pose du modèle au cours du temps**. Cette méthode, permettant **une capture de gestuelles temps réel** par le biais de plusieurs flux caméras, a été développée par [Delamarre 03]. Ces travaux présentent une approche originale permettant **de minimiser**, **par le biais de forces dynamiques**, **la distance entre la position réelle du sujet filmé et de son modèle 3D**.

Dans le cas du ***Cyberdôme***, l'utilisateur est caractérisé principalement par ses **silhouettes**, sur chaque point de vue et pour chaque *frame*. Il est donc logique que **l'image du modèle** soit l'ensemble des silhouettes 2D de ce dernier, estimées sur chaque point de vue et pour chaque *frame*. Ce calcul est rendu possible grâce à la première phase de calibrage permettant d'estimer les matrices caméras, permettant de faire le lien mathématique entre un pixel à l'écran et un point 3D de l'environnement virtuel mis en jeu par le processus de capture. Ainsi, sur chaque point de vue, la silhouette du modèle projeté sur le plan image est calculée. Pour chaque nouvelle *frame*, l'image du modèle est l'ensemble des silhouettes du modèle, dont le positionnement a été calculé à la *frame* précédente.
Les silhouettes de l'utilisateur et du modèle permettent de calculer **la force** et **le couple** à appliquer sur chaque primitive composant le modèle 3D, le but étant que les silhouettes sur chaque plan image soient aussi confondues que possible.
Tout d'abord, sur chaque plan image, des forces, « ressorts » ou « normales » et obéissant aux lois de la physique, sont calculées. Les forces « ressorts » permettent de rapprocher les

silhouettes de l'utilisateur et du modèle et les forces « normales » poussent la silhouette du modèle au sein de la silhouette de l'utilisateur. Chaque force relie un pixel de la silhouette de l'utilisateur avec le pixel le plus proche appartenant à la silhouette du modèle, et est associée à une primitive particulière du modèle (voir Fig. 4.32.).

Figure 4.32. : Calcul des forces dynamiques

A partir de là, pour chaque point de vue et pour chaque primitive, sont calculées les force et couple 2D à appliquer. Cette force et ce couple sont estimés à partir des forces normales et ressorts appliquées sur la primitive. De plus, un facteur d'échelle intervient dans le calcul, tenant compte ainsi de la distance réelle entre la caméra et la primitive considérée. Enfin survient un changement de repère pour se placer dans un repère de référence 2D à partir du repère 2D défini par le plan image.
Enfin, la force et le couple 3D à appliquer sur chaque primitive est calculée, sous plusieurs hypothèses que [Delamarre 03] justifie (chapitres 12 & 13 de la thèse). L'ensemble des forces et des couples 3D est alors appliqué sur le modèle par itérations successives. Le modèle est finalement positionné, lorsque la somme des normes des forces est inférieure à un seuil défini par le concepteur (voir Fig. 4.33.).

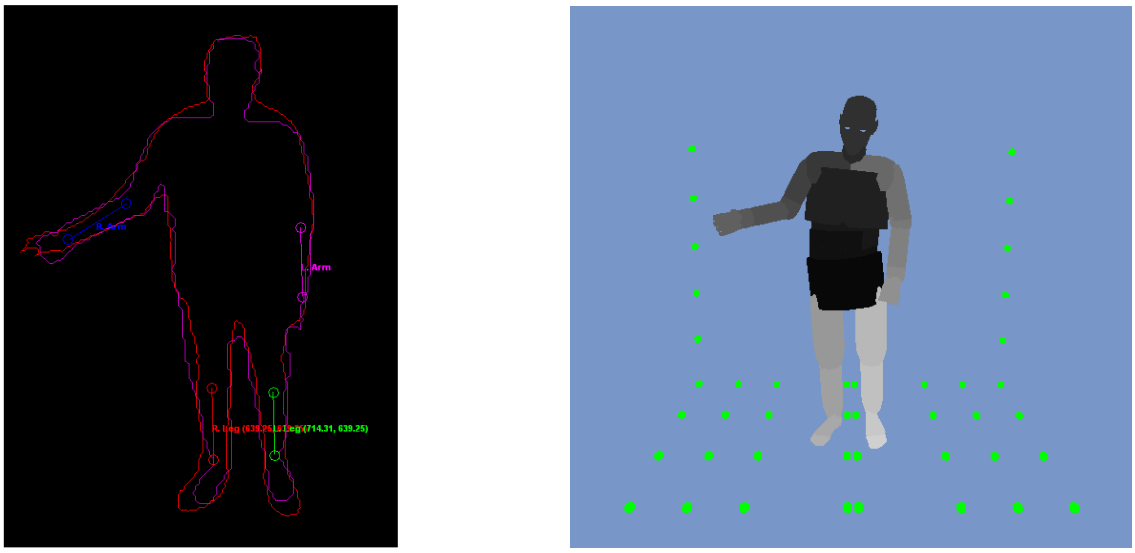


Figure 4.33. : Positionnement du modèle

Nous pouvons remarquer ici que la phase de calibrage, permettant l'ajustement interactif des dimensions du modèle vis-à-vis de l'utilisateur, permet que les silhouettes de l'utilisateur et du modèle soient similaires, impliquant ainsi la création de forces faibles et rapidement calculées. Plus les dimensions du modèle seront proches de celles de l'utilisateur, plus la convergence du modèle vers la bonne pose sera rapide.
A noter qu'il aurait été possible de calculer les forces et couples directement en 3D, en confrontant le modèle avec une reconstruction 3D de l'utilisateur. Cependant, comme le remarque et le justifie [Delamarre 03], cette approche serait au mieux aussi bonne que celle des forces 2D, et se révèle en pratique plus difficile à mettre en œuvre.

L'utilisation de marqueurs au cours du processus permet à la fois de mettre en correspondance directement les primitives 'bras' et 'jambes' (droite et gauche) du modèle, avec les zones images, correspondant à ces marqueurs, détectées pour chaque nouvelle *frame*.
Deux méthodes permettent le calcul des forces et couples des primitives du modèle associées aux marqueurs. La première méthode est la même que celle employée pour calculer les forces et couples vis-à-vis des silhouettes de l'utilisateur et du modèle. Autrement dit, les forces sont tout d'abord calculées en 2D puis en 3D. La deuxième méthode consiste à calculer directement les forces et couples en 3D. Les extrémités des marqueurs, repérées sur chacun des points de vue, sont estimées dans la scène virtuelle. Il est alors possible de calculer les forces et couples reliant les extrémités des marqueurs avec celles des primitives associées.

3.3.2.5. Caractérisation

Le calcul des poses des différentes primitives composant le modèle 3D permet **la caractérisation de la gestuelle** au sein de la scène virtuelle, relativement à un repère 3D de référence associée à cette dernière. Le logiciel ***CoolCap*** de la société permet la caractérisation de la gestuelle de l'utilisateur sous deux formats : un format propriétaire ***XTR*** et le format standard ***BVH***. La seule différence de codification existant entre ces deux formats est que le premier repère les primitives du modèle de manière absolue par rapport au repère de la scène

virtuelle, alors que le second estime les positions des primitives les unes à partir des autres, décrivant alors une hiérarchie à partir d'un élément racine (le 'bassin' dans le cas échéant).

3.4. Evaluation du système

En accord avec les Sections 3.3.2.3. et 3.3.2.4. de ce chapitre, le processus de capture, mis en jeu par le ***Cyberdôme***, peut se résumer par le schéma suivant :

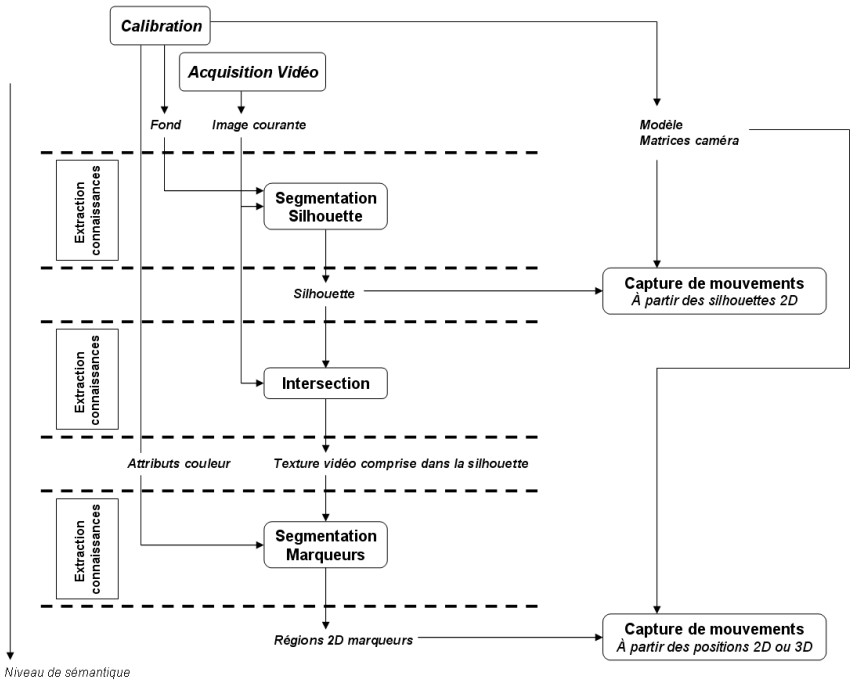


Figure 4.34. : Résumé du processus de capture de gestuelles

Le processus mis en œuvre au sein du ***Cyberdôme*** permet **une capture temps réel des gestuelles de l'utilisateur** (25 images/seconde pour un *frame rate* de caméra similaire). Il permet aussi bien de capturer les gestuelles lentes que rapides de l'utilisateur.
L'approche utilisée pour estimer les poses du modèle, couplée avec **l'utilisation des marqueurs**, est efficace, robuste, rapide (une dizaine d'itérations permet la convergence du modèle) et permet de résoudre les occultations. Bien que mathématiquement complexe, l'algorithme de calcul des forces peut intégrer facilement d'autres types de forces que les forces ressorts et normales.

D'un point de vue matériel, le nombre de caméras nécessaires au système et l'utilisation de marqueurs sont justifiés principalement par le fait que les silhouettes 2D de l'utilisateur constituent le point de départ du processus de capture.
Une silhouette 2D seule, c'est-à-dire non couplée avec d'autres caractéristiques images, peut générer des ambiguïtés et des incertitudes quant à son interprétation. En effet, une silhouette peut avoir **une forme similaire pour différentes poses de l'utilisateur**. Par exemple, la différenciation des silhouettes sera difficile si l'utilisateur est face à une caméra, les bras le long du corps et les jambes serrées l'une contre l'autre, ou s'il est toujours face à cette même

caméra, les jambes serrées, mais cette fois un des deux bras en avant. De plus, une silhouette seule ne permettra pas de dire si l'utilisateur est face à la caméra ou s'il regarde dans sa direction, ni de savoir où est sa droite et sa gauche (voir Fig. 4.35.).

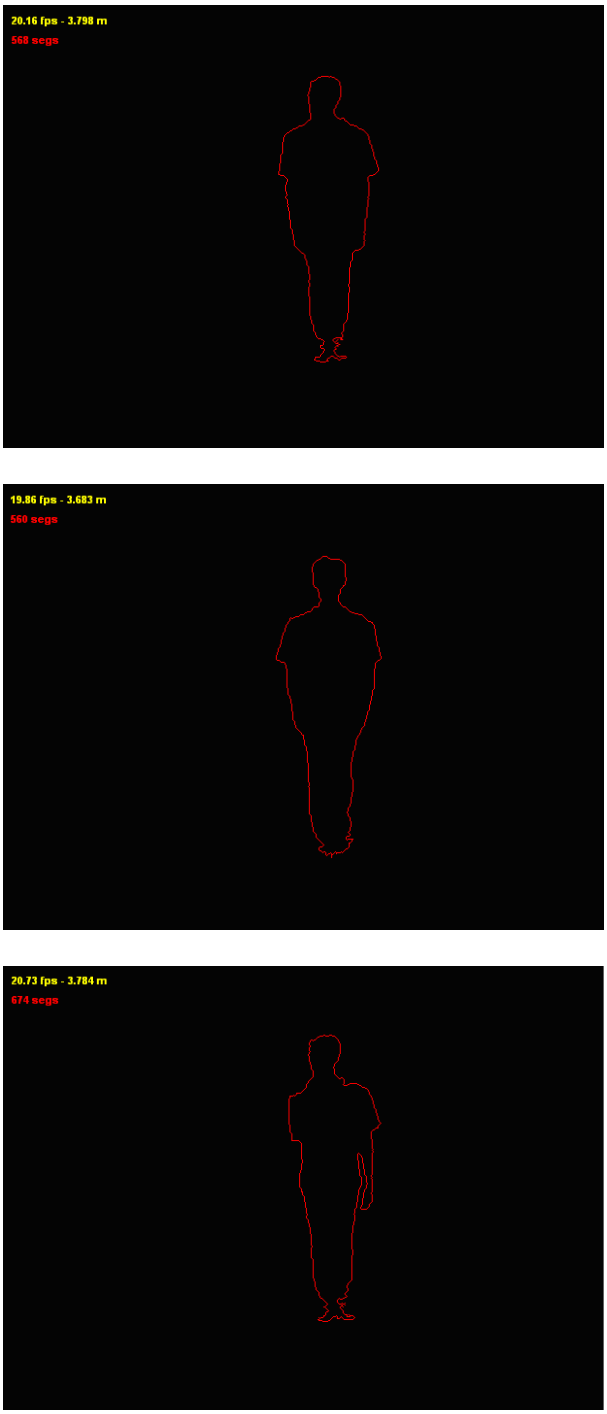


Figure 4.35. : Exemples de silhouettes ambigües : de haut en bas, l'utilisateur est de face, l'utilisateur est de dos, et l'utilisateur lève un bras devant lui

D'une part, plusieurs caméras, disposées autour de l'utilisateur, peuvent permettre la détection des parties du corps non visibles pour un point de vue donné. Il est à noter que l'observation

de l'utilisateur sous plusieurs points de vue peut permettre également la résolution des occultations, i.e. le fait qu'une partie du corps de l'utilisateur en cache une autre. D'autre part, les marqueurs permettent de caractériser immédiatement les bras des jambes et la gauche de la droite. Le système sait en effet *a priori* quelle couleur est portée par quel membre et le processus de capture serait évidemment beaucoup moins fiables s'il s'agissait de mettre en correspondance uniquement les silhouettes de l'utilisateur et du modèle. Les marqueurs permettent également d'accélérer le processus de capture et de calculer plus rapidement les poses des primitives composant le modèle. En effet, de par le fait qu'il n'existe aucune ambiguïté entre une couleur et une primitive particulière, des forces et couples plus importants peuvent être appliqués, permettant d'atteindre un état d'équilibre plus rapidement.

En accord avec nos objectifs, nous désirons supprimer deux contraintes que présente le système : le contrôle total de la scène observée et l'utilisation de marqueurs. De plus, nous voulons étendre le processus à une capture plus générale de l'activité globale au sein de la scène réelle. Les modifications que nous avons apportées au système sont l'objet de la prochaine section.

3.5. Modifications apportées au système

Cette section présente **les solutions que nous avons envisagées**, dans le but d'améliorer le processus de capture, mis en place au sein du ***Cyberdôme***, en accord avec les objectifs de ces travaux de thèse.

Dans le cadre des travaux de recherche de l'équipe ***ImagIN***, l'utilisateur interagit avec le système par le biais d'interfaces **non invasives** et **ne nécessitant pas le contrôle total de la scène observée** au sein de laquelle il évolue. C'est pourquoi le système de capture considéré ici présente deux contraintes majeures vis-à-vis de nos objectifs : il implique **un contrôle total de l'environnement de capture** (éclairage puissant, fond connu, uni et statique) et demande à l'utilisateur **de porter un équipement particulier** (marqueurs de couleur sur les bras et les jambes). Nous avons donc souhaité, avant toute chose, nous libérer au maximum de ces deux contraintes.
Cette thèse s'effectuant dans le cadre d'une convention ***CIFRE***, le système utilisé est commercial et développé par une compagnie privée. Il n'était donc pas possible ni question de modifier les grandes lignes suivies par le processus de capture de gestuelles. De plus, ce processus est tout à fait efficace et vérifie plusieurs contraintes, telles qu'une capture temps réel ainsi qu'une mise en œuvre rapide. Il nous fallait donc trouver des solutions pertinentes, s'appliquant de manière locale au cours du processus de capture, pour améliorer ce dernier, en accord avec nos besoins et objectifs.
Enfin, les modifications que nous avons apportées au système sont hors tout contexte applicatif particulier. Il ne s'agit ici que du système de capture développé au sein de la société XD Productions. Les améliorations que nous allons présenter ici ont cependant été développées au cours de ces travaux de thèse. La Figure 4.36. illustre les modifications induites par nos travaux sur le processus de capture.

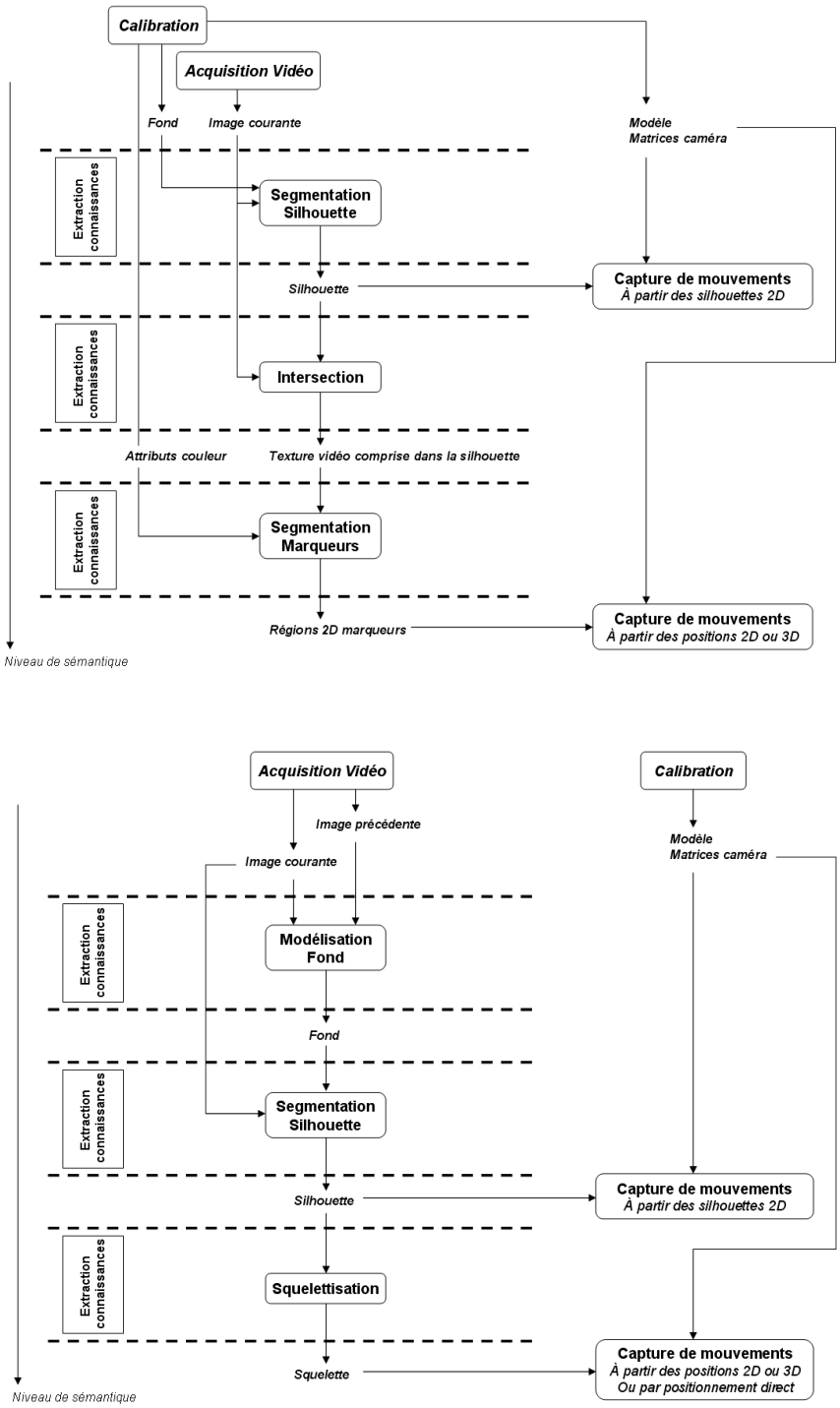


Figure 4.36. : En haut, processus de capture initial. En bas, processus modifié dans ces travaux de thèse

Un des objectifs de ces travaux de thèse est d'exécuter le processus de capture de mouvements au sein d'**un environnement peu contrôlé**, aussi bien du point de vue de l'éclairage que du fond de la scène. De plus, le processus de capture doit être robuste face **au dynamisme possible de cet environnement**.
Or, avec le processus tel qu'il est décrit précédemment, il n'est pas possible d'atteindre ce but. L'algorithme de segmentation des silhouettes, trop simple, se révèle inefficace face à un dynamisme potentiel de la scène, même léger. Les apparitions, disparitions et déplacement

d'objets, l'hétérogénéité du fond, les changements d'éclairage au cours de la capture, etc. doivent en effet être rapidement pris en compte au cours du processus. Dans le cas contraire, l'algorithme génère alors un ensemble de silhouettes perturbées, dont les artefacts génèrent la création de forces 2D indésirables, résultant en une pose finale du modèle 3D imprécise, en plus d'un tremblement persistant du aux propriétés oscillatoires des forces dynamiques. De plus, l'environnement est totalement clos, empêchant toute présence et observation autres que celles de l'utilisateur. Enfin, l'éclairage puissant, nécessaire pour diminuer les ombres, rend impossible toute projection de la scène virtuelle et donc toute immersion de l'utilisateur au sein de celui-ci.
Il nous a donc fallu envisager une nouvelle approche pour alléger au maximum ces contraintes. Avant d'entreprendre une modification de l'algorithme de segmentation des silhouettes, il nous est apparu naturel, aux vues de nos objectifs, de nous diriger vers **une modélisation du fond de manière dynamique**, c'est-à-dire au cours du processus de capture de gestuelles, plutôt que d'un simple et unique enregistrement, antérieurement à ce dernier. La Section 3.4.1. présentera un état de l'art sur cette problématique et exposera la solution que nous avons adoptée. Comme le montre la Figure 4.36., nous somme intervenus à un niveau de sémantique antérieur à celui de la segmentation des silhouettes.

En accord avec les caractéristiques voulues du système présenté dans ces travaux, l'interface avec l'utilisateur ne doit pas contraindre ce dernier avec **un équipement supplémentaire à porter**.
Il est clair que l'interface initiale présente une transparence importante. Les marqueurs sont des bandes tissus peu encombrantes, légères, faciles à porter et à enlever, passives (ne générant pas de signaux) et contraignant peu les mouvements de l'utilisateur. Cela dit, ces marqueurs obligent l'utilisateur à manipuler un équipement matériel supplémentaire, ce qui est contraire à la philosophie de recherche de l'équipe ***ImagIN***.
C'est pourquoi nous abordons au cours de la Section 3.4.2. la solution que nous avons envisagée pour nous libérer au maximum de cette contrainte. La suppression des connaissances apportées par les marqueurs de couleur ajoute bien sûr au processus initial de capture de mouvements de nouvelles incertitudes et possibilités d'erreurs, liées aux ambiguïtés des silhouettes. Il nous fallait donc trouver un ensemble de marqueurs **implicites**, extraits uniquement à partir de l'observation de la scène, pour remplacer les marqueurs explicites que sont les bandes de tissu colorées. Ces marqueurs implicites constituent **un nouveau support pour la création de forces dynamiques supplémentaires**, à l'image des marqueurs de couleur. En ne prenant en compte que les silhouettes de l'utilisateur, un moyen intuitif de déterminer ce nouveau support fut de **squelettiser** ces silhouettes. Un squelette de l'utilisateur permet de connaître les extrémités des différents membres, l'orientation de ces derniers, les différentes articulations, etc. chaque élément du squelette rendant alors possible le positionnement de la primitive associé à celui-ci, du modèle.

Nous verrons dans le prochain chapitre comment l'introduction de la gestion de contexte de au sein de notre système et la mise en place de mécanismes adaptatifs au cours de l'interactivité a permis, d'une part, **de compléter nos solutions** présentées au cours de cette section et, d'autre part, **d'étendre le processus de capture à une capture plus globale de la scène réelle**.

3.5.1. Environnement de capture contrôlé

La modélisation du fond fait partie intégrante de la 2ème étape du processus de capture : **le suivi** (ou *tracking*). Cette modélisation permet **la segmentation**, comprendre la séparation, du sujet d'intérêt ou de l'avant-plan de la scène (*foreground*), ici l'utilisateur, du reste de la scène, du fond ou encore arrière-plan (*background*). [Moeslund & al. 06] intitule cette étape *figure-ground segmentation*.
Dans notre système, cette étape du processus est cruciale car elle permet **la segmentation et la codification des silhouettes de l'utilisateur**. Une mauvaise modélisation du fond résulte en une mauvaise segmentation des silhouettes de l'utilisateur et donc en une mauvaise estimation de la pose du modèle 3D. La mauvaise modélisation du fond entraine donc une mauvaise capture de mouvements.
D'après nos objectifs, **l'environnement de capture ne doit pas être contrôlé** : dynamisme de la scène, éclairage changeant, etc. C'est pourquoi notre modélisation **temps réel** du fond doit être **dynamique** et **adaptative**, au cours du processus de capture.

La Section 3.4.1.1. proposera un état de l'art concernant la problématique de la modélisation dynamique et adaptative du fond. Après avoir discuté et comparé les différentes méthodes au cours de la Section 3.4.1.2., nous présenterons notre contribution (Section 3.4.1.3.).

3.5.1.1. Etat de l'art

Nous proposons ici **un état de l'art** concernant **la modélisation de scène**, majoritairement **dynamiques**, **en temps réel**, **dans le but d'en extraire le sujet d'intérêt**. De nombreuses avancées ont été faites dans le domaine, à partir du moment où des systèmes de surveillance automatiques, aussi bien de scènes intérieures qu'extérieures, ont été développés.
Il existe dans la littérature de nombreux états de l'art concernant cette problématique. Nous appuyons le notre sur les travaux suivants : [Moeslund & al. 06, Porikli 06, Bouwmans & al. 08, Elhabian & al. 08].
La section suivante sera générale et s'articulera autour des concepts englobés par cette problématique : qu'est ce qu'un modèle de fond ? comment les différentes approches peuvent être classées (classification de [Bouwmans & al. 08]) ? quels sont les aspects de cette problématique que le concepteur doit prendre en compte, avant d'adopter ou d'élaborer une méthode ? et enfin, quels sont les processus à élaborer pour correctement générer et gérer un modèle de fond ? La Section 3.4.1.1.2. exposera alors les approches les plus citées et intéressantes de la littérature, selon la classification de [Bouwmans & al. 08]. Enfin, la dernière section présentera quelques méthodes pour évaluer un processus visant à extraire le sujet d'intérêt d'un *background*.

Modélisation basique			**Valeur moyenne, valeur médiane, histogramme, etc.**	√ Très simple à mettre en œuvre √ Temps réel X Fond contrôlé et statique X Ne gère pas le dynamisme
Modélisation statistique	***Hypothèse sur la loi***	GAUSSIENNE	**[Wren & al. 97] *Pfinder***	√ Gère les changements lents X Contrôle de la scène pour la phase d'apprentissage X Ne gère pas les scènes extérieures X Ne gère pas les changements rapides X Sensibilité face aux erreurs d'apprentissage X Sensibilité face aux ombres
			[Horprasert & al. 99, Horprasert & al. 00]	√ Fusion de deux informations (chromaticité & luminosité) donc plus de précision √ Méthode plus automatique √ Classification multiclasses des pixels √ Gère les changements d'illumination locaux et globaux √ Scènes intérieures et extérieures X Ne gère pas les mouvements au sein de la scène (pas de mise à jour) X Plus coûteuse en termes de mémoire et de traitement
		MoG	**[Grimson & al. 98, Stauffer & Grimson 99]**	√ Phase d'apprentissage permettant la présence d'objets mobiles, jusqu'à une certaine limite √ Scènes intérieures & extérieures √ Gère les mouvements et les changements d'illumination lents √ Démonstrateur fonctionnant en continu pendant 16 mois X Ne gère pas les mouvements rapides et les ombres X Approche coûteuse ne permettant pas toujours le temps réel
			[KaewTraKulPong & Bowden 01, KaewTraKulPong & Bowden 03]	√ Mise à jour plus rapide et plus précise d'une scène stable √ Post-traitements fiabilisant et précisant les résultats √ Scènes intérieures & extérieures √ Temps réel
			[Porikli & Tuzel 05]	√ Mise à jour par mécanismes bayésiens optimisant et fiabilisant le processus
			[Hasenfratz & al. 04]	√ Espace YUV permettant plus de sensibilité face aux ombres
			[McKenna & al. 00, Javed & al. 02]	√ Fusion des informations relatives à la chromaticité avec celles relatives aux gradients √ Processus plus précis
			[Javed & al. 02]	√ 3 niveaux de traitements : niveau pixel, niveau région, niveau image √ Processus plus robuste et capable de gérer plus de dynamisme au sein de las scène
		AUTRES	**[Cucchiara & al. 03] *Sakbot*, fonction médiane**	√ Fusion des informations chromaticité et luminosité (espace RGB) √ Ajouts de connaissances sémantiques, améliorant le processus de manière générale √ Classification multiclasses √ Sensibilité aux ombres
			[Haritaoglu & al. 98, Haritaoglu & al. 00] *W4*, distribution bimodale à partir des statistiques d'ordre des valeurs des pixels du fond	√ Approche hiérarchisée : niveau pixel & niveau objet √ Objets mobiles pouvant être présents lors de la phase d'apprentissage √ Scènes intérieures & extérieures relativement dynamiques X Phase d'apprentissage importante X Ne gère pas les ombres et les changements rapides au sein de la scène
	Pas d'hypothèse		**[Elgammal & al. 00, Elgammal & al. 02] *Non-parametric Kernel Density Estimation***	√ Approche plus sensible de manière générale √ Gère les changements lents et rapides au sein de la scène X Méthode coûteuse en termes de mémoire et de traitement X Contrainte temps réel non respectée
Modélisation floue			**[El Baf & al. 08abc]**	√ Fusion réelle des informations couleur et texture (intégrale de Choquet) √ Mise à jour adaptée au niveau du pixel au cours du temps √ Méthode précise et robuste, facile à mettre en œuvre √ Conditions variées, scènes extérieures X Peut s'avérer coûteuse en termes de mémoire et de traitement X Contrainte temps réel ?
Estimation	***Filtre prédictif***		**[Toyama & al. 99] *Wallflower***	√ Approche hiérarchique : niveau pixel (filtre de Wiener), niveau région, niveau image √ A montré son efficacité par rapport à plusieurs autres méthodes connues
	Observations passées		**[Yang & al. 04a]**	√ Approche hiérarchique : niveau pixel & niveau image √ Pas de phase d'apprentissage √ Simple, très rapide, robuste √ Permet de faire face à un grand nombre de situations problématiques (fantômes, manipulation d'objets, déplacements erratiques, changements soudains, etc.)
			[Chalidabhongse & al. 03, Kim & al. 05] *Codebook*	√ Flux vidéo temps réel & vidéo compressé √ Modèle compact du fond √ Fusion des informations relatives à la chromaticité et à la luminosité √ Proposition d'un nouvel espace couleur √ Pas hypothèses sur le modèle à priori √ Scènes intérieures & extérieures √ Gère les changements d'illumination locaux et globaux, les ombres, les mouvements erratiques... X Peut s'avérer coûteuse en termes de mémoire et de traitement
			[Porikli & Wren 05] *Wave-back*	√ Gère les mouvements pseudopériodiques √ Approche temps réel X Nécessite une très importante phase d'apprentissage
			[Porikli 05]	√ Approche temps réel √ Contextes d'observation 'extrêmes' (changements d'illumination abruptes, mouvements erratiques, etc.)

3.5.1.1.1. Définitions et concepts associés

Un modèle du fond de la scène permettra **l'extraction du sujet d'intérêt**, ici l'utilisateur, **de l'image courante**. Le concepteur devra tout d'abord définir ce qu'est à proprement parler son modèle de fond, la représentation de ce dernier que le processus d'extraction du sujet d'intérêt va manipuler.
Une représentation du fond, sa précision et sa gestion (gestion plus ou moins rapide, facilité de création, de manipulation et d'application) dépendent des objectifs que doit remplir ce fond. L'ensemble des processus opérant ou utilisant ce fond dépend de cette représentation. De plus, les faiblesses d'un modèle pourront être compensées par les différents processus le manipulant. Une représentation plus grossière pourra par exemple être compensée par une mise à jour avancée. Ou encore, une mise à jour rapide et adaptée sera pertinente pour opérer sur un modèle de fond très précis.

Depuis un peu plus d'une dizaine d'années maintenant, différentes approches ont été développées, visant à traiter de la problématique étudiée dans cette section. [Bouwmans & al. 08] donne une classification de ces méthodes, que nous allons adopter.
Ces différentes méthodes peuvent en effet être regroupées en **4 classes** bien distinctes :

- **Modélisation basique du fond** (*Basic Background Modeling*) : ces modélisations sont relativement simplistes et n'utilisent que des concepts relativement simples à mettre en œuvre tels que des calculs de moyennes, de valeurs médianes ou d'histogrammes. Ces méthodes sont utilisées dans le cas de fond contrôlé (fond uni et statique).
- **Modélisation statistique du fond** (*Statistical Background Modeling*) : Ces approches sont celles qui sont le plus adoptées actuellement. Elles consistent à créer un modèle probabiliste, plus ou moins complexe, du fond, généralement à partir de plusieurs observations effectuées dans le temps. Ces observations, qu'elles soient à l'échelle du pixel, d'une région ou d'une image à part entière, sont alors associées à un ensemble de variables aléatoires, associées à différentes lois de probabilité. Il est alors vérifié que chaque nouvelle observation suit ces lois, permettant ainsi de différencier les éléments du fond de ceux de l'avant-plan.
- **Modélisation floue du fond** (*Fuzzy Background Modeling*) : ces méthodes sont récentes et impliquent l'utilisation d'intégrales floues pour fusionner les mesures de similarité hétérogènes, établies entre le modèle du fond et l'image courante.
- **Estimation du fond** (*Background Estimation*) : le fond attendu est estimé en tant qu'image 2D et est alors comparé aux images provenant des différents flux vidéo. Tout élément de l'image courante n'obéissant pas au modèle attendu est alors considéré comme appartenant au *foreground*. Cette estimation peut se faire par le biais de filtres prédictifs (Kalman, Wiener, etc.), de construction d'historiques pour analyser les propriétés passées d'un pixel, etc.

Lors de l'élaboration ou de l'adoption d'une méthode pour générer et gérer un modèle de fond, le concepteur doit se mettre en accord l'ensemble de ses conditions de travail et ses objectifs.
Il doit tout d'abord déterminer sur quelles données le processus va s'appuyer pour construire son modèle et agir sur ce dernier. La plupart des approches ne considèrent que la couleur des pixels de l'image pour générer et manipuler le modèle de fond. Pourtant, certaines méthodes

peuvent également s'appuyer sur d'autres données, brutes i.e. issues directement des flux vidéo ou de plus haut niveau i.e. construite à partir des données brutes : textures, contours, profondeur, gradient, flux, etc. Ces informations pourront être traitées de manière indépendante ou fusionnées, pour obtenir un résultat cohérent vis-à-vis de l'hétérogénéité des données. Plus l'approche prend en compte d'informations en entrée, plus elle pourra atteindre un degré de précision important. Cependant, elle risque alors d'atteindre des coûts de traitement et de mémoire trop importants, relativement aux contraintes matérielles que présente le système.
Il est ensuite possible de générer et gérer le modèle à plusieurs niveaux de granularité. Tout d'abord, le processus peut agir au niveau du pixel. Mais il est également possible de travailler au niveau de la région ou de l'objet. Et enfin, il peut élaborer et manipuler le modèle du fond au niveau image à proprement parler. Il est possible de rencontrer dans la littérature des approches élaborées à partir de la hiérarchisation des décisions et des traitements. Typiquement, le modèle du fond sera établi au niveau pixel, affiné et mis à jour au niveau de la région et de l'objet et, dans le cas de l'absence de mouvement et/ou du sujet d'intérêt au sein de la scène, remplacé par l'image courante (niveau image).
Les approches rencontrées dans la littérature se différencient également par les contextes matériels et relatifs à la scène observée les englobant. En effet, les approches seront distinctes selon si la scène observée est une scène intérieure ou extérieure, statique ou dynamique et si les caméras observant la scène sont fixes (ou stationnaires) ou mobiles. Elles seront fonctions des capacités du système en termes de mémoire et de traitement et obéiront ou non à la contrainte temps réel.
Enfin, la plupart des approches sont développées dans le but de répondre à un ensemble délimité de problèmes. Ces problèmes doivent donc être précisément identifiés, de manière à adopter ou à élaborer une méthode adéquate pour y répondre. Certaines approches s'appliqueront ainsi à gérer les mouvements lents au sein de la scène ; les mouvements rapides ; les mouvements périodiques ou erratiques ; l'apparition/la disparition/le déplacement d'objets au sein de la scène ; les ombres ; les mouvements potentiels de la caméra ainsi que le bruit généré par les fluctuations du flux vidéo ; les changements lents ou rapides d'illumination ; etc.

Chaque méthode, qu'elle que soit la classe à laquelle elle appartient suit, comprend **4 processus** bien distincts :

1. *Modélisation du fond*
   Ce processus génère **la modélisation du fond**, correspondante à la représentation qu'a adoptée le concepteur. Cette modélisation dépend des objectifs du concepteur, des différents contextes et contraintes s'articulant autour du processus, etc.

2. *Mise à jour (ou maintenance) du fond*
   Dans nos travaux, la scène est dynamique. Visuellement, les éléments composant la scène ou compris dans la scène vont évoluer, entrainant la modification au cours du temps des valeurs des pixels. De manière à être en accord permanent avec le fond de la scène observée et à être capable d'en extraire le sujet d'intérêt à tout instant, il est donc nécessaire **de mettre à jour le modèle de fond au cours du temps**.

3. *Initialisation du fond*
   L'initialisation du modèle peut se faire avant ou pendant l'interactivité. Typiquement, le modèle de fond est construit à partir d'une scène vide (scène statique et sans l'utilisateur).

Cela dit, il peut s'avérer qu'il ne soit pas possible que la scène soit vide et dynamique. L'initialisation du modèle doit alors prendre en compte, dans une certaine mesure, le dynamisme potentiel de la scène. Cette initialisation, plus ou moins complexe, peut s'effectuer sur une ou plusieurs *frames*. Cette opération peut alors être couplée avec d'autres traitements la supportant : filtrage de petites zones, détection des personnes présentes au sein de la scène (par le biais de leur peau par exemple), classification des pixels ayant des attributs similaires, etc.

4. *Extraction du sujet d'intérêt (ou de l'avant-plan)*
   Le processus d'extraction du sujet d'intérêt du fond est à élaborer précisément. Il est composé d'une étape de différenciation entre le modèle du fond et l'image courante, suivie d'un ensemble de traitements pour fiabiliser les résultats de cette différenciation.
   Le processus de différenciation entre l'image courante et le modèle du fond est à définir par le concepteur. Il consiste à mesurer la différence qui peut exister entre le modèle du fond et les images issues du flux vidéo (simple différence, différence sur plusieurs *frames*, autre ?). Cette différence est alors soumise à un seuillage, permettant alors de mettre en évidence les objets composant l'avant-plan de l'arrière plan.
   Par la suite, selon le résultat de la différenciation et des objectifs que doit atteindre le processus d'extraction, un ensemble de traitements supplémentaires est exécuté de manière **à mettre en évidence et préciser le sujet d'intérêt**. C'est en effet généralement nécessaire car l'étape de différenciation avec le modèle du fond peut générer deux types d'erreurs : erreurs de type faux négatif, *false negatives* (éléments du *foreground* ayant été considéré comme du fond) ou erreurs de type faux positifs, *false positives* (éléments du fond ayant été considéré comme du *foreground*). Ces erreurs sont majoritairement dues aux ombres des différents éléments composant la scène observée. Des opérations de filtrages ou des opérations morphologiques peuvent donc être adoptées pour nettoyer le bruit ou pour supprimer les zones de trop petites tailles. De plus, certaines approches permettent d'envisager l'appartenance d'un pixel de l'image à une sous-classe intermédiaire particulière, se situant entre les deux classes principales *background* et *foreground* : fond inchangé, ombre, surexposition, objet mobile appartenant au fond, objet fantôme, ombre fantôme, etc.

3.5.1.1.2. Classification des méthodes existantes

1. *Modélisation basique du fond*
   Nous détaillerons peu cette classe qui comprend l'ensemble des techniques basiques de modélisation et de manipulation du fond. Ces techniques sont généralement adoptées dans le cas de scènes simples, contrôlées et statiques, n'impliquant donc pas la mise à jour du modèle de fond. L'initialisation du fond se fait lorsque le sujet d'intérêt ne se trouve pas dans la scène.
   La méthode la plus simple pour modéliser un fond est de considérer un ensemble de N *frames* et de moyenner sur ces N *frames* les N valeurs que prend chaque pixel. Une approche similaire consiste à attribuer à chaque pixel sa valeur médiane. L'extraction du sujet se fait alors par simple différence, pondérée ou non, de l'image courante avec le fond. Il est également possible de comparer les histogrammes du fond et de l'image courante. L'extraction du sujet d'intérêt se fait alors en appliquant un seuil à cet histogramme. Tous les pixels dont la valeur se trouve en dessus de ce seuil sont considérés comme appartenant au *foreground*.

Typiquement, ces techniques sont incapables de faire face aux effets fantômes provoqués par les objets mobiles, aux changements d'illumination, aux ombres ou à tout type de mouvements au sein de la scène. Il est à noter que la méthode utilisée au sein du ***Cyberdôme*** (voir Section 3.3.2.3.) appartient initialement à cette classe de méthodes.

2. *Modélisation statistique du fond*
Cette forme de modélisation est certainement la plus répandue actuellement. Il s'agit de considérer le pixel, ou tout autre niveau image, comme une variable aléatoire, suivant une densité de probabilité particulière. Le fond peut alors être représenté sous la forme d'un modèle statistique. La construction de ce modèle se fait lors d'une phase d'apprentissage sur plusieurs *frames*.

Il est tout d'abord possible de considérer que la distribution que suivent les pixels du *background* est une loi de probabilité connue. Les paramètres de cette loi sont alors déterminés à partir des données observées lors de la phase d'apprentissage.

Tout d'abord, il a été proposé que chaque pixel suive une unique distribution, généralement une loi normale, dont les paramètres sont calculés lors de la phase d'apprentissage, à partir des valeurs RGB du pixel.
Pfinder [Wren & al. 97] est généralement considéré comme étant le premier système utilisant un modèle statistique de fond. Dans ce modèle, chaque pixel suit une simple distribution gaussienne, dont les paramètres sont déterminés à partir de la valeur du pixel dans l'espace couleur YUV. Le contexte d'observation est une scène intérieure, qui doit être vide (sans le sujet d'intérêt) lors de la phase d'apprentissage. Cette méthode permet de gérer les mouvements lents au sein de la scène mais pas les petits mouvements rapides. Elle reste sensible aux erreurs qui peuvent se produire lors de la phase d'apprentissage et son efficacité n'a pas été prouvée dans le cas de l'observation d'une scène extérieure.
[Horprasert & al. 99, Horprasert & al. 00] utilisent également une simple distribution gaussienne pour modéliser le comportement d'un pixel. Cependant, chaque distribution est construite à partir de la chromaticité et de la luminosité d'un pixel, informations calculées à partir des valeurs RGB de ce pixel. A partir de l'analyse de ces distributions, le *foreground* peut être extrait (valeur de seuil déterminé automatiquement lors de la phase de différenciation avec l'image courante). De plus, l'approche propose une classification multiclasses des pixels : *background* original, ombre, surexposition, objet mobile. Cette méthode est efficace car offrant plus de sensibilité quant à la détection du *foreground*. Elle permet de faire face à des changements d'illumination locaux et globaux. De plus, c'est une approche temps réel, capable de gérer des scènes intérieures et extérieures. Cependant, cette approche suppose que le fond soit statique, le processus ne comprenant donc pas de procédure de mise à jour du modèle, et est plus coûteuse en temps de calcul.

Par la suite, devant les inconvénients que peuvent présenter les méthodes modélisant l'évolution du pixel par une unique distribution, il a été proposé de la modéliser par plusieurs distributions. La modélisation du fond par le biais de mixture de gaussiennes, *Mixture of Gaussian* (MoG) (ou encore mélange de gaussiennes, mixture gaussienne, mélange gaussien et leurs équivalents anglais) est certainement l'approche la plus

utilisée aujourd'hui. Le comportement d'un pixel est décrit par K distributions gaussiennes pondérées, typiquement avec K compris entre 3 et 5. Lors de la phase d'apprentissage ou de mise à jour du modèle, pour chaque pixel, les n dernières valeurs du pixel permettent de calculer les paramètres des K distributions, ainsi que leurs poids associés (par des algorithmes tels que l'algorithme ***Expectation-Maximization*** (EM) ou une de ses approximations, par exemple). Tout pixel dont l'évolution s'écarte trop des K distributions (3 déviations standards à partir de la moyenne) est considéré comme appartenant au *foreground*.
[Grimson & al. 98, Stauffer & Grimson 99] furent les premiers à proposer une telle approche, dans le cadre d'un système de surveillance vidéo. L'algorithme EM étant trop coûteux pour une procédure de mise à jour du modèle, une approximation *online* est employée. Si la valeur du pixel ne correspond à aucune distribution, la distribution la moins probable est remplacée par une distribution dont la moyenne sera cette valeur, la variance sera importante et le poids très faible. Plus les valeurs des pixels observés supporteront cette nouvelle distribution, plus cette dernière aura un poids important. Lors de la phase d'apprentissage, les objets mobiles peuvent être acceptés, jusqu'à une certaine limite, puisque chaque élément du *foreground* sera représenté par un ensemble de distributions associées à des poids peu importants, qui seront alors vite remplacées au cours du processus. Le système de surveillance a fonctionné en extérieur, pendant 16 mois en continu. L'approche a montré son efficacité devant des changements d'illumination et des mouvements lents, ainsi qu'à des apparitions et des disparitions d'objets. Des fonds complexes peuvent être modélisés, mais la méthode ne gère pas les ombres et les variations rapides au sein de la scène. L'approche est coûteuse en temps de calcul, ce qui peut se ressentir lors de la phase d'apprentissage et qui ne permet pas forcément d'assurer un processus temps réel.
Plusieurs améliorations ont été apportées à cette dernière méthode. [KaewTraKulPong & Bowden 01, KaewTraKulPong & Bowden 03] proposent une procédure de mise à jour du fond légèrement différente, permettant d'apprendre plus rapidement et plus précisément une scène stable. De plus, plusieurs traitements, visant à supprimer les faux positifs (filtrage, opérations morphologiques, etc.), sont employés au cours du processus d'extraction du *foreground*. L'approche est temps réel et est efficace dans le cas de scènes extérieures. [Porikli & Tuzel 05] modifie également la procédure de mise à jour en remplaçant l'algorithme EM par des mécanismes bayésiens, pour déterminer les paramètres et poids des distributions gaussiennes. Egalement, l'adoption d'espaces couleur autres que le standard RGB permet de fiabiliser le processus, particulièrement face aux ombres. [Hasenfratz & al. 04] présente ainsi une approche où les distributions gaussiennes sont déterminées à partir des valeurs des pixels dans l'espace YUV. Le lecteur intéressé pourra lire l'état de l'art proposé par [Kristensen & al. 06], qui énumère les méthodes basées MoG, utilisant d'autres espaces couleur que l'espace RGB. [McKenna & al. 00, Javed & al. 02] présentent des approches permettant de fusionner aux informations relatives à la chromaticité, des informations relatives aux gradients, pour rendre le processus plus robuste et plus précis. [Javed & al. 02] expose également une technique hiérarchisée. Au niveau du pixel, des modèles statistiques de couleur et de gradients sont utilisés séparément pour classer chaque pixel comme appartenant au *background* ou au *foreground*. Au niveau région, les pixels du *foreground* sont groupés en région cohérente, par le biais du processus de différenciation basée chromaticité, et le processus de différenciation basée gradient permet de tester la validité de ces régions. Enfin, au niveau image, une analyse est effectuée pour détecter les modifications globales d'illumination.

Pour finir, le lecteur intéressé pourra se référer à [Bouwmans & al. 08], qui proposent un état de l'art sur les méthodes basées MoG. [Zivkovic04, Lee 05], quant à eux, présentent des méthodes pour optimiser les calculs, mathématiquement et informatiquement, des modèles basées MoG.

D'autres types de distributions que les gaussiennes ont été envisagés pour modéliser un fond dynamique et en extraire les éléments de l'avant-plan.
Plus que la gaussienne, [Cucchiara & al. 03] observent de meilleurs résultats avec la fonction médiane. Cette méthode, appelé ***Sakbot*** (*Statistical And Knowledge Based ObjecT detection*), se base sur des informations relatives à la chromaticité et la luminosité, dans l'espace RGB, pour construire, pixel à pixel, son modèle statistique de fond. L'introduction au sein du système de connaissances concernant les objets à segmenter permettent l'amélioration des processus de modélisation du fond et d'extraction du *foreground*. Des traitements supplémentaires lors de ce dernier processus permettent la fiabilisation des résultats et la classification des pixels au sein de diverses classes, telles que celles concernant les objets mobiles, les fantômes et les ombres.
Le système ***W4*** de surveillance vidéo [Haritaoglu & al. 98, Haritaoglu & al. 00] est certainement l'un des systèmes les plus cités, lors de l'étude de la problématique de modélisation du fond et d'extraction du *foreground*. L'évolution de chaque pixel est traduite par une distribution bimodale, construite à partir des statistiques d'ordre des valeurs des pixels du fond, observées lors d'une phase d'apprentissage. A l'issue de cette phase d'apprentissage, chaque pixel du fond est associé à sa valeur minimale, sa valeur maximale et la différence d'intensité maximale qui a pu être observée entre deux *frames* consécutives. La phase d'apprentissage, qui implique l'observation d'une séquence importante d'images (>100), se déroule en deux étapes : la première étape permet la différenciation des pixels mobiles et stationnaires au cours du temps (application d'un filtre médian appliqué pendant 20 à 40 secondes de vidéo ; la deuxième ne considère que les pixels stationnaires et les associe aux trois valeurs énumérées précédemment. La mise à jour du modèle se fait au niveau du pixel (mise à jour périodique pour s'adapter aux changements d'illumination) et de l'objet (un objet, qui fut mobile, sera mis à jour dans le fond s'il est immobile pendant une période importante). Ces deux mises à jour utilisent 3 cartes 2D, construits dynamiquement au cours de l'observation, qui définissent des historiques d'appartenance à une classe pour chaque pixel : combien de fois le pixel a appartenu au fond, combien de fois il a appartenu à l'avant-plan et depuis combien de *frames* il appartient à l'avant-plan. L'extraction du *foreground* se fait en fonction au niveau du pixel, en prenant en compte les 3 valeurs qui lui sont associées. Des traitements supplémentaires permettent la suppression de faux positifs et la labellisation des différentes régions détectées. Le système ***W4*** permet de considérer des scènes extérieures, où évoluent plusieurs individus. Il gère un dynamisme relatif de la scène, sans prendre en compte les ombres et les changements soudains d'illumination.

Contrairement aux méthodes présentées précédemment, certaines approches ont cherché à établir des modèles statistiques du fond, sans faire d'hypothèses concernant le type de distributions que suivent les pixels de l'image. Ceci évite d'avoir à choisir un modèle et d'estimer ses différents paramètres.

[Elgammal & al. 00, Elgammal & al. 02] présente une modélisation du fond en utilisant une méthode baptisée ***Non-parametric Kernel Density Estimation***. Un pixel du fond est modélisé à partir de n observations au cours du temps. La méthode estime les contributions d'un ensemble de fonctions noyaux, en se servant de toutes les valeurs enregistrées, plutôt que de mettre à jour le modèle de manière itérative, observation après observation. L'extraction du *foreground* est accompagnée de l'étude de la cohérence spatiale des régions détectées. Cette méthode, plus sensible que la plupart des autres méthodes et permettant de faire face aussi bien aux mouvements lents et rapides qui ont lieu au sein de la scène, peut s'avérer très coûteuse en termes de mémoire et de traitement au niveau du système. La contrainte réelle n'est pas respectée.

3. *Modélisation floue du fond*

Les méthodes appartenant à cette classe étant récentes, nous ne donnerons ici qu'un exemple de modélisation floue du fond.

La méthode décrite dans [El Baf & al. 08abc] n'emploie pas un modèle complexe de fond. Ce modèle peut être une image vidéo, ou une moyenne sur plusieurs *frames*, de préférence sans le sujet d'intérêt au sein de la scène observée, représenté dans un espace de couleur donné.

Le fond et l'image courante sont alors comparés en calculant des mesures de similarité entre les deux images. Ces mesures de similarité, qui sont au départ indépendantes, sont effectuées sur la couleur et la texture. L'intégrale floue de Choquet permet alors de fusionner les deux mesures. Cette intégrale donne alors une mesure de similarité entre les deux images, pour chaque pixel et relativement aux données couleur et texture qui sont alors considérées de manière corrélée. Un seuil peut alors être appliqué sur cette mesure pour déterminer si le pixel considéré appartient au *background* ou au *foreground*.

La mise à jour du fond s'effectue au niveau du pixel et en fonction de la mesure donnée par l'intégrale de Choquet. Cette mise à jour consiste à mixer, de manière pondérée vis-à-vis de la mesure donnée par l'intégrale floue, le fond et l'image courante. Les poids sont également fonction de l'historique d'appartenance du pixel (poids différent si le pixel appartenant à la *frame* précédente au *background* ou au *foreground*). La mise à jour est donc adaptée pour chaque pixel et au cours du temps.

Différents traitements supplémentaires sont employés pour améliorer les résultats. Le bruit et les zones trop petites sont supprimés par filtrage et labellisation des pixels connexes.

L'approche est robuste car deux types de données sont utilisés en entrée de l'algorithme. Ce dernier se révèle d'ailleurs facile à mettre en œuvre. Plusieurs espaces de couleur et plusieurs ensembles d'images, impliquant des scènes extérieures nuageuses et pluvieuses, ont été testés. L'algorithme a montré son efficacité devant les changements d'illumination, les ombres, les modifications légères au sein de la scène.

4. *Estimation du fond*

Les méthodes appartenant à cette classe cherchent à modéliser l'image 2D du fond qui sera alors à comparer à l'image réelle issue des flux caméra. Cette modélisation peut donc se faire de manière prédictive, par l'intermédiaire de filtres par exemple, ou en se référant simplement à ou aux observations passées.

Les filtres prédictifs les plus utilisés sont ceux de Kalman, Wiener et Tchebychev.

La méthode ***Wallflower*** [Toyama & al. 99], par exemple, est un algorithme de modélisation et de mise à jour hiérarchisée du fond. Au niveau du pixel, la méthode utilise le filtre de Wiener pour prédire la valeur, dans l'espace RGB, d'un pixel, en fonction de valeurs passées de ce dernier. Les éléments du *foreground* peuvent alors être extraits et le fond mis à jour au niveau du pixel. Au niveau région, les différents pixels détectés comme appartenant au *foreground* sont regroupés en régions cohérentes. Enfin, au niveau image, les changements soudains d'illumination au sein de la scène globale sont détectés, permettant la mise à jour rapide et précise du modèle de fond.
Cette méthode démontre son efficacité par rapport à de nombreuses autres méthodes, comme le montre l'étude comparative qu'effectuent les auteurs. En effet, la méthode ***Wallflower*** est confrontée à 8 autres algorithmes connus.

D'autres méthodes ne cherchent pas à prévoir forcément la valeur du fond à un instant donné, mais modélise le fond en fonction des évolutions de la scène qui ont déjà été observées.

[Yang & al. 04a] propose une méthode permettant la modélisation du fond aussi bien au niveau du pixel que de l'image globale, en se basant sur l'hypothèse que la valeur, dans l'espace RGB, d'un pixel appartenant à un objet mobile évolue plus rapidement que s'il appartient au fond. Cette méthode a été développée dans le cadre de l'élaboration d'un système intelligent de surveillance vidéo de scènes aussi bien intérieures qu'extérieures.
La technique de mise à jour du fond au niveau du pixel se base sur l'introduction au sein du processus d'une matrice 2D dynamique, dont la valeur pour une position donnée (correspondant à celle d'un pixel dans l'image) permet de savoir depuis combien de *frames* un pixel n'a pas changé de valeur. Lorsqu'un pixel voit sa valeur évoluer, entre l'image courante et la *frame* précédente (ou une autre *frame* passée), la matrice dynamique est alors mise à jour au niveau des coordonnées images du pixel, attribuant à ce dernier une valeur correspondant à une durée en *frames*. Au cours de la *frame* suivante, si le pixel a encore évolué, la même durée lui est attribuée au sein de la matrice dynamique. En revanche, dans le cas contraire, la durée diminue d'une unité. D'une *frame* à une autre, si aucun changement n'est observé, cette durée tend progressivement vers une valeur nulle. Lorsque cette durée est nulle, la valeur du pixel est mélangée de manière pondérée à celle du pixel correspondant du fond. Au niveau de l'image globale, la mise à jour du fond se fait par l'intermédiaire d'une quantification du mouvement au sein de l'image courante. Si cette mesure est en deçà d'un certain seuil, alors le fond est mis à jour directement grâce à l'image courante (mélange pondéré de l'image courante avec le fond), sans analyse de la matrice dynamique.
Aucune phase d'initialisation ou d'apprentissage n'est requise. En effet, la procédure de mise à jour permet de construire progressivement un modèle du fond, au sein duquel le sujet d'intérêt est potentiellement présent. A partir du moment où le sujet d'intérêt devient mobile, il sera détecté et le fond se mettra alors progressivement à jour, sans le prendre en compte.

Le processus de différenciation entre l'image courante et le modèle de fond est un simple OU logique (le pixel de l'image courante appartient au *foreground*, à partir du moment où une de ses valeurs R, G ou B est différente, vis-à-vis d'un seuil particulier, de la valeur correspondante du même pixel du fond). Plusieurs traitements basiques suppriment alors les zones détectées trop petites.
Cette méthode est simple, très rapide, robuste et ne nécessite pas de phase d'apprentissage. La procédure de mise à jour du fond permet de faire face à un grand nombre de situations problématiques (fantômes, manipulation d'objets, déplacements erratiques, changements soudains, etc.). En effet, l'introduction au sein du processus d'une matrice dynamique permet de ne pas mettre à jour dans le fond, pendant un certain temps, un sujet mobile qui reste immobile. De plus, l'analyse au niveau de l'image globale permet de faire face aux changements soudains au sein de la scène.

[Chalidabhongse & al. 03, Kim & al. 05] présentent la méthode ***Codebook*** pour extraire les objets du *foreground* par modélisation et mise à jour du fond. Cette méthode a été développée dans un contexte particulier de traitement de vidéos compressées. Mais elle est également largement applicable pour le traitement de flux vidéo temps réel. La méthode s'applique au niveau du pixel et utilise des informations relatives à la chromaticité et la luminosité, extraites à partir des valeurs RGB du pixel. Dans le cadre de cette méthode, les auteurs ont proposé un nouvel espace couleur (plus ou moins une extension de l'espace RGB normalisé) au sein duquel la couleur et la luminosité d'un pixel sont différenciables et comparables, par le biais d'une métrique précise, avec celles d'un autre pixel.
Chaque pixel est représenté par un *codebook*, forme compressée reflétant son évolution au cours d'une séquence donnée d'images. Un *codebook* comprend un ou plusieurs *codewords*. Un *codeword* est construit à partir d'un ensemble d'observations (pas forcément toutes les observations) concernant les valeurs qu'a pu prendre le pixel au cours de la séquence. Un *codeword* est caractérisé, entre autres, par une valeur précise pour la couleur et un intervalle de valeurs pour la luminosité. Tous les pixels ne sont pas représentés par le même nombre de *codewords*. De plus, les *codewords* ne traduisent pas une distribution de probabilité particulière.
Pour construire le modèle initial, aucune donnée ni hypothèse n'est requise *a priori*. Ce modèle initial est construit au cours d'une très longue phase d'apprentissage. Au départ, le *codebook* d'un pixel donné est vide. La première valeur que prend ce pixel permet la construction du premier *codeword* du *codebook*. Le *codebook* se remplit alors au fur et à mesure des observations, chaque observation mettant à jour un *codeword* déjà existant, ou créant un nouveau *codeword*. La confrontation d'une observation avec un *codeword* s'effectue au sein de l'espace couleur développé dans ces travaux (calcul de la distance entre les deux valeurs couleur et vérification que la valeur de luminosité observée est bien comprise dans l'intervalle de valeurs défini par le *codeword*).
La procédure de mise à jour est similaire à celle décrite lors de la phase d'apprentissage. La différence résulte dans le fait qu'un filtre temporel est appliqué sur l'ensemble des *codewords* de chaque *codebook*. Un *codeword* construit à partir d'un faible nombre d'observations (la notion de 'faible' est définie par un seuil) et/ou non mis à jour depuis un certain nombre de *frames* est éliminé.
La détection et extraction des pixels du *foreground* s'effectue en comparant les valeurs couleur et de luminosité d'un pixel de l'image courante avec celles du *background*, au sein de l'espace couleur développé dans ces travaux.

La méthode ***Codebook*** permet la modélisation compacte de fonds complexes, aussi bien en extérieur qu'en intérieur. La méthode est temps réel et permet de faire face aux changements globaux et locaux, en particulier d'illumination, des ombres et surexpositions, et des différents mouvements erratiques, au sein de la scène. Cependant, plus le fond est complexe, ou se complexifie, plus l'approche peut s'avérer coûteuse en termes de traitement et de mémoire. Une optimisation possible est de classer les *codewords* au sein d'un *codebook* en fonction du temps entre leur dernière mise à jour. Il est en effet tout à fait probable qu'un même *codeword* soit mis à jour sur plusieurs *frames* consécutives. Cette optimisation permet d'accélérer les traitements.

Pour finir, et pour ne citer qu'eux, les travaux de Porikli [Porikli & Wren 05, Porikli 05] exposent d'autres possibilités originales quant à la modélisation et mise à jour de fond.
[Porikli & Wren 05] présentent une représentation du fond se basant sur la décomposition fréquentielle de l'image, approche baptisée ***Wave-back***. La modélisation et la mise à jour du fond s'effectuent au niveau du pixel. Les différentes observations relatifs à un pixel donné permettent le calcul des coefficients de la transformée en cosinus discrète associée. Pour un pixel donné, chaque nouvelle valeur permet le calcul de nouveaux coefficients, qui seront alors confrontés à ceux pixel du fond correspondant et permettant ainsi la génération d'une carte de distances (*distance map*). Un seuillage sur ces cartes permet l'extraction des pixels appartenant au *foreground*. Cette approche est particulièrement pertinente lorsque la scène est composée d'éléments effectuant des mouvements pseudopériodiques, tels que des branches balancées par le vent. L'approche est temps réel, pour des images 320x240, et nécessite une très importante phase d'apprentissage (>3000 images).
[Porikli 05] propose une méthode se basant sur le fait qu'une scène peut être décomposée en plusieurs parties statiques et dynamiques. Pour cela, l'auteur se base sur le fait qu'une scène peut être vue comme une image intrinsèque, multiplication de plusieurs images (ici des images dynamiques, correspondant au *foreground*, et des images statiques, correspondant au *background*). Par l'analyse spatiotemporelle des gradients calculés sur une séquence d'images, cette méthode permet la modélisation du fond et l'extraction du *foreground*. Cette méthode, qui peut s'appliquer tout à fait dans un contexte temps réel, est robuste face à conditions extrêmes telles que des changements d'illumination abruptes et des mouvements fortement erratiques.

3.5.1.1.3. Evaluation

La qualité des résultats d'un algorithme de modélisation de fond et d'extraction du sujet d'intérêt dépendent fortement du type de système et d'application développés. Elle est donc très liée à la subjectivité du jugement des différents concepteurs. En fonction de cette qualité, les concepteurs pourront décider de changer d'approches pour atteindre leurs objectifs. Il est donc intéressant, dans ce cas, d'évaluer différents algorithmes, en fonction d'objectifs particuliers, de manière à obtenir la meilleure qualité possible quant aux résultats obtenus.
Nous ne serons pas exhaustifs dans cette section car, là encore, les travaux sont nombreux. Nous ne citerons que quelques références connus, illustrant ainsi nos propos.

La meilleure façon, et la plus répandue, d'évaluer un algorithme est de soumettre ce dernier à un ensemble d'images (fond et sujet d'intérêt), mises à disposition sur Internet. Chaque image

est associée au résultat 'idéal' (*ground truth*) que doit produire l'algorithme. Il est ainsi possible visuellement de comparer le résultat de l'algorithme testé avec celui lié à l'image d'entrée.
Les ensembles d'images les plus utilisés sont les suivants : ***Wallflower Dataset***[37], ***ATON Dataset***[38], ***OTCBVS Dataset***[39], ***VSSN Dataset***[40], ***PETS Dataset***[41], ***IPPR Dataset***[42], etc.
Ces ensembles d'images peuvent également être utilisés pour comparer les performances de plusieurs algorithmes existants, comme par exemple [Hall & al. 05] qui comparent, entre autres, les performances des algorithmes présentés dans [Stauffer & Grimson 99, Toyama & al. 99, Wren & al. 97].

Plusieurs méthodes ont bien sûr ont été élaborées pour évaluer un algorithme selon d'autres critères. [Chalidabhongse & al. 03], par exemple, présente une méthodologie, appelée la méthode ***PDR*** (*Perturbation Detection Rate*), pour mesurer les performances d'un algorithme de soustraction de fond. Cette approche vise à quantifier la sensibilité d'un algorithme à détecter une cible faiblement contrastée, présente au sein d'une scène modélisée. Le processus d'évaluation se base sur l'hypothèse que le *foreground* présente une distribution statistique localement similaire, mais légèrement perturbée ou décalée, à celle du fond. L'analyse est alors effectuée en perturbant ou décalant les distributions que présente le *background*, dans des directions aléatoires et uniformes dans l'espace RGB. L'algorithme de soustraction de fond est alors exécuté et son taux mesuré de détection (du *foreground*) est alors comparé au taux idéal attendu. Ce taux est une fonction du contraste et de l'amplitude du décalage ou de la perturbation. Cette méthode est intéressante car aucune connaissance sur le *foreground a priori* n'est requise pour évaluer un algorithme de soustraction de fond.
Dans ces travaux, 4 algorithmes connus sont confrontés [Stauffer & Grimson 99, Horprasert & al. 99, Kim & al. 05, Elgammal & al. 00].

3.5.1.2. Positionnement

[37] http://research.microsoft.com/en-us/um/people/jckrumm/WallFlower/TestImages.htm
[38] http://cvrr.ucsd.edu/aton/shadow/
[39] http://www.cse.ohio-state.edu/otcbvs-bench/
[40] http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/
[41] http://pets2006.net/
[42] http://media.ee.ntu.edu.tw/Archer_contest/

Modélisation basique
Valeur moyenne, valeur médiane, histogramme, etc.

- Méthodes trop basiques
- C'est justement ce que nous voulons améliorer

Modélisation statistique

- Très étudiées et améliorées
- Méthodes précises et robustes
- Permet de gérer tous types de scènes, avec différents degrés de dynamisme
- Plus coûteuses de manière générale
- Implique l'usage de statistiques
- Pas de méthodes résolvant tous les problèmes
- Nécessite une phase d'apprentissage particulière, plus longue et plus complexe

Modélisation floue
[El Baf & al. 08abc]

- Approches trop récentes ?
- Méthodes qui semblent faciles à mettre en œuvre
- Traitements adaptés au niveau du pixel : similarité, intégrale, modélisation & mise à jour
- Traitements nombreux au niveau du pixel
- Contrainte temps réel ? Peut être difficile pour des images importantes
- Sensibilité aux changements rapides ?
- Ne gère peut être pas un dynamisme un peu plus important au sein de la scène

Estimation

- Nombreuses méthodes
- Répond à de nombreuses problématiques
- Il existe toujours une solution pour améliorer une méthode
- Obéit à la contrainte temps réel
- Contextes d'observation 'extrêmes'
- Peut-être plus efficace et performante que les méthodes statistiques
- N'implique pas de calculs statistiques (plus intuitive et moins coûteuse)

Hypothèse sur la loi

- Modèles de fond précis
- Extraction relativement simple du *foreground*
- La loi doit être vérifiée
- Adoption d'un modèle prédéfini, qui peut ne pas refléter la vérité terrain
- Nécessite une phase d'apprentissage

Pas d'hypothèse
[Elgammal & al. 00, Elgammal & al. 02]

- Pas d'hypothèses à faire concernant les distributions
- Pas d'estimation des paramètres
- Répond aux mêmes problématiques, parfois même mieux
- Méthodes sans modèle prédéfini donc moins simple
- Méthodes plus coûteuses (temps réel)

Filtre prédictif
[Toyama & al. 99]

- Hypothèses à faire sur l'évolution des valeurs d'un pixel au cours du temps
- Phase d'apprentissage
- A fait ses preuves dans le cadre de nombreux développements

Observations passées
[Yang & al. 04a]
[Chalidabhongse & al. 03, Kim & al. 05] Codebook
[Porikli & Wren 05] Wave-back
[Porikli 05]

- Très étudiées et étendues
- Répond à un grand nombre de problématiques
- Mise à jour la plupart du temps uniquement à partir de la *frame* précédente

Gaussienne
[Wren & al. 97], [Horprasert & al. 99, Horprasert & al. 00]

- Méthodes 'trop' simples
- Suffisante pour une scène intérieure
- Reste insuffisante face à des degrés faibles de dynamisme

MoG
[Grimson & al. 98, Stauffer & Grimson 99]
[KaewTrakulPong & Bowden 01, KaewTrakulPong & Bowden 03], [Porikli & Tuzel 05] (Processus)
[Hasenfratz & al. 04] (Espace couleur)
[McKenna & al. 00, Javed & al. 02] (Fusion)
[Javed & al. 02] (Hiérarchie)

- A prouvé son efficacité dans le cadre de plusieurs systèmes performants
- Très étudié donc de nombreuses améliorations et extensions
- Bien adapté aux scènes extérieures
- Permet de faire face à plus de dynamisme au sein de la scène
- Permet d'être sensible aux ombres grâce aux améliorations (espace couleur particulièrement)
- Gère difficilement les changements rapides (mouvements, illumination) au sein de la scène
- Peux vite s'avérer coûteux (temps réel)

Autres distributions
[Cucchiara & al. 03]
[Haritaoglu & al. 98, Haritaoglu & al. 00]

- Solutions équivalentes vis-à-vis des MoG et de leurs améliorations
- Complexification vis-à-vis des MoG
- Moins évidente à mettre en œuvre

1. Contexte de travail

Avant de discuter les différentes méthodes exposées précédemment, il nous faut bien identifier le contexte de travail au sein duquel notre processus est élaboré.

Nous travaillons avec des images 2D issues directement des flux vidéo des caméras. Ces images, une fois le filtre ***Bayer*** appliqué, sont des images couleur, la valeur de chaque pixel étant calculée dans l'espace RGB. Nous pouvons donc traiter tout type d'informations extraites ou construites à partir de ces images.
Cela dit, la modélisation et la soustraction du fond n'est qu'une courte étape du processus de capture. Nous désirons que cette étape soit la plus rapide possible afin de ne pas trop ralentir le processus global. C'est pourquoi nous ne cherchons pas forcément à construire un maximum de connaissances pour modéliser notre fond, engendrant alors plus de traitements. Notre point de départ est l'ensemble des valeurs RGB des pixels. Nous cherchons donc à n'utiliser que celles-ci pour nos processus de modélisation et mise à jour du fond.

La remarque est sensiblement la même en ce qui concerne le degré de hiérarchisation de notre algorithme. Une importante hiérarchie de traitements entraine forcément plus de traitements. Cela dit nous n'excluons pas une approche hiérarchique, plus précise et plus robuste.
Nous pouvons d'ores et déjà affirmer qu'un traitement au niveau 'objet' n'est pas forcément nécessaire, dans la mesure où nous savons ce que nous voulons segmenter et détecter, l'utilisateur, qui est notre seul et unique *foreground*. En revanche, un traitement au niveau image ou *frame* nous semble pertinent car, d'après notre état de l'art, les approches employant ce niveau de traitement peuvent faire face aux changements importants, particulièrement concernant l'illumination, au sein de la scène.

Concernant les aspects matériels et logiciels de notre système, nous disposons bien sûr de ressources limitées, quoiqu'importantes. Chaque flux vidéo est traité par un ordinateur disposant de ressources matérielles et logicielles importantes et le processus de capture à proprement parler est exécuté sur une machine dédiée. Chaque machine traitant un flux vidéo ne s'occupe donc que de la gestion du fond et de l'extraction des silhouettes. Nous disposons donc de ressources pour exécuter correctement notre processus, toujours en respectant la contrainte temps réel bien évidemment.
Nos caméras sont fixes et ne sont pas supposées bouger au cours de l'interaction. Il existe également peu de raisons pour qu'elles soient soumises à un tremblement particulier. En revanche, toute fluctuation vidéo est à prendre en compte, bien que n'arrivant très rarement.

La scène intérieure que nous observons est une unique salle de classe. L'utilisateur est le seul sujet d'intérêt, donc *foreground*, à extraire de la scène. L'utilisateur peut effectuer tous types de mouvements : périodiques, erratiques, lents, rapides, etc.
La scène est dynamique de par la suppression du contrôle initial du *background*. Le fond n'est plus uni, comprend des zones sombres ou éclairées, avec des contrastes plus ou moins importants. En plus de l'utilisateur, plusieurs observateurs mobiles peuvent se trouver dans la scène. De plus, la scène peut comprendre certains objets qui peuvent être manipulés pendant l'interactivité (pas par l'utilisateur, en tout cas dans ce contexte de thèse).
L'éclairage initial ne permet pas la projection d'une scène 3D et donc l'immersion de l'utilisateur au sein de celle-ci. Il est donc nécessaire de diminuer cet éclairage. Cependant, la scène est sombre et les images vidéo de qualité relativement médiocre. Dans le cas d'une scène peu éclairée, les couleurs et les contrastes ne se distinguent plus. Concernant les conditions d'éclairage, un compromis doit donc être établi pour employer un éclairage permettant l'immersion de l'utilisateur et la distinction suffisante de celui-ci au sein de la

scène. De plus, la salle de classe donnant partiellement sur l'extérieur, les conditions d'illumination peuvent varier au cours de l'interactivité. Donc, le processus de modélisation et de soustraction de fond doit pouvoir gérer l'éclairage dynamique défini par ces dernières remarques.
Pour finir, dans le contexte d'un éclairage moins puissant qu'initialement, l'utilisateur génère un certain nombre d'ombres, que le processus de modélisation et de soustraction doit éventuellement prendre en compte.

2. Etapes du processus
La scène pouvant être dynamique au cours de l'interactivité, nous nous dirigeons vers une approche comportant une étape de mise à jour. L'initialisation du modèle doit bien sûr avoir lieu, mais nous souhaitons que cette étape ne soit pas trop longue et rébarbative, pour de simples raisons de praticité. Pour finir, dans notre cas, le processus d'extraction de l'utilisateur vis-à-vis du fond est déjà plus ou moins établi. Le fond est soustrait à l'image courante et la silhouette de la zone détectée est calculée. Nous ne cherchons pas ici à modifier ces deux procédures. Avant celles-ci, toute étape supplémentaire pour fiabiliser le résultat est envisageable.

3. Les différentes méthodes
Même si relativement bref, l'état de l'art présenté au cours de la section précédente nous permet de nous positionner par rapport aux différentes approches exposées.

La première classe, la modélisation classique du fond, regroupe un ensemble de méthodes trop simples, ne permettant pas de faire face au dynamisme potentiel au sein de la scène observée. L'algorithme initial, utilisé dans le cadre du ***Cyberdôme***, se situe dans cette classe. C'est justement ce que nous voulons améliorer. Nous ne nous dirigerons donc pas vers ce type de solutions.

La modélisation du fond par le biais d'une unique distribution gaussienne par pixel est peut-être une modélisation un peu trop simple, quoiqu'apparemment suffisante dans le cas d'une scène intérieure. Cependant, cette modélisation fait difficilement face à un dynamisme de scène important, aux changements rapides au sein de celles-ci et reste sensible vis-à-vis des problèmes générés par une variation d'illumination.
La modélisation du fond par le biais de MoG a permis de grandement fiabiliser le processus et a prouvé son efficacité par le biais de l'élaboration de plusieurs systèmes performants, expérimentés dans des cadres réels. Cette modélisation est bien adaptée aux scènes extérieures et permet de gérer un dynamisme important. De plus, la modélisation par le biais de MoG est très étudiée, il existe donc un grand nombre d'améliorations et d'extensions, qui ont prouvé leurs pertinence et efficacité à leur tour. Le processus global a été optimisé, fiabilisant les résultats ; différents espaces couleur ont été employés, permettant de rendre le système capable de détecter et segmenter les ombres ; la modélisation a été supportée par la fusion de plusieurs informations, augmentant la précision des résultats ; et la hiérarchisation des traitements a permis d'accroitre la robustesse du processus, tout en permettant à ce dernier de faire face à plus de dynamisme au sein de la scène. Cependant, les modélisations via MoG ne permettent pas forcément au système de gérer les changements rapides (illumination et mouvements) au sein de la scène. De plus, plus les méthodes se complexifient, plus elles peuvent s'avérer coûteuses en termes de mémoire et de traitement, ne permettant pas alors d'obéir à la contrainte temps réel.

Concernant l'adoption de lois statistiques, autres que la loi normale, il est possible de trouver les solutions équivalentes dans la littérature, en tout cas aux méthodes modélisant le fond via MoG, améliorations comprises. Cela dit, l'adoption d'une modélisation via MoG n'est pas innocente. Vis-à-vis de ces dernières techniques, l'adoption d'autres distributions entraine presque toujours une complexification du processus global du processus de modélisation et de soustraction du fond. De plus, une autre loi est souvent moins évidente à mettre en œuvre, la loi normale étant très connue et facile à implémenter.
L'adoption d'une distribution particulière permet, dans le cas où celle-ci est vérifiée, d'obtenir des modèles de fond précis et implique une extraction relativement simple des objets composant l'avant-plan de la scène. Cependant, cette adoption va de paire avec celle d'un modèle prédéfini, qui peut ne pas refléter la vérité terrain et nécessite une phase d'apprentissage particulière pour déterminer les différents paramètres de ce modèle. Ce qui n'est pas le cas si le concepteur décide ne pas faire d'hypothèses sur la distribution suivi par les pixels du fond. Cette forme de modélisation statistique est également efficace et les méthodes regroupées dans cette classe permettent souvent d'obtenir de meilleurs résultats que les méthodes employant une distribution particulière. Cela dit, ces méthodes sont plus complexes, plus difficiles à implémenter et plus coûteuses et donc pas forcément adaptées à la contrainte temps réel.
De manière générale, les modélisations statistiques sont très intéressantes et très répandues, donc très étudiées et améliorées. Elles sont précises et robustes et permettent d'acquérir le modèle de tous types de scènes. De plus, les méthodes basées sur ce type de modélisation peuvent faire face à différents degrés de dynamisme au sein de la scène observée. En revanche, ces méthodes basées statistique sont coûteuses, en termes de mémoire et de traitement. Elles nécessitent une phase d'apprentissage particulière, plus longue et plus complexe. Enfin, une méthode est généralement dédiée à la résolution d'un dynamisme particulier. Autrement dit, il n'existe pas de méthodes résolvant tous les problèmes.

Les techniques basées modélisation floue sont particulièrement récentes et manquent donc peut être encore d'études pour prouver leur réelle efficacité. Cela dit, ces méthodes ne semblent pas difficiles à mettre en œuvre. Un des aspects très intéressant de ce type de techniques est que le processus entier est adapté au niveau du pixel : mesures des similarités, calcul de l'intégrale floue, modélisation et mise à jour du fond. Le désavantage associé est bien sûr que l'ensemble des traitements est répété autant de fois qu'il y a de pixels dans l'image. Nous pouvons donc nous poser la question du respect de la contrainte temps réel de ce type de méthodes, dans le cas d'images de dimensions importantes à traiter. Enfin, ces méthodes ne permettent *a priori* pas forcément de faire face à des degrés importants de dynamisme au sein de la scène.

Concernant les méthodes d'estimation du fond basées filtres prédictifs, les remarques sont sensiblement les mêmes que pour les méthodes statistiques adoptant une loi de distribution particulière. Ces techniques ont souvent montré leur efficacité dans le cas de plusieurs développements. Cela dit, un modèle doit être choisi et vérifié au préalable et le processus d'extraction du *foreground* nécessite une phase d'initialisation, visant à calculer les différents paramètres du modèle prédictif.
Les autres méthodes d'estimation du fond, qui se basent sur les observations passées de la scène, sont, à l'instar des techniques statistiques, très étudiées et étendues. Il est effectivement possible de rencontrer dans la littérature un très grand nombre de travaux, accompagnés de tout autant d'améliorations : optimisation & fiabilisation des processus, utilisation de plusieurs espaces couleur, fusion d'informations diverses, hiérarchisation des traitements, etc.

La plupart des méthodes mettent à jour le fond à partir uniquement de la *frame* précédente, ce qui peut se révéler insuffisant dans certains contextes d'observation.
Les méthodes basées estimation du fond sont nombreuses et permettent de répondre à de nombreuses problématiques, à l'instar des méthodes statistiques. Ces méthodes n'ont pas de désavantages globaux : il existe toujours une meilleure solution, permettant de résoudre un problème particulier. Par rapport aux modèles statistiques, les méthodes basées estimation du fond permettent peut être plus d'atteindre le temps réel et de faire face à des situations d'observation un peu plus extrêmes. De plus, du fait qu'elles n'impliquent pas de calculs statistiques, ces méthodes se révèlent plus faciles à comprendre et assimiler et moins coûteuses en termes de traitement et de mémoire.

4. Evaluation d'un algorithme
Chaque solution permettant d'évaluer un algorithme répond à un objectif et un contexte d'observation particuliers. Aucune solution ne permet d'évaluer tous les aspects d'une méthode de modélisation et de soustraction du fond. La meilleure évaluation est encore l'évaluation visuelle du résultat d'une méthode (extraction du *foreground*), en fonction des objectifs et des besoins des concepteurs.

5. Notre conclusion
Nous nous dirigeons dans ces travaux vers une méthode basée estimation de fond. Ces méthodes nous semblent en effet plus faciles à prendre en main, assurant un processus de modélisation et de soustraction de fond rapide et moins coûteux. Les travaux de [Yang & al. 04a] nous semble un excellent point de départ. Cette solution est complète, rapide, robuste, très intuitive, n'impliquant pas obligatoirement de phase d'apprentissage et permet de répondre à un ensemble de situations problématiques diversifiées.
Concernant l'évaluation de notre algorithme, nous n'adoptons pas une technique particulière, l'ensemble des méthodes présentées précédemment étant tout à fait envisageable. Cela dit, notre contexte d'observation est particulier et nous ne demandons pas à ce que la méthode adoptée réponde à toutes les problématiques. Nous évaluons notre algorithme vis-à-vis de séquences d'images que nous avons enregistrées, dans notre contexte d'observation particulier.

3.5.1.3. Contribution

L'algorithme développé est directement inspiré des travaux de Yang [Yang & al. 04a, Yang & al. 04b]. Nous décrivons ici l'algorithme tel qu'il est développé et exécuté au sein du ***Cyberdôme***. Cet algorithme a été décrit dans [Picard & Estraillier 08ab] et est résumé par la Figure 4.37.

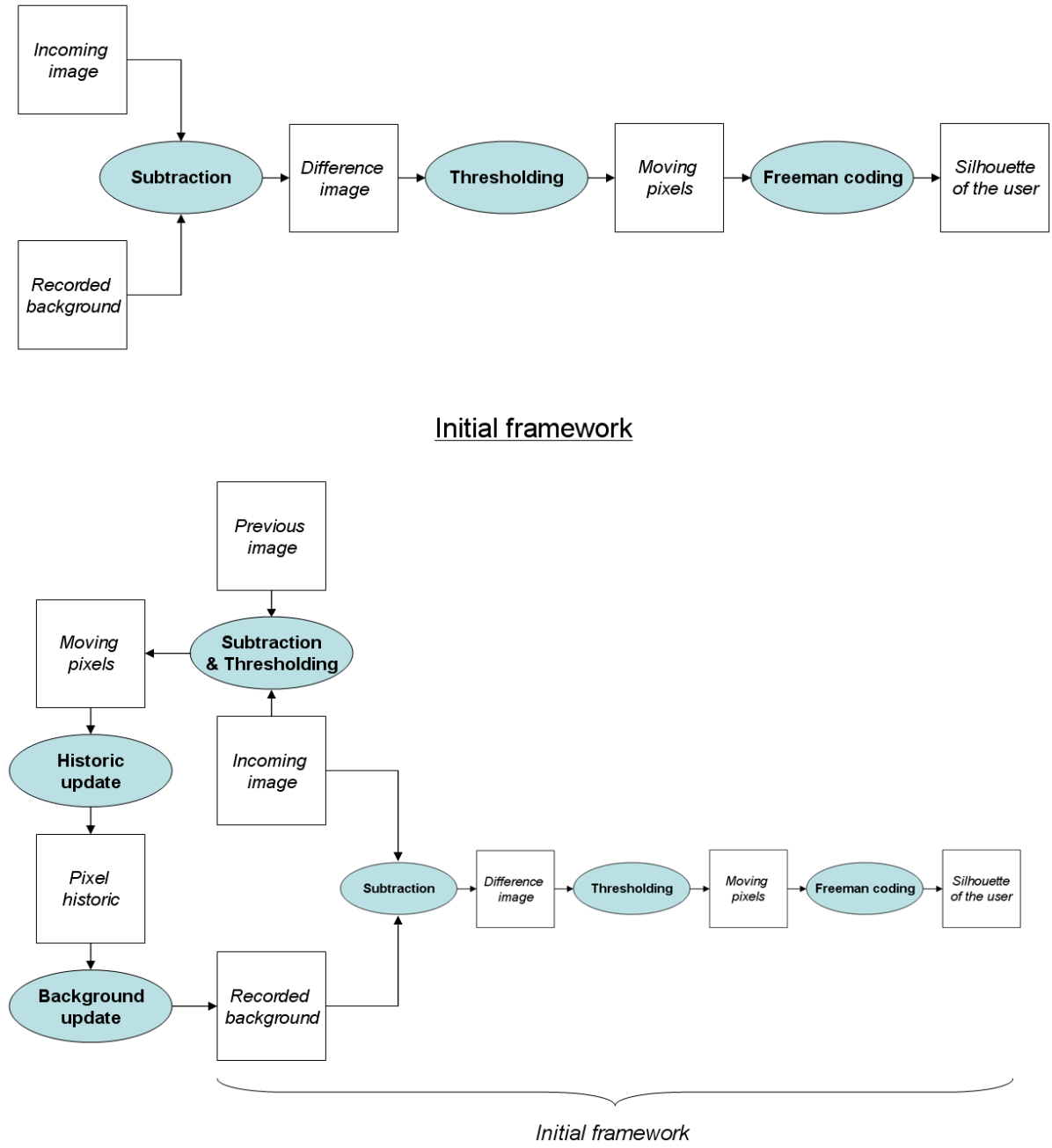


Figure 4.37. : Processus initial et modifié de segmentation des silhouettes de l'utilisateur

1. Modèle du fond

Le fond de la scène est une **image en deux dimensions**, dont les valeurs des pixels sont exprimées **dans l'espace RGB**.

2. Initialisation du fond

Une image de la scène, vide qui plus est (i.e. sans l'utilisateur au sein de celle-ci) **n'est pas forcément nécessaire** pour initialiser le processus.

3. Mise à jour du fond
Le *background* est mis à jour par cet algorithme **au niveau du pixel**, et **au niveau de l'image** dans sa globalité.

La mise à jour du fond, **au niveau pixel**, commence par le calcul de **l'image différence** entre les *frames* courante et précédente (valeurs RGB des pixels). Un OU logique est alors appliquée sur l'image différence, nous permettant ainsi de détecter les pixels ayant évolué d'une *frame* à une autre.
L'algorithme tel qu'il est présenté dans les travaux de Yang stipule qu'une *frame* passée, antérieure à celle immédiatement précédente, peut être considérée. En considérant la *frame* précédente, tout mouvement, qu'il soit du à l'utilisateur ou à un dynamisme autre au sein du *background*, est immédiatement détecté.
Le seuil associé au OU logique est déterminé au cours de phases d'expérimentations. Un compromis doit être établi entre un seuil trop sévère, entrainant une perte de détection au niveau du mouvement engendré par l'utilisateur, et un seuil trop souple, qui aura pour conséquence une détection de mouvements parasites.

Entre cette image différence et le fond modélisé se trouve **une matrice dynamique 2D** intermédiaire.
L'image différence indique si la valeur d'un pixel a évolué depuis la *frame* précédente, fixant ainsi la valeur du pixel correspondant, au sein de la matrice dynamique, à un seuil, déterminé au cours de phases d'expérimentations. Ce seuil traduit le nombre de *frames* pendant lequel le pixel n'interviendra pas dans le processus de mise à jour du fond.
Au cours de la *frame* suivant celle pendant laquelle un pixel a été détecté comme mobile, le processus observe la valeur de ce pixel au sein de l'image différence. S'il caractérise de nouveau un mouvement au sein de la scène, sa valeur au sein de la matrice dynamique reste ou devient égale au seuil mentionné précédemment. Dans le cas contraire (le pixel n'a pas 'bougé'), sa valeur au sein de la matrice dynamique est décrémentée de 1 *frame*. Ainsi, pour une *frame* particulière, la matrice dynamique indique le nombre de *frames* depuis lequel la valeur d'un pixel donné a cessé de caractériser du mouvement au sein de la scène.

La matrice dynamique est utilisée par le processus de mise à jour du fond. Pour une *frame* particulière, la valeur d'un pixel, appartenant au fond modélisé, est mise à jour si sa valeur, au sein de la matrice dynamique, non nulle au cours de la *frame* précédente, devient nulle. Autrement dit, si un pixel n'a pas traduit de mouvement au sein de la scène au cours d'un nombre donné de *frames*, sa valeur au sein du *background* est mise à jour. La valeur seuil, attribuée à un pixel donné lorsque celui-ci traduit du mouvement au sein de la scène, dépend des gestuelles observées. Un seuil compris entre 5 et 10 *frames* permet une mise à jour rapide du fond et, donc, de faire face à des mouvements rapides au sein de la scène.
La valeur d'un pixel du fond résulte de la combinaison linéaire de son ancienne valeur (au sein du fond) avec sa valeur au sein de l'image courante. Le coefficient linéaire traduit la vitesse avec laquelle un pixel du fond sera remplacé par le pixel correspondant au sein de l'image courante. Dans le cadre de nos travaux, nous remplaçons directement le pixel mis à jour du fond par le pixel de l'image courante.

La Figure 4.38. illustre le processus de modélisation du fond et l'évolution de la matrice dynamique et du *background*, au cours de *frames* successives.

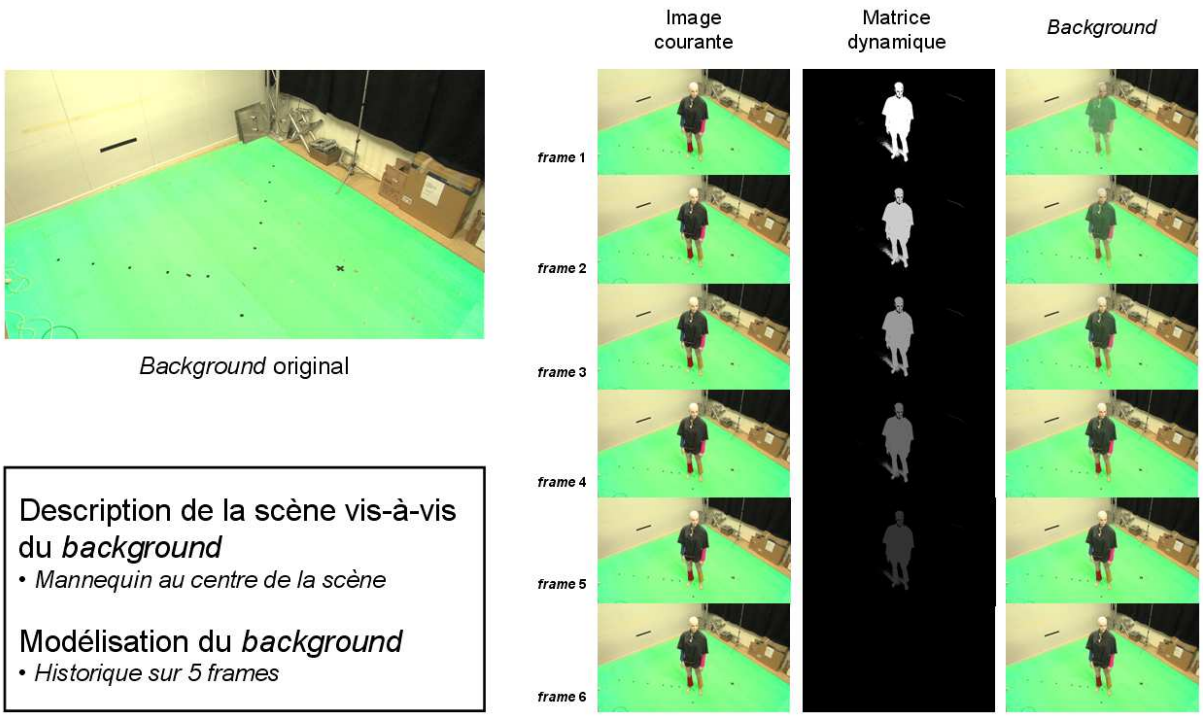


Figure 4.38. : Processus de modélisation dynamique du fond

La mise à jour **au niveau de l'image** s'effectue au cours de chaque nouvelle *frame*. Elle consiste, tout d'abord, à calculer le pourcentage de pixels mobiles, vis-à-vis du nombre total de pixels, au sein de l'image courante, par le biais de l'image différence. Si ce pourcentage est faible (en dessous d'un certain seuil, déterminé au cours de phases d'expérimentations), traduisant ainsi le fait que peu de mouvement est observé au sein de la scène, l'image de fond est tout simplement remplacée immédiatement par l'image courante.

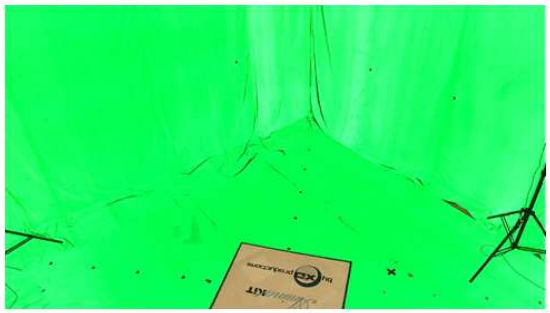
4. Extraction du sujet d'intérêt

Une fois le fond modélisé, la soustraction de ce dernier à l'image courante, suivi d'un OU logique, permet **d'extraire l'utilisateur et d'en calculer la silhouette**. Le processus de segmentation des silhouettes est le même que celui établi initialement par le ***Cyberdôme***.

Le processus de modélisation du fond comporte toutefois un inconvénient non négligeable. Si l'utilisateur cesse d'être mobile au cours de l'interactivité, **il est alors évidemment mis à jour dans le fond**. Il n'est ainsi plus détecté et segmenté, le processus de capture devenant alors impossible. De plus, s'il redevient mobile, le processus de segmentation des silhouettes sera inefficace et non significatif, dans la mesure où l'utilisateur fait dorénavant partie intégrante du fond.

Pour faire face à ce problème, nous introduisons, au sein du processus, l'initialisation et la mise à jour d'une zone 2D entourant l'utilisateur. Le processus ne met alors pas à jour le fond au sein de cette zone 2D, empêchant l'utilisateur d'être mis à jour au sein du *background*. Par la suite, le processus de segmentation s'exécute au sein de cette zone.

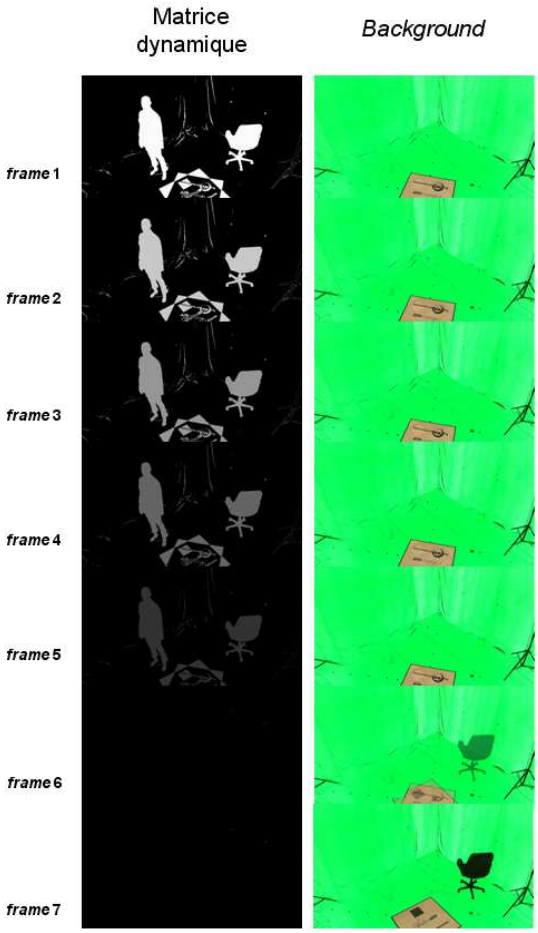
Pour chaque *frame*, et chaque point de vue, la zone 2D englobant la silhouette de l'utilisateur, **SAS** (*Safe Area around Segmentation*), est calculée. Il en est de même pour la zone 2D englobant la silhouette projetée du modèle 3D, **SAM** (*Safe Area around Model*). Ces deux zones sont fusionnées en permanence, calculant ainsi **la zone 2D SASM** (*Safe Area around Segmentation & Model*), englobant, à coup sûr, l'utilisateur.
L'exploitation de la zone SASM permet la modélisation optimale du fond de la scène, tout en assurant la segmentation de l'utilisateur, même si ce dernier reste immobile. La Figure 4.39. illustre, par le biais de 2 exemples, l'utilisation de la zone SASM dans le processus de modélisation du fond.



Background original

Description de la scène vis-à-vis du *background*
- *Mannequin (sujet d'intérêt)*
- *Tapis tourné*
- *Siège*

Modélisation du *background*
- *Historique sur 5 frames*



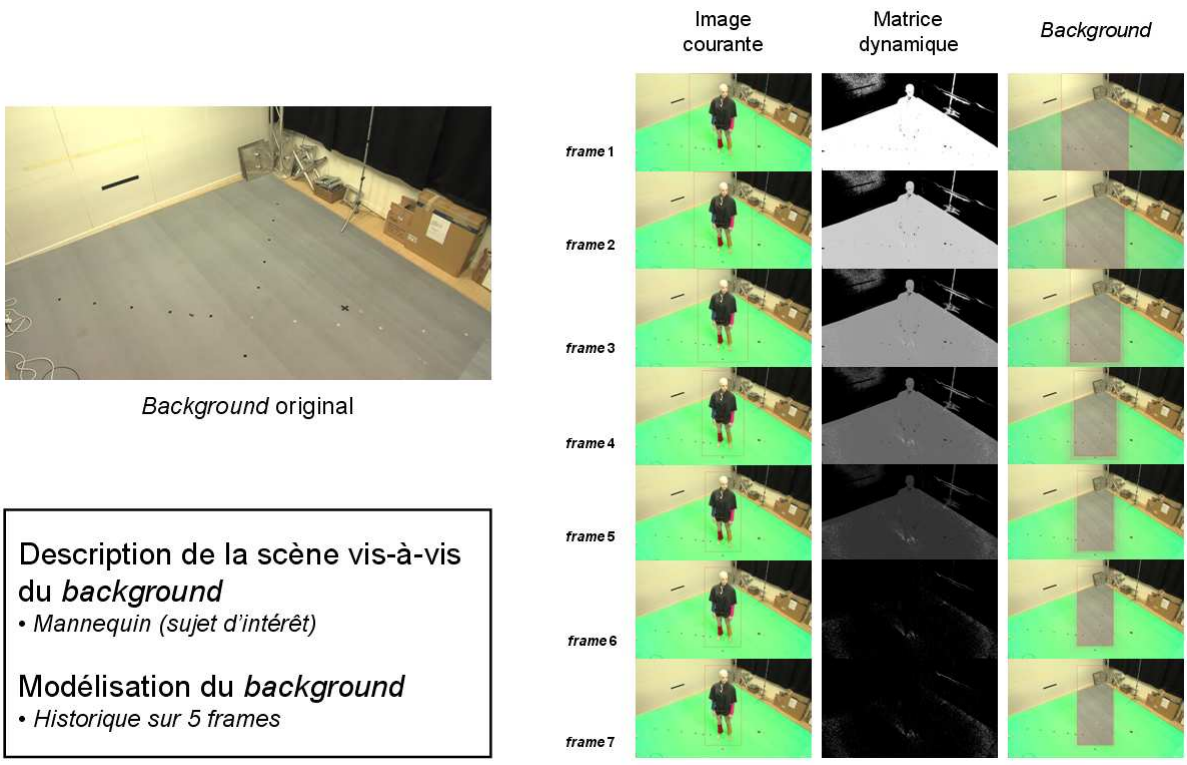


Figure 4.39. : Processus de modélisation dynamique du fond – Prise en compte de la zone SASM

Les dimensions de la zone SASM s'adaptent également en fonction des gestuelles de l'utilisateur. Entre deux *frames* consécutives, les dimensions de la zone sont comparées. Un fort écart entre celles-ci indique une gestuelle importante de la part de l'utilisateur entre ces deux *frames*. Les dimensions de la zone SASM sont alors volontairement augmentées d'une longueur de sécurité, déterminée au cours de phases d'expérimentations. Il est ainsi assuré que la zone SASM englobe complètement l'utilisateur. Au cours des *frames* suivantes, si l'utilisateur bouge moins, voire devient immobile, les dimensions de la zone SASM diminuent progressivement de manière à ce qu'elles soient les plus petites possibles (mise à jour du fond 'autant que possible').

3.5.2. Marqueurs

Au sein du ***Cyberdôme***, l'utilisateur doit porter sur lui **4 marqueurs de couleur**, un à chaque bras et un à chaque jambe. Ces marqueurs sont des bandes de tissu coloré, chaque marqueur étant d'une couleur différente. Le système sait *a priori* quelle couleur est associée à quel membre de l'utilisateur. Ces marqueurs permettent la distinction immédiate et sans ambigüité des bras et des jambes de l'utilisateur, ainsi que la droite et la gauche de ce dernier.
D'après nos objectifs, la capture des gestuelles de l'utilisateur doit être **non invasive**, i.e. celui-ci ne doit avoir à ne porter aucun équipement matériel supplémentaire. C'est pourquoi nous désirons supprimer, ou tout du moins l'alléger au maximum, la contrainte que constitue le port de ces marqueurs.
Nous pensons que **la squelettisation des silhouettes 2D** de l'utilisateur nous permettra de construire de nouvelles connaissances, de manière à atteindre cet objectif. En effet, une fois le squelette de l'utilisateur déterminé pour un point de vue donné, il nous sera possible d'une

part d'extraire les différentes extrémités et articulations, 1er pas vers une identification sémantique des parties du corps correspondantes de l'utilisateur, et d'autre part d'avoir un support supplémentaire pour mieux capturer les gestuelles de celui-ci.
Nous verrons que notre approche ne suffit pas en elle-même à résoudre le problème du port des marqueurs. L'apport au sein du système d'informations contextuelles, en fonction du scénario de l'application, nous permettra de nous rapprocher plus de notre objectif. Nous exposons notre méthode au cours du chapitre suivant.

La Section 3.4.2.1. proposera **un état de l'art** sur les différentes méthodes existantes pour squelettiser les silhouettes de l'utilisateur. Après nous être positionné par rapport à ce dernier au cours de la Section 3.4.2.2., **notre contribution** vis-à-vis de cette problématique sera présentée (Section 3.4.2.3.).

3.5.2.1. Etat de l'art

Au cours de cette section, nous allons proposer **un état de l'art** concernant la problématique de **la squelettisation d'objets**, tels que des silhouettes dans un espace en 2D. Cette problématique est depuis longtemps étudiée et il existe un nombre très important de recherches traitant de celle-ci. Notre état de l'art est basé sur plusieurs sources [Ivanov & al. 00, Corlouer & al. 08, Girard 08].
La squelettisation est utilisée dans de nombreux domaines d'application (voir Fig. 4.40.), de la médecine à la biologie en passant par la minéralogie, l'architecture et l'urbanisme. Ce procédé est évidemment grandement étudié dans les domaines relatifs à l'image : étude et la reconnaissance de formes en particulier au niveau des écritures manuscrites ou typographiques (OCR), modélisation de solides pour la conception et la manipulation de formes, organisation de nuages de points, recherche et planification de parcours, animation 3D, réalité virtuelle (VR), compression-décompression d'images, indexation d'images ou de modèles 3D dans les bases de données, analyse morphologique, etc.

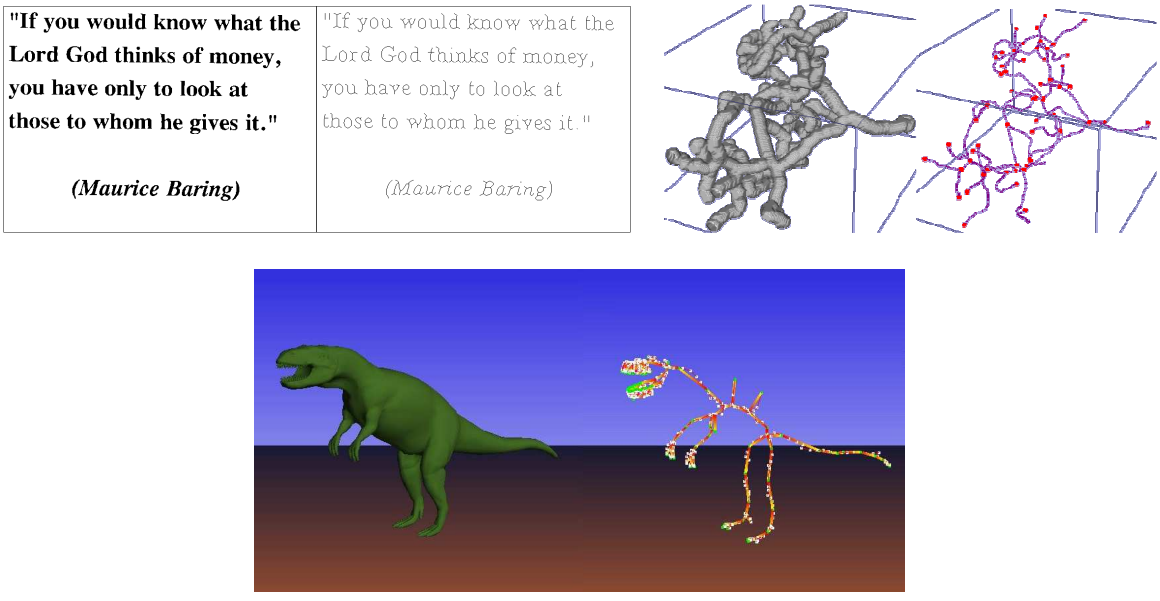


Figure 4.40. : Quelques domaines d'application de la squelettisation

La section suivante présentera les différentes définitions du squelette d'un objet et exposera les différents aspects et concepts qu'englobe cette problématique. Différentes méthodes seront alors présentées au cours de la Section 3.4.2.1.2.

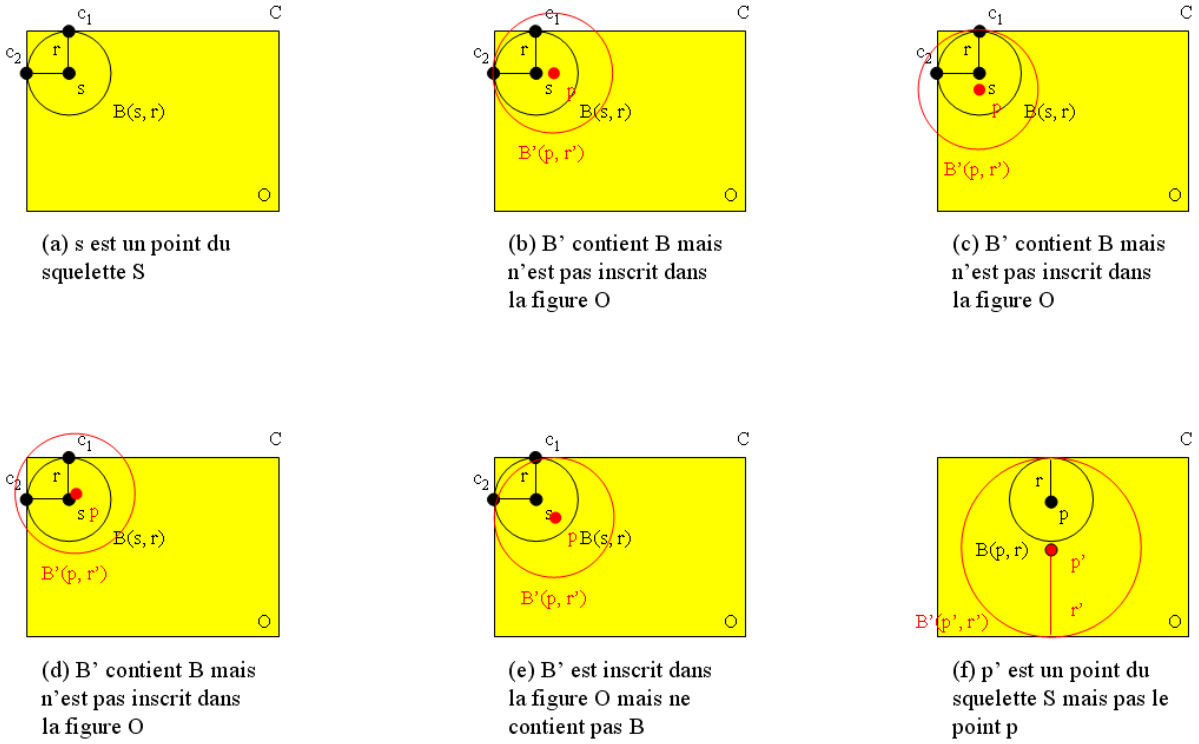
3.5.2.1.1. Définitions et concepts associés

1. Les deux définitions mathématiques du squelette d'un objet
Le squelette d'un objet peut être défini de la façon suivante :

> Soit un objet O, borné et topologiquement fermé, représenté dans un espace de dimension n.
> Soit C, l'ensemble des points qui constituent la forme, le bord, le contour de l'objet O.
> Soit d(x, y), une métrique ou fonction de distance permettant de mesurer la distance entre un point x et un point y de l'espace.
> Soit DT(d, p, C), la distance d'un point p, appartenant à O, à C, selon la métrique d.
>
> Soit S l'ensemble des points du squelette de O. s est un point du squelette S de l'objet O, si il existe m points c_i de C, $1 \leq i \leq m$, avec $m \geq n$, tels que :
>
> $$DT(d, s, C) = d(s, c_1) = d(s, c_2) = \ldots = d(s, c_n)$$

Cette définition est généralisée à trois niveaux : au niveau de **la dimension de l'espace de représentation** ; au niveau de **la métrique d utilisée** ; et au niveau de **la transformée en distance DT**. Il est donc important de préciser ces aspects en fonction du problème étudié.
Premièrement, l'espace de représentation peut être celui d'une image, donc de dimension 2 ; celui du monde réel ou d'un monde virtuel donc de dimension 3, etc.
Deuxièmement, il existe de nombreuses métriques permettant le calcul de la distance entre deux points : la distance euclidienne qui est la plus connue ; la distance de Manhattan ; la distance de Mahalanobis, etc.
Enfin, par le biais d'une métrique d, il est possible de mesurer toutes les valeurs de distance entre un point p de O donné et tous les points c de C. La transformée en distance DT est la sélection spécifique d'une de ces valeurs. Ainsi, elle peut par exemple être définie comme la valeur de distance maximale (le choix le plus logique et donc usité) ; la valeur de distance minimale, etc.

Le squelette d'un objet peut également se définir d'une manière légèrement différente. Elle est basée sur le constat suivant. D'après la définition précédente, un point s du squelette S est le centre d'une boule, notée B(s, r), de rayon r = DT(d, s, C) tangente en m points c_i de C, $1 \leq i \leq m$, avec $m \geq n$, n étant la dimension de l'espace de représentation. Le constat est qu'il n'existe pas de boule B'(p, r'), avec $p \neq s$, $r' > r$, telle que $B(s, r) \subset B'(p, r') \subset O$, avec O l'objet d'étude. La boule B est alors appelée la boule d'inclusion maximale, ou la boule maximale. La Figure 4.41. illustre ce constat avec un rectangle dans un espace à deux dimensions. La métrique choisie est la métrique euclidienne tandis que la DT donne la valeur de distance minimale d'un point p appartenant à O par rapport à l'ensemble des points du contour C.



(a) s est un point du squelette S

(b) B' contient B mais n'est pas inscrit dans la figure O

(c) B' contient B mais n'est pas inscrit dans la figure O

(d) B' contient B mais n'est pas inscrit dans la figure O

(e) B' est inscrit dans la figure O mais ne contient pas B

(f) p' est un point du squelette S mais pas le point p

Figure 4.41. : Boule d'inclusion maximale

La figure (a) illustre le concept du disque B(s, r) inscrit dans le rectangle O, dont le centre est un point s du squelette S et le rayon r = d(s, c_1) = d(s, c_2) = DT(d, s, C). Les figures (b), (c), (d) et (e) illustrent le fait qu'il n'existe pas de disque B'(p, r'), avec $p \neq s$, $r' > r$ qui soit à la fois inscrit dans le rectangle O et qui contienne B(s, r). Dans la figure (f), B' est une boule d'inclusion maximale (donc p' est un point du squelette) alors que B ne l'est pas.

Grâce au constat préalablement exposé, **il est possible alors de définir le squelette d'un objet de la façon suivante :**

> Soit un objet O, borné et topologiquement fermé, représenté dans un espace de dimension n.
> Soit C, l'ensemble des points qui constituent la forme, le bord, le contour de l'objet O.
> Soit d(x, y), une métrique ou fonction de distance permettant de mesurer la distance entre un point x et un point y de l'espace.
> Soit DT(d, p, C), la distance d'un point p, appartenant à O, à C, selon la métrique d.
>
> Soit S l'ensemble des points du squelette de O. s est un point du squelette S de l'objet O, s'il est le centre d'une boule B(s, r) d'inclusion maximale avec r = DT(d, s, C).
> B(s, r) est tangent à C en m points c_i, $1 \leq i \leq m$, avec $m \geq n$.
> O est alors la réunion des boules B(s, DT(d, s, C)) pour tous les points s de S.

La Figure 4.42. illustre les deux définitions que nous venons de donner du squelette d'un objet. Il est à remarquer que bien que très similaires et corrélées, les squelettes décrits par ces deux définitions peuvent ne pas être égaux. Il est clair que le squelette S_1 décrit par la

première définition est inclus dans S_2, squelette décrit pas la deuxième définition. Mais, par exemple dans le cas d'un rectangle, les coins de ce dernier appartiennent à S_2 et non à S_1.

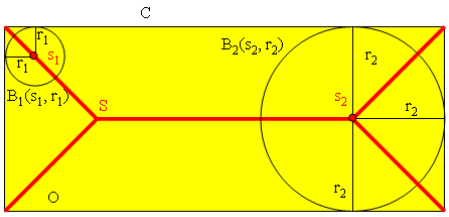


Figure 4.42. : Squelette d'un rectangle

A noter que la squelettisation s'effectue aussi bien à l'intérieur qu'à l'extérieur de la forme de l'objet d'étude. Les **endosquelettes** (désignés plus communément par le terme « squelettes »), centrés à l'intérieur de la forme, et **exosquelettes**, à l'extérieur, sont généralement calculés simultanément. Les endosquelettes décrivent la topologie et les métriques de l'objet tandis que les exosquelettes traduisent les relations d'adjacence entre l'objet et ses voisins. Ainsi ces derniers peuvent être exploités pour des opérations telles que le regroupement d'objets ou la planification de chemins.

2. Processus de squelettisation

En accord avec notre problématique, nous nous cantonnerons dorénavant **au processus de squelettisation d'un objet en deux dimensions**, **dans un espace discrétisé** tel qu'une image vidéo. Cela dit, nos propos peuvent tout à fait s'appliquer dans le cas d'espaces de dimensions plus importantes.

Le processus de squelettisation réduit celui-ci en une représentation minimaliste comprenant les caractéristiques topologiques et géométriques essentielles de l'objet. Cet ensemble de données doit être traduit sous une forme simple à extraire et facilement manipulable.
Le résultat de cette réduction est appelé **un squelette** ou encore **axe médian** (MA pour *Medial Axis*). Le squelette est une représentation complémentaire à celle décrite par la forme globale de l'objet. En effet, tandis que la forme de l'objet décrit les frontières physiques de celui-ci, le squelette traduit les relations de proximité entre ces dernières (voir Fig. 4.43.). Des études ont montré que, dans le processus de perception visuelle humaine, notre sensibilité inconsciente est maximale au niveau du squelette.

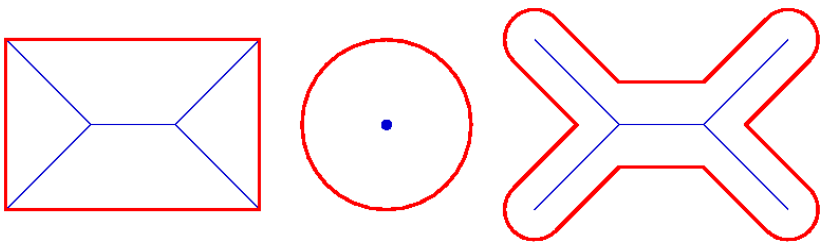


Figure 4.43. : Quelques exemples de squelettes de formes basiques, tirés de [Corlouer & al. 08]

Pour reprendre la terminologie adoptée jusqu'alors, soit O, l'objet étudié dans l'image et soit C son contour, à partir duquel sera effectué le processus de squelettisation. d(X, Y) est la métrique, ou fonction de distance, mesurant la distance entre le pixel X de coordonnées X(x_1, x_2) et le pixel Y de coordonnées Y(y_1, y_2). La transformée en distance DT(d, C) est une image en niveau de gris donnant, pour chaque pixel de l'image, sa valeur de distance par rapport au contour C de l'objet O, selon la métrique d. Cette valeur est généralement la valeur de la distance minimale mesurée entre un pixel donné de l'image et l'ensemble des pixels du contour C. Le squelette de l'objet se trouve alors le long des discontinuités de courbure de la DT. Autrement dit, si la transformée en distance est affichée comme une surface 3D, la troisième dimension représentant la valeur de la distance du point au contour C, alors les pixels du squelette sont ceux dont la valeur représente la crête de cette surface (voir Fig. 4.44.).

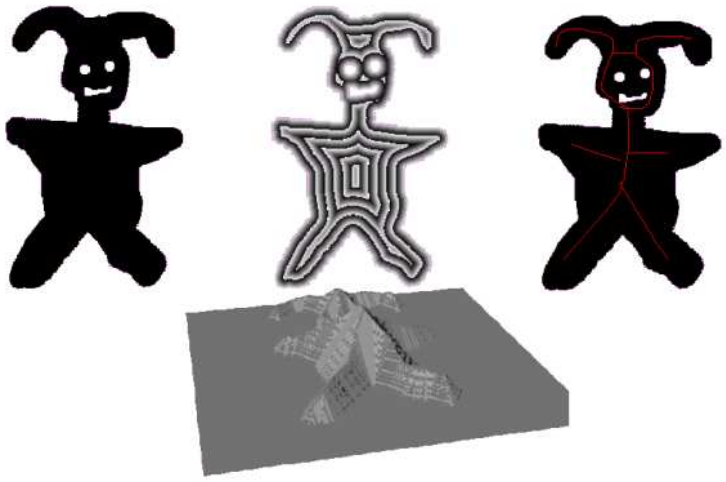


Figure 4.44. : De gauche à droite et de haut en bas : l'objet étudié ; la DT ; l'axe médian (en rouge) ; la DT représentée dans un espace en 3D. La crête de la surface (les discontinuités de courbure) permet d'identifier les pixels composant l'axe médian.

Il est alors possible de différencier **la transformée de l'axe médian** (MAT pour *medial axis transform*) et **l'axe médian** (MA pour *medial axis*) à proprement parler. Le MA peut être représenté par une image binaire, la valeur des pixels du squelette étant à 255, le reste étant à 0. La MAT peut être représenté par une image en niveaux de gris, représentant également le squelette mais dont la valeur des pixels indique la distance de ces derniers au contour de l'objet (voir Fig. 4.45.). La MAT est parfois appelé le squelette pondéré ou l'axe médian pondéré.

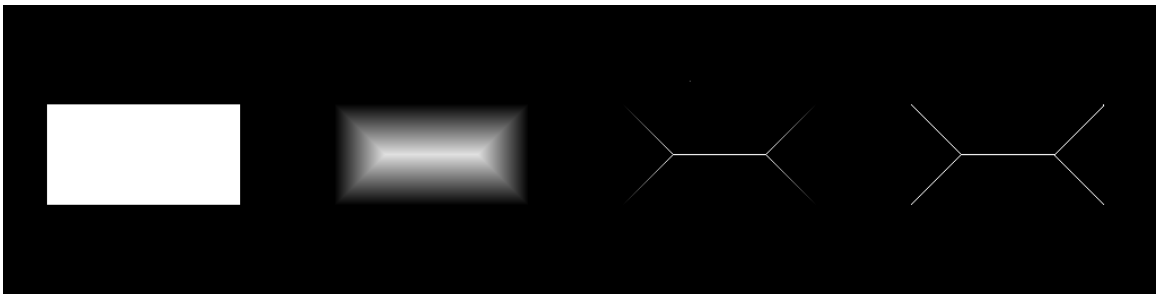


Figure 4.45. : De gauche à droite : l'objet étudié ; la DT (niveaux de gris) ; la MAT (niveaux de gris) ; le MA (binaire)

3. Propriétés & Limites

Le squelette doit représenter explicitement à la fois les propriétés **topologiques** (relations de connexité, etc.) et **géométriques** (ramifications, parties allongées, etc.) de l'objet. Autrement dit, certaines propriétés topologiques et géométriques sont conservées lors du calcul du squelette.

Les squelettes finaux sont **d'épaisseur minimale**. Par exemple, l'épaisseur d'un squelette d'objet 2D est de 1 pixel. En 3D, la squelettisation d'un objet pourra générer un squelette surfacique (*medial surfaces*), constitués d'un ensemble de voxels qui forment une surface d'épaisseur unité, ou encore un squelette curviligne, souvent appelé squelette homotopique, ensemble de voxels formant une courbe, dans l'espace, d'épaisseur unité.

Les squelettes sont théoriquement **invariants vis-à-vis de transformations linéaires** (translation, rotation, et changement d'échelle) effectuées sur l'objet d'étude.

Ils doivent permettre également **de reconstruire l'objet O**, propriété très étudiée dans le domaine de reconnaissance de formes.

Cela dit, la squelettisation étant une transformation semi-continue, la moindre perturbation au niveau des frontières de l'objet entraîne l'ajout au squelette final **de branches supplémentaires souvent indésirables** (voir Fig. 4.46.). C'est un des problèmes majeurs, l'idée étant alors de rendre 'insensible' le squelette calculé aux perturbations, même très petites, qui peuvent avoir lieu au niveau de la forme de l'objet d'étude. Cette opération de simplification est appelé **l'ébarbulage** (*pruning*), ou encore **la régularisation du squelette** (*regularization*).

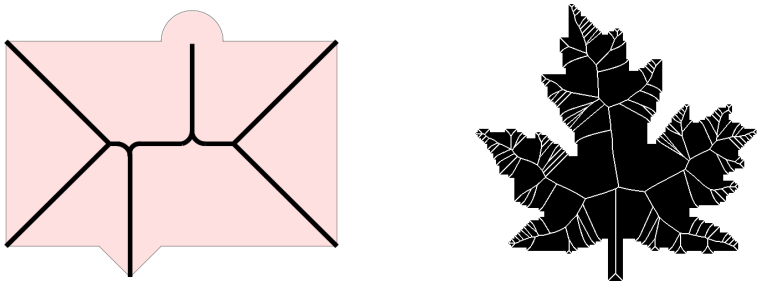


Figure 4.46. : Sensibilité du squelette aux perturbations du contour

Il est également fortement possible de rencontrer, lors du développement d'un processus de squelettisation, des problèmes **au cours du passage d'un espace continu dans un espace**

discret. En effet, les théories mathématiques derrière un algorithme de squelettisation s'expriment majoritairement dans un espace continu. Ces théories sont les mêmes dans un espace discret, tel qu'une image vidéo. Cependant, ce passage continu-discret nécessitent généralement des traitements supplémentaires pour faire face aux challenges, bien connus dans le domaine du traitement d'images, qu'il génère. Tout d'abord, le point d'intersection de deux droites en continu n'existe pas forcément en discret (voir Fig. 4.47.). Ce problème est connu et de plusieurs solutions permettent de le résoudre, en tout cas partiellement.

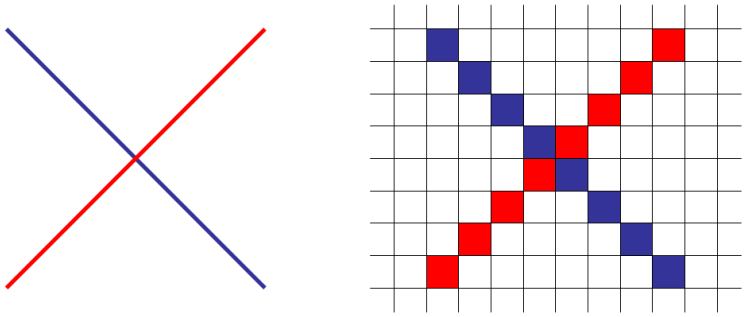


Figure 4.47. : Intersection de deux droites dans un espace continu (à gauche) et dans un espace discret (à droite)

Deuxièmement, une valeur flottante (« à virgule ») doit nécessairement être tronquée en une valeur de type entière, lorsqu'il s'agit d'une coordonnée pixel ou d'une valeur de pixel. Par exemple, un segment allant du point X(3.125, 4.668) au point Y(3.459, 5.112) sera en fait un segment de longueur de 1 pixel en (3, 5). Enfin, le concepteur aura à faire plusieurs choix lors de son développement, qui influenceront l'ensemble du processus : choix du type de connexité d'un pixel (4-connexité : les voisins du pixel considéré sont les 4 pixels horizontaux et verticaux ; 8-connexité : les voisins du pixel considéré sont les 8 pixels autour de ce dernier, etc.) ; choix de la métrique (une valeur de distance diagonale entre deux pixels consécutifs vaudra 1 eu utilisant la métrique discrète mais $\sqrt{2}$ en utilisant la métrique euclidienne) ; etc. Concernant le choix de la métrique, la distance euclidienne est considérée comme la 'vraie' distance entre deux points de l'image. Cela dit, bien qu'elle puisse être calculée de manière exacte ([Fabbri & al. 08] présente une étude comparative de 6 approches pour ce faire), cette distance est difficilement applicable dans un espace discret. Plusieurs approximations de celle-ci peuvent être alors utilisées : la *city-block distance*, la *chessboard distance*, la *distance pondérée (3, 4)*, la *distance pondérée (5, 7, 11)*, etc.

3.5.2.1.2. Classification des méthodes existantes

Amincissement topologique		[Xia 89]	√ Implémentation directe de l'analogie du feu de prairie dans l'espace discret X Pas de métrique euclidienne
Cartes de distances	Calcul des cartes	[Rosenfeld & Pfaltz 68]	√ Algorithme basée sur de la morphologie récursive. Construction de la carte en deux passes √ Utilise la métrique euclidienne
	Extraction du squelette	[Davies & Plummer 81]	√ Combinaison d'un algorithme d'amincissement topologique avec une méthode d'extraction du squelette √ Connexité des points du squelette X Hérite des inconvénients des méthodes basées amincissement topologique
		[Arcelli & Sanniti di Baja 85]	√ Extraction combinée avec un algorithme de suivi X Traitement des *saddle points*
		[Ivanov & al. 00]	√ Algorithme basé entiers √ Moins de problèmes relatifs à la discrétisation de l'espace √ Moins de coûts mémoire et traitement, simplification des calculs, diminution des vitesses d'exécution √ Possibilité de reconstruire la forme initiale de l'objet
		[Montanari 69]	√ Optimisation de l'extraction au niveau de l'ébarbulage √ Méthode efficace, applicable dans un espace discret X Connexité des points du squelette non préservée
		[Dill & al. 87, Pizer & al. 87]	√ Optimisation de l'extraction au niveau de l'ébarbulage √ Moins de sensibilité du squelette vis-à-vis des perturbations du contour X Processus exécuté sur plusieurs résolutions donc moins rapide et plus coûteux X Connexité des points du squelette effectuée en *post processing* supplémentaire
		[Ho & Dyer 86]	√ Optimisation de l'extraction au niveau de l'ébarbulage √ Moins de sensibilité du squelette vis-à-vis des perturbations du contour X Résultats peu intuitifs X Résultats trop ébarbulés dans le cas de formes complexes
Diagramme de Voronoï	Construction du diagramme (sites de type points 2D)	[Shamos & Hoey 75] ***Divide & Conquer***	√ Gestion du passage continu-discret √ Construction efficace et optimale du diagramme de Voronoï X Fusion des sous-diagrammes difficiles X Difficile à implémenter
		[Fortune 86] ***Sweepline***	√ Gestion du passage continu-discret √ Construction robuste, stable et optimale du diagramme de Voronoï √ Approche plus facile à implémenter
		[O'Sullivan 03]	√ Algorithme basé sur celui de [Fortune 86] √ Applications temps réel possibles
		[Inagaki & al. 92, Sugihara & Iri 94]	√ Robuste √ Diagrammes de Voronoï cohérents topologiquement
		[Hoff III & al. 00]	√ Accélération matérielle √ Implémentation simple √ Identification et énumération des approximations et sources d'erreurs (précision ajustable) √ Sites 2D et 3D √ Généralisation de l'algorithme au niveau des sites X Coûteux en termes de traitement et de calcul X Temps de calcul conséquents à partir d'une trentaine de sites (points 2D)
	Extraction du squelette	[Ogniewicz & Ilg 92, Ogniewicz & Kübler 95]	√ Optimisation de l'extraction au niveau de l'ébarbulage √ Robuste √ Insensibilité du squelette aux symétries et aux changements d'échelle X Coûteux en temps de calcul, en mémoire et en traitement (temps réel)
		[Gold & al. 99]	√ Optimisation de l'extraction au niveau de l'ébarbulage X Modification du contour de l'image

La plupart des méthodes visant à calculer le squelette d'un objet cherchent à simuler aussi précisément que possible une analogie bien connue : **l'analogie du feu de prairie** (*fire front paradigm*). Cette analogie, considérée dans un espace continu, permet de bien comprendre le processus de squelettisation d'une forme et traduit l'obtention d'un squelette idéal vis-à-vis d'un objet étudié.

En considérant une prairie dont les propriétés sont les mêmes en tout point, des feux (aux propriétés identiques et constantes : vitesse de propagation, intensité, etc.) sont allumés simultanément en tout point du contour de celle-ci. Le squelette de cette prairie est alors l'ensemble des points, les points d'extinction, où les feux se sont rencontrés.

[Blum 67] est à l'origine de cette analogie. Sa motivation première fut de fournir des descripteurs 2D stables de la forme d'objets qu'il qualifiait de « biologiques ». Sa définition de l'axe médian et de sa génération, est concise, simple et très intuitive.

De nombreux auteurs ont cherché à développer cette analogie. L'implémentation de cette dernière s'avère toutefois difficile, au sein d'un espace discret, typiquement une image vidéo

en 2D. Deux problèmes majeurs sont alors rencontrés : la connexité obligatoire des points appartenant au squelette et l'exactitude de la métrique euclidienne.
Le lecteur intéressé pourra se référer à [Smith 87], entre autres, pour un survol des méthodes permettant d'approximer l'analogie des feux de prairie.

Il est courant de classer les différentes méthodes existantes suivant 3 catégories : **les méthodes dites d'amincissement topologique** (*topological thinning*), encore appelées les méthodes topologiques ; **les méthodes basées carte de distances ou basées relief** ; **les méthodes basées calcul analytique**, parfois appelées les méthodes basées diagramme de Voronoï. Dans certains travaux, l'analogie du feu de prairie est considérée comme une catégorie à parti entière. Nous pensons que de nombreuses techniques, appartenant à une des trois catégories ci-dessus, cherchent à simuler cette analogie, faisant plus d'elle un concept, un objectif à atteindre, qu'une catégorie de méthodes donnée.

1. Amincissement topologique
Ces méthodes tendent à simuler au mieux l'analogie du feu de prairie. En adoptant cette technique pour extraire l'axe médian d'un objet, la pertinence topologique des pixels de ce dernier est étudiée, plutôt que les propriétés métriques de sa forme. A partir du contour de l'objet, les pixels sont testés les uns après les autres puis retirés, dans la mesure où leur suppression n'altère pas les caractéristiques topologiques de la forme amincie de l'objet. Ces opérations sont effectuées grâce à des opérateurs morphologiques ainsi qu'à leurs diverses combinaisons (érosion, dilatation, ouverture, fermeture), appliqués typiquement sur des images binaires.
Cette méthode est rapide, relativement fiable dans le cas de silhouettes allongées. De plus, elle garantit la connexité du squelette. Toutefois, le résultat d'un amincissement topologique d'un objet de diamètre important donne des résultats peu intuitifs pour l'œil humain. En effet, la topologie d'une grille rectangulaire implique que ce genre d'amincissement conduise à des calculs de métriques régulières. Autrement dit, l'application d'un algorithme de squelettisation de type amincissement topologique entraîne la perte de la métrique euclidienne. De plus, l'axe médian résultant n'est pas toujours d'épaisseur minimale, ce qui est contraire à sa définition mathématique.

[Fisher & al. 96] est une bonne référence pour comprendre comment fonctionne les opérateurs morphologiques et donc les mécanismes d'amincissement topologique. Le lecteur intéressé pourra s'en inspirer dans le but d'implémenter ces techniques pour traiter une image 2D.
[Xia 89] propose quant à lui une implémentation directe de l'analogie dans un espace discret. Cependant, du fait de ne pas utiliser une métrique euclidienne mais de favoriser plutôt une métrique régulière travaillant dans des espaces topologiques réguliers, cette implémentation est sujette aux inconvénients typiques, cités précédemment

2. Carte de distances
Les méthodes englobées dans cette classe cherchent à estimer le squelette d'un objet, par extraction des lignes de crêtes d'une carte de distances. Une carte de distances est une image, où chaque point de l'objet est associé à la valeur de la distance entre ce point et le bord de l'objet le plus proche. Les lignes de crêtes représentent les maxima locaux.
Les méthodes basées carte de distances sont divisés en deux étapes : la première étape consiste à calculer la carte de distances à partir d'une image binaire, au sein de laquelle l'objet

est représenté, la deuxième étape étant l'extraction du squelette à partir de la carte de distances précédemment calculée.

La première étape de ce type de méthode consiste à évaluer à partir d'une image de l'objet une carte des distances (*distance map*), ou encore une transformée en distance. Cette carte, une image en niveaux de gris, traduit la proximité des points de l'objet vis-à-vis des points appartenant à son contour. Autrement dit, la transformée en distance associe à chaque pixel de l'image la distance au point du contour de l'objet le plus proche. La carte de distances la plus répandue est basée sur la métrique euclidienne mais il existe évidemment d'autres types de cartes se basant sur des métriques différentes. La qualité de cette carte des distances s'avère décisive quant à celle du squelette extrait.
De même que pour l'amincissement topologique, [Fisher & al. 96] peut permettre de faire ses premiers pas à l'utilisateur intéressé, concernant l'implémentation d'algorithmes de calcul de cartes de distances à partir d'images binaires 2D, suivi de l'extraction des axes médians à partir de celles-ci.
Une fois la métrique choisie, le calcul de la transformée en distance se fait généralement suivant deux méthodes. La première consiste en une série d'érosions avec un élément structurant adéquat (généralement un carré 3x3 pour la métrique L∞), jusqu'à ce que tous les pixels du premier plan soient érodés. La deuxième, plus ingénieuse et surtout plus rapide (ne nécessitant que deux passes) est basée sur la notion de morphologie récursive (voir [Rosenfeld & Pfaltz 68] qui utilise la métrique euclidienne et essaye de l'approximer au mieux. Les objectifs principaux des algorithmes exposés étaient l'analyse de formes et de chemins).
Les algorithmes de génération de cartes de distances sont en général assez simples mais les squelettes finaux extraits ne sont généralement pas en adéquation avec l'analogie du feu de prairie proposée initialement par [Blum 67]. De plus, la transformée en distance est très sensible au bruit. En effet, une perturbation sur le contour entraîne immédiatement une modification de la carte des distances.

L'axe médian de l'objet est extrait à partir des cartes de distances au cours de la deuxième étape.
[Davies & Plummer 81] propose une approche prometteuse combinant un algorithme d'amincissement topologique avec une méthode d'extraction des points du squelette. Ainsi, le processus de *thinning* assure la bonne connexité des points de l'axe médian. Cependant, la méthode hérite de tous les problèmes connus, relatifs à l'exécution d'algorithmes d'amincissement topologique, influant ainsi grandement sur la qualité des résultats.
[Arcelli & Sanniti di Baja 85], entre autres, essaye de compléter le squelette extrait en exécutant des algorithmes de suivi sur les arêtes de la carte des distances. Cependant, même dans le cas de métriques régulières, ce type de méthode doit faire face à des situations particulières, telles que le traitement de points col (*saddle point* : point en lequel les dérivées s'annulent mais qui n'est pas extremum local).
L'approche présentée dans [Ivanov & al. 00] est uniquement basée sur la manipulation d'entiers, permettant ainsi de passer outre les problèmes relatifs à la troncation des valeurs flottantes dans un espace discret. Cette approche se base pour ce faire sur la définition d'un nouvel espace métrique. L'extraction de l'axe médian est suivie d'une étape de simplification du squelette obtenu et d'un filtrage supplémentaire pour supprimer les parties du squelette générées par du bruit. Cette méthode a été développée dans le but de réduire les vitesses de traitements, la complexification des calculs et les coûts en termes de mémoire et de traitements. Le squelette obtenu permet de reconstruire la forme initiale de l'objet étudié.
De nombreuses méthodes ont été élaborées dans le but de rendre le squelette plus stable et moins sensible aux petites variations qui peuvent avoir lieu sur le contour de l'objet.

[Montanari 69] propose une méthode d'ébarbulage basée sur l'analyse des rayons des disques inscrits le long des branches du squelette. Son approche est totalement analytique et reste dans le domaine continu (même si elle fut programmée). Elle est efficace mais ne permet pas de préserver la connexité du squelette. D'autres méthodes telles que [Dill & al. 87] ou encore [Pizer & al. 87] travaillent directement sur le contour de l'objet (respectivement en y appliquant un filtre passe bas ou en le floutant) pour que celui-ci ne génère pas de branches parasites. Ce type de méthode implique d'exécuter le processus de squelettisation à différents niveaux de résolutions. De plus, les correspondances entre les segments de l'axe médian sur différents niveaux de résolution doivent être faites en *post processing*. Une autre solution est d'attribuer à chaque point du squelette, associé à une partie du contour de l'objet, une mesure symbolisant sa stabilité. Ainsi, [Ho & Dyer 86] donne une mesure de proéminence qui permet, après filtrage, de supprimer les branches les moins importantes. Cependant, l'application directe de cette méthode donne des résultats peu intuitifs et trop ébarbulés dans le cas de formes complexes.
L'extraction de l'axe médian peut poser plusieurs difficultés. Tout d'abord, le squelette ne doit présenter aucun élément redondant. D'un autre côté, si les points du squelette sont considérés comme étant les centres des plus grands disques inscrits, ceux-ci sont alors peu nombreux et distribués, plus ou moins de manière dispersée. Par conséquent, d'autres méthodes appliquent des algorithmes de géométrie différentielle pour extraire le squelette, permettant l'usage de la métrique ou de la pseudo-métrique euclidienne. Cependant, les problèmes de connexité ne sont pas résolus. Dans tous les cas, en raison de la nature discrète d'une image 2D, le résultat final à savoir le squelette extrait reste une approximation grossière du concept de [Blum 67].

3. Calcul analytique
Ces méthodes cherchent à extraire l'axe médian par le biais de calculs analytiques. L'objet est divisé au cours de ces calculs en un ensemble de sous-objets, dont les squelettes sont connus. Le squelette global est alors l'union des sous-squelettes calculés.
La méthode la plus répandue, que nous allons présenter ici, consiste à estimer l'axe médian de l'objet via le diagramme de Voronoï calculé à partir du contour de ce dernier.

Etant donné un ensemble de primitives, appelées les sites de Voronoï, un diagramme de Voronoï partitionne l'espace continu en régions. Chaque région, liée à une primitive donnée, est définie par l'ensemble des points les plus proches de celle-ci.
Le diagramme de Voronoï peut être vu comme un graphe. Chaque sommet de ce graphe (en vert dans la Figure 4.48.), équidistant de trois sites (les sites sont en bleu foncé dans la Figure 4.48.), est le point d'intersection des côtés des différentes régions du diagramme générées par ces derniers. C'est pourquoi ce point d'intersection est le centre d'un cercle passant par ces trois sites. Chaque arête (en bleu clair dans la Figure 4.48.) de ce graphe est équidistante de deux sites et constitue un des côtés des régions générées par ces derniers. Chaque point d'une arête est le centre d'un cercle passant par ces deux sites.

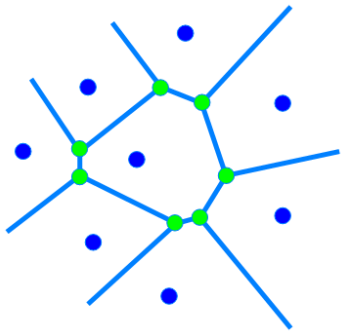


Figure 4.48 : Diagramme de Voronoï de points 2D

Un diagramme de Voronoï de n sites distincts comprend n cellules. Une cellule peut avoir au maximum n-1 frontières. Le nombre d'intersections entre les frontières ne dépassera pas 2n-5, et le nombre de frontières, 3n-6.
Les sites de Voronoï ne sont pas toujours des points, en 2D ou en 3D, et la métrique euclidienne n'est pas toujours adoptée. Des algorithmes théoriques et pratiques ont été élaborés dans le but de généraliser le calcul de diagrammes de Voronoï vis-à-vis du type de ses sites (primitives de type points, lignes, segments, courbes, etc.), de sa dimension (2D, 3D ou de dimension n) ou encore de la métrique utilisée. La complexité des algorithmes n'en est que plus grande puisque les frontières des régions du diagramme ne restent plus linéaires mais deviennent des courbes et surfaces algébriques (ou plus encore, relativement à la dimension de l'espace d'étude) dont les intersections doivent être calculées. Il est donc tout à fait normal que de nombreux auteurs aient proposé des algorithmes permettant le calcul d'une approximation des diagrammes de Voronoï généralisés. Dans notre contexte de thèse, nous n'irons pas aussi loin, les sites du diagramme étant les points 2D de la silhouette de l'utilisateur.
Les diagrammes de Voronoï sont exploités dans de nombreux domaines tels que la biologie, la chimie, en mécanique, en géographie, en mathématiques, dans le domaine médical, en synthèse d'images, en robotique et bien sûr en traitement d'images. Ils constituent un excellent point de départ pour la recherche de plus proches voisins, que ce soit au niveau des sites, au niveau d'objets divers compris dans l'espace d'étude, l'analyse et la reconnaissance de formes, la planification de chemins, la modélisation de structures, la segmentation d'images, la création d'effets visuels originaux, l'optimisation de processus, etc.

Le concept de diagrammes de Voronoï est connu depuis au moins 4 siècles. Descartes, dans « Le Monde de Mr Descartes, ou le Traité de la Lumière », publié en 1644, utilise des représentations similaires à celles des diagrammes de Voronoï, pour décrire la disposition de la matière dans le système solaire. [Dirichlet 50] et [Voronoï 08] sont les premiers à présenter véritablement ce concept. Depuis les années 70, des algorithmes pour calculer les diagrammes de Voronoï de différentes primitives géométriques ont été développés en géométrie algorithmique et autres domaines relatifs (voir les études de [Aurenhammer 91] pour un état de l'art concernant le calcul numérique et robuste de diagrammes de Voronoï de points 2D ainsi que [Okabe & al. 00]).

De nombreux algorithmes ont été proposés pour la construction de diagrammes de Voronoï de points 2D, 3D et de dimension supérieure. Une construction naïve d'un diagramme de Voronoï de points 2D est de construire les médiatrices des segments reliant un site à tous les

autres. Une cellule de Voronoï est alors l'intersection de tous les demi-plans définis par ces médiatrices. Chaque cellule de Voronoï est un polygone convexe qui peut ne pas être fermé. Lorsqu'il s'agit d'implémenter informatiquement cette technique, celle-ci étant alors incrémentale (ajout d'un site à la fois et construction progressive du diagramme), le concepteur doit faire face à un temps de calcul prohibitif, dont la complexité est $O(n^2)$. De plus, dans un espace discret, typiquement une image numérique, l'intersection entre 2 droites d'épaisseur 1 pixel n'est pas toujours assurée (voir Figure 4.47.), paradoxe bien connu en géométrie discrète. La discrétisation des médiatrices et la recherche d'intersections entre elles peuvent donc s'avérer problématiques.

Il existe cependant un certain nombre d'implémentations d'algorithmes de calcul de diagrammes continus (valeurs exactes ou de type flottantes) de points, en général 2D, utilisables dans le cadre d'un traitement d'images numériques. Les principaux problèmes rencontrés lors de l'élaboration de ces algorithmes sont les problèmes relatifs au passage continu-discret et les coûts en termes de mémoire et de traitement. Cependant, ces problèmes ont été traités avec succès dans le cadre de certains travaux. Les deux algorithmes les plus connus et les plus efficaces sont les algorithmes *divide-and-conquer* (diviser et conquérir) de [Shamos & Hoey 75] et *sweepline* (selon une ligne de balayage) de [Fortune 86].
Le premier, de complexité O(nlogn), sépare les sites en deux sous-ensembles de taille similaire. Les diagrammes de Voronoï de ces deux sous-ensembles sont par la suite calculés récursivement. La dernière étape, qui peut se révéler difficile, consiste alors à fusionner ces deux diagrammes pour donner le diagramme final. Cette technique reste difficile à programmer informatiquement.
La deuxième approche, de complexité maximale O(nlogn), consiste à balayer le plan par une ligne (la *sweep line*) soit verticalement, soit horizontalement. Au fur et à mesure que cette ligne évolue, le diagramme se construit. Chaque site, dépassé par la ligne de balayage, génère une *beach line* (appelée également *wavefront*, front d'onde) qui est la ligne d'équidistance entre le site et la ligne de balayage. Il s'agit donc d'une parabole. Chaque intersection des paraboles constitue un point des arêtes des régions de Voronoï. Cette dernière approche est plus facile à implémenter. Elle est robuste et stable numériquement.

De nombreuses implémentations sont accessibles via internet, en C, C++ ou en Java.
[O'Sullivan 03], dans le cadre d'un projet *open source* nommé ***MapViewer***, programmé en C++, propose un algorithme intéressant, basé sur celui de [Fortune 86]. En fonction du nombre de sites traités, l'approche peut permettre le calcul temps réel des diagrammes de Voronoï.
[Inagaki & al. 92] et [Sugihara & Iri 94] ont quant à eux proposé des algorithmes robustes pour construire des diagrammes de Voronoï cohérents topologiquement.
[Hoff III & al. 00] présente une méthode pour calculer des approximations discrètes de diagrammes de Voronoï de points 2D, entre autres (également de points 3D, segments 2D ou 3D, de polygones 2D ou 3D ou de courbes 2D ou 3D), en utilisant le GPU de cartes graphiques standards. La transformée en distance associée à un site est approximée par un *mesh* coloré. Grâce à l'utilisation du Z-buffer, les régions de Voronoï sont calculées et identifiées par le biais de leurs couleurs respectives. Une pondération de l'influence de chaque site est également possible. Enfin, des algorithmes simples, basés sur la couleur des régions, permettent de déterminer les frontières de ces dernières, ainsi que leurs régions adjacentes. Plusieurs applications, telles que la planification de mouvements rapides dans des environnements statiques et dynamiques, sont proposées par les auteurs. L'approche est facilement implémentable, rapide puisque accélérée matériellement (applications 2D temps réel), et efficace, chaque approximation et source d'erreurs étant spécifiquement identifiée et

énumérée (précision ajustable). Cependant, les temps de calcul deviennent vite conséquents au fur et à mesure que le nombre de points augmente (à partir d'une trentaine de points, l'algorithme montre ses limites).
Enfin, un diagramme de Voronoï est le dual d'une triangulation de Delaunay. Certaines approches utilisent cette propriété pour calculer précisément le diagramme de Voronoï.

Pour finir, sans rentrer dans les détails, des algorithmes ont été proposés pour calculer les diagrammes exacts de Voronoï de sites de plus haut niveau. Deux approches basées sur des techniques incrémentales et de type *divide-and-conquer* ont été résumées par [Okabe & al. 00]. Le lecteur intéressé pourra trouver plusieurs autres algorithmes, concernant des sites de Voronoï de type polygones, polyèdres, courbes, etc. Dans tous ces cas, le calcul du diagramme implique la représentation et la manipulation de courbes et de surfaces algébriques de dimensions importantes, ainsi que la recherche de leurs intersections. Il en résulte qu'il n'existe pas encore d'algorithmes efficaces et robustes pour construire une représentation continue et topologiquement cohérente de diagrammes de Voronoï généralisés.
Une approximation des diagrammes généralisés est possible, permettant ainsi d'éviter certains problèmes propres au calcul exact de ces derniers. Cependant, les algorithmes de calcul approximatif de diagrammes de Voronoï généralisés [Lavender & al. 92, Sheehy & al. 95, Teichmann & Teller 97, Hoff III & al. 00] restent limités et sont très coûteux en temps de calcul et en mémoire, suivant le nombre, la dimension et le type des sites de Voronoï.

A l'instar des cartes de distances précédemment présentées, le diagramme de Voronoï, calculé à partir des points 2D du contour d'un objet (les sites de Voronoï), permet l'extraction de l'axe médian de ce dernier (*Voronoï Medial Axis*). Ce squelette consiste en une succession de segments, chacun appartenant à une cellule de Voronoï. Chaque segment représente l'axe de symétrie local de deux sites de Voronoï et est en parfaite cohérence avec l'analogie du feu de prairie proposée par [Blum 67]. Le squelette est caractérisé par de vraies distances euclidiennes et une topologie correcte.
Le squelette de Voronoï doit être ébarbulé par la suite. En effet, le nombre de segments composant le squelette reste conséquent et la plus infime perturbation au niveau des sites de Voronoï entraine la création de nouvelles branches indésirables. Les parties du squelette qui se trouvent le plus en profondeur dans l'objet sont moins sensibles aux changements au niveau des frontières de ce dernier. Ces segments, qui décrivent les relations topologiques globales, doivent être préservés au cours du processus d'ébarbulage. Ils ont été créés au cours de la dernière phase de la propagation du feu de prairie, quand les fronts opposés se rapprochent.
[Ogniewicz & Ilg 92, Ogniewicz & Kübler 95] proposent un algorithme de squelettisation basé sur le calcul des diagrammes de Voronoï des points composant le contour d'un objet 2D. L'ébarbulage du squelette est exécuté en attribuant aux points du squelette une mesure de proéminence et de stabilité. Le constat est que plus le contour entre deux sites Voronoï est grand, plus le segment que ces derniers génèrent est stable. Plusieurs fonctions sont proposées, calculant le résidu entre la longueur réel du contour entre les deux sites et une estimation de celui-ci (par un arc de cercle, par plusieurs arcs de cercle, par un segment). Un seuillage sur la valeur du résidu permet de supprimer les artefacts. Par la suite, le squelette est de nouveau simplifié par une opération de regroupement hiérarchique de ses branches, aboutissant à une représentation multi-résolution de celui-ci. Cette opération est effectuée par un algorithme à deux passes dont le principe est de propager un label le long des branches. Plusieurs applications illustrent cette méthode, comme la reconnaissance d'objets ou encore l'interprétation de cartes routières. L'approche est robuste, les squelettes sont insensibles aux symétries et aux changements d'échelle, en plus de représenter efficacement la forme de

l'objet. La régularisation et la hiérarchisation permettent bien de préserver les segments stables et d'établir une représentation du squelette de la forme, à plusieurs niveaux de résolution. Les objets testés sont nombreux et variés et les exemples d'applications sont concrets et mettent même en jeu des notions de sémantique en ce qui concerne la reconnaissance d'objets. Cela dit, les algorithmes présentés ne sont pas temps réel, même si les performances dénotent des traitements rapides, et nécessitent plusieurs passes au niveau du squelette. Dans le cas d'objets complexes, la tâche se révèle être coûteuse en temps de calcul. Les algorithmes demandent la configuration de nombreux paramètres.
[Gold & al. 99] agissent directement sur le contour de l'objet, pour ébarbuler le squelette généré par le calcul du diagramme de Voronoï. Ceci permet d'éviter la plupart des artefacts. Le but est d'établir des cartes de réseaux routiers ou encore de rivières, dans les domaines de la géographie et de l'étude de cartes. L'approche est efficace et intelligente mais le principe de cette dernière (modification directe du contour de l'objet) n'est pas toujours applicable.

3.5.2.2. Positionnement

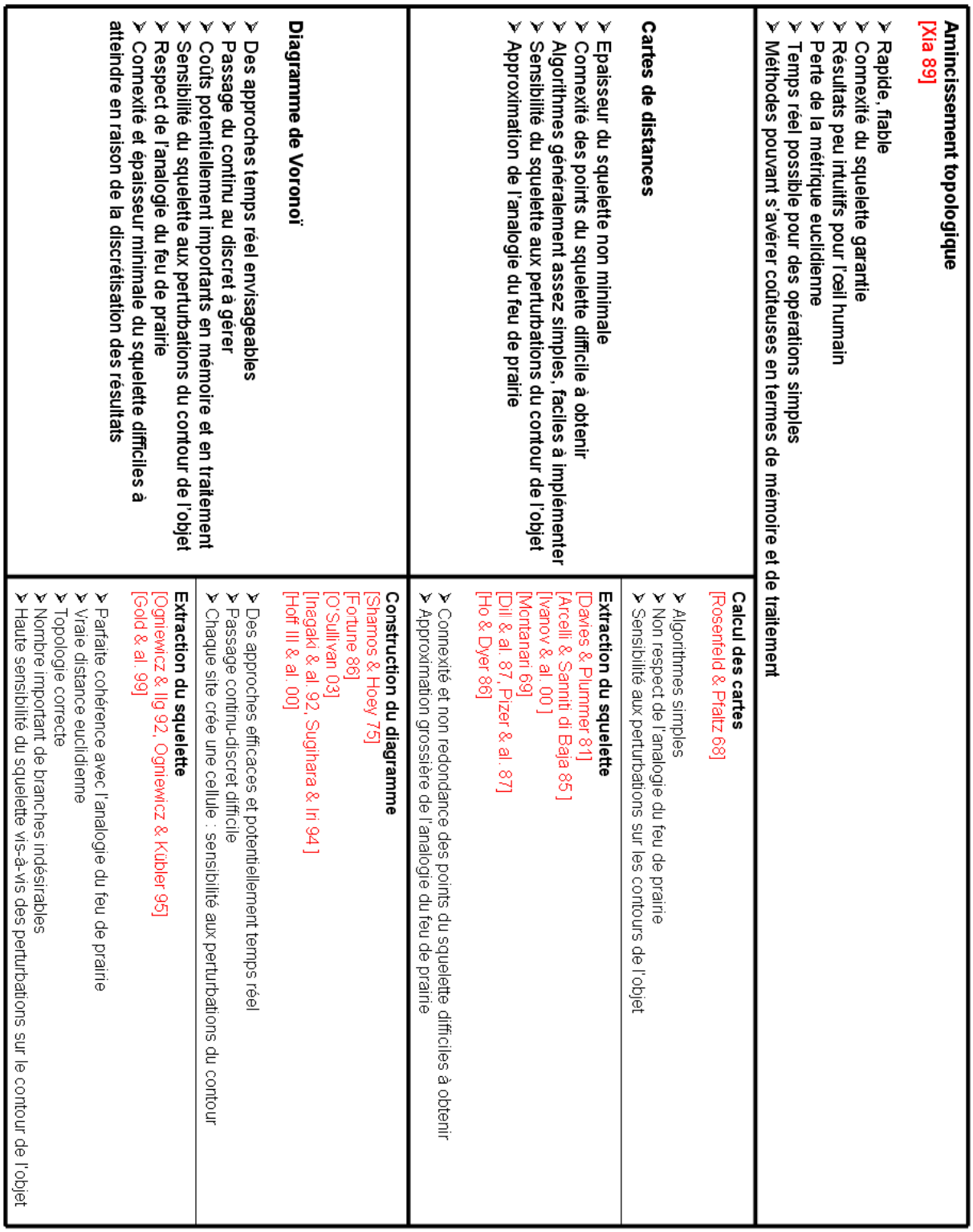
Nous allons discuter ici les différentes catégories de méthodes, de manière à choisir celle qui s'applique le mieux dans notre contexte de travail.

Les méthodes d'amincissement topologique sont rapides et fiables. Elles permettent de conserver la topologie et la connexité des points du squelette à coup sûr, ce qui n'est pas le cas pour les autres catégories de techniques. Le temps réel est tout à fait envisageable pour des opérations d'amincissement simples et peu nombreuses pour atteindre le résultat escompté.
En revanche, outre le fait que la métrique euclidienne n'est pas respectée, la combinaison de plusieurs opérateurs morphologiques s'avère vite coûteuses en mémoire et traitement. De plus, les vitesses d'exécution s'en trouvent être fortement augmentées, ne permettant alors pas de respecter la contrainte temps réel.

Les méthodes basées cartes de distances sont simples, faciles à comprendre et à implémenter. De nombreux travaux ont prouvé que ces méthodes sont efficaces et optimisables suivant différents aspects.
Ces techniques ne garantissent cependant pas la connexité des points du squelette. Les distances de chaque pixel au contour dépendent de la forme de l'objet. Un seuillage global pour extraire la crête, c'est-à-dire le squelette, ne peut pas être adapté à tous les reliefs de celle-ci. Une partie de la crête non sélectionnée entraîne une perte de la connexité des points du squelette. De manière similaire, les pixels du squelette ne sont pas uniques, la crête pouvant être un plateau de plusieurs pixels ayant la même distance au contour de l'objet. Le squelette extrait ne sera alors pas d'épaisseur minimale. Le squelette extrait est également sensible aux perturbations qui peuvent exister sur le contour de l'objet. Chaque pixel de la perturbation a également une influence immédiate sur le calcul des distances, influant ainsi sur la qualité du squelette extrait. Enfin, les méthodes regroupées dans cette catégorie ne permettent qu'une approximation, parfois grossière, de l'analogie du feu de prairie de [Blum 67].

Les méthodes analytiques, basées sur la construction du diagramme de Voronoï, contrairement aux méthodes précédentes, sont en accord parfait avec l'analogie du feu de prairie. Plusieurs algorithmes pour construire le diagramme sont parfaitement étudiés et

optimisés de nos jours. Une approche temps réel est tout à fait envisageable par le biais de ce type de méthodes.



Le premier désavantage que peuvent présenter ces techniques concerne la sensibilité du squelette aux perturbations du contour de l'objet. En effet, chaque pixel de la perturbation constitue un nouveau site de Voronoï, responsable de la création d'une branche du squelette supplémentaire. C'est pourquoi de nombreux auteurs ont cherché à supprimer ces branches

supplémentaires. De plus, l'efficacité de ces méthodes est très dépendante de leur capacité à gérer le passage du continu au discret. Une mauvaise gestion de ce passage entraîne vite en effet une perte de la connexité des points du squelette et la non-obtention d'un squelette d'épaisseur minimale. Enfin, d'un point de vue informatique, ces méthodes peuvent s'avérer coûteuses du point de vue mémoire.

Dans le cadre de ces travaux de thèse, nous choisissons d'utiliser une méthode basée sur la construction du diagramme de Voronoï. Ces méthodes, les plus utilisées de nos jours, sont rapides (temps réel), respectent l'analogie du feu de prairie et permettent l'extraction efficace d'un squelette respectant les propriétés géométriques et topologiques de l'objet. De plus, il est possible d'utiliser un certain nombre d'algorithmes sur ***Internet***, tels que celui décrit dans [O'Sullivan 03], se basant sur l'algorithme performant et optimisé de [Fortune 86]. C'est pourquoi cet algorithme constituera notre point de départ. Le fait que ce dernier est *open source* nous permettra de veiller aux performances du processus de squelettisation, que cela soit au niveau du passage continu-discret, qu'au niveau des coûts en termes de mémoire et de traitement.

Une remarque pour conclure et justifier notre approche. Nous cherchons, par le biais de la suppression des marqueurs colorés, à extraire des images de nouvelles informations nous permettant d'une part d'identifier les différentes parties (articulations, extrémités, etc.) du corps de l'utilisateur et, d'autre part, de déterminer un nouveau support pour capturer les mouvements de ce dernier.
Il existe bien évidemment d'autres méthodes que la squelettisation des silhouettes de l'utilisateur. Certaines méthodes cherchent à reconstruire un modèle 3D (ou directement son squelette 3D) de l'utilisateur par le biais des différentes observations 2D (silhouettes, textures, mouvements, etc.). La squelettisation de ce modèle pourrait permettre la comparaison du squelette extrait avec le squelette du modèle 3D utilisé au cours de la capture. Ces approches ne sont pour la plupart pas adaptées à des applications réelles. D'autres méthodes permettent d'identifier directement les différentes parties du corps de l'utilisateur. Ces méthodes requièrent un modèle d'apparence de l'utilisateur établi à partir des textures vidéo. Nous partons du principe que les silhouettes, uniquement, sont nos données principales, point de départ pour pallier les problèmes liés à la suppression des marqueurs. Cela dit, ces dernières méthodes sont envisageables et leur étude constitue une perspective à ce travail.

3.5.2.3. Contribution

Un algorithme a été développé, dont le point de départ sont les travaux de [O'Sullivan 03]. Au cours de cette section, nous présentons **les 4 étapes** qui composent cet algorithme, **les différentes problématiques** que nous avons rencontrées, ainsi que **différents résultats**. Ces étapes sont : **la génération du diagramme de Voronoï** ; **l'extraction et la complétion du squelette** ; **l'ébarbulage du squelette** ; **la simplification du squelette**.

1. Génération du diagramme de Voronoï
Notre contribution est moindre ici puisque nous avons utilisé l'algorithme fourni par [O'Sullivan 03]. Cet algorithme se base sur les travaux de [Fortune 86] et génère le diagramme par la méthode de la *sweepline*, que nous avons brièvement décrit précédemment. Le diagramme est calculé à partir d'une liste de points 2D, les sites de Voronoï, et est paramétré par la zone 2D englobant les sites. Dans notre cas, les points 2D sont les points appartenant à la silhouette de l'utilisateur et la zone est la zone SASM, englobant l'utilisateur,

zone que nous calculons lors de la mise à jour dynamique du fond (pour ne pas prendre en compte l'utilisateur dans cette mise à jour). La Figure 4.49. montre le diagramme de Voronoï calculé à partir d'une silhouette humaine, par le biais de l'algorithme de [O'Sullivan 03]. L'algorithme considère itérativement chaque paire de sites. 2 sites génèrent ainsi un segment faisant partie du côté commun des deux cellules de Voronoï leur correspondant. C'est ici que s'arrête l'algorithme de [O'Sullivan 03], que nous avons alors commencé de modifier.

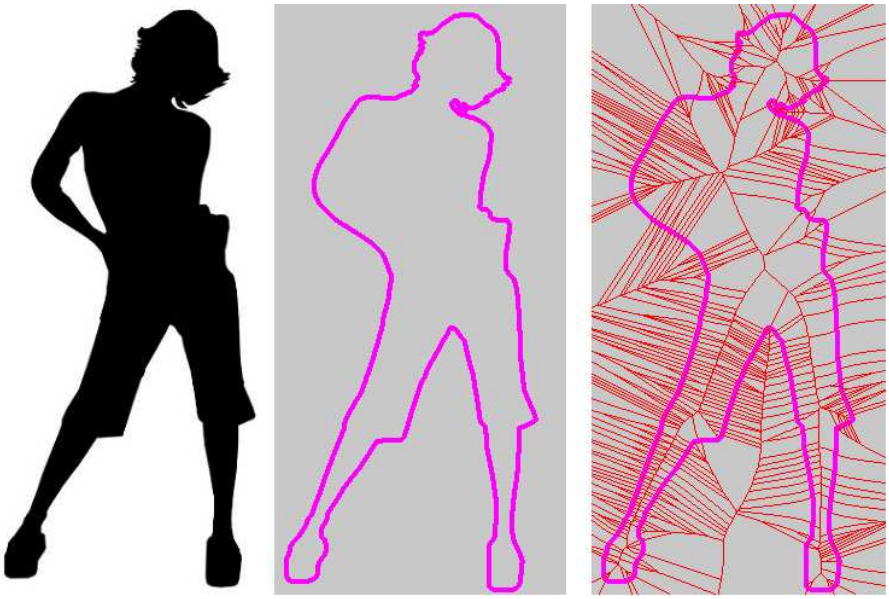


Figure 4.49. : De gauche à droite : image traitée ; silhouette extraite ; diagramme de Voronoï

Nous avons tout d'abord mis en place une première phase simple d'ébarbulage, consistant à ne considérer que les paires de sites qui sont distants d'un nombre paramétrable de pixels (vis-à-vis de la silhouette), pour générer le diagramme de Voronoï. Autrement dit, deux pixels, appartenant à la silhouette, séparés d'(au moins) un nombre paramétrable de pixels sur la silhouette, génèrent un segment. Dans notre exemple, en ne considérant que les paires de pixels séparés d'une cinquantaine de pixels sur la silhouette, nous obtenons un premier ébarbulage efficace (voir Fig. 4.50., les sites considérés pour la génération du diagramme sont en bleu clair). Dans notre exemple, nous obtenons alors 6 squelettes, un étant à l'intérieur de la silhouette (endosquelette), les autres à l'extérieur (exosquelettes).

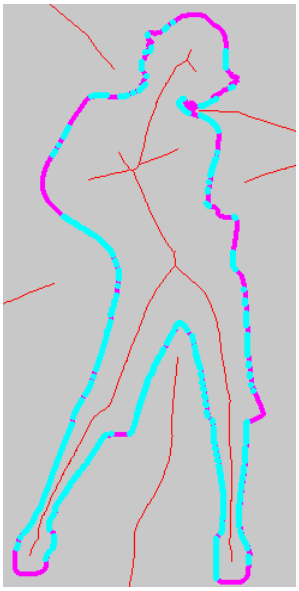
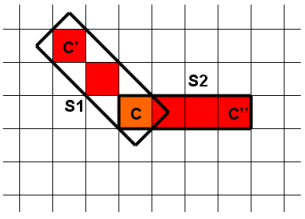


Figure 4.50. : Première phase d'ébarbulage

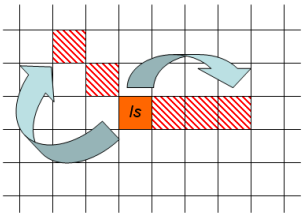
Nous avons alors mis en place un mécanisme de labellisation des segments au cours de leur génération. Ainsi, chaque squelette extrait comprend un label unique. Cette labellisation se fait par le biais d'une table de correspondance ***labelLUT***. Cette table nous indique, pour chaque position 2D d'un pixel appartenant à un squelette donné, la valeur du label du squelette et lie la position 2D du pixel considéré aux positions 2D des pixels suivants sur le squelette (voir Fig. 4.51.). Ainsi, pour chaque segment généré, la ***labelLUT*** est mise à jour. Cette table nous permet de savoir si un nouveau squelette doit être crée (le segment généré n'appartient encore à aucun squelette), si un squelette doit être étendu (une des extrémités du segment généré appartient déjà à un squelette) ou si deux squelettes existants doivent être fusionnés (le segment généré lie deux squelettes). A la fin de la génération du diagramme, chaque squelette est labellisé, la ***labelLUT*** nous permettant de connaître les positions 2D des points les composant.



Génération de deux segments **S1** et **S2**, dont **C** ***(x, y)*** est le point commun

S1 -> [**C**, **C'**] avec **C'** ***(x', y')***
S2 -> [**C**, **C''**] avec **C''** ***(x'', y'')***

Diagramme de Voronoï



labelLUT (**C** ***(x, y)***) = { ***ls*** ; ***(x', y')*** ; ***(x'', y'')*** }

Avec ***ls*** = label du squelette auquel **C** appartient

labelLUT

Figure 4.51. : Table de correspondance ***labelLUT***

Notre processus de labellisation nous permet également d'attribuer à chaque squelette un ensemble d'extrémités (les derniers points du squelette) et de nœuds (les carrefours, d'où partent plusieurs segments du squelette). Ces extrémités et nœuds sont calculés et mis à jour lorsqu'un segment est ajouté au squelette. Ce calcul et cette mise à jour se fait par l'intermédiaire de la ***labelLUT***. Les différents cas de figure sont illustrés par le biais de la Figure 4.52. La Figure 4.53. montre le résultat du processus de détermination des extrémités (en vert) et des nœuds (en bleu) du squelette intérieur à la silhouette, dans le cadre de notre exemple.

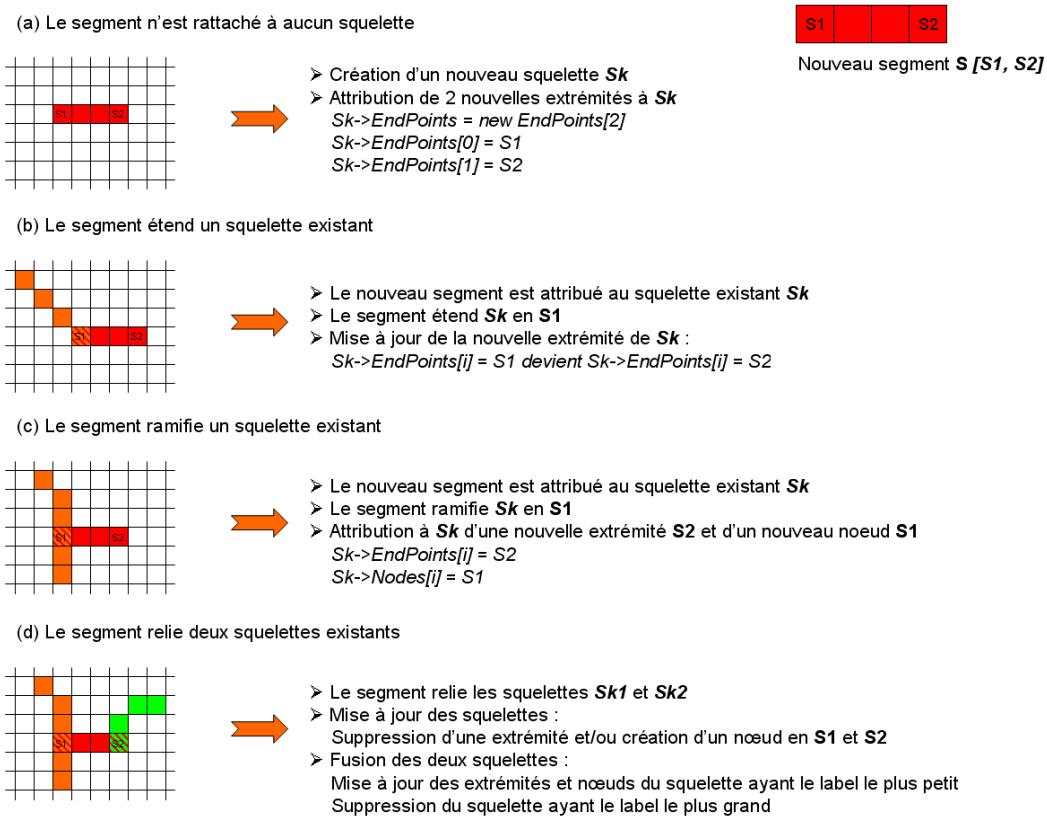


Figure 4.52. : Attribution d'extrémités et de nœuds aux squelettes

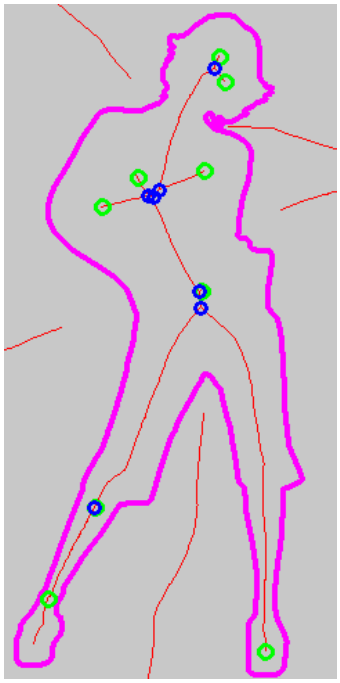


Figure 4.53. : Extrémités (en vert) et nœuds (en bleu) du squelette

Au cours du processus de génération du diagramme de Voronoï, nous cherchons également à savoir si le squelette est à l'intérieur ou à l'extérieur de la silhouette. Plusieurs solutions, plus ou moins complexes, sont possibles, comme, par exemple, un raisonnement à partir des normales à la silhouette vis-à-vis des différents points appartenant à un squelette donné. Nous avons cherché une alternative simple à ces raisonnements, pouvant s'avérer coûteux. Pour ce faire, nous sommes partis du principe que les squelettes se trouvant à l'intérieur de la silhouette ne touchaient pas les bords de la zone 2D SASM, englobant les sites qui paramètre l'algorithme de [O'Sullivan 03]. Autrement dit, un squelette sera considéré « à l'extérieur de la silhouette si un des segments qui le composent touche le bord de la zone 2D SASM, englobant les sites. Cette hypothèse peut s'avérer fausse si, pour un point de vue donné, la silhouette de l'utilisateur ne se trouve pas entièrement dans l'image. En ce cas, le ou les différents squelettes extraits du diagramme seront considérés comme extérieurs à la silhouette et ne seront alors pas exploités par la suite.

Pour finir, nous faisons face à la discrétisation des segments en arrondissant dès le départ (avant le processus de labellisation) les coordonnées de leurs 2 extrémités à l'entier le plus proche. Nous vérifions alors que nous ne traitons pas un point (arrondissement des coordonnées des extrémités d'un segment d'une longueur de moins de 1 pixel) ou un segment déjà labellisé, par le biais de la ***labelLUT***.

2. Extraction et complétion du squelette

Notre processus d'extraction du squelette se résume au fait que nous ne traitons que les squelettes se trouvant « à l'intérieur » de la silhouette. La sélection des squelettes se trouvant à l'intérieur de la silhouette est immédiate, le marquage d'un silhouette comme étant « à l'intérieur » ou « à l'extérieur » s'effectuant au cours de la génération des différents segments

du diagramme de Voronoï. Si, pour un point de vue donné, il n'existe aucun endosquelette, le processus de squelettisation de la silhouette se termine.

Théoriquement, il n'existe qu'un squelette à l'intérieur de la silhouette. Or, de par le processus de discrétisation exposé précédemment, à la fin de la génération du diagramme de Voronoï, plusieurs squelettes peuvent être marqués comme étant « à l'intérieur » de la silhouette (voir Fig. 4.54.). Les différents endosquelettes doivent être fusionnés au cours d'une phase de complétion. Nous pouvons constater ce phénomène sur la Figure 4.54. : le processus de détermination des extrémités et nœuds du squelette a extrait une extrémité alors que la branche du squelette n'est pas terminée (jambe se trouvant à notre gauche en regardant l'image). Ceci est du au fait qu'il existe deux endosquelettes qui théoriquement devraient être joints.

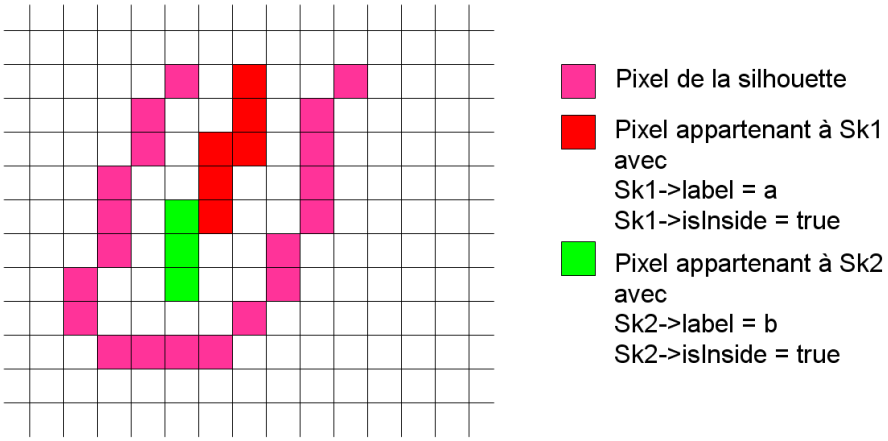


Figure 4.54. : 2 endosquelettes à fusionner

La fusion des squelettes se trouvant à l'intérieur de la silhouette se déroule en deux étapes.
Tout d'abord, les distances en pixel (distance euclidienne) entre les extrémités des endosquelettes sont calculées. A partir du moment où une distance calculée est en deçà d'un certain seuil (5 pixels dans notre cas), les deux endosquelettes sont fusionnés aux deux extrémités considérées (avec mise à jour de la ***labelLUT***).
Deuxièmement, il peut tout à fait arriver que deux endosquelettes doivent être fusionnés en des points autres que leurs extrémités. Le plus sûr moyen d'obtenir une fusion correcte de l'ensemble des endosquelettes serait alors de calculer toutes les distances entre tous les points appartenant à tous les endosquelettes. Les points les plus proches seraient alors fusionnés. Cette procédure n'est pas acceptable dans la mesure où elle engendrerait des temps de calcul prohibitifs. De manière à résoudre ce problème, nous décidons qu'une fusion ne s'exécute qu'à partir d'une extrémité d'un endosquelette donné. Par le biais de la ***labelLUT***, nous considérons le voisinage de cette extrémité (4-connexité) sur une distance égale au seuil de fusion donné précédemment. Si un point appartenant à un endosquelette autre que celui considéré est rencontré, les deux squelettes sont fusionnés.
La Figure 4.55. illustre notre algorithme par le biais de notre exemple. L'extrémité de la jambe a, cette fois-ci, correctement été extraite par le biais de la fusion des endosquelettes.

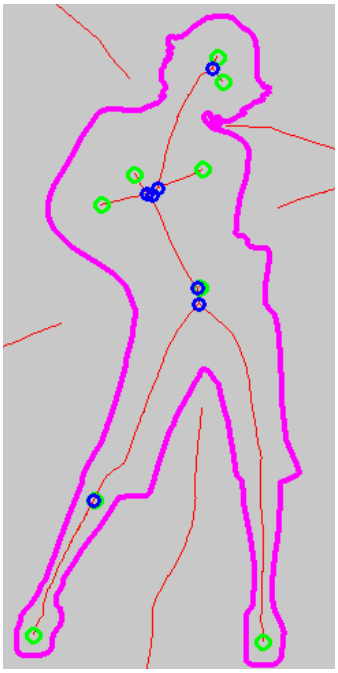


Figure 4.55. : Complétion du squelette

3. Ebarbulage du squelette

La phase d'ébarbulage à proprement parler comprend deux étapes : la fusion des nœuds proches et la suppression des branches trop petites du squelette. Cette phase est en fait une deuxième phase d'ébarbulage, la première s'effectuant au cours de la génération du diagramme de Voronoï.

Au cours de la première étape, le nombre de segments séparant deux nœuds est déterminé. Si ce nombre est inférieur ou égale à un seuil donné, alors les deux nœuds sont fusionnés. Notre algorithme est récursif et plusieurs nœuds peuvent être fusionnés au cours de la même étape. La Figure 4.56. illustre la fusion des nœuds proches dans notre exemple.

La 2ème étape consiste à supprimer les branches du squelette trop petites. Cette notion de « trop petit » est défini d'une part par un nombre de segments et d'autre part par la longueur cumulée des segments de la branche considérée. Le seuillage par rapport au nombre de segments de la branche considérée doit se faire en accord avec le seuillage utilisé au cours de la fusion des nœuds proches. Autrement dit, ce seuil doit être au moins égal à celui utilisé précédemment pour la fusion des nœuds. Le seuillage sur la longueur de la branche est nécessaire car certaines branches du squelette sont grandes en longueur mais ne sont composées que de quelques segments, alors qu'au contraire, certaines branches parasites comprennent un nombre important de très petits segments. La combinaison de ces deux tests nous permet d'obtenir un ébarbulage efficace, comme le montre la Figure 4.56.

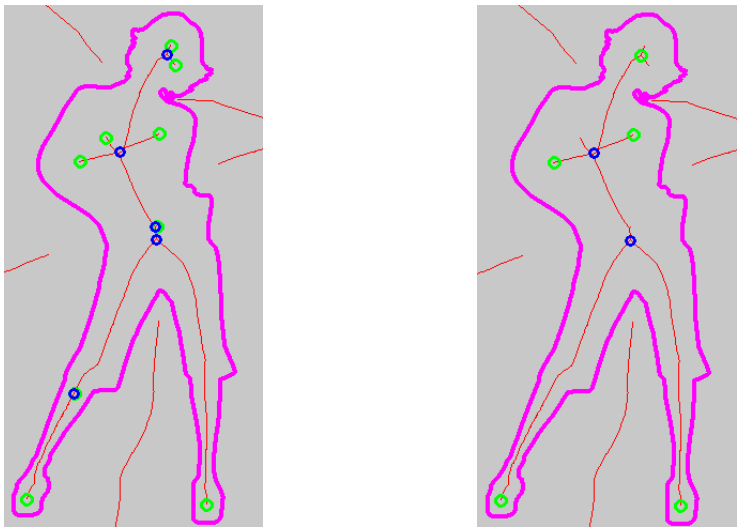
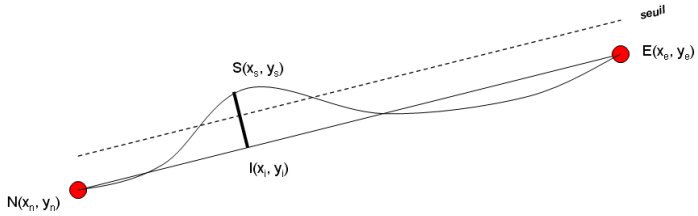


Figure 4.56. : Fusion des nœuds proches (à gauche) et suppression des petites branches (à droite)

4. Simplification du squelette

Cette étape de simplification du squelette consiste à modéliser ce dernier sous la forme de droites simples plutôt que par un ensemble de petits segments. Cette étape nous permet également de calculer les différentes articulations se trouvant entre un nœud et une extrémité. Ainsi, le squelette devient un ensemble de droites allant d'un nœud à une articulation ; d'un nœud à une extrémité ; d'une articulation à une articulation ; d'une articulation à une extrémité.

Une branche du squelette est considérée comme allant d'un nœud à une extrémité. Chaque branche du squelette est considérée une à une et ne comprend au départ aucune articulation. L'algorithme calcule tout d'abord l'équation de la droite passant par le nœud et l'extrémité de la branche. Puis, il parcourt la branche et calcule la distance euclidienne entre chaque point de la branche et la droite calculée précédemment. Le point le plus éloigné de la droite, et dont la distance à celle-ci est supérieure à un seuil donné, devient une articulation. L'algorithme s'arrête alors et recommence son traitement sur la branche étudiée. La différence réside dans le fait que, cette fois-ci, les droites allant du nœud à l'articulation et de l'articulation à l'extrémité sont considérées. Le processus se termine une fois que la branche ait été parcourue et qu'aucune articulation n'ait été extraite. La Figure 4.57. expose les différentes étapes du premier parcours de la branche.



(1) Equation de la droite ***(N, E)***
(N, E) : $y = mx + d$
avec $m = (y_e - y_n) / (x_e - x_n)$
et $d = (y_n x_e - y_e x_n) / (x_e - x_n)$

(2) Droite perpendiculaire à ***(N, E)*** et passant par ***S***
$y = m'x + d'$
avec $m' = -1 / m$
et $d' = y_s - m'x_s$

(3) Coordonnées du point ***I***
$x_i = (m(d' - d)) / (m^2 + 1)$
$y_i = mx_i + d$

(4) Longueur **|| *SI* ||** et seuillage
$|| SI || = \sqrt{(xs - xi)^2 + (ys - yi)^2}$
if || SI || >= seuil then Sk->articulations[i] = new articulation

Figure 4.57. : Calcul des articulations des branches du squelette

Une fois l'ensemble des articulations d'une branche calculée, le squelette se voit attribuer un objet 'branche', composé d'une ou plusieurs équations de droites et une liste d'articulations. La Figure 4.58. illustre notre algorithme dans le cas de notre exemple. Les différentes branches sont en jaune et les articulations en rouge.

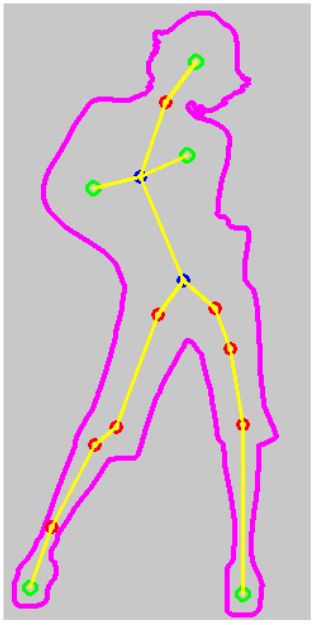


Figure 4.58. : Simplification du squelette

4. Conclusion

Nos objectifs de thèse mettent en avant le besoin d'un système dédié à la capture, non invasive et dans un environnement non contrôlé, de l'activité prenant place au sein de la scène. Cette activité, une fois caractérisée et interprétée, permet au système de répondre à celle-ci, à la fois de manières interactive et adaptative.

Le cadre industriel de cette thèse a impliqué l'adoption, comme point de départ autour duquel s'articulent ces travaux de thèse, d'un système commercial de capture des mouvements d'un unique utilisateur, le ***Cyberdôme***. Outre le fait qu'il met en œuvre un processus de capture ne pouvant pas capturer l'activité de manière générale, ce système initial présente deux contraintes matérielles majeures. Le processus de capture implique, d'une part, le total contrôle de l'environnement de capture et, d'autre part, contraint l'utilisateur à porter des marqueurs colorés aux bras et aux jambes.

La première partie de ce chapitre a proposé plusieurs définitions concernant les notions de 'capture', 'mouvement' et 'corps. Ces définitions nous ont apparu nécessaires, dans la mesure où elles nous ont permis de bien comprendre et expliciter le processus de capture que nous voulions implémenter au cours de ces travaux de thèse. Les définitions s'articulant autour de la notion de 'mouvements' sont sujettes à discussion, et ont été établies en accord avec notre sujet de thèse, qui ne considère que la modalité 'gestuelle' pour interagir avec le système. Si d'autres modalités devaient être considérées, au cours de futurs travaux, ces définitions devraient être précisées.
Puis, plusieurs systèmes de capture ont été présentés, dans le but de mettre en relief les différents aspects d'un processus de capture, ainsi que les problématiques rencontrées suivant les approches envisagées. Nous nous sommes particulièrement intéressés aux processus de capture non invasifs, qui permettent une interaction plus libre et plus naturelle, mais qui impliquent des problématiques importantes et non triviales.

Nous avons présenté, en seconde partie de ce chapitre, nos contributions, visant à étendre et à améliorer le processus de capture initial, mis en œuvre par le ***Cyberdôme***, en accord avec nos objectifs de thèse. Ces contributions visent à soulager le système initial de ses deux contraintes matérielles principales : le total contrôle de l'environnement de capture & la nécessité pour l'utilisateur de porter des marqueurs de couleur. L'extension du processus de capture de mouvements à celui d'une capture plus globale de l'activité est exposée au cours du chapitre suivant, car impliquant la gestion d'informations contextuelles dans un cadre applicatif bien particulier. Nous proposons ici une évaluation de nos contributions.

1ère contribution : Modélisation dynamique du fond
Pour permettre au système de capturer les gestuelles utilisateur, nous avons introduit au sein du processus **un algorithme de modélisation dynamique du fond de la scène**. Cet algorithme permet la modélisation d'un fond de bonne qualité, suffisante en tout cas vis-à-vis des gestuelles que nous avons étudiées dans ces travaux.
En paramétrant le processus comme nous l'avons décrit précédemment, les mouvements au sein de la scène sont immédiatement pris en compte lors de la modélisation du fond.
De plus, les problèmes dus aux effets fantômes, typiques dans ce type de processus, sont rapidement résolus. Au pire, ces effets fantômes sont observés pendant le nombre maximal de *frames*, qui paramètre l'attribution de valeurs aux pixels de la matrice dynamique. Un seuillage ajusté permet également de supprimer les artefacts, générés par ces effets fantômes,

au cours du processus d'extraction du *foreground*, au bout de quelques *frames* (3 à 4 *frames* dans notre cas).
Le temps réel est largement respecté, les processus de modélisation et d'extraction du *foreground* étant très rapide. De plus, ces derniers peuvent facilement être exécutés sur le GPU des cartes graphiques des différents PCs.

L'algorithme que nous avons implémenté, avec le support de la zone SASM, a montré son efficacité et sa robustesse **face à de nombreuses situations problématiques**. La Figure 4.59. illustre la robustesse de notre algorithme au cours d'une de ses situations. Le sujet d'intérêt est le mannequin. Au cours du processus de modélisation, une personne devient mobile dans le fond de la scène et passe derrière le mannequin. La zone SASM reste centrée sur le mannequin tandis que la personne mobile est progressivement mise à jour dans le fond. Le processus de capture n'est donc pas perturbé par la présence d'une personne mobile dans le fond de la scène et continue de se focaliser sur le traitement du sujet d'intérêt.

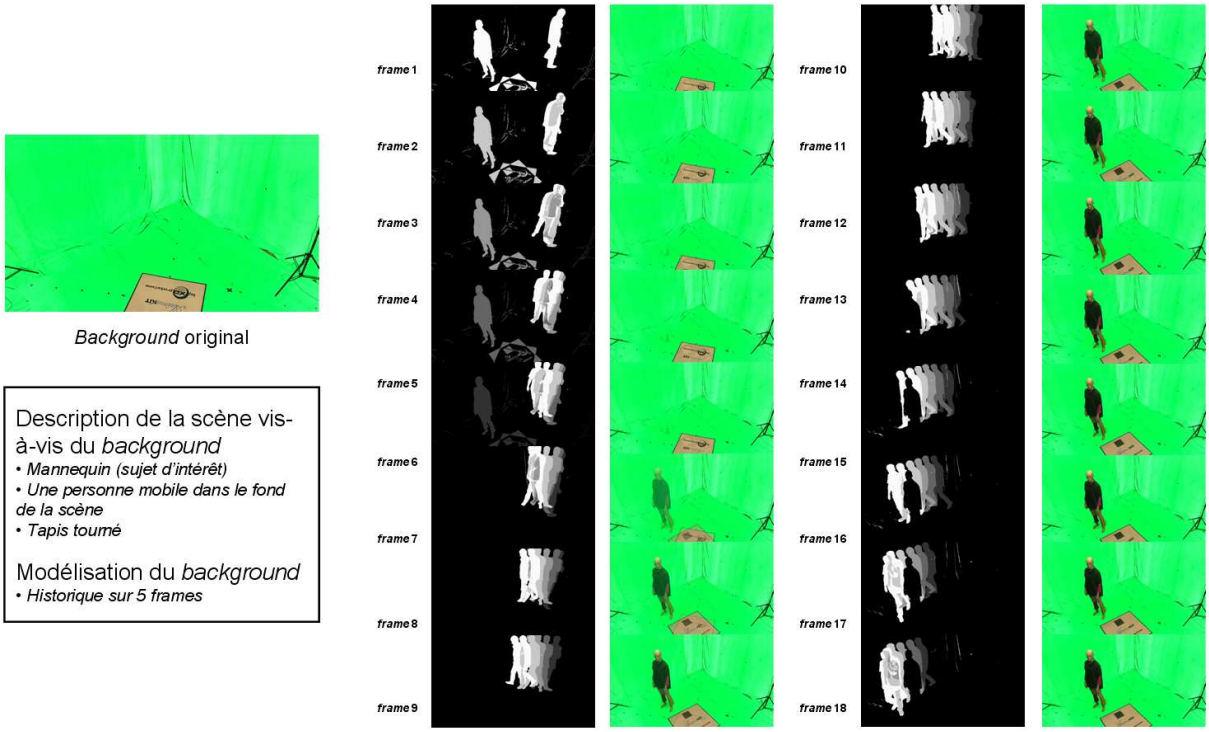


Figure 4.59. : Robustesse de notre processus face à une scène dynamique

L'algorithme présenté est une amélioration importante au sein de notre système, mais souffre toutefois **de plusieurs inconvénients**.
Tout d'abord, le nombre de paramètres est finalement assez important et les calibrage et mise au point du processus restent difficiles et longs à effectuer. Elles nécessitent un nombre important d'expérimentations, qui peuvent vite s'avérer fastidieuses. Ces étapes sont, pour le moment inévitables, et pourraient être allégées par le calcul automatique de certains paramètres, ou par la mise en place de profils, un profil correspondant à un type de scène particulier, par exemple. Dans notre cas, le fait que notre contexte d'observation soit unique et ne change que très rarement est un avantage important.

Deuxièmement, notre algorithme nous permet de modéliser dynamiquement un fond de bonne qualité, mais nous n'avons pas travaillé extensivement sur le processus d'extraction en lui-même. Cette étape, la différenciation simple de l'image courante avec le fond, suivi d'un OU logique, est basique et s'avère parfois insuffisante pour extraire correctement les silhouettes de l'utilisateur.
Enfin, nous n'avons pas proposé de solutions vis-à-vis des ombres, qui restent problématiques. Toute solution est cependant envisageable.

2ème contribution : Squelettisation des silhouettes de l'utilisateur
Nous avons présenté au cours de cette section **notre algorithme de squelettisation des silhouettes de l'utilisateur**. Cette squelettisation se fait par le biais d'**un diagramme de Voronoï** généré à partir des points 2D constituant la silhouette. Vis-à-vis de la définition mathématique du squelette d'un objet, la métrique utilisée est **euclidienne** et les points du squelette correspondent à **une transformée en distance maximale** relativement à la frontière de l'objet (la silhouette de l'utilisateur).

Nous avons testé notre algorithme tout d'abord sur quelques images choisies aléatoirement sur ***Internet*** (voir Fig. 4.60.), puis sur des séquences d'images, enregistrées au cours d'une séance de capture au sein du ***Cyberdôme***, et enfin, en temps réel (voir Fig. 4.61.).

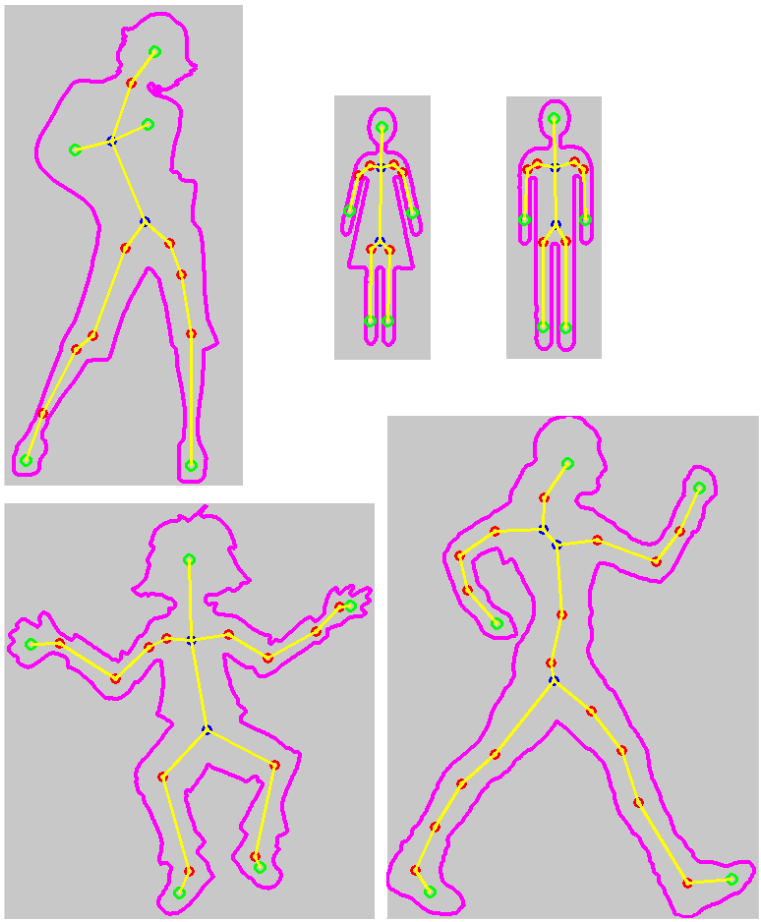


Figure 4.60. : Exemples de squelettisation (images ***Internet***)

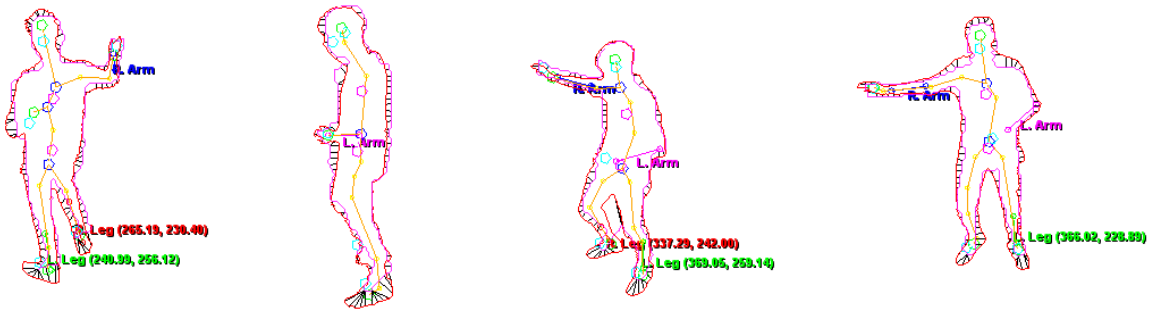


Figure 4.61. : Exemples de squelettisation (***Cyberdôme***)

Il est tout à fait possible d'évaluer visuellement le squelette que nous extrayons des différentes silhouettes. Le squelette extrait nous semble tout à fait cohérent, géométriquement et topologiquement, avec une silhouette donnée. Le processus de squelettisation nous permet de déterminer les différentes extrémités, nœuds et articulations que nous pouvons supposer intuitivement, en observant la silhouette étudiée. Ce squelette est la structure la plus stable (vis-à-vis des perturbations du contour) que nous pouvons extraire de la silhouette de l'utilisateur. Sans aucune connaissance supplémentaire que celles apportées par la silhouette, **c'est le meilleur résultat que nous pouvions escompter**.

Le processus de squelettisation est **temps réel**. A noter que l'algorithme tiré de [O'Sullivan 03] comportait plusieurs fuites mémoire qui s'avéraient problématiques lors du traitement continu d'une large séquence d'images vidéo. Ces problèmes ont été résolus au cours de nos travaux.

Le processus de squelettisation **requiert peu de paramètres**. Ces paramètres sont déterminés au terme d'une courte phase d'expérimentations et sont généralement adaptés sur une large période de temps. Au cours d'une séance de capture, ils ne nécessitent pas l'intervention d'un superviseur.

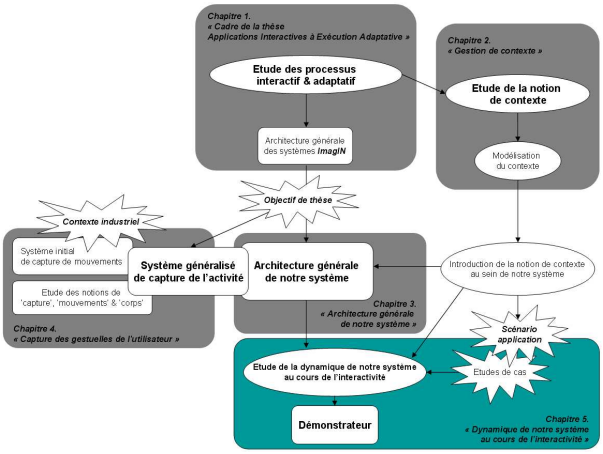
Enfin, **la gestion du passage continu-discret** est évidemment inévitable de par la nature même de notre approche (les diagrammes de Voronoï sont calculés dans un espace continu). En revanche, nous y faisons face très tôt dans le processus et de manière efficace. Jusqu'à présent aucun problème, tel qu'un squelette incomplet ou disjoint, n'a été observé.

Chapitre 5. Dynamique de notre système au cours de l'interactivité

Luck favors the prepared mind
- Pasteur -

On pourra alors se focaliser sur des choses plus utiles. De toute façon, ça sera pas compliqué de faire plus utile.
- RR -

Résumé



Le Chapitre 5. présente **la dynamique du processus interactif**, après la retranscription de l'activité au sein de la scène virtuelle.

Les différentes connaissances permettant la caractérisation du contexte d'interaction au sein duquel l'activité prend place sont tout d'abord identifiées. Ces différentes connaissances sont **extraites de la scène virtuelle** par le biais **de mécanismes d'observation** particuliers.
Il est alors possible **de caractériser ce contexte d'interaction** et les sous contextes qu'il regroupe. Nous présentons, dans le cadre d'un contexte applicatif bien spécifique, notre modélisation.
Un module dédié se charge par la suite **d'interpréter ce contexte d'interaction**, en évaluant **la distance** pouvant exister entre **le contexte d'interaction observé** et celui **attendu** par le scénario.
Cette distance permet l'évolution du scénario et donc **la mise en place des réponses interactives et adaptatives du système**. Nous nous arrêtons particulièrement sur l'étude du processus adaptatif au sein de notre système.

Pour illustrer nos travaux de thèse, nous avons développé une application, immergeant l'utilisateur au sein d'**un entrainement de tennis**. Nous illustrons nos travaux par le biais de 3 études de cas bien particulières.

Plan du chapitre

Nous nous intéressons, dans ce chapitre, à **la dynamique du processus interactif**, mis en œuvre par notre système, et se déroulant avec l'utilisateur. **Plusieurs exemples** nous permettront d'exposer et d'illustrer les différents aspects de cette dynamique.
A partir de **la scène virtuelle**, composée des modélisations de **la scène réelle capturée** et de **la scène 3D immersive**, le système, par le biais d'un gestionnaire dédié, extrait différentes connaissances hétérogènes sur **l'activité** observée. Ces connaissances lui permettent alors de caractériser **le contexte d'interaction** englobant l'activité observée, relative particulièrement **aux gestuelles utilisateur**. Suit alors **le processus d'interprétation de l'activité**, supporté par **le scénario de l'application**, qui décide de la validité **du contexte d'interaction observé**, vis-à-vis **du contexte d'interaction attendu** par le scénario de l'application. Enfin, en plus de sa réponse **interactive**, modifiant la scène virtuelle, le système met en place sa réponse **adaptative**, adaptant ainsi son fonctionnement global, à tous les niveaux de son architecture. Ces réponses interactives et adaptatives sont également déterminées à partir du scénario de l'application et sont mises en place en fonction de l'interprétation de l'activité observée.

Nous organisons nos propos, de l'extraction des connaissances à partir de la scène virtuelle, à la mise en place des mécanismes adaptatifs, en accord avec l'architecture de notre système (voir Figure 5.1.). Le lecteur pourra se référer au chapitre relatif à l'architecture de notre système pour plus de détails sur cette dernière.

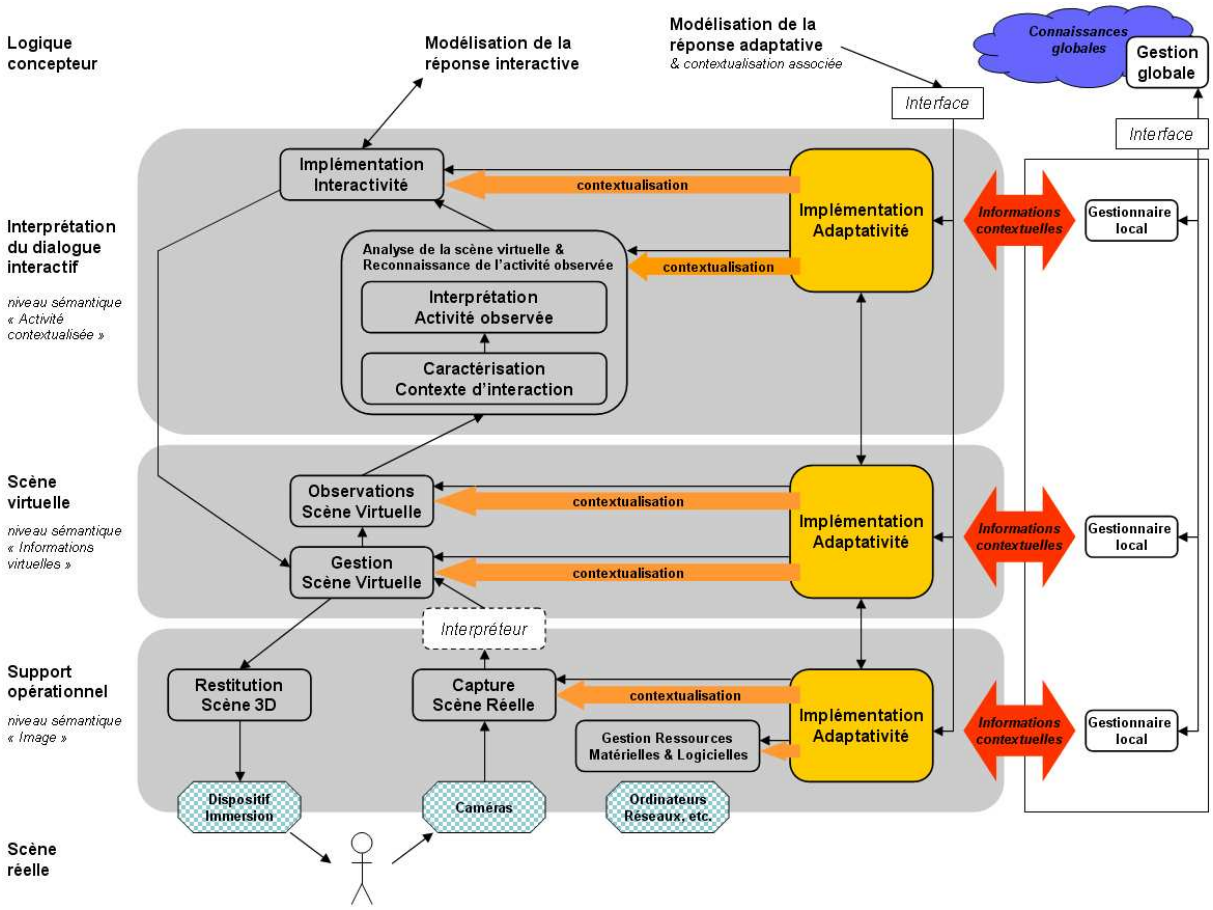


Figure 5.1. : Architecture du système conçu dans le cadre de ces travaux de thèse

Nous nous plaçons ici en permanence dans le contexte de notre étude, à **savoir l'immersion de l'utilisateur au sein d'un entrainement virtuel de tennis**. Nous décrivons au cours de la première section de ce chapitre, **le scénario de notre application**. Ce scénario nous permettra d'illustrer certains de nos propos, par quelques exemples de situations et de contextes d'interaction.

Le processus interactif débute par **la modélisation de la scène virtuelle**, à partir de laquelle différentes connaissances sont **extraites** et **construites**, au cours de l'interactivité avec l'utilisateur. Ces connaissances s'ajoutent à celles déjà **introduites *a priori*** au sein du système, lors des phases de conception, de calibrages et d'expérimentations. Elles sont caractérisées sur plusieurs niveaux sémantiques, les connaissances de plus haut niveau étant construites à partir des connaissances de plus bas niveau. La Section 2. identifie l'ensemble de ces connaissances.

A partir des connaissances extraites de la scène virtuelle, il est possible **de caractériser les contextes d'interaction** au sein desquels l'utilisateur évolue. Ces contextes d'interaction traduisent **l'activité** observée au sein de la scène virtuelle, cette activité regroupant **les gestuelles utilisateur** et **d'autres événements réels**, dont l'utilisateur n'est pas responsable (des mouvements de spectateurs par exemple). La Section 3. présente cette caractérisation.
Le contexte d'interaction que nous modélisons a été présenté au cours de la Section 6.4. du Chapitre 2. Il comprend les contextes **« Utilisateur »**, **« Système/Application »**, **« Scène réelle »**, **« Interface »**, **« Observation »** et **« Temps »**. Ces contextes sont caractérisés à partir des connaissances, introduites au sein du système *a priori* ou extraites à partir des observations de la scène virtuelle. Par le biais de ces contextes, l'objectif de notre caractérisation est de principalement traduire l'activité, observée ou attendue, au sein de la scène virtuelle, qui sera alors soumise, par la suite, à l'interprétation du système. De plus, les informations contextuelles, fournies par un contexte d'interaction donné, servent à paramétrer les mécanismes adaptatifs, mis en place par le système après l'interprétation de la scène. Comprise et décrite principalement au sein du contexte « Utilisateur », nous décrivons principalement **notre caractérisation de la gestuelle utilisateur**. Nous proposons une caractérisation basée sur :

- l'ensemble des **règles qualitatives** et **quantitatives** suivies par la représentation virtuelle (ou certains éléments de celle-ci) de l'utilisateur ;
- **les relations spatio-temporelles** existant entre la représentation de l'utilisateur (ou certains éléments de celle-ci) et la scène virtuelle (ou certains éléments de celle-ci) ;
- **les événements interactifs** générés par les éléments actifs composant la scène virtuelle.

La Section 4. traite la problématique de **l'interprétation de l'activité au sein de la scène virtuelle**, à partir de la modélisation du contexte d'interaction au sein duquel l'utilisateur évolue.
Le processus dédié du système interprète l'activité **en comparant** les informations contextuelles construites à partir de cette dernière avec celles extraites du scénario de l'application, traduisant l'activité attendue par le système.
D'une part, à partir des connaissances extraites de la scène virtuelle au cours de l'interactivité, **le contexte d'interaction observé** est modélisé par le système. Ce contexte comprend, entre autres, la gestuelle utilisateur caractérisée. D'autre part, pour chaque nouvelle étape du scénario, **le contexte d'interaction attendu**, que le système est sensé observer, est extrait du scénario. Ce contexte d'interaction, défini en accord avec les objectifs de l'utilisateur,

comprend la gestuelle utilisateur et les autres événements particuliers que le système est en droit attendre, relativement à l'étape scénarisée face à laquelle l'utilisateur se trouve.
Du processus d'interprétation, il est alors possible de définir **une distance**, quantifiant la différence pouvant exister entre le contexte d'interaction observé et celui attendu par le scénario. Cette distance permet alors de déterminer la véritable signification de l'activité au sein de la scène virtuelle, signification à partir de laquelle **les différents mécanismes adaptatifs** seront par la suite mis en place par le système. En effet, le processus d'interprétation est l'étape nécessaire pilotant l'adaptativité du système. Avant de pouvoir s'y adapter, le système doit tout d'abord comprendre et reconnaître l'activité au sein de la scène observée.

Après avoir interprété la scène observée, le système met en œuvre **sa réponse**. Cette réponse a une double nature. D'une part, elle est **interactive**, définissant alors la réponse visuelle du système, restituée à l'utilisateur et fonction de l'activité observée au sein de la scène virtuelle. D'autre part, elle est **adaptative**, adaptant ainsi le fonctionnement interne et global du système vis-à-vis de l'interprétation de cette activité. C'est cette réponse, qui traduit **l'adaptativité** de notre système, que nous étudions au cours de la Section 5.
La réponse adaptative du système se concrétise par la mise en place de plusieurs **mécanismes adaptatifs**. Ces mécanismes sont, d'une part, **pilotés par l'évolution du scénario** de l'application et, d'autre part, **paramétrés par les différentes informations contextuelles** présentes au sein du système. Ces mécanismes sont exécutés à tous les niveaux de l'architecture du système et sont dédiés aux différents processus mis en œuvre au cours de l'interactivité : **capture de la scène réelle**, **observation de la scène virtuelle**, **caractérisation & interprétation de la scène virtuelle**, **mise en place de la réponse visuelle restituée à l'utilisateur**, et **gestion des ressources matérielles et logicielles**.
Au cours de l'interactivité, par l'accumulation des effets des différents mécanismes adaptatifs, un ensemble de boucles logicielles, appelées **boucles vertueuses**, se met en place au sein du système. L'objectif de ces boucles vertueuses est d'améliorer constamment l'interactivité avec l'utilisateur.

Nous présentons les Sections 2., 3., 4. et 5., par le biais de la Figure 5.2., qui situe ces dernières au sein de notre architecture et au cours du déroulement de l'interaction.

La dernière section de ce chapitre présente **3 études de cas**, dans le cadre de notre scénario, illustrant ainsi le fonctionnement de notre système, en fonction de l'activité observée au sein de la scène.

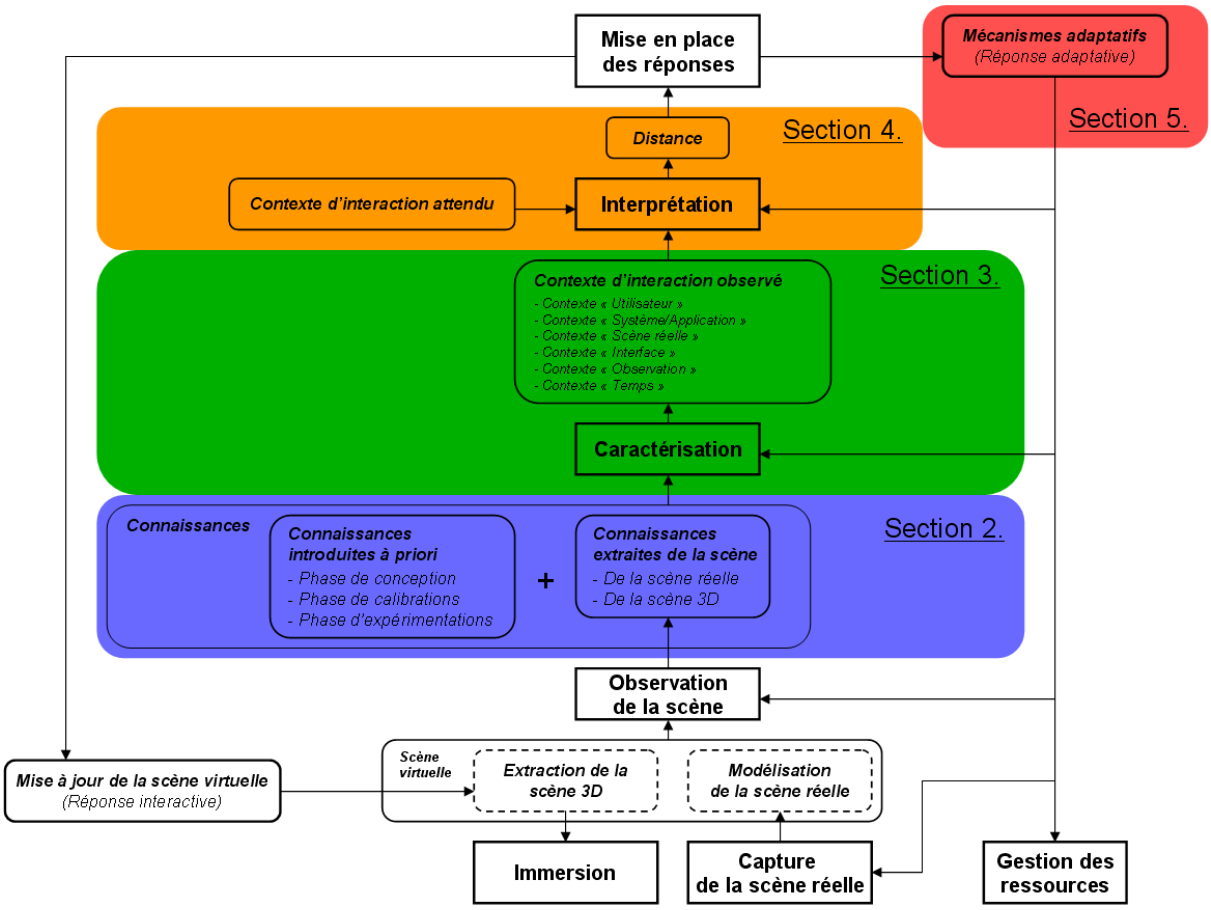


Figure 5.2. : Présentation des différentes sections de ce chapitre

1. Etude de cas

Pour illustrer notre approche, nous avons développé une application, de type *exergame* (*exercise* + *entertainment*), immergeant l'utilisateur **au sein d'un entrainement de tennis**.

Une partie de jeu comprend **15 lancers de balles**. Chaque partie est divisée en **3 phases de jeu** de **5 lancers**. L'utilisateur a pour objectif d'**intercepter les balles**, lancées par le canon se trouvant en face de lui, avec l'aide de **ses mains**. Dans la mesure où il s'agit d'un entrainement au tennis, l'utilisateur doit intercepter les balles lancées par le biais de **coups droits** (interception avec la main se situant du même côté que la balle lancée) et de **revers** (interception avec la main se situant du côté opposé à celui de la balle lancée). La partie se termine à l'issue des 15 lancers.

L'utilisateur évolue au sein de l'espace physique délimité par la structure du ***Cyberdôme***. Il est libre d'adopter toutes gestuelles qu'il juge adéquates, en accord avec le scénario de l'application. D'éventuels spectateurs peuvent se situer dans le fond de la scène, derrière l'utilisateur.
De plus, l'utilisateur est représenté visuellement par un avatar, évoluant au sein d'un terrain de tennis 3D. L'utilisateur se trouve à une extrémité du terrain. En face de lui, à l'autre extrémité du terrain, se trouve un canon à balles. Ce canon lance, selon des intervalles de temps déterminés par le scénario et contrôlés par l'application, des balles de tennis, que

l'utilisateur doit intercepter. Le dispositif d'immersion, utilisé dans le cadre de nos travaux, restitue à l'utilisateur le rendu 3D du terrain de tennis, au sein duquel son avatar évolue.

Notre application est basique et son scénario est simple, facilement analysable et développable. L'utilisateur doit effectuer des gestuelles simples, faciles à apprendre et à exécuter. Cependant, ce scénario nous a permis de valider notre approche, rapidement, et non forcément de manière complexe. Il a permis de mettre en avant les différentes problématiques, auxquelles nous avons fait face et d'illustrer l'ensemble de nos solutions de manières efficace et correcte.
Il est à remarquer que, dans le cadre de notre application sportive, le scénario de cette dernière implique, de la part de l'utilisateur, l'usage global de son corps. L'emploi d'un système de capture de mouvements du corps de l'utilisateur dans sa globalité s'avère donc pertinent.

Enfin, par le biais de cette application, nous illustrons **l'adaptativité** de notre système, vis-à-vis de **contextes d'interaction identifiés et caractérisés**. Les différentes informations contextuelles et les mécanismes d'adaptation sont simples à définir et à mettre en œuvre. Après avoir exposé notre approche dans les prochaines sections, nous appliquons cette dernière à notre scénario, au cours de la Section 6.

2. Connaissances utilisées par notre système

Cette section répertorie l'ensemble des connaissances utilisées au cours de l'interactivité, introduites lors **d'une phase antérieure** à cette dernière ou **construites au cours de celle-ci** (voir Figure 5.3.). Ces connaissances, une fois considérées par le processus associé, permettront **la caractérisation des différents contextes d'interaction**, au sein desquels l'utilisateur évolue.

Définies sur plusieurs niveaux sémantiques, ces connaissances peuvent être introduites au sein du système ***a priori***, i.e. lors de phases antérieures (conception, calibrages, réglages et expérimentations) à l'interactivité à proprement parler avec l'utilisateur (Section 2.1.).
Elles peuvent également être **construites au cours de l'interactivité**, à partir de la scène virtuelle modélisée par le système (Section 2.2.).

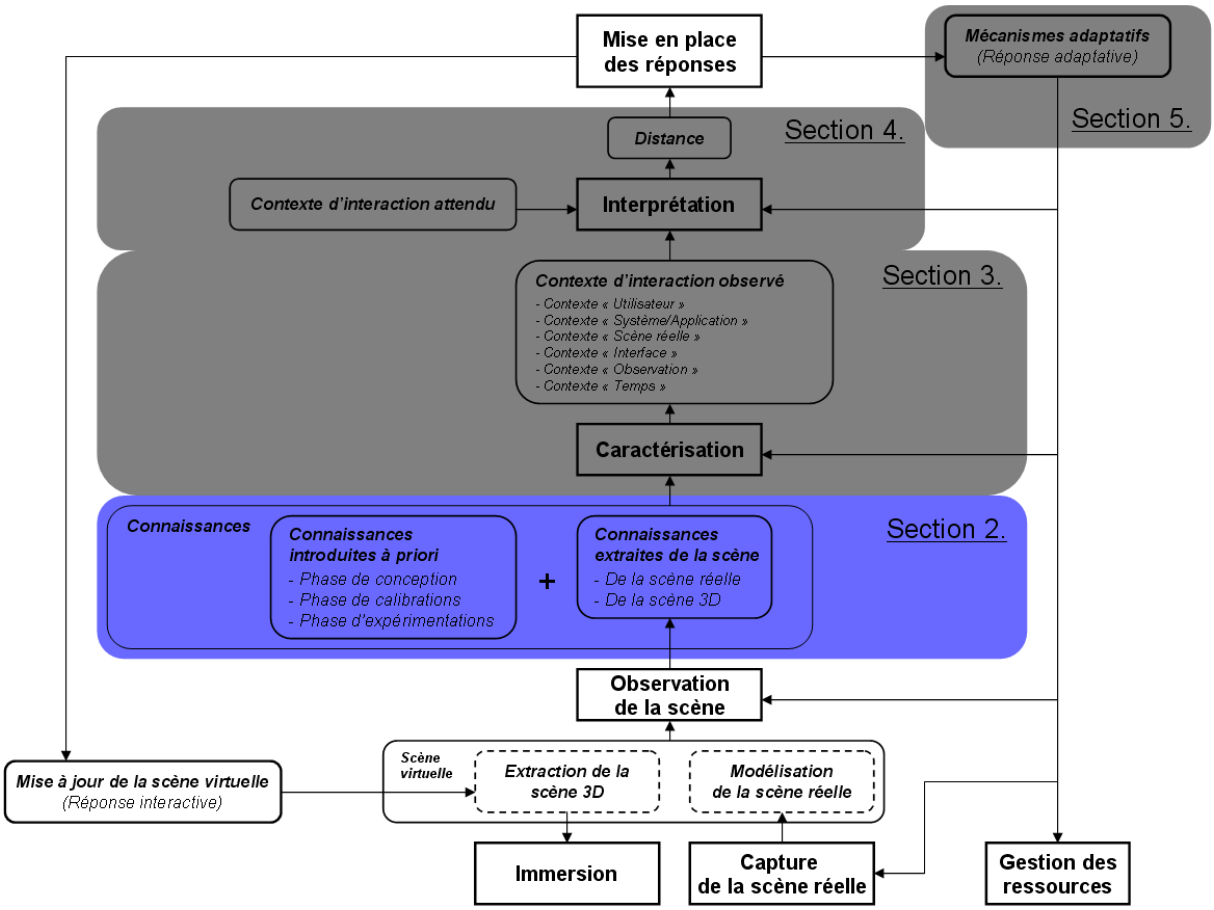


Figure 5.3. : Connaissances utilisées par notre système

2.1. Connaissances introduites a priori

Ces connaissances sont déterminées au cours de **la phase de conception** du système, ou sont établies lors **de phases de calibrages et d'expérimentations**, antérieures à l'interactivité. Cet ensemble de connaissances est utilisé et modifié par les processus composant le système, au cours des différentes interactions avec l'utilisateur.

La capture de la scène réelle, mise en œuvre par le ***Cyberdôme*** augmenté de nos contributions, nécessite **trois étapes de calibrages**, apportant un ensemble de connaissances au processus.
La première étape de calibrage permet de calculer les matrices caméras, permettant le passage mathématique entre un espace 2D (l'image) et un espace 3D (la scène 3D). Nos caméras étant fixes, ces matrices ne varient pas au cours de l'interactivité.
La seconde étape calibre les dimensions de la représentation virtuelle de l'utilisateur en accord avec celles de son corps. De même, pour un même utilisateur au cours de l'interactivité, ces connaissances ne varient pas.
La dernière étape de calibrage concerne la reconnaissance des marqueurs colorés au cours de la capture. Nous ne considérons plus ces connaissances dans le cadre de ce chapitre, car un de nos objectifs est de développer un processus de capture non invasive, du point de vue de l'utilisateur.

Le scénario de l'application constitue également un ensemble important d'informations, apportées à l'ensemble du système lors de la phase de conception. Bien que le scénario évolue au cours de l'interactivité, la structure et le contenu global de celui-ci ne varient pas, tout du moins dans nos travaux. Nous partons également du principe qu'une seule application est envisagée pour une séquence interactive donnée. De ce scénario peuvent alors être extraits les différentes situations et contextes d'interaction au sein desquels l'utilisateur doit évoluer, les objectifs qu'il doit atteindre, les réponses du système, qu'elles soient interactives ou adaptatives, etc.
De plus, c'est du scénario que sont extraites les différentes connaissances relatives à la description de la scène virtuelle. En effet, pour chaque nouveau scénario, une nouvelle scène virtuelle est modélisée et mise à jour en fonction de l'évolution de ce dernier.

Dans le cadre de nos travaux, l'ensemble des **aspects matériels** du système et la grande majorité des **aspects logiciels** de ce dernier représentent un ensemble de connaissances introduites *a priori*, au cours de la phase de conception. Par exemple, certains algorithmes sont dépendants du matériel utilisé, ou de la version logicielle d'une bibliothèque de développement donnée. Ces informations sont déterminées par les concepteurs au moment de l'élaboration du système. Elles ne varient pas au cours de l'interactivité (dans nos travaux en tout cas).
De plus, au cours de l'interactivité, les différents algorithmes, mis en œuvre pour assurer la bonne réponse du système vis-à-vis de la scène interprétée, sont **configurés** et **pilotés** par un **ensemble de paramètres variables**. Ces paramètres, tout du moins les paramètres initiaux et leurs intervalles d'évolution, sont déterminés lors de phases d'expérimentations et de réglages du système. C'est le cas, par exemple, pour les algorithmes de segmentation et de capture, mis en jeu par le ***Cyberdôme***. Ces paramètres peuvent ne pas être modifiés au cours de l'interactivité, ou bien varier en fonction de ce qui est capturé au sein de la scène réelle.

Peu de connaissances sont *a priori* nécessaires concernant **la scène réelle**. La plupart des aspects de celle-ci ne changent pas au cours de l'interactivité et ne sont pas utiles au système. C'est le cas par exemple de la configuration spatiale des différents objets fixes de la scène réelle, des lumières, etc. L'algorithme de modélisation dynamique du fond, présenté au cours de la Section 3.5.1.3. du chapitre Chapitre 4., permet de faire face au dynamisme relatif de la scène réelle.
Certaines connaissances peuvent tout de même s'avérer utiles pour la bonne capture de la scène réelle. L'utilisateur évolue dans un espace limité, ce qui se traduit, au niveau de l'image, par une zone 2D particulière et, au niveau de la scène 3D, par une zone 3D spécifique. Il en est de même pour l'emplacement de spectateurs éventuels. Bien sûr, ces zones ne sont pas sûres à 100% au cours du temps, l'utilisateur ou le spectateur pouvant évoluer où bon lui semble. Mais, ces connaissances s'avèrent vraies et vérifiables au cours de la majeure partie de l'interactivité, à partir du moment où l'utilisateur suit le scénario de l'application.
Enfin, plusieurs connaissances relatives à l'interface matérielle du système peuvent potentiellement être utilisées au cours de l'interactivité. Ainsi, une de nos 4 caméras a été sélectionnée et indexée logiciellement. Cette caméra a été élue pour deux raisons. Tout d'abord, c'est elle qui observe le mieux notre scène réelle. De plus, se trouvant au dessus de la zone d'immersion, c'est en face d'elle que l'utilisateur évolue. Nos caméras étant fixes, cette connaissance ne varie pas au cours de l'interactivité.

2.2. Connaissances construites à partir de la scène virtuelle

Comme nous l'avons mentionné précédemment, **la scène virtuelle**, construite par le système, comprend la modélisation de la scène capturée par le ***Cyberdôme***, augmenté de nos contributions (voir chapitre précédent), et celle de la scène 3D immersive, restituée à l'utilisateur au cours de l'interactivité. La scène virtuelle rassemble donc les différentes connaissances, que le gestionnaire dédié du système établit et gère au cours de l'interactivité avec l'utilisateur, **pour caractériser le contexte d'interaction**, au sein duquel l'activité observée prend place.

La Section 2.2.1. énumère les connaissances établies à partir de la scène réelle capturée, telle qu'elle est modélisée au sein de la scène virtuelle.
La Section 2.2.2. énumère celles construites à partir de l'exploitation de la scène 3D, par le gestionnaire de la scène virtuelle.

2.2.1. Connaissances établies à partir de la scène réelle

Nous considérons ici les connaissances relatives à la gestion de **la scène réelle**, telle qu'elle est capturée par le ***Cyberdôme*** augmenté de nos contributions. Nous avons déjà partiellement traité de cette problématique lors de notre présentation du ***Cyberdôme*** au cours du chapitre précédent. Nous rappelons que nous nous intéressons uniquement ici aux connaissances liées à la scène réelle et construites au cours de l'interactivité. Les connaissances *a priori* sont détaillées au cours de la Section 2.1.

Tout d'abord, le processus de capture s'effectue dans le temps. Il est donc tout à fait possible de prendre en compte un intervalle temporel entre deux instants particuliers pour, par exemple, simplement observer la gestuelle de l'utilisateur au cours de cet intervalle. Cet intervalle de temps peut être mesuré en termes d'unités temporelles standards (la seconde, la minute, etc.) ou en *frames*.

Ensuite, la scène réelle est capturée et modélisée à partir de l'analyse de 4 points de vue différents. Ces points de vue sont matérialisés par 4 flux vidéo 2D temps réel. Ainsi, toute caractéristique 2D pouvant être extraite d'images 2D, représentant notre scène réelle sous plusieurs angles de vue, peut être utilisée par le processus de caractérisation du contexte d'interaction. Dans nos travaux, nous extrayons les silhouettes de l'utilisateur et un ensemble de zones 2D, calculées à partir de critères spécifiques que nous détaillerons par la suite.

2.2.2. Connaissances construites à partir de l'exploitation de la scène 3D

Les gestuelles qu'adopte l'utilisateur animent un avatar, personnage 3D, représentant l'utilisateur au sein de la scène 3D. Cet avatar, modélisé à partir de sa description extraite du scénario de l'application, est lié, sémantiquement et numériquement, au modèle 3D utilisé par le processus de capture de gestuelles.
Comme nous l'avons mentionné précédemment, le processus de capture de gestuelles mis en œuvre par le ***Cyberdôme*** suit une approche basée modèle. Ce modèle 3D est composé d'une hiérarchie de simples primitives rigides (non déformables au cours du temps), articulées les unes avec les autres et sémantiquement différentiables. Chaque primitive est associée à une partie spécifique du corps de l'utilisateur, comme une épaule ou un avant bras par exemple.

Ainsi, le système peut considérer uniquement une partie du corps de l'utilisateur, plusieurs parties de son corps (de manière synchronisée ou non), ou son corps dans sa globalité. De plus, à l'instar des primitives, le système peut considérer les articulations de l'avatar, ces dernières étant modélisées précisément (nature, limites, etc.).
Enfin, chaque primitive et articulation de l'avatar est positionnée et orientée dans un repère 3D, au cours du temps. Autrement dit, pour une *frame* donnée, il est possible de calculer les positions et rotations d'un élément particulier du modèle représentant l'utilisateur. A partir de ces positions et rotations, les règles quantitatives et qualitatives qui régissent la gestuelle de l'utilisateur, au cours étape scénarisée donnée, peuvent être établies.

En plus de l'ensemble des connaissances uniquement liées à l'avatar, il est également possible de construire, à partir de la scène 3D telle qu'elle est gérée par le gestionnaire de la scène virtuelle, plusieurs autres connaissances spécifiques, nécessaires à la bonne caractérisation des contextes d'interaction au sein desquels l'activité prend place.
En fonction du scénario de l'application, il nous est possible de mettre en relation spatialement et temporellement les différentes parties du corps de l'utilisateur avec les différents éléments composant la scène 3D. Chaque élément de la scène 3D, élément interactif ou appartenant à l'avatar, est sémantiquement identifiable et positionnable dans l'espace 3D. A un instant donné ou au cours d'un intervalle de temps particulier, le système peut considérer une ou plusieurs zones 3D au sein de la scène 3D, pour établir les relations spatiales et temporelles entre ces éléments.
De plus, les différents éléments, composant la scène 3D, peuvent être associés à des comportements interactifs, déclenchés par des gestuelles utilisateur particulières. Ces ressources interactives, et leurs comportements associés, sont développées lors de la phase de conception et sont prévues (mise à jour, activation / désactivation, etc.) lors du déroulement du scénario de l'application. Le gestionnaire de la scène virtuelle est alors chargé de gérer ces ressources et les différents événements interactifs que leurs comportements génèrent au sein de la scène 3D. Par exemple, si l'avatar, animé par l'utilisateur, interagit avec un bouton 3D, l'événement (la collision entre l'élément composant l'avatar et la ressource interactive) est typiquement détecté par le gestionnaire de la scène virtuelle, mettant alors en œuvre les comportements spécifiques associé à ce bouton 3D. Les écoutes et détections de ce type d'événements interactifs, par le gestionnaire de la scène virtuelle, sont prévues lors du développement de l'application.

3. Caractérisation du contexte d'interaction

Les différentes connaissances, énumérées précédemment, nous permettent **de caractériser les différents contextes d'interaction**, au sein desquels l'utilisateur évolue et, de manière générale, l'activité prend place, au cours de l'interactivité. Cette caractérisation est proposée au cours de cette section (voir Figure 5.4.).
Notre caractérisation reflète **l'activité**, observée et attendue, au sein de la scène virtuelle, et est **le point de départ du processus d'interprétation**, supporté par le scénario de l'application. Le contexte d'interaction fournit également un ensemble d'informations contextuelles **qui paramètre les mécanismes adaptatifs**, mis en place par le système au cours de l'interactivité, après son interprétation de l'activité.
La Section 3.1. présente notre caractérisation générale d'un contexte d'interaction. Elle englobe donc la définition des contextes détaillés au cours de la Section 6.4. du Chapitre 2. : contexte **« Utilisateur »**, contexte **« Système/Application »**, contexte **« Scène réelle »**, contexte **« Interface »**, contexte **« Observation »** et contexte **« Temps »**.

Nous détaillons particulièrement, au cours de la Section 3.2., la caractérisation de **la gestuelle utilisateur**. Cette caractérisation est comprise au sein du contexte **« Utilisateur »**, lui-même compris dans un contexte d'interaction donné. Notre définition de la gestuelle utilisateur est fonction de 3 caractéristiques :

- **Des règles qualitatives et quantitatives** suivies par la représentation virtuelle de l'utilisateur (ou certains éléments de celle-ci) ;
- **Des relations spatio-temporelles** existant entre la représentation virtuelle de l'utilisateur (ou certains éléments de celle-ci) et la scène virtuelle (ou certains éléments de celle-ci) ;
- **Des événements interactifs générés par les éléments actifs** au sein de la scène virtuelle.

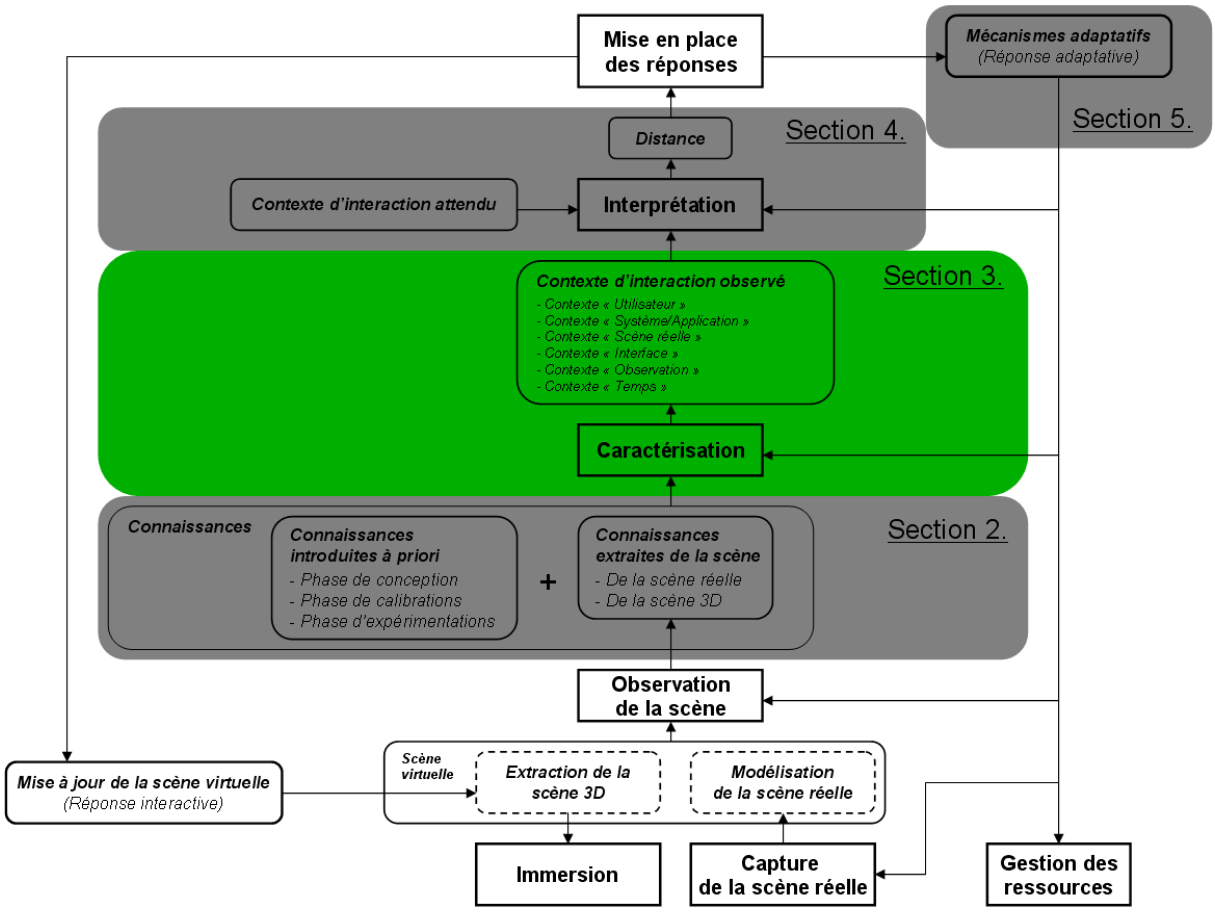


Figure 5.4. : Caractérisation du contexte d'interaction

3.1. Contexte d'interaction

Dans nos travaux, **le contexte d'interaction** est modélisé à partir **des observations effectuées sur la scène virtuelle**, qui traduit l'activité observée en cours.
Ce contexte d'interaction regroupe à la fois les caractérisations des activités observées et attendues par le scénario de l'application. Nous mentionnerons donc un contexte d'interaction **observé**, et un contexte d'interaction **attendu**, ce dernier étant une connaissance établie *a*

priori et faisant partie de notre modélisation du contexte d'interaction observé. Le contexte d'interaction observé et le contexte d'interaction attendu comprennent les mêmes sous-contextes, sous-contextes caractérisant respectivement ce qui est construit au cours de l'interactivité et ce qui est attendu par le scénario de l'application au cours de celle-ci.
Par la suite, **le processus d'interprétation** utilise le contexte d'interaction modélisé pour confronter les caractérisations observées et attendues, dans le but de reconnaître l'activité au sein de la scène.

La section précédente identifie les différentes **connaissances**, à partir desquelles nous caractérisons un contexte d'interaction. Un contexte d'interaction, comme nous l'avons présenté au cours de la Section 6.4. du Chapitre 2., comprend plusieurs sous-contextes.
Cette section décrit donc ces différents sous-contextes, et les informations contextuelles comprises au sein de ceux-ci, en nous plaçant dans le cadre de nos travaux de thèse :

- Contexte **« Utilisateur »** (Section 3.1.1.).
- Contexte **« Système/Application »** (Section 3.1.2.).
- Contexte **« Scène réelle »** (Section 3.1.3.).
- Contexte **« Interface »** (Section 3.1.4.).
- Contexte **« Observation »** (Section 3.1.5.).
- Contexte **« Temps »** (Section 3.1.6.).

Il est à noter que certains contextes, dans nos travaux, se révèlent peu importants et nécessaires pour illustrer notre approche. C'est pourquoi nous nous attarderons, de manière plus approfondie, sur les contextes que nous avons jugés plus pertinents dans le cadre de nos travaux.
La Figure 5.5. résume les propos de cette section.

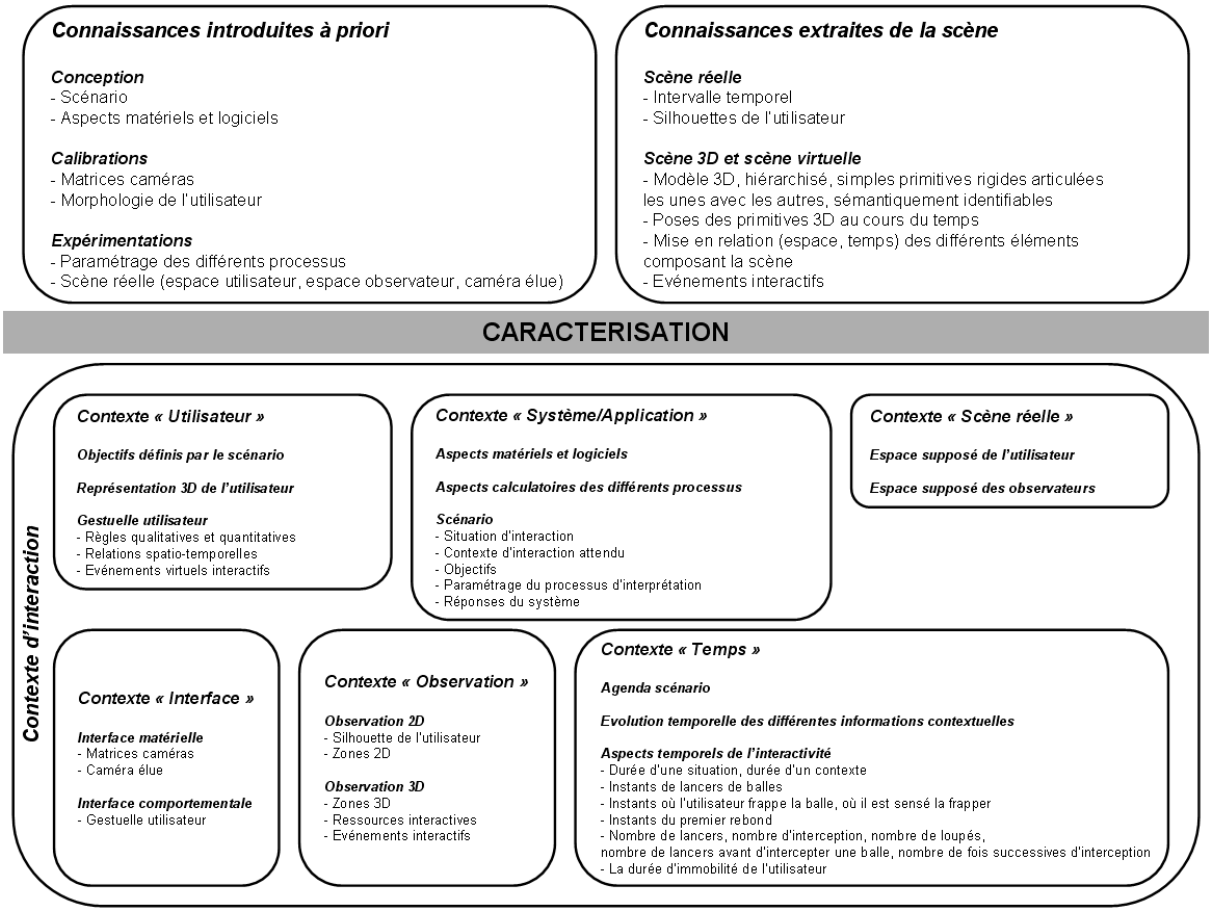


Figure 5.5. : Caractérisation du contexte d'interaction à partir des connaissances du système

3.1.1. Contexte « Utilisateur »

Le contexte **« Utilisateur »**, dans le cadre de nos travaux, comprend, d'une part, **les objectifs** de l'utilisateur au cours des situations et contextes d'interaction dans lesquelles il évolue, et, d'autre part, **les gestuelles** qu'il doit adopter pour résoudre ces situations et contextes.

Les objectifs de l'utilisateur sont définis par le scénario de l'application. Ces connaissances caractérisent ce qui est attendu par le système au cours de l'interactivité.
Les gestuelles utilisateur, quant à elles, sont caractérisées comme il le sera exposé au cours de la Section 3.2. Les caractérisations des gestuelles peuvent traduire aussi bien l'activité observée au cours de l'interactivité (caractérisations calculées au cours de l'interactivité), que l'activité attendue. Dans ce dernier cas, les caractérisations des gestuelles sont déterminées lors de la phase de conception du système et sont des connaissances inclues dans le scénario de l'application. Elles seront donc comprises dans les contextes d'interaction attendus par le système.
Le contexte « Utilisateur » comprend également **la description de la représentation de l'utilisateur** (identité sémantique des parties du corps, hiérarchie, etc.) au sein de la scène virtuelle. Ces connaissances sont établies lors de la phase de conception.

Nous caractérisons donc ce sous-contexte vis-à-vis plutôt de l'activité, observée et attendue, de l'utilisateur, sans prendre en compte son identité propre ou son état interne. Cela dit, il sera

bien sûr possible par la suite de complexifier ce sous-contexte, en prenant en compte, par exemple, des profils, concernant son identité ou sa représentation au sein de la scène virtuelle.

3.1.2. Contexte « Système/Application »

Les sous-contextes généraux, compris au sein du contexte « Système/Application », relatifs aux **aspects matériels**, **logiciels**, **réseau**, etc. varient d'un système à un autre et d'une application scénarisée à une autre. Ils sont établis à partir des connaissances logicielles (bibliothèques utilisées, etc.) et matérielles (configuration réseau, matériel utilisé, etc.), introduites *a priori* au sein du système lors de la conception.
Ces contextes sont nombreux et variés mais ne présentent que peu d'importance dans le cadre de nos travaux, dans la mesure où nous n'utilisons qu'un système de capture et qu'une application pour illustrer ces travaux de thèse. Nous avons toutefois, au cours de la phase de conception, cherché à optimiser l'ensemble des algorithmes vis-à-vis d'un grand nombre d'aspects matériels et logiciels.

Les propos sont sensiblement les mêmes concernant **les sous-contextes calculatoires** du système. Chaque couche de l'architecture comprend un contexte calculatoire particulier, défini à un niveau sémantique donné. De plus, tous les modules d'une couche prennent en compte des informations contextuelles particulières, de même que tous les processus mis en jeu au cours de l'interaction. Ces contextes, nombreux et complexes, sont déterminés et initialisés par le biais de phases d'expérimentations, mais peuvent évoluer au cours de l'interactivité, en fonction des scénarios que suivent les différents processus mis en jeu.
Cela dit, seuls certains sous-contextes calculatoires sont particulièrement importants, dans le cadre de nos travaux. Généralement, ces contextes sont fortement liés à d'autres contextes plus parlants et plus 'visuels', comme l'ensemble des zones 2D, au niveau du contexte « Observation », qui pilotent les algorithmes de capture de la scène réelle et ceux d'observation de la scène virtuelle. Ces zones 2D, traduites en termes de paramètres pour ces algorithmes, sont plus explicites dans le cadre de l'étude du contexte « Observation », car visuellement observables. Nous détaillerons donc plus précisément ces informations contextuelles au cours de la Section 3.1.5.

Enfin, **le sous-contexte lié à l'application** comprend principalement les différentes informations contextuelles établies à partir du scénario de l'application. Ce sous-contexte définit à un instant donné la situation d'interaction face à laquelle l'utilisateur se trouve, les objectifs qu'il doit remplir pour la résoudre, le ou les contextes d'interaction au sein desquels il est supposé évoluer (vis-à-vis du scénario de l'application), les paramètres devant configurer le processus d'interprétation de la scène et les réponses (interactives et adaptatives) que le système doit mettre en place en fonction de l'activité au sein de la scène. Le contexte d'interaction attendu par le système, via le scénario de l'application, est ainsi défini.
Ces informations contextuelles permettent véritablement l'interaction adaptée du système avec l'utilisateur. Pour une étape scénarisée donnée, les objectifs de l'utilisateur sont définis. Ce dernier, cherchant à les atteindre, évolue au sein de la scène virtuelle. De cette scène sont extraites des connaissances spécifiques, qui, ajoutées aux connaissances introduites *a priori*, permettent la caractérisation du contexte d'interaction. Ce contexte d'interaction observé est confronté, au cours du processus d'interprétation de la scène, au contexte d'interaction attendu par le scénario. Le processus d'interprétation, configuré par les différents paramètres fournis par le sous-contexte lié à l'application, permet la mise en place des réponses du

système. Le scénario évolue alors, impliquant la mise à jour du sous-contexte lié à l'application. Et le processus interactif recommence.
Le scénario, dans le cadre de nos travaux, étant fixe, les différentes informations contextuelles décrites ci-dessus sont déterminées au cours de la phase de développement de l'application. En revanche, au cours de l'interactivité, elles sont introduites au sein du processus interactif, par le biais du sous-contexte lié à l'application, en fonction de l'évolution du scénario.
Le contexte lié à l'application, par le biais du scénario, regroupe également les caractéristiques descriptives de la scène 3D immersive, caractéristiques définies au cours du développement de l'application et évoluant au cours de l'interactivité.

3.1.3. Contexte « Scène réelle »

Concernant **le contexte « Scène réelle »**, notre scène réelle est une salle de classe. Cet environnement évolue peu en termes de configuration spatiale et de luminosité ambiante. Nous pouvons donc considérer ces connaissances comme étant absolues, bien que nous ne les utilisions pas véritablement dans le cadre de nos travaux.

Au sein de notre système, l'utilisateur évolue dans un espace physique, précis et bien délimité. Cet espace constitue une connaissance, comprise au sein du contexte « Scène réelle », introduite au sein du système lors de la phase de conception et ajustée lors de phases d'expérimentations.

Dans notre contexte d'étude, les variations (couleur, luminosité, etc.), pouvant avoir lieu au sein de la scène réelle, peuvent être générées par la présence éventuelle de spectateurs. Ces spectateurs se situent dans le fond de la scène réelle, derrière et autour de l'utilisateur, face aux caméras, regardant la projection de la scène 3D immersive.
Cela dit, le système physique de capture de mouvements est imposant et la surface où évolue l'utilisateur bien délimitée. Les emplacements des spectateurs éventuels sont donc localement délimitables. Ce sont typiquement des connaissances comprises au sein du contexte « Scène réelle », connaissances introduites *a priori*, et ajustées lors de phase d'expérimentations.

Le contexte « Scène réelle » peut supporter la caractérisation à la fois de l'activité observée au sein de la scène virtuelle, et l'activité attendue par le scénario de l'application.

3.1.4. Contexte « Interface »

Le sous-contexte, compris dans le contexte « Interface » et lié à **l'interface logicielle**, n'est pas nécessaire à l'illustration de ces travaux de thèse. Les caractéristiques de simulation de la scène virtuelle, de visualisation de cette dernière, etc. ne varient pas au cours de l'interactivité, dans le cadre de nos travaux. Toutefois, ce sont des aspects intéressants, qui pourront être considérés dans le futur, pour illustrer la capacité du système à s'adapter de par lui-même au cours de l'interactivité.

Le sous-contexte lié à **l'interface matérielle** est plus important et présente un ensemble d'informations utilisées dans nos travaux. En effet, notre interface matérielle ne varie pas au cours de l'interactivité. Ainsi, certaines connaissances, pouvant être considérées comme absolues i.e. vraies à tout moment, peuvent être déterminées lors de phases antérieures à l'interactivité (comme la phase de calibrage des caméras, par exemple). Ces connaissances

regroupent à la fois des connaissances obligatoires pour le bon fonctionnement de notre système, notamment du processus de capture (typiquement, les matrices caméras) et des connaissances qui peuvent s'avérer décisives lors de l'interprétation de l'activité observée. Ainsi, nous avons sélectionné et indexé logiciellement une de nos caméras, considérant cette information comme vraie tout au long de l'interactivité. Cette caméra est celle se trouvant la plus en face de l'utilisateur, englobe complètement la scène réelle et l'utilisateur et est située au niveau de la projection immersive, impliquant que l'utilisateur regarde dans sa direction. Plusieurs algorithmes sont dédiés au point de vue défini par cette caméra élue, tels que l'algorithme de détection du visage de l'utilisateur, qui doit nous permettre, ensuite, de distinguer la droite de la gauche relativement aux parties du corps de celui-ci.

Enfin, concernant **l'interface comportementale**, les techniques d'interaction adoptées par l'utilisateur sont empruntées à celle du tennis. Elles sont établies en accord avec le scénario de l'application, lors de l'écriture de ce dernier.
Le contexte lié à l'interface comportementale est évidemment fortement lié au contexte « Utilisateur », au sein duquel sont décrites les gestuelles caractérisées de l'utilisateur (voir Section 3.2.).

L'ensemble des informations contextuelles comprises au sein du contexte « Interface » peut servir à la caractérisation des contextes d'interaction observés et attendus.

3.1.5. Contexte « Observation »

Dans nos travaux, le contexte « Observation » est essentiel. Ce sous-contexte regroupe un ensemble de connaissances principalement **dédiées à l'observation de la scène virtuelle**. Ces connaissances permettent d'orienter les différents traitements et d'ainsi optimiser et rendre plus efficace **le processus d'observation de la scène virtuelle**. Dans nos travaux, nous utilisons également une partie des informations contextuelles du contexte « Observation » (observation 2D) pour améliorer **la capture de la scène réelle**.
Les informations contextuelles comprises au sein de ce sous-contexte sont gérées à la fois par les scénarios de l'application et ceux des différents processus les manipulant. Ainsi, leur mise à jour peut dépendre du déroulement de l'application et de l'activation d'une situation et/ou d'un contexte particulier (observation d'une partie du corps de l'utilisateur particulière, par exemple), ou de l'activité observée au sein de la scène réelle (agrandissement d'une zone d'observation en fonction du mouvement enregistré au sein de la scène réelle, par exemple). Ces connaissances sont donc déterminées au cours de la phase de conception du système et de l'interactivité.
Quoiqu'il en soit, l'ensemble des connaissances, regroupées dans ce sous-contexte, peut supporter les caractérisations des contextes d'interaction observés et attendus.

Le contexte « Observation » comprend un ensemble d'informations, **2D** (sous-contexte lié à l'observation 2D de la scène réelle capturée) ou **3D** (sous-contexte lié à l'observation 3D des scènes 3D et virtuelle). Ces informations sont principalement un ensemble de régions d'intérêt, de zones, combinables les unes avec les autres, et dédiées à la redirection, au sein de ces dernières, des processus d'observation de la scène virtuelle et de capture de la scène réelle. Ainsi, ces zones permettent la capture et l'observation focalisées de la scène virtuelle (**gestion multiple-focus**), tout en continuant de capturer et d'observer, dans le même temps et de manière globale, l'ensemble de celle-ci.

Le contexte lié à l'observation 2D de la scène réelle comprend la majorité des informations contextuelles que nous utilisons dans le cadre de ces travaux. Il rassemble, pour un point de vue donné :

1. La silhouette de l'utilisateur
L'ensemble des silhouettes de l'utilisateur est le point de départ obligatoire du processus de capture de gestuelles. Une silhouette 2D est extraite de chaque point de vue. Le processus d'extraction de la silhouette de l'utilisateur est détaillé au cours du Chapitre 4. La silhouette de l'utilisateur est donc une connaissance construite au cours de l'interactivité, et mise à jour pour chaque nouvelle *frame*.

2. Un ensemble de zones 2D
Ces zones 2D sont liées à l'utilisateur (ou à certaines parties de son corps), et aux autres événements, prenant place au sein de la scène réelle, et dont l'utilisateur n'est pas la source (mouvement généré par un spectateur, par exemple).

Les zones sont construites à partir de nombreuses connaissances, aussi bien introduites *a priori* au sein du système (configuration spatiale de la scène, matrices caméra calculées à partir des phases de calibrage, scénario de l'application, ajustement des connaissances par des phases d'expérimentations), que construites à partir des processus de capture de mouvements et de modélisation dynamique du fond (Section 3.5.1.3. du chapitre précédent). Ces connaissances sont utilisées pour la construction de plusieurs autres sous-contextes (contextes « Scène réelle », « Interface », « Système/Application »).

Dans le cadre de nos travaux, nous construisons les zones 2D suivantes :

- **La zone SASM** (*Safe Area around Segmentation & Model*) englobant l'utilisateur. Cette zone a été définie au cours de la Section 3.5.1.3. du Chapitre 4. Cette zone est la combinaison de **la zone SAS** (*Safe Area around Segmentation*) englobant la silhouette segmentée de l'utilisateur, et de **la zone SAM** (*Safe Area around Model*) englobant la silhouette projetée du modèle 3D utilisé par le processus de capture de gestuelles. Ainsi, cette zone SASM englobe à coup sûr l'utilisateur. Cette zone 2D est calculée pour chaque nouvelle *frame*.
- **Les zones SABE** (*Safe Areas around Body Elements*) qui englobent certains éléments spécifiques de la représentation virtuelle de l'utilisateur, suivant différents niveaux : le niveau *Body Part*, le niveau *Member* et le niveau *Body* (voir Section 3.2.1.). Ces zones sont déterminées par le scénario de l'application (contexte « Système/Application ») et mettent en relief certaines parties du corps de l'utilisateur. Elles sont calculées, pour chaque nouvelle *frame*, à partir de la projection du modèle 3D utilisé par le processus de capture de gestuelles, qui est effectuée partie du corps par partie du corps.
- Le contexte « Scène réelle » nous permet de déterminer l'ensemble des **zones SAO** (*Safe Areas around Observers*) englobant l'emplacement 2D d'éventuels spectateurs, dans la scène réelle, pour un point de vue donné.
- **La zone SAMP** (*Safe Area around Moving Pixels*) est construite à partir de l'image différence calculée à partir de la *frame* à l'instant t et celle à l'instant t-1. Cette image différence, calculée par le biais de l'algorithme de modélisation du fond de la scène réelle ([Yang & al. 04]), nous indique les pixels dont l'intensité a varié entre deux

frames consécutives. Autrement dit, ces pixels appartiennent aux éléments mobiles de la scène réelle, ayant évolué entre deux *frames* consécutives. La zone SAMP traduit le dynamisme de la scène réelle à un instant donné.
La zone SAMP peut être combinée avec les zones SASM, SABE et SAO pour déterminer les zones englobant, respectivement, les parties en mouvement (entre deux *frames* consécutives) du corps de l'utilisateur ; les parties en mouvement (entre deux *frames* consécutives) du corps de l'utilisateur mises en relief par le scénario de l'application ; les zones où se trouvent d'éventuels spectateurs dénotant un mouvement (entre deux *frames* consécutives) de leur part. Ainsi combinées, ces zones peuvent améliorer le processus d'interprétation de la scène virtuelle.

- Il est tout à fait possible, au lieu de considérer les zones précédemment décrites, de considérer **leurs complémentaires**. Ainsi, il sera possible d'observer et de traiter particulièrement la zone où ne se trouve pas l'utilisateur, les parties du corps de l'utilisateur qui ne sont pas en mouvement, etc. C'est au sein de cette zone que des perturbations dans la scène réelle peuvent avoir lieu, perturbations n'ayant pas pour origine la gestuelle de l'utilisateur. Ces zones sont à surveiller au cours de l'interactivité et ne doivent pas perturber le déroulement de celle-ci.
- Enfin, de par la suppression des marqueurs colorés du processus de capture de mouvements initial, le système n'est pas capable de distinguer la droite de la gauche de l'utilisateur (bras droit – bras gauche, jambe droite – jambe gauche), uniquement par le biais des silhouettes de ce dernier. Le contexte « Interface » nous permet de connaître la caméra se situant la plus en face de l'utilisateur (Section 3.1.4.). Un algorithme de détection du visage de l'utilisateur, exécuté sur le point de vue défini par cette caméra, peut permettre de diviser l'image en 2 zones, une zone correspondant à **la partie droite de l'image**, l'autre à **la partie gauche**. En effet, la détection du visage permet de faire de fortes hypothèses sur sa position vis-à-vis du point de vue considéré et donc de résoudre l'ambiguïté droite-gauche que peut présenter une simple silhouette 2D. Ces 2 zones sont calculées chaque fois que le scénario l'implique.

Le contexte « Observation » comprend, d'autre part, un sous contexte dédié à **l'observation 3D de la scène virtuelle** (et donc de la scène 3D restituée à l'utilisateur). Les informations contextuelles comprises dans ce contexte permettent principalement la caractérisation de la gestuelle utilisateur. Ces informations contextuelles sont à coupler avec les objectifs de l'utilisateur. En effet, selon l'objectif de l'utilisateur au cours d'une situation ou d'un contexte d'interaction particulier, ce dernier adoptera une gestuelle appropriée, qui modifiera les différentes informations comprises dans le contexte lié à l'observation 3D.

Ce contexte comprend, tout d'abord, **un ensemble de zones 3D**, permettant la caractérisation des relations spatiales, à un instant donné ou au cours d'un intervalle de temps particulier, entre le modèle représentant l'utilisateur (ou certains éléments de celui-ci) et la scène (ou certaines éléments de celle-ci). En effet, en considérant une zone dans la scène, à un instant donné ou au cours d'un intervalle de temps particulier, il est possible de savoir si elle englobe un élément, sémantiquement identifiable et repéré dans l'espace, du modèle et/ou de la scène (ressource interactive). Cette connaissance peut permettre de déduire une relation spatio-temporelle entre différents éléments de la scène virtuelle, nécessaire à la caractérisation de la gestuelle utilisateur au cours d'une étape d'interaction particulière.

Ces zones 3D sont construites à partir des connaissances sur la scène réelle, l'interface matérielle (matrices caméra) et les objectifs définis par le scénario de l'application. Elles sont mises à jour par le processus de capture et l'évolution du scénario.
Il est à noter que ces zones 3D peuvent être projetées sur un ou plusieurs points de vue particuliers, créant ainsi de nouvelles régions d'intérêt 2D, pouvant être combinées avec les zones 2D précédemment décrites.

Le contexte lié à l'observation 3D comprend également **les différentes ressources interactives et leurs comportements associés**, mises en relief par le scénario de l'application, à un instant donné ou au cours d'un intervalle de temps particulier. Ces ressources, et plus particulièrement les événements interactifs qu'elles déclenchent, sont nécessaires à la bonne caractérisation des gestuelles utilisateur. Ces informations contextuelles sont déterminées lors de l'écriture du scénario, au cours de la phase de conception. Dans notre contexte d'étude, il pourra s'agir des balles de tennis à intercepter, simulées physiquement, et des collisions qu'elles génèrent, pouvant avoir lieu avec les différents éléments du modèle 3D (les mains de ce dernier par exemple).

3.1.6. Contexte « Temps »

Le contexte « Temps », et les informations contextuelles qu'il comprend, permettent de mettre en évidence certains aspects de l'interactivité, **à un instant donné ou au cours d'un intervalle de temps particulier**, et pouvant se révéler nécessaire pour la bonne interprétation de l'activité. Les informations contextuelles, comprises dans ce sous-contexte, sont en grande partie associées avec d'autres informations contextuelles, issues d'autres sous-contextes, de manière à mettre en relief, du point de vue temporel, certains aspects de l'interactivité. Les autres informations contextuelles constituent généralement des comptes, relatifs au nombre d'occurrences d'un événement spécifique qui est mis en relief par le scénario de l'application.

Les différentes informations contextuelles, relatives au contexte « Temps », sont établies lors de la phase de conception et ajustées par des expérimentations, antérieurement à l'interactivité. Comme nous l'avons exposé au cours de la Section 6.5.6. du Chapitre 2., l'écriture du scénario permet la mise en place d'un agenda, qui rythme et contrôle temporellement l'interactivité selon plusieurs niveaux de granularité. Cet agenda permet de suivre, au cours du temps, l'évolution du scénario et des différents éléments d'interaction (situation et contexte).
Les différentes informations contextuelles sont mises à jour au cours de l'interactivité, par le biais des différents scénarios mis en jeu (scénario de l'application et scénarios que suivent les différents processus).

Une information contextuelle est définie et sélectionnée, lors de la phase de conception, si elle s'avère pertinente et nécessaire à la bonne interprétation de la scène observée. Elles doivent également être significatives du point de vue de l'adaptativité du système. En effet, ces informations contextuelles doivent avoir un sens, du point de vue du scénario, dans la mesure où elles paramètrent des mécanismes adaptatifs spécifiques. Autrement dit, si une information contextuelle n'ajoute pas une signification du point de vue de l'interprétation de la scène virtuelle et de la pertinence d'une mesure adaptative du système, elle n'est alors pas à prendre en compte, et par le concepteur lors de l'élaboration du système, et par le système au cours de l'interactivité.

Dans le cadre de nos travaux, nos situations d'interaction sont simples et leur évolution est pilotée par des objectifs basiques, tels que l'interception de la balle par la main de l'utilisateur et le renvoi de celle-ci, de l'autre côté du terrain.
Du point de vue de l'interactivité, nous nous intéressons donc aux différents aspects temporels suivants :

- La durée, supposée ou observée, d'une situation d'interaction ou d'un contexte d'interaction (en secondes ou en *frames*). Nous rappelons que notre agenda est modélisé sous la forme d'une séquence organisée et hiérarchisée de situations et contextes d'interaction. Ces situations ou contextes peuvent être des phases de jeu, par exemple, entre deux lancers, entre deux niveaux de difficulté, entre le début et la fin d'une interaction particulière ou de l'interactivité dans sa globalité, etc. La confrontation entre la durée supposée et celle observée, d'une situation ou d'un contexte d'interaction, peut déclencher la mise en place de mécanismes adaptatifs, comme l'adaptation du niveau de difficulté ou la répétition de la situation ou du contexte.
- Les instants à partir desquels sont lancées les balles par le canon. Ces instants nous permettront de calculer des intervalles de temps spécifiques, entre deux lancers par exemple, ou entre un lancer et une interception.
- Les instants où l'utilisateur est sensé frappé la balle, et où il la frappe réellement (avant le premier rebond de la balle sur le terrain). Les paramètres de lancer étant totalement contrôlés par le système, il est possible de calculer une approximation de l'instant d'interception de la balle par l'utilisateur. En plus du calcul d'intervalles de temps supplémentaire (durée d'un lancer supposé ou observé), la comparaison entre l'instant d'interception supposé avec celui observé pourra permettre l'évaluation du niveau de l'utilisateur.
- Les instants où les balles effectuent leur premier rebond, après avoir été lancées par le canon, qu'elles aient été interceptées par l'utilisateur ou non. Considérant l'instant à partir duquel a été lancée une balle donnée, il est possible de mesurer la durée de ce lancer, jusqu'au premier rebond de la balle. Ainsi, en analysant cet intervalle de temps, il est possible de savoir si l'utilisateur a intercepté la balle ou non.
- Au cours d'une situation ou d'un contexte d'interaction donné, le nombre de lancers ; le nombre de fois que l'utilisateur a intercepté une balle ; le nombre de fois qu'il a loupé une balle ; le nombre de lancers qu'il a fallu avant que l'utilisateur n'intercepte une balle ; le nombre de fois successives qu'il a réussi à intercepter une balle. Ces mesures permettent l'évaluation du niveau de l'utilisateur, impliquant une adaptation du niveau du jeu de la part du système.
- La durée (en secondes ou en *frames*) durant laquelle l'utilisateur n'a pas effectué de mouvements, alors que plusieurs balles ont été lancées. Si cette durée est importante, l'utilisateur ne veut plus jouer ou se désintéresse du jeu (mesure de l'intérêt de l'utilisateur). Le système doit alors intervenir, la mesure adaptative la plus simple de la part de ce dernier étant alors de mettre l'application en état de pause.

Du point de vue des différents processus mis en œuvre au cours de l'interactivité, les aspects temporels sont nombreux. Ils évoluent en accord avec le déroulement des scénarios que suivent ces processus (et donc avec le déroulement du processus interactif). Par exemple, l'ensemble des zones 2D et 3D du contexte « Observation », détaillées au cours de la section précédente, sont mises à jour au cours du temps en fonction de l'évolution du scénario que

suivent les processus de capture de la scène réelle et d'observation de la scène virtuelle, et donc, à plus haut niveau, de l'évolution du scénario de l'application.
Il est à remarquer que nous assurons une cohérence temporelle, d'une *frame* par rapport à la suivante, d'un grand nombre d'informations contextuelles, particulièrement ces zones 2D et 3D, qui sont regroupées dans le contexte « Observation ». Autrement dit, pour la plupart des informations contextuelles, prises en compte au cours d'une *frame* t par les processus de capture et d'observation de la scène, la notion de passé ne concerne que la *frame* précédente.

3.2. Gestuelle utilisateur

Nous présentons, au cours de cette section, **notre caractérisation de la gestuelle utilisateur**. Nous rappelons que nous définissons la gestuelle de l'utilisateur comme étant la logique dans laquelle s'inscrivent les **'mouvements'**, les **'gestes'**, les **'actions'** et les **'comportements'** que ce dernier peut adopter. La gestuelle utilisateur, caractérisé par le processus dédié de notre système, est une information contextuelle, comprise au sein du contexte « Utilisateur » (Section 3.1.1.).

Dans nos travaux, nous caractérisons la gestuelle utilisateur par le biais :

- **De règles quantitatives et qualitatives**, suivies par certaines parties du corps de l'utilisateur sur un intervalle de temps donné (Section 3.2.1.) ;
- **De relations spatio-temporelles** entre les différents éléments de la représentation virtuelle de l'utilisateur et ceux de la scène virtuelle (Section 3.2.2.) ;
- **D'événements interactifs**, générés par les éléments actifs de la scène (Section 3.2.3.).

La Section 3.2.4. se placera dans le contexte de notre étude et définira les mouvements, gestes, actions, comportements et gestuelles de l'utilisateur, en fonction des éléments de notre caractérisation.

3.2.1. Règles quantitatives et qualitatives

La gestuelle de l'utilisateur peut être caractérisée par le biais **de règles quantitatives et qualitatives**. Ces règles sont suivies par des parties du corps de l'utilisateur bien particulières et sont définies sur des intervalles de temps précis.

Tout d'abord, le modèle de l'utilisateur, utilisé par le processus de capture (contexte « Utilisateur ») nous permet de décrire la gestuelle d'un ou de plusieurs éléments particuliers de celui-ci, qu'ils soient une primitive ou une articulation. De manière à décrire plus précisément et plus efficacement la gestuelle utilisateur, nous définissons **3 niveaux de représentation** vis-à-vis de ce modèle. Il est donc possible de décrire la gestuelle utilisateur à différents niveaux de granularité.
Ces 3 niveaux de représentation (voir Figure 5.6.) sont :

- Le niveau **« Partie élémentaire »** (*Body Part Level*). C'est le niveau de description le plus bas. Une partie élémentaire du corps de l'utilisateur n'est pas décomposable et est indépendante. Il s'agit, par exemple, de l'élément « main droite » du modèle.

- Le niveau **« Membre »** (*Member Level*). Ce niveau définit les hiérarchies de parties élémentaires représentant les membres dédiés aux facultés de préhension et de locomotion, rattachés aux troncs inférieur et supérieur du modèle. Autrement dit, les « bras » et les « jambes ».
- Le niveau **« Corps »** (*Body Level*). C'est le niveau de description le plus haut. Ce niveau définit les hiérarchies de parties élémentaires et de membres, représentant le haut et le bas du corps.

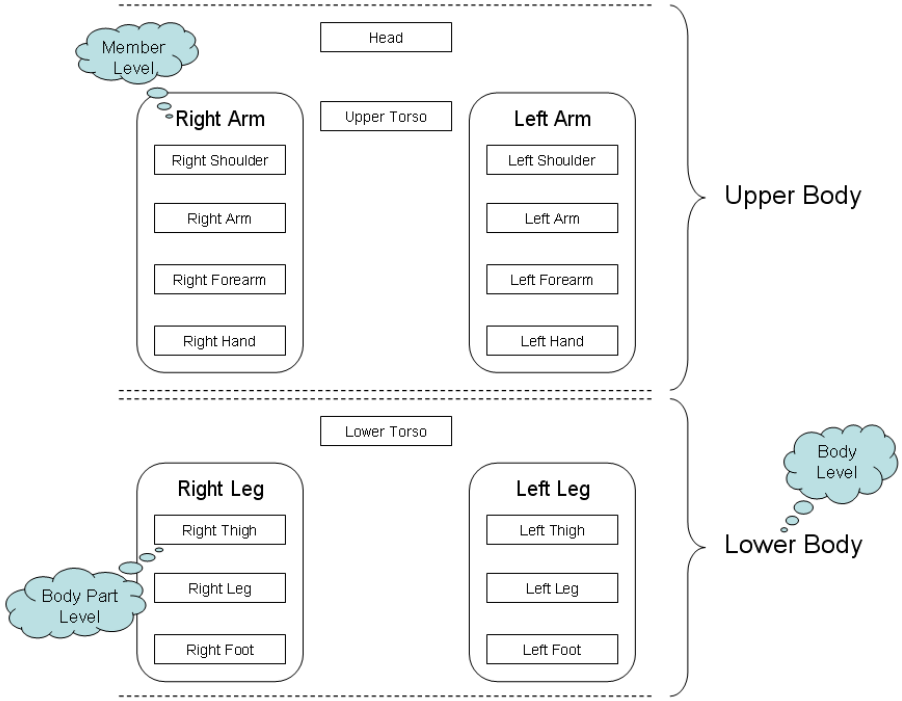


Figure 5.6. : Les 3 niveaux de représentation du modèle de l'utilisateur

La gestuelle utilisateur peut être décrite comme un ensemble de règles qualitatives et quantitatives. Ces règles sont alors définies en fonction d'un niveau de représentation du modèle, d'un intervalle de temps d'observation donné, etc. Elles sont donc calculées principalement à partir des informations issues de la capture de gestuelles et peuvent s'exprimer de manière absolue (par rapport au repère absolu de la scène, de manière générale par rapport à l'échelle de temps, etc.) ou relative (par rapport à un autre élément du modèle 3D représentant l'utilisateur, par rapport à une valeur temporelle mesurée antérieurement, etc.).
Les règles qualitatives permettent une description sémantique de la gestuelle utilisateur. Elles sont déterminées à partir d'une interprétation logique et déductive des données de capture. Ces règles peuvent, par exemple, apporter des connaissances sur :

- La localisation spatiale d'un élément du modèle représentant l'utilisateur : « le bras droit est devant l'utilisateur », « la main droite se trouve à une hauteur plus élevée que la main gauche », « l'utilisateur tend le bras gauche vers la gauche », etc. ;
- La vitesse de la gestuelle décrite par un élément du modèle, que ce soit par rapport à une gestuelle passée, à la vitesse de la gestuelle décrite par un autre élément : « le bras

gauche bouge plus vite que la jambe droite », « la gestuelle de la jambe droite est plus lente que précédemment », etc. ;
- La force exercée par un élément du modèle 3D, en comparaison avec celle exercée par un autre élément, ou avec la force exercée par le même le élément mais antérieurement : « l'utilisateur a frappé la balle plus fort que précédemment », « l'utilisateur a frappé la balle plus fortement la balle avec la main gauche, qu'avec la main droite » ;
- Et autres ! Nous voyons ici que la construction de ces règles dépend fortement du scénario de l'application. Elles sont en effet fortement dépendantes des objectifs que doit remplir l'utilisateur.

Les règles quantitatives permettent d'analyser mathématiquement et physiquement la gestuelle utilisateur. Elles sont construites à partir de l'observation et de l'interprétation des données de capture dans le temps. Ces règles quantitatives peuvent par exemple s'exprimer sous la forme :

- De courbes 2D (vis-à-vis d'une image) ou 3D (vis-à-vis de la scène virtuelle), exprimant l'évolution d'une position, d'une rotation, d'une vitesse (équations de mouvement), etc. Encore une fois, ces trajectoires peuvent s'exprimer de manière absolue ou de manière relative ;
- De modèles statistiques, modélisant et prédisant les différents états au cours du temps d'une variable donnée : ***Filtre de Kalman***, ***Modèle de Markov Caché*** (*Hidden Markov Model*, HMM), ***Réseau Bayésien Dynamique*** (*Dynamic Bayesian Network*, DBN), ***Champs Conditionnel Aléatoire*** (*Conditional Random Field*, CRF), etc. ;
- De volumes spatiotemporels exprimant l'évolution de données 2D au cours du temps (silhouettes, région, position particulière, etc.). Ces volumes peuvent, par exemple, d'étudier visuellement l'évolution d'une gestuelle au cours du temps ;
- Et bien d'autres encore. A l'instar des règles qualitatives, la construction et la détermination des règles quantitatives dépendent grandement des objectifs de l'utilisateur, des types de gestuelles à observer, des méthodes utilisées pour la reconnaissance des gestuelles, etc.

La caractérisation de la gestuelle utilisateur, à un niveau particulier de représentation, est rendu possible grâce aux informations contextuelles du contexte « Observation », notamment par le biais des zones 2D et 3D, confrontées au modèle 3D représentant l'utilisateur (contexte « Utilisateur »). De plus, cette caractérisation se fera en fonction du scénario de l'application et des objectifs qu'il définit pour une situation ou un contexte d'interaction donné (contexte « Système/Application »). Enfin, la caractérisation de la gestuelle utilisateur se fera sur un ou des intervalles de temps particuliers (contexte « Temps »), définis à partir de notre agenda scénarisé (Section 6.5.6. du Chapitre 2.).
Ces règles qualitatives et quantitatives sont mises à jour au cours du processus de capture de mouvements et de l'évolution du scénario.

3.2.2. Relations spatio-temporelles

Nous caractérisons également la gestuelle adoptée par l'utilisateur par le biais des **relations spatio-temporelles**. Ces relations expriment les connexions dans l'espace et dans le temps,

entre les différents éléments composant la représentation virtuelle de l'utilisateur et les différents éléments composant la scène virtuelle.

Ces connexions expriment les relations logiques entre la scène virtuelle et l'utilisateur. Elles peuvent, par exemple, exprimer la proximité d'un membre de l'utilisateur par rapport à un élément interactif de la scène.
Elles peuvent bien sûr être mises en relation avec la durée d'un intervalle de temps d'observation, un instant précis, etc. Par exemple, en accord avec le scénario de l'application et toujours dans le but de caractériser sa gestuelle, il peut s'avérer important de savoir combien de temps l'utilisateur est resté positionné à un endroit spécifique de la scène.
En conséquence, ces relations sont déterminées à partir d'opérations de raisonnement déductif, à partir de la description spatiale de la scène et de son évolution au cours du temps, de la description de la représentation virtuelle de l'utilisateur et des mesures au cours du temps, relatives au processus de capture de gestuelles.

La caractérisation de ces relations spatio-temporelles se fait en fonction des informations contextuelles comprises dans le contexte « Utilisateur » (éléments composant la représentation de l'utilisateur), le contexte « Système/Application » (objectifs de l'utilisateur, description de la scène virtuelle, description de l'avatar représentant l'utilisateur), le contexte « Observation » (zones 3D) et le contexte « Temps » (intervalles de temps d'observation).
Les relations spatio-temporelles évoluent en fonction de l'évolution du scénario de l'application et du processus de capture de gestuelles.

3.2.3. Evénements interactifs

Les événements interactifs, qui ont lieu **au sein de la scène virtuelle** et **générés par les éléments actifs** composant celle-ci, constituent le dernier élément de caractérisation de la gestuelle adoptée par l'utilisateur. Il peut par exemple s'agir des événements générés par un élément interactif de la scène, comme un bouton 3D ou une balle de tennis, manipulé par l'utilisateur.

La scène virtuelle, au sein de laquelle l'utilisateur évolue, est typiquement composée d'éléments 3D pouvant générer de manière autonome un certain nombre d'événements interactifs. Ces éléments actifs, et leurs comportements associés, sont gérés par **le moteur de l'application** (un moteur de rendu, un moteur de jeu, etc.) et sont conçus lors de la phase de développement. Au cours de l'exécution de l'application, une boucle permanente observe ces éléments et les différentes interactions qui peuvent prendre place autour de ceux-ci. En fonction de ces interactions, un comportement spécifique, propre à ces éléments, est déclenché. Ce sont ces comportements, et les événements qu'ils génèrent, qui nous permettent de caractériser la gestuelle utilisateur. Le moteur de l'application, au niveau de notre architecture de système, se situe au niveau des couches « support opérationnel » (module « Restitution de la scène 3D ») et « Scène virtuelle » (modules « Gestion de la scène virtuelle » et « Observation de la scène virtuelle »). Ce moteur est complété par nos développements, particulièrement au niveau de la gestion et de l'observation de la scène virtuelle, en accord avec nos objectifs.
Ces événements sont propres à la scène virtuelle mais ne sont pas forcément liés à l'utilisateur. Ils peuvent résulter d'une observation faite au sein de la scène réelle, non dépendante de la gestuelle adoptée par l'utilisateur.

La caractérisation des événements interactifs se fait en fonction des informations contextuelles comprises dans le contexte « Système/Application » (situations et contextes d'interaction, description de la scène virtuelle), le contexte « Observation » (ressources et événements 3D interactifs) et le contexte « Temps » (instants et intervalles de temps d'observation).
Les relations spatio-temporelles varient et sont mises à jour en fonction de l'évolution du scénario.

3.2.4. Mouvement, geste, action, comportement et gestuelle

Les sections précédentes définissent les éléments de caractérisation d'une gestuelle, et des différents éléments pouvant la composer. La difficulté réside bien sûr, pour un élément particulier d'une gestuelle, dans la nature des éléments de caractérisation et dans la complexité de leur association. Nous avons choisi de ne pas proposer de définition stricte, vis-à-vis de ces éléments de caractérisation, car celle-ci dépend fortement de la nature du scénario de l'application. En effet, un **mouvement**, un **geste**, une **action**, un **comportement** ou bien encore une **gestuelle**, pourra avoir une signification (et donc une précision et une complexité) différente d'un scénario à un autre. Au sein de l'équipe ***ImagIN***, des travaux sont actuellement en cours pour développer une ontologie, dans laquelle seraient précisés ces différents termes et les relations qui peuvent exister entre eux, en accord avec un scénario, ou une classe de scénarios donnée. Un tel travail sort du cadre que nous nous sommes fixés dans ces travaux de thèse.
Il est à noter que nous ne considérons, dans ces travaux, qu'une interaction de l'utilisateur par le biais de ses gestuelles. Nous ne considérons pas d'autres modalités, telles que sa voix par exemple. Si, dans des travaux futurs, tel était le cas, il faudrait redéfinir la notion de gestuelles, et de ses composantes.

Dans notre contexte d'étude, notre scénario est basique et s'accompagne de gestuelles simples de la part de l'utilisateur.
Nous définissons le **« mouvement »** comme étant l'ensemble des règles quantitatives et qualitatives que suivent les différentes parties du corps de l'utilisateur au cours de l'interactivité.
A partir du moment où les objectifs de l'utilisateur sont définis, tout mouvement cherchant à les atteindre devient un **« geste »**.
Dans nos travaux, nous définissons majoritairement des actions, dans la mesure où l'utilisateur agit sur l'environnement pour atteindre ses objectifs. Une **« action »** implique donc l'influence de l'utilisateur sur une ressource interactive (principalement les balles simulées) et sera alors caractérisée par les événements interactifs résultant de cette influence.
Un **« comportement »** impliquera alors plus la considération d'autres caractéristiques, propres au jeu en lui-même, telles que le niveau de difficulté de la situation d'interaction courante, le niveau de l'utilisateur, etc. et mettra en jeu des séquences organisées de mouvements, gestes et actions, relativement à une situation et/ou un contexte d'interaction. L'ensemble des caractéristiques décrites au cours des deux sections précédentes pourra être considéré, pour décrire le comportement de l'utilisateur.
De par la simplicité de notre scénario, impliquant des phases de jeu courtes au cours desquelles les gestuelles adoptées sont relativement basiques, notre définition de la **« gestuelle »** est fortement similaire à celle du comportement. Nous parlerons de gestuelle pour décrire la logique complète que suivent les mouvements, gestes, actions et

comportements pour un répondre à une situation d'interaction ou un contexte d'interaction particulier.

4. Interprétation de la scène virtuelle observée

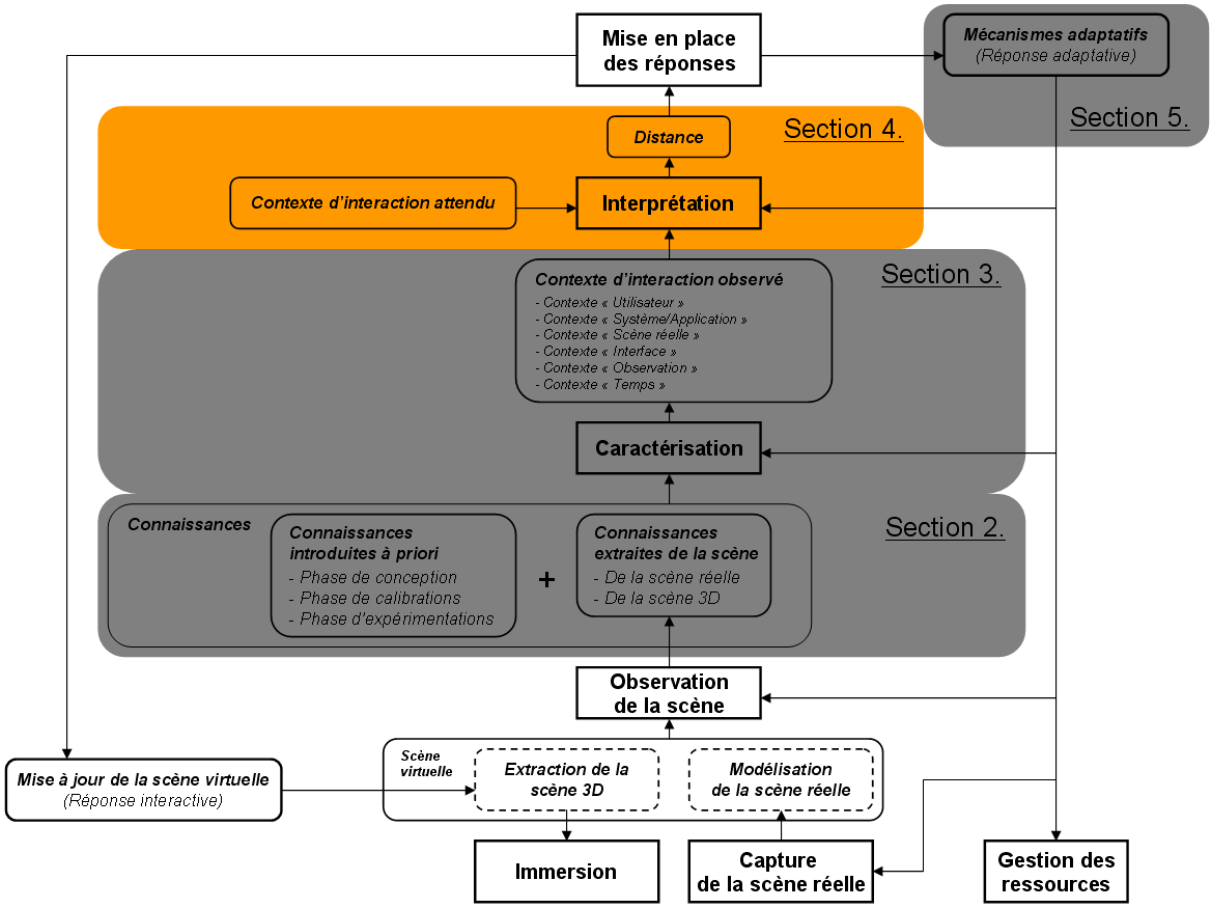


Figure 5.7. : Interprétation de la scène observée

Antérieurement à **l'interprétation de la scène virtuelle** observée (voir Figure 5.7.), l'activité et les différents événements en résultant sont caractérisés et conjugués, dans le but de construire le contexte d'interaction au sein duquel l'utilisateur évolue. Ce contexte d'interaction est en fait un ensemble de sous-contextes, dédiés à différents aspects du système et des processus le composant, de son environnement et de l'interactivité en elle-même. Les sections précédentes énumèrent les différentes connaissances, introduites *a priori* et/ou construites au cours de l'interactivité, à partir desquelles un contexte d'interaction peut être caractérisé, caractérisation que nous avons également définie.

Notre caractérisation du contexte d'interaction englobe notamment les caractérisations des gestuelles utilisateur et des différents événements, générés, ou non, par l'activité de l'utilisateur. De plus, un contexte d'interaction comprend un sous-contexte lié à l'application (contexte « Système/Application », Section 3.1.2.). Ce sous-contexte comprend plusieurs informations contextuelles, construites à partir du scénario de l'application. Parmi ces

informations contextuelles se trouvent **le contexte d'interaction attendu**, en accord avec les objectifs que doit atteindre l'utilisateur pour une situation d'interaction donnée.
Un contexte d'interaction attendu, et les informations contextuelles qu'il comprend (particulièrement la gestuelle utilisateur et les événements ayant lieu au sein de la scène), est caractérisé comme exposé au cours de la Section 3. C'est au sein de ce contexte d'interaction que l'utilisateur est sensé évoluer, pour atteindre les objectifs définis par une étape du scénario donnée. Un contexte d'interaction attendu traduit donc ce qui est supposé être observé par le système (processus d'observation de la scène virtuelle) à un instant donné. Autrement dit, par le biais d'un contexte d'interaction attendu, le système sait ce qu'il doit observer, dans quelle partie de la scène virtuelle il doit les observer, et à quel moment. Ce contexte décrit un ensemble de suppositions fiables, significatives quant à l'interprétation de la scène virtuelle, concernant ce qui doit être observé par le système.

Au cours d'une situation d'interaction particulière, le processus de caractérisation modélise le contexte d'interaction à partir de ses différentes observations et des connaissances qu'il possède. Par la suite, le processus d'interprétation de la scène virtuelle extrait de ce **contexte d'interaction observé**, le contexte d'interaction attendu, défini lors de la phase de conception du système et extrait du scénario.
Une fois les contextes d'interaction observés et attendus considérés, et avant de pouvoir s'adapter, le processus d'interprétation se charge de reconnaitre l'activité au sein de la scène observée. L'interprétation de la scène virtuelle se traduit donc par une compréhension et une reconnaissance du contexte d'interaction observé caractérisé. Le processus d'interprétation de la scène effectue alors **la comparaison** entre le contexte d'interaction observé et celui attendu par le système. Par le biais du contexte d'interaction attendu, le scénario de l'application supporte ce processus.

Le processus d'interprétation calcule donc la différence, pouvant exister entre le contexte d'interaction observé et celui attendu par le scénario de l'application. Cette différence, ou **distance**, traduit l'écart entre les objectifs attendus par le scénario et ceux réellement atteints par l'utilisateur. Cette distance est en fait un ensemble de distances, les caractéristiques comparées étant nombreuses et hétérogènes.
D'une part, la distance représente **le niveau de compréhension de l'utilisateur quant aux objectifs qu'il doit atteindre** et à la situation d'interaction courante au sein de laquelle il évolue. En effet, plus cette distance sera grande, plus elle traduira le fait que l'utilisateur n'a pas compris non seulement ses objectifs, définis par le scénario de l'application, mais également comment il doit interagir avec le système.
D'autre part, la distance calculée traduit **le niveau de compréhension du processus d'interprétation, vis-à-vis de la scène virtuelle observée**. Elle indique si l'activité au sein de la scène observée est comprise par le processus, si elle présente des ambiguïtés de sens, ne permettant pas toujours d'aboutir à une interprétation correcte de la part du processus, ou encore si elle n'est pas reconnue par le système.
Enfin, cette distance **pilote l'évolution du scénario de l'application** et est donc responsable **de l'adéquation et de la qualité des réponses du système**, mises en jeu au cours de l'interactivité. Au cours des différentes étapes du scénario, la distance est soumise à plusieurs conditions. Ces conditions d'acceptation sont extraites du scénario de l'application (comprises donc en tant qu'information contextuelle au sein du contexte « Système/Application »), chaque condition étant spécifique à un élément particulier composant la distance. En fonction du scénario, seuls certains éléments de la distance peuvent être décisifs quant à l'évolution du scénario. C'est pourquoi, si les différents éléments composant la distance ne peuvent pas tous être calculés, ou si certains de ces éléments ne remplissent pas les conditions d'acceptation

associées, des compromis peuvent tout de même être faits pour assurer la bonne évolution du scénario.

A la suite de l'interprétation de la scène, le processus dédié du système peut mettre en place ses **réponses interactives et adaptatives**. Ces réponses sont en adéquation avec le contexte d'interaction observé, et donc de la gestuelle qu'a adopté l'utilisateur au cours de la situation d'interaction en cours.

5. Adaptativité

Après avoir interprété la scène virtuelle qu'il observe et l'activité au sein de celle-ci, le système, au niveau de la logique concepteur, met en place **deux réponses** de natures différentes.
La première réponse, que nous qualifions d'**interactive**, se matérialise visuellement via l'immersion, transmettant ainsi à l'utilisateur la réponse du système vis-à-vis de la gestuelle qu'il a adoptée et de l'activité générale au sein de la scène virtuelle. Par le biais de cette réponse, l'environnement 3D, restitué à l'utilisateur, est mis à jour au cours de l'interactivité.
La deuxième réponse caractérise **l'adaptativité** du système (« réponse adaptative ») et se concrétise sur l'ensemble du fonctionnement du système, à tous les niveaux de son architecture, dans le but permanent d'améliorer l'interactivité avec l'utilisateur. Ces différents mécanismes adaptatifs, dans le cadre de nos travaux, sont prévus lors de la phase de conception du système.
Cette section traite de l'adaptativité de notre système et de la mise en place des mécanismes adaptatifs au cours de l'interactivité avec l'utilisateur (Voir Figure 5.8.).

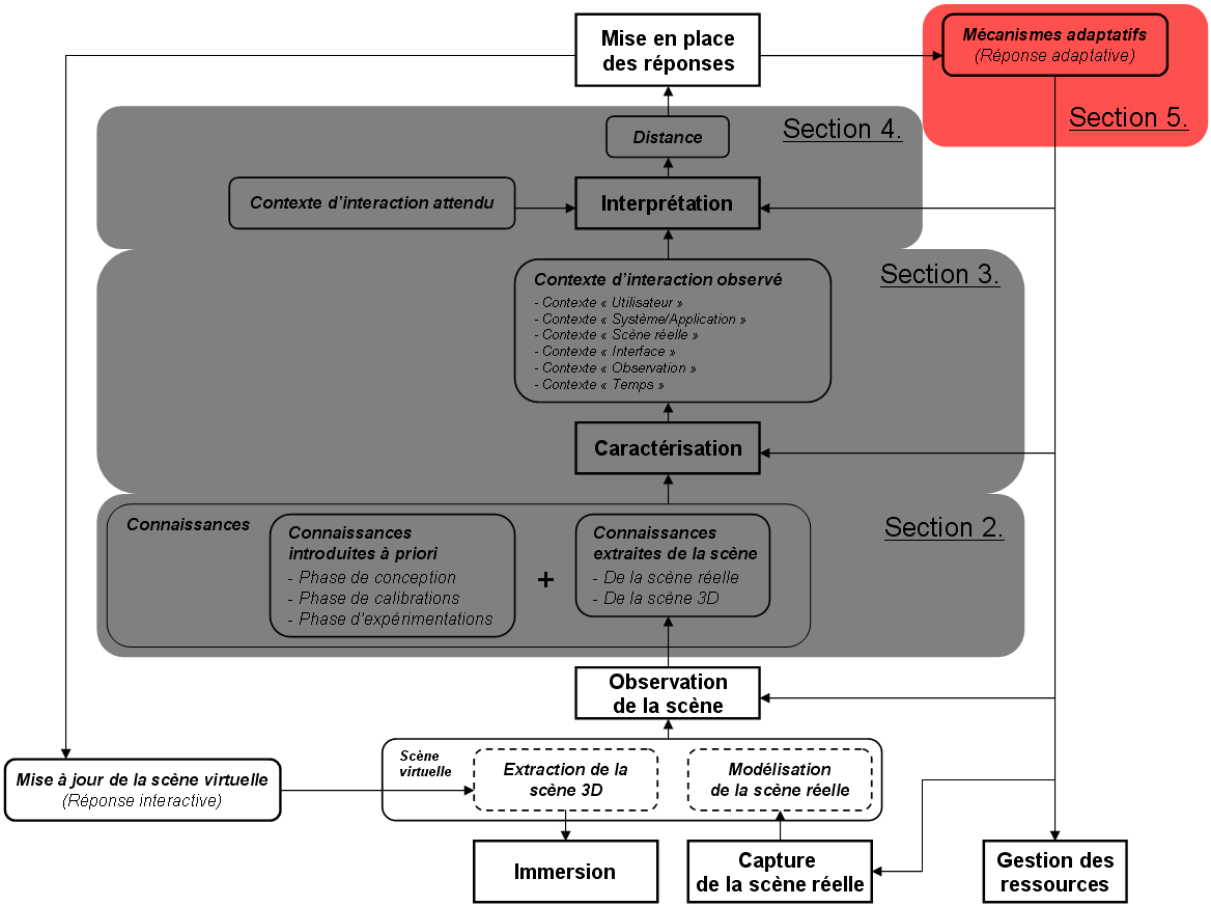


Figure 5.8. : Adaptativité

La Section 5.1. décrit **le processus de pilotage de ces mécanismes adaptatifs**, par le biais de l'évolution du scénario de l'application. Ce pilotage permet l'exécution d'une réponse adéquate vis-à-vis de l'interprétation du système de la scène virtuelle, et particulièrement de la gestuelle utilisateur caractérisée.
La Section 5.2. présente à la fois l'influence **des informations contextuelles** sur **les mécanismes adaptatifs**, les paramétrant en accord avec l'interprétation de la scène virtuelle, et l'adaptation de ces dernières, illustrant un mécanisme adaptatif subjacent, mis en place par le système.
Les différents mécanismes adaptatifs, mis en place au sein de notre système, seront détaillés dans la Section 5.3. Ces différents mécanismes adaptatifs agissent sur la capture de la scène réelle, l'observation de la scène virtuelle, la caractérisation et l'interprétation de la scène virtuelle, la réponse visuelle restituée à l'utilisateur et la gestion des différentes ressources matérielles et logicielles.
L'accumulation des effets de l'adaptativité au cours de l'interactivité met en avant la présence de boucles logicielles au sein de notre système, que nous appelons **boucles vertueuses**. La Section 5.4. définit et détaille ces boucles.

5.1. Pilotage par le biais du scénario

Dans nos travaux, l'adaptativité de notre système est supportée par l**e scénario de l'application**. En effet, les mécanismes adaptatifs sont mis en place au cours de la phase de conception du système et sont pilotés par l'évolution du scénario, au cours de l'interactivité. Leur activation, toujours planifiée par le scénario, est fonction du processus d'interprétation et dépend donc de la distance pouvant exister entre le contexte d'interaction modélisé à partir de l'observation de la scène virtuelle et celui attendu au cours d'une étape du scénario donnée.

Par conséquent, les réactions mises en place par le système sont cohérentes et adéquates vis-à-vis de la gestuelle adoptée par l'utilisateur, et de l'activité au sein de la scène de manière générale. En un sens, l'utilisateur n'est plus restreint par le scénario de l'application. Mais, dans le but d'améliorer les interactions avec l'utilisateur, le système adapte, en permanence, l'exécution de ses différents processus à sa gestuelle observée et interprétée, dans les limites du cadre scénaristique.

5.2. Paramétrage par les informations contextuelles

Le système interactif s'adapte à la situation d'interaction courante, dont les différents aspects pertinents permettant d'améliorer l'interactivité, ont été mis en relief par les informations contextuelles. En effet, **l'adaptativité de notre système est paramétrée par l'ensemble des informations contextuelles**, fournies par le contexte d'interaction modélisé à partir de l'observation de la scène virtuelle. Ces informations sont présentes à tous les niveaux de l'architecture du système.

Au cours de l'interactivité, les informations contextuelles sont modifiées de manière à être cohérentes et significatives vis-à-vis de l'activité observée au sein de la scène virtuelle. Cet ajustement des informations contextuelles traduit la capacité adaptative du système.
L'ajustement des informations contextuelles, qui s'effectue en accord avec le scénario de l'application (et donc des scénarios que suivent les différents processus au sein du système), constitue un mécanisme adaptatif à part entière. En effet, les informations contextuelles paramètrent les autres mécanismes adaptatifs mis en place par le système au cours de sa réponse à l'utilisateur. Meilleur est cet ajustement, meilleure sera l'adaptativité du système. Cet ajustement s'effectue de manières parallèle et synchronisée avec la plupart des mécanismes adaptatifs présentés au cours des prochaines sections.

Ainsi, les informations contextuelles paramètrent l'adaptativité du système, mais sont également adaptées au cours de l'interactivité. Autrement dit, d'une part, les informations contextuelles permettent de mieux expliquer une situation d'interaction particulière, le système pouvant s'adapter alors efficacement à cette dernière. D'autre part, le système s'adapte à une situation d'interaction donnée, en ajustant les informations contextuelles pertinentes, de manière à s'adapter de mieux en mieux au fur et à mesure de l'interactivité.

Nous avons énuméré les différentes informations contextuelles, que nous utilisons dans le cadre de nos travaux, au cours de la Section 3.

5.3. Mécanismes adaptatifs au sein de notre système

Nous présentons, au cours de cette section, les différents mécanismes mis en place au sein de notre système, relativement à, d'une part, l'architecture de notre système et, d'autre part, notre contexte d'étude.

Les différents mécanismes adaptatifs de notre système sont dirigés sur :

- **la capture de la scène réelle**, la modélisation de la scène 3D n'étant pas adaptative, dans le cadre de nos travaux (Section 5.3.1.) ;
- **l'observation de la scène virtuelle** (Section 5.3.2.) ;
- **la caractérisation et l'interprétation de la scène virtuelle** (Section 5.3.3.) ;
- **la réponse visuelle restituée à l'utilisateur** (Section 5.3.4.) ;
- **la gestion des différentes ressources matérielles et logicielles** (Section 5.3.5.).

5.3.1. Capture de la scène réelle

Par **« capture de la scène réelle »**, nous rappelons que nous considérons bien sûr la capture de la gestuelle utilisateur, mais également la capture de tout autre événement non généré par l'utilisateur (par des spectateurs, par exemple). Cette capture inclut **l'acquisition vidéo de la scène réelle** ; le processus de **capture de gestuelles**, présenté au cours du chapitre précédent ; et le processus de **capture de tout autre événement observable au sein de la scène réelle**, dont les gestuelles de l'utilisateur ne sont pas directement responsables.

La capture de la scène réelle est supportée par les informations contextuelles comprises au sein des contextes « Utilisateur » (modèle 3D utilisé par le processus de capture), « Système/Application » (scénario qui pilote l'évolution des informations contextuelles), « Scène réelle » (configuration spatiale de la scène réelle, emplacements de l'utilisateur et des spectateurs éventuels), « Interface » (matrices caméra, caméra en face de l'utilisateur), « Observation » (l'ensemble des zones 2D) et « Temps » (cohérence temporelle des données vis-à-vis de la *frame* précédente, mesure du mouvement au sein de la scène réelle en fonction de l'état de cette dernière au cours des *frames* précédentes).
Les zones 2D, comprises au sein du contexte « Observation » sont particulièrement utilisées pour adapter la capture de la scène réelle. Par le biais de ces zones (**gestion multiple-focus**), le processus de capture est capable de rediriger et focaliser ses traitements dans des zones de l'image bien particulières. Ce processus assure cependant, dans le même temps, une vision plus globale de la scène réelle, assurant ainsi un parallélisme permanent de vision. Ce **parallélisme de vision** est géré au niveau du scénario, et est adapté pertinemment en accord avec la situation d'interaction courante observée.

Tout d'abord, les processus de segmentation des silhouettes et de squelettisation de ces dernières ne sont effectués qu'au sein de la zone SASM, englobant de manière certaine l'utilisateur. Parallèlement, le fond de la scène réelle n'est mis à jour qu'à l'extérieur de cette zone. Ainsi, tout mouvement au sein de la scène réelle, dont l'utilisateur n'est pas responsable, n'influe jamais sur la segmentation et la squelettisation des silhouettes de l'utilisateur. De plus, l'utilisateur n'est jamais mis à jour avec le fond de la scène réelle. Même si l'utilisateur reste immobile, il n'est jamais modélisé avec le fond et reste détecté et suivi. De la même manière, le processus de capture de gestuelles n'est exécuté, au minimum, qu'au sein de la zone SASM. Ainsi, la zone SASM assure une robustesse des différents processus, vis-à-vis de tout ce qui peut se passer à l'extérieur de celle-ci et qui n'est pas dépendant des gestuelles de l'utilisateur.

Deuxièmement, le processus de capture de mouvements, en fonction du scénario de l'application, peut n'être exécuté que sur certaines parties spécifiques du corps de l'utilisateur, mises en valeur par le biais de zones 2D. Ces zones peuvent être celles autour d'une partie du corps de l'utilisateur mise en relief par le scénario de l'application (zones SABE) ou en mouvement au cours de la situation d'interaction courante (zone SAMP couplée avec la zone SASM).
La Figure 5.9. donne des exemples de parallélisme de vision de la scène réelle. L'utilisateur est englobé au sein de la zone SASM (en bleu) où le fond n'est pas mis à jour et où les processus de segmentation, squelettisation et capture de gestuelles prennent place. De plus, la zone SAMP (en rouge) permet la détection du mouvement de certains membres de l'utilisateur, permettant ainsi la focalisation des observations sur ceux-ci.
La Figure 5.10. illustre le processus de création de forces 2D, uniquement au sein de zones 2D particulières (SAMP ici). Il est à noter que le processus de capture peut être également exécuté de manière pondérée (forces dont l'intensité est plus importante) au sein de ces zones.

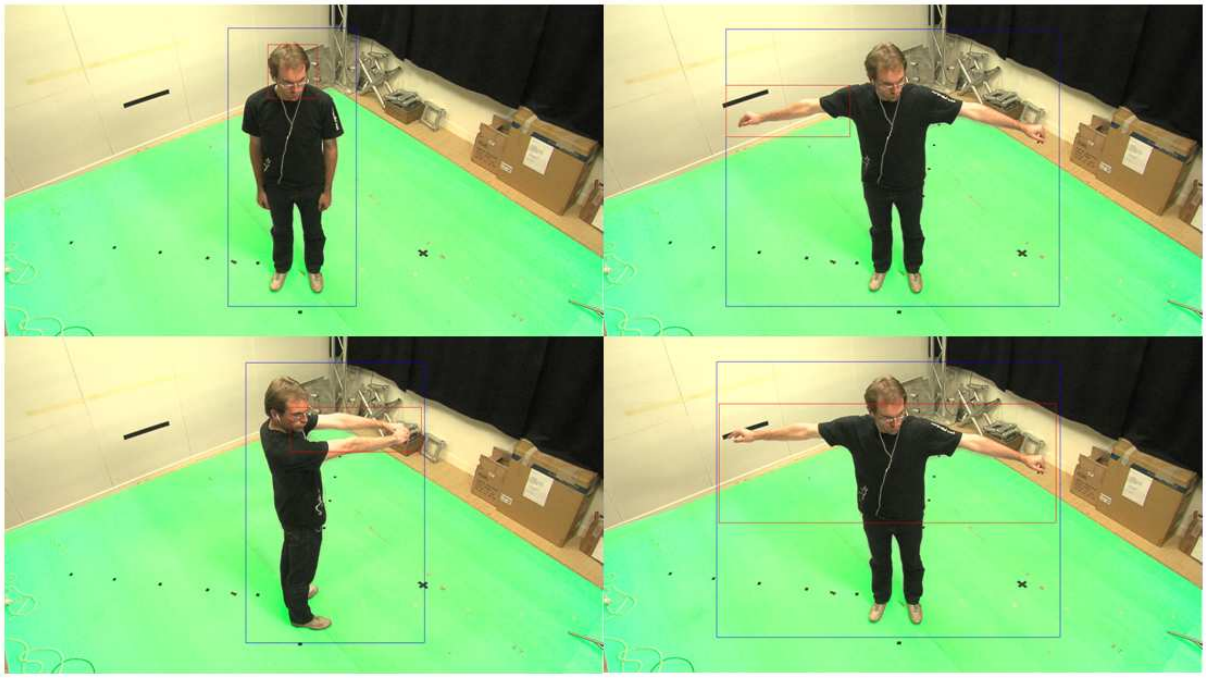


Figure 5.9. : Exemples de parallélisme de vision au niveau de la capture de la scène réelle

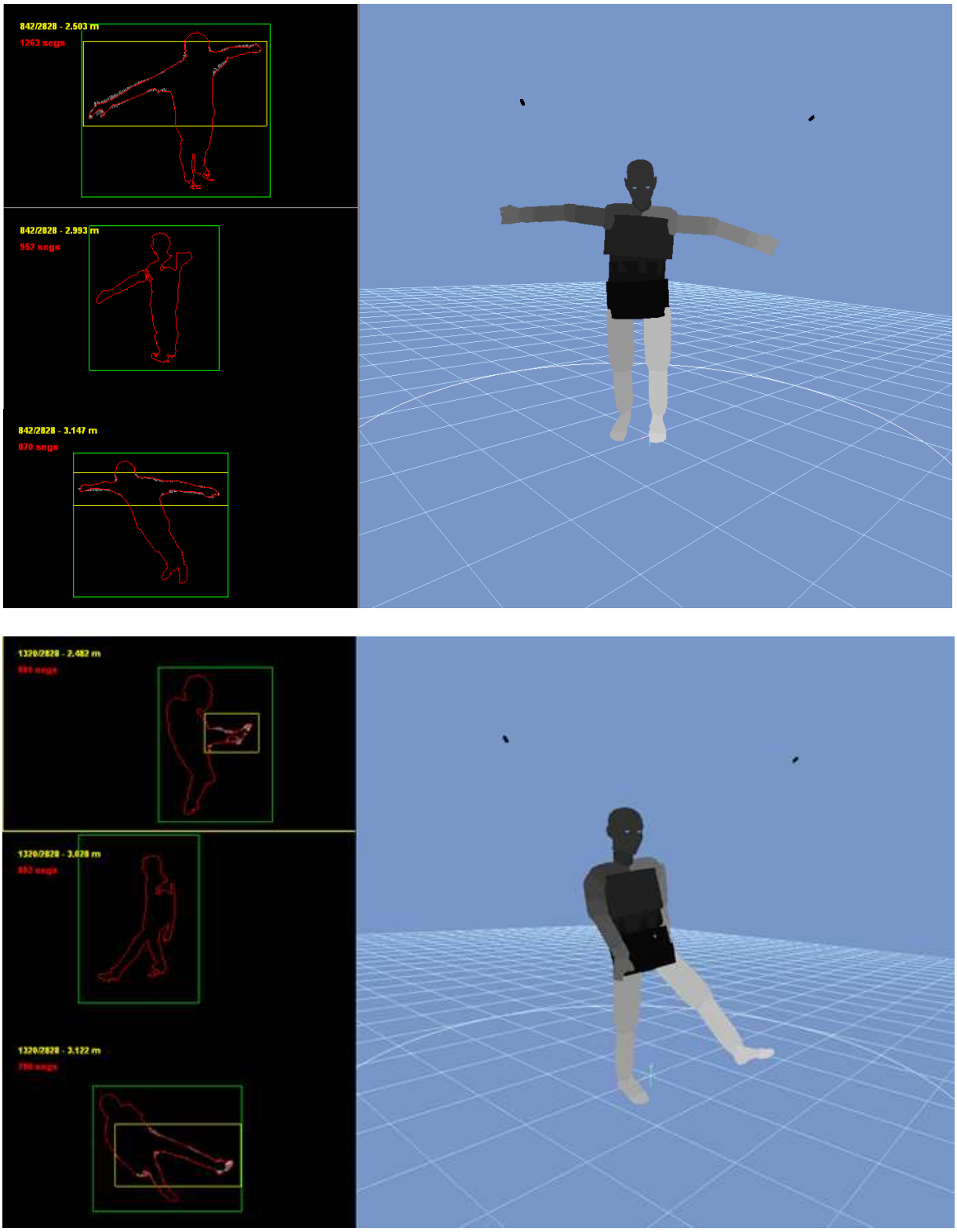


Figure 5.10. : Processus de capture de gestuelles exécuté uniquement au sein de zones 2D (SAMP)

Dans ces travaux de thèse, il nous a fallu trouver des solutions pour désambiguïser les silhouettes de l'utilisateur et pallier la suppression des marqueurs de couleur. En effet, selon les poses de l'utilisateur, une silhouette peut ne pas donner assez d'informations pour repérer

toutes les parties du corps de celui-ci. Par exemple, si l'utilisateur a les bras le long du corps et les jambes jointes, sa silhouette ne permettra pas de dire s'il regarde vers les caméras ou dans la direction opposé, de déterminer où se trouve son bras droit et son bras gauche, etc. De plus, le squelette généré par ce type de silhouette sera un simple trait 2D vertical. Les marqueurs de couleur, utilisés initialement pour la capture des gestuelles de l'utilisateur, permettait de résoudre ces situations.
Il est possible, antérieurement et/ou au cours de l'interactivité, de désigner une caméra comme étant celle la plus en face de l'utilisateur (information contextuelle comprise dans le contexte « Interface »). Pour ce point de vue donné, en couplant la zone SASM et la zone SABE liée à la tête de l'utilisateur, le système peut à tout moment exécuter un processus de détection de visage. Ce processus peut permettre de désambiguïser une silhouette, relativement aux côtés droit et gauche de celle-ci, dans la mesure où un visage détecté signifie que l'utilisateur regarde vers les caméras. Ainsi, si les bras et les jambes de l'utilisateur sont visuellement observables (i.e. distincts vis-à-vis de la silhouette), il est possible d'orienter la capture de gestuelles en positionnant directement les bras et jambes gauches et droits de l'avatar. Dans le contexte de notre application, l'utilisateur regarde vers les caméras (l'immersion étant au niveau de ces dernières) et ne croise que très rarement ses bras et ses jambes.

Le contexte « Scène réelle » nous permet d'émettre de fortes hypothèses sur les emplacements au sein de la scène réelle d'éventuels spectateurs. Ces emplacements, une fois projetés sur un point de vue donné, génèrent un ensemble de zones 2D, les zones SAO, combinables avec les autres zones 2D comprises au sein du contexte « Observation ».
Théoriquement, l'algorithme de mise à jour du fond doit pouvoir gérer rapidement le mouvement de spectateurs au sein de la scène, mettant ainsi à jour les spectateurs dans le fond. Cela dit, il peut s'avérer insuffisant face à des mouvements continus de leur part. La détection de mouvement, dont l'utilisateur n'est pas la source, peut parasiter le processus de capture de gestuelles de ce dernier.
La zone SAMP permet d'englober le mouvement dans la scène globale et pour une *frame* donnée. En combinant cette zone avec la zone SASM (englobant l'utilisateur), il est possible d'isoler les zones 2D, au sein desquelles du mouvement est observable mais dont l'utilisateur n'est pas la source. Ainsi, ces zones peuvent être exclues rapidement de tous types de traitements (remplacement immédiate de ces zones dans le fond, capture de gestuelles non influencée par ces zones, etc.). Cette approche est validée tant que l'emplacement de l'utilisateur (informations contextuelles comprises dans le contexte « Scène réelle ») n'est pas concurrent avec ceux des spectateurs.

Enfin, les différentes zones concernant l'utilisateur (SASM, SASM combinée avec SAMP, SABE) pourront directement influer sur l'acquisition des flux vidéo. Il est en effet possible de n'acquérir que certaines parties de l'image par le biais de ces zones. Cette acquisition réduite n'est possible que si l'interprétation de la scène réelle ne présente pas d'ambigüités quant aux attentes du scénario, et si les zones 2D liées à l'utilisateur ne sont pas concurrentes avec celles liées aux autres événements pouvant survenir au sein de la scène réelle.

5.3.2. Observation de la scène virtuelle

L'observation de la scène virtuelle englobe aussi bien l'observation de la scène réelle capturée que l'exploitation de la scène 3D. C'est à partir de ces observations que les gestuelles utilisateur seront caractérisées.

Ce sont principalement les connaissances apportées par le contexte « Observation », qui permettent l'adaptation du processus d'observation de la scène virtuelle. En effet, elles permettent de focaliser les processus observateurs au sein de celles-ci (zones 2D, zones 3D, ressources interactives), assurant ainsi de multiples observations simultanées (gestion multiple-focus et vision globale de la scène). Le processus d'observation assure donc un parallélisme permanent de vision, lui permettant de repasser à une observation globale de la scène virtuelle à tout moment. Les différentes informations contextuelles du contexte « Observation », ainsi que le parallélisme de vision mis en œuvre par le processus d'observation, sont pilotés par le scénario et son évolution. Ainsi, le processus d'observation adapte ses traitements en accord avec l'évolution du scénario et les interactions passées.

Les différentes zones 2D sont définies et mises à jour par l'évolution du scénario (les zones SABE par exemple) et par des associations simples entre zones (la zone SASM englobant l'utilisateur calculée à partir de l'union de la SAS et de la SAM, par exemple).
Parmi les différentes zones 2D, la zone SAMP est la seule zone d'observation dont les dimensions s'ajustent de manière adaptative au cours de l'interactivité. De manière générale, cette zone, calculée à partir de l'image différence entre la *frame* courante et la *frame* précédente, englobe précisément les pixels dont l'intensité a varié pendant les deux *frames* consécutives.
Si le mouvement observé est ample et rapide, les dimensions de la zone varient de manière significative entre les deux *frames* consécutives. Ainsi, un algorithme, utilisant cette zone indépendamment de sa mise à jour, pourra potentiellement prendre en compte la zone à l'instant t-1, dont les dimensions ne sont plus du tout adéquates à l'instant t.
Pour pallier ce problème, une marge de sécurité, en pixels, est ajoutée systématiquement aux dimensions de la zone SAMP. Cette marge, comprise entre une valeur minimale et une valeur maximale paramétrables, est mise à jour de manière adaptative. Si les dimensions de la zone SAMP sont 'trop' différentes (au delà d'un seuil donné) d'une *frame* à une autre, la marge de sécurité est augmentée significativement (passage à une valeur maximale). Ainsi, la zone SAMP englobe de manière certaine le mouvement au sein de la scène.
Pour finir, si les dimensions de la zone SAMP varient peu entre deux frames et si la marge de sécurité n'a pas atteint sa valeur minimale, la valeur de cette dernière est décrémentée progressivement.

Au niveau de l'observation 3D, le système gère, par l'intermédiaire du scénario de l'application, les zones 3D et les écoutes des événements générés par les différents éléments interactifs composant la scène virtuelle.
Dans le cadre de nos travaux, ces zones 3D et ces écoutes sont gérées en accord avec le scénario de l'application : du point de vue du scénario, seules les zones mises en relief et seules les écoutes pertinentes sont actives au cours d'une situation d'interaction donnée. Les autres zones et écoutes ne sont alors pas prises en compte par le système.
Par exemple, si l'utilisateur doit intercepter une balle 3D particulière, l'ensemble des autres balles, déjà présentes au sein de la scène virtuelle, sont désactivées (logiciellement, physiquement, interactivement), restreignant ainsi les possibilités d'interactions avec l'utilisateur.

5.3.3. Caractérisation et de l'interprétation de la scène observée

L'adaptativité du système se traduit également au niveau de **la caractérisation et de l'interprétation de la scène observée**. Les mécanismes adaptatifs mis en place à ce niveau

sont pilotés par les contextes d'interaction attendus, extraits du scénario de l'application au cours des différents situations et contextes d'interaction.

Que cela soit de manière globale ou par le biais d'informations contextuelles, il est possible d'extraire de la scène virtuelle un très grand nombre d'observations que le système doit caractériser et interpréter. Il est évident que plus les observations sont ciblées et peu nombreuses, plus le système sera amène à caractériser et à interpréter rapidement et efficacement celles-ci.
Grâce aux connaissances apportées par le scénario de l'application, il est possible d'attendre à un instant donné des observations spécifiques. Ces observations sont précisément identifiées lors de la phase du développement de l'application et sont limitées en nombre. Pour une situation ou un contexte d'interaction donné, un ensemble d'observations correspond à un ensemble de caractérisations et d'interprétations, qui peuvent être extraites du scénario de l'application. Ainsi, le processus dédié ne caractérise et n'interprète que ce qui est attendu par le scénario, optimisant ainsi son propre fonctionnement. Par exemple, si un mouvement du bras droit est attendu de la part de l'utilisateur, seules les poses des éléments correspondant aux bras droit du modèle représentant l'utilisateur, seront considérées par le processus de caractérisation et d'interprétation de la gestuelle utilisateur.

5.3.4. Réponse visuelle restituée à l'utilisateur

L'adaptation de la réponse visuelle restituée à l'utilisateur permet la mise en évidence des aspects pertinents de l'interactivité à un instant donné. Cette mise en évidence est dédiée à l'utilisateur, de manière à ce que ce dernier adopte la gestuelle adéquate en accord avec le scénario de l'application. Une réponse adaptée de la part du système permettra donc, d'une part, de conforter l'utilisateur dans sa gestuelle, ses objectifs ayant été confirmés par le système, et, d'autre part, de replacer l'utilisateur, dont la gestuelle indique qu'il n'a pas compris les objectifs, dans le contexte d'interaction attendu par le processus d'interprétation.
Ainsi, cette forme d'adaptativité est un apport important dans nos travaux, car elle contribue majoritairement à améliorer l'interactivité avec l'utilisateur. Elle est supportée principalement par le contexte d'interaction attendu à un instant donné, informations contextuelles extraites du contexte « Système/Application ».

La première forme d'adaptation consiste à évaluer la difficulté ou la facilité de l'utilisateur à interagir avec le système, en adéquation avec le scénario. Dans le cadre de nos travaux, l'utilisateur doit intercepter avec ses mains, des balles de tennis lancées par un canon en face de celui-ci. En fonction de sa capacité à intercepter les balles, le système détermine le niveau de l'utilisateur. En fonction de ce niveau, plusieurs mécanismes adaptatifs (tels que ceux que nous décrivons par la suite) pourront être mis en place, en fonction du scénario de l'application.

La deuxième forme d'adaptation du système se matérialise sous la forme d'affichages supplémentaires permettant d'aider l'utilisateur à adopter la bonne gestuelle et à mieux comprendre le contexte d'interaction au sein duquel il évolue. Ces mécanismes adaptatifs découlent généralement de l'évaluation du niveau de l'utilisateur (donc de sa capacité à intercepter les balles).
Ainsi, dans notre contexte d'étude et suivant le niveau de l'utilisateur, les trajectoires des balles seront affichées antérieurement ou au début du lancer de balles, permettant à l'utilisateur de se préparer correctement à intercepter la balle.

De plus, les balles et les mains de l'avatar représentant l'utilisateur sont englobées au sein de boites 3D qui déterminent l'espace physique réel occupé par l'élément considéré. Ces boites permettent la détection et la simulation des collisions entre les différents éléments physiques modélisés au sein de la scène virtuelle. Suivant la capacité de l'utilisateur à intercepter les balles, ces boites, englobant aussi les mains de l'utilisateur que les balles, peuvent être affichées. Ainsi, l'utilisateur peut adopter la gestuelle adéquate, permettant ainsi la collision visible entre la boite englobant une balle donnée et celle englobant une de ses mains.

Troisièmement, le système peut permettre à l'utilisateur d'adopter la gestuelle adéquate en ajustant la difficulté du jeu. Cet ajustement peut résulter en une modification de la réponse visuelle restituée à l'utilisateur, mais peut également être transparente pour ce dernier. Dans tous les cas, et dans notre contexte d'étude, l'adaptativité du système se traduit par l'adaptation des éléments interactifs au sein de la scène.
Ainsi, en fonction de la capacité de l'utilisateur à intercepter les balles, les dimensions des boites physiques englobant les balles et les éléments composant l'avatar (particulièrement les mains et les bras) peuvent varier, de manière à accroitre ou à diminuer la difficulté du jeu. Dans le même but, les ajustements peuvent également se concrétiser au niveau des propriétés du lancer de balles : paramètres du lancer (angle de lancer, force du lancer, etc.), ralentissement de la balle au cours du lancer, etc.

Enfin, la gestuelle de l'utilisateur pourra influencer directement l'évolution du scénario et donc la réponse visuelle immersive, restituée à ce dernier.
Dans nos travaux, le jeu se mettra en état de pause, si l'utilisateur ne bouge pas (pas de mouvement, de la part de l'utilisateur, détecté au sein de la scène réelle) alors que plusieurs balles ont été lancées (évaluation de l'intérêt de l'utilisateur pour le jeu).

5.3.5. Gestion des ressources matérielles et logicielles

L'optimisation de la gestion des ressources, **matérielles et logicielles**, s'effectue parallèlement et manière synchronisée à la plupart des mécanismes adaptatifs présentés précédemment. Elle est principalement supportée par l'utilisation des connaissances comprises au sein du contexte « Observation » : zones 2D, zones 3D et écoutes d'événements au sein de la scène virtuelle. En effet, ces informations contextuelles permettent aux traitements de ne pas obligatoirement considérer la scène dans sa globalité mais seulement parties de celles-ci. Les traitements peuvent donc être redirigés et focalisés par l'intermédiaire de ces informations contextuelles, limitant les calculs sur de faibles portions de la scène et permettant au système d'utiliser ses ressources pour d'autres traitements. De plus, grâce à cette optimisation, la contrainte temps réel est assurée au cours de l'interactivité.

Comme nous l'avons mentionné au cours de la Section 3.1.5., le contexte « Observation » permet de mettre en place la gestion multiple-focus de la scène virtuelle au cœur du processus adaptatif du système. Cette gestion permet un parallélisme de vision important, dans la mesure où les processus utilisant ce parallélisme ont à la fois une vision globale de la scène virtuelle et une vision plus focalisée de cette dernière (en accord avec le scénario). Grâce à ce parallélisme, ces processus peuvent ne traiter que les différentes parties de la scène virtuelle qui s'avéreraient pertinentes, vis-à-vis du scénario de l'application. La gestion de la scène virtuelle est optimisée, optimisant du même coup la gestion des différentes ressources matérielles et logicielles.

Les zones 2D ne constituent généralement qu'environ 20 à 30 % de l'image 2D globale, permettant ainsi la suppression, du reste de l'image, de toutes zones pouvant parasiter les traitements, notamment le processus de capture. Autrement dit, à l'extérieur de toutes régions d'intérêt, le reste de l'image n'est pas considéré et ne peut générer aucun traitement supplémentaire. Ainsi, les différents traitements, s'exécutant au sein de ces zones, sont cadrés et limités. Les résultats sont obtenus plus rapidement et sont plus précis, optimisant ainsi ces traitements et donc l'interactivité dans sa globalité. Par exemple, en accord avec l'interprétation de la scène observée au cours d'une situation courante, le processus de capture peut ne créer des forces dynamiques 2D qu'au sein des zones SABE. Ainsi, la capture ne s'effectue qu'au niveau de certaines parties du corps de l'utilisateur, mises en relief par le scénario. Le processus de capture, délimité et ciblé, est plus rapide et précis et le système plus allégé en terme de gestion des ressources matérielles et logicielles. Pour finir, l'optimisation matérielle est particulièrement mise en avant, lorsque les différentes zones 2D influent sur l'acquisition même du flux vidéo. C'est au niveau matériel que l'image n'est plus considérée dans sa globalité, influant alors sur l'ensemble de la chaine de traitements au cours de l'interactivité.

Les zones 3D permettent également de focaliser l'ensemble des traitements, assurant une exploitation limitée et ciblée de la scène virtuelle. Les résultats des différents traitements sont plus précis et rapidement obtenus. Il en est de même pour les écoutes des événements, générés par les éléments interactifs composant la scène. Les écoutes non mises en avant par le scénario sont désactivées, permettant au système de se focaliser sur celles d'intérêt. Par exemple, dans notre contexte d'étude, nous nous intéressons principalement aux collisions générées par les balles de tennis et les parties du corps, notamment les mains, de l'utilisateur. Tout autre élément, physiquement modélisé au sein de la scène virtuelle, pourra être désactivé, libérant ainsi les ressources, au niveau du moteur physique utilisé.
Le système ne considère plus tout et à tout moment. Il ne traite que les différentes zones de la scène et écoutes d'événements, mises en relief par le scénario de l'application. Le reste de la scène n'influe alors pas sur l'interaction en cours. Les différents traitements, notamment ceux relatifs à l'observation de la scène virtuelle et la gestion des ressources matérielles et logicielles sont ainsi optimisés.

Enfin, le processus dédié ne caractérise et n'interprète que l'ensemble des observations qui sont attendus par le scénario de l'application. Seules les données nous permettant de comprendre la gestuelle utilisateur, pour une situation d'interaction donnée, sont considérées. Par exemple, si seul le mouvement du bras droit doit être caractérisé et interprété pour une situation d'interaction donnée, les mouvements du reste des éléments composant l'avatar ne sont pas pris en compte par le processus. Ainsi, le processus de caractérisation et d'interprétation devient plus simple et rapide, allégeant l'ensemble de la chaine de traitements au sein du système.

5.4. Boucles vertueuses

La Section 5.4.1. définit précisément ce que nous appelons, dans ces travaux, une **« boucle vertueuse »**. Nous verrons que nous empruntons cette notion à notre capacité propre à nous adapter aux différentes situations que nous rencontrons.

La Section 5.4.2. expose en quoi les boucles vertueuses, résultant de l'adaptation progressive de notre système, **optimise** ce dernier.
Enfin, la dernière section présente **les différentes boucles vertueuses mises en jeu dans notre système au cours de l'interactivité**.

5.4.1. Définition

Cette notion de **« boucle vertueuse »** illustre le fait que le système considère et utilise les interactions passées pour établir et mettre en place les mécanismes adaptatifs au cours de l'étape scénarisée courante. Les résultats des interprétations de l'activité passées sont réinjectées au sein des processus traitant l'étape scénarisée courante. Par le biais et en fonction des interactions passées, les traitements des différents modules composant le système, devenus interdépendants, sont cadrés et orientés. En effet, l'analyse des différents résultats permet la redirection progressive et de plus en plus précise (au fur et à mesure des interactions) des différents traitements et processus (capture, gestion de la scène virtuelle, caractérisation & interprétation, etc.), vers les objectifs définis par l'étape scénarisée courante. Autrement dit, une meilleure adaptation du système résulte en une meilleure interaction avec l'utilisateur, adoptant alors la gestuelle adéquate vis-à-vis des objectifs du scénario de l'application, gestuelle attendue par le système et qui sera alors mieux interprétée par celui-ci (voir Figure 5.11.).

Ce processus de « boucle vertueuse » n'est pas anodin. C'est ce que chaque être humain fait lorsqu'il analyse, interprète et répond à une situation d'interaction particulière. Dans ces travaux, nous avons introduit cette notion de « boucle vertueuse » au sein de notre système.
Notre système, par le biais de l'immersion et du rendu de la scène 3D, transmet une image de son état à l'utilisateur. A l'image de l'être humain, notre système adapte son état et essaye donc de changer l'image que l'utilisateur se fait de celui-ci à un instant donné. Ce faisant, il tente de rendre cohérents les objectifs du scénario de l'application avec ceux que s'est défini mentalement l'utilisateur, de les préciser, les compléter, les redéfinir, etc. En adaptant son comportement, le système cherche à diminuer la distance existant entre les objectifs du scénario et ceux que l'utilisateur peut se définir.
Par la suite, l'utilisateur adopte la gestuelle qu'il juge adéquate, en fonction des objectifs qu'il s'est défini à partir des réponses précédentes du système. Le système, quant à lui, attend, par le biais du scénario, une gestuelle particulière de la part de l'utilisateur. En adaptant correctement son comportement et répondant de manière adéquate aux gestuelles utilisateur, le système cherche à réduire la distance pouvant exister entre la gestuelle utilisateur observée et celle attendue par le scénario.
Ainsi, par l'intermédiaire de ces boucles vertueuses, le système réduit progressivement, au fur et à mesure des interactions avec l'utilisateur la distance existant, d'une part, entre les objectifs du scénario et ceux que s'est défini l'utilisateur et, d'autre part, la gestuelle utilisateur observée et celle attendue par le scénario. Plus ces distances seront petites, plus la qualité de l'interactivité sera importante.

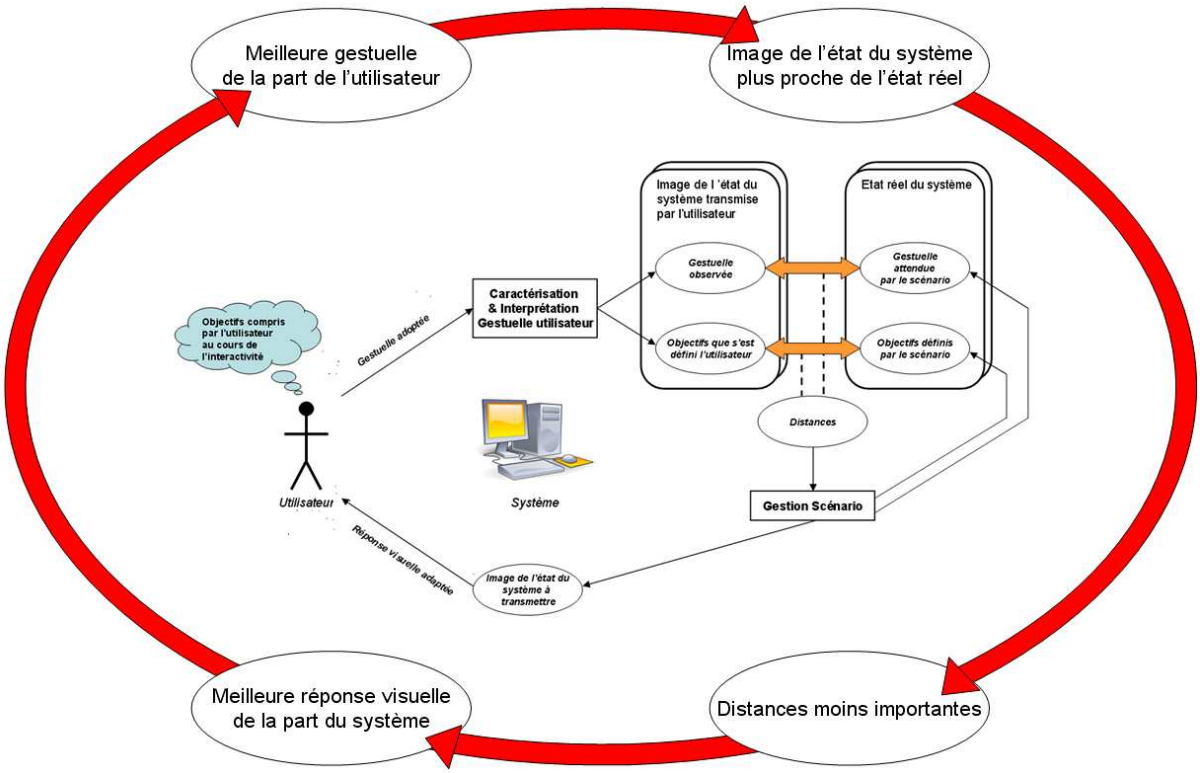


Figure 5.11. : Boucle vertueuse

5.4.2. Optimisation du système

Les boucles vertueuses mises en jeu au sein de notre système ont pour objectif **d'améliorer en permanence l'interactivité avec l'utilisateur**. Elles permettent l'optimisation progressive du fonctionnement global et interne du système interactif.
Les boucles vertueuses permettent tout d'abord de **simplifier** et **d'alléger** l'ensemble du processus interactif (Section 5.4.2.1.).
Deuxièmement, elles permettent une **interactivité plus précise et efficace** (Section 5.4.2.2.).
Enfin, elles mettent une place **des interactions de plus en plus rapides** (Section 5.4.2.3.).

5.4.2.1. Une interactivité de plus en plus simple

Au fur et à mesure que le système s'adapte aux gestuelles utilisateur, les boucles vertueuses permettent **une contextualisation de plus en plus efficace** et donc **une simplification et un allégement progressifs des processus** mis en œuvre au cours de l'interactivité.

Notre système est complexe et de nombreux processus sont mis en jeu au cours des interactions. Plus la qualité de l'adaptation du système à l'utilisateur sera importante, plus le système comprendra et reconnaitra l'activité au sein de la scène observée. Le système pourra donc focaliser ses traitements sur ce qui est attendu par le scénario, et ne pas considérer le reste de la scène. Ainsi, le nombre de traitements diminue au fur et à mesure des interactions. De plus, les différents processus participant à une interaction donnée rentrent en jeu au cours de celle-ci, si et seulement si tous les paramètres d'entrée qu'ils nécessitent pour effectuer

leurs traitements sont disponibles. Seuls les traitements nécessaires au cours de cette interaction sont donc effectués. Par conséquent, les traitements mis en jeu au cours de l'interactivité sont gérés de manières plus économique et plus sélective, utilisant des ressources matérielles et logicielles limités et particulières. Les boucles vertueuses permettent un allégement significatif et progressif des processus mis en œuvre au cours de l'interactivité, assurant de manière sûre la contrainte temps réel, que nous nous imposons dans le cadre de nos travaux.

5.4.2.2. Une interactivité de plus en plus précise

Les boucles vertueuses rendent, au fur et à mesure de l'interactivité, les traitements **plus ciblés et plus précis**, résultant en une résolution plus efficace des différentes situations d'interaction.

Plus l'adaptation du système à l'utilisateur sera bonne, meilleures seront ses réponses à ce dernier. En ce cas, le système peut donc de plus en plus émettre l'hypothèse que l'utilisateur comprend les contextes d'interaction au sein desquels il évolue. En conséquence, le système pourra attendre de plus en plus sûrement les événements attendus par le scénario de l'application, et donc mettre en place des traitements de plus en plus focalisés et précis. Face à une situation d'interaction donnée ambigüe, cette forme d'optimisation ne donne pas de solutions particulières. En revanche, elle diminue significativement le nombre d'ambigüités de sens que cette situation peut présenter.

En suivant ce même raisonnement, le système peut même aller jusqu'à prévoir les prochaines situations d'interaction, au fur et à mesure que l'interactivité se déroule en accord avec le scénario de l'application. Connaissant ce qui doit être attendu, le système peut anticiper les traitements et ressources matérielles et logicielles, dont il aura besoin lorsque la situation d'interaction, qui fera évoluer le scénario, aura à se dérouler.

5.4.2.3. Une interactivité de plus en plus rapide

La dernière forme d'optimisation qu'instaurent les boucles vertueuses concerne l**a rapidité de l'interactivité**, qui devient de plus en plus importante au fur et à mesure des réponses adaptatives du système.

Le système s'adaptant de mieux en mieux au fur et à mesure des interactions, les distances, définies au cours de la Section 5.4.1., « objectifs scénario – objectifs utilisateur » et « gestuelle observée – gestuelle attendue », deviennent de plus en plus faibles. Par conséquent, le système répondra à l'utilisateur de plus en plus en adéquation avec sa gestuelle, et de plus en plus immédiatement et rapidement. Chaque situation d'interaction sera rapidement résolue, assurant de manière certaine une interactivité temps réel.

5.4.3. Les boucles vertueuses au sein de notre système

En accord avec nos propos, nous décomposons le processus interactif en **6 étapes** :

- **Modélisation de la scène virtuelle** (capture de la scène réelle & modélisation de la scène 3D) ;
- **Mise en place des observateurs et observation de la scène virtuelle** ;
- **Caractérisation du contexte d'interaction** ;
- **Interprétation de la scène observée** ;
- **Mise en place des réponses interactives et adaptatives** ;
- **Réponse de l'utilisateur**.

La mise en place des mécanismes adaptatifs au sein de notre système permet de mettre en évidence **une boucle vertueuse** importante, liant ces différentes étapes (voir Figure 5.12.). Cette boucle vertueuse principale améliore significativement l'interactivité entre le système et l'utilisateur, au fur et à mesure que les interactions se succèdent. Ces améliorations sont dépendantes les unes des autres, l'une entrainant l'amélioration des étapes interactives suivantes.
D'une part, cette boucle vertueuse permet la propagation des bonnes interprétations du système sur l'ensemble du fonctionnement de celui-ci. D'un autre côté, si la distance, existante entre les contextes d'interactions observées et attendues, est importante, le système adapte progressivement l'ensemble de son comportement, par le biais de cette boucle vertueuse principale, dans le but de traiter efficacement les événements non attendus par le scénario, les exceptions, etc. Dans tous les cas, l'interactivité entre l'utilisateur et le système est améliorée grâce au support de cette boucle vertueuse.

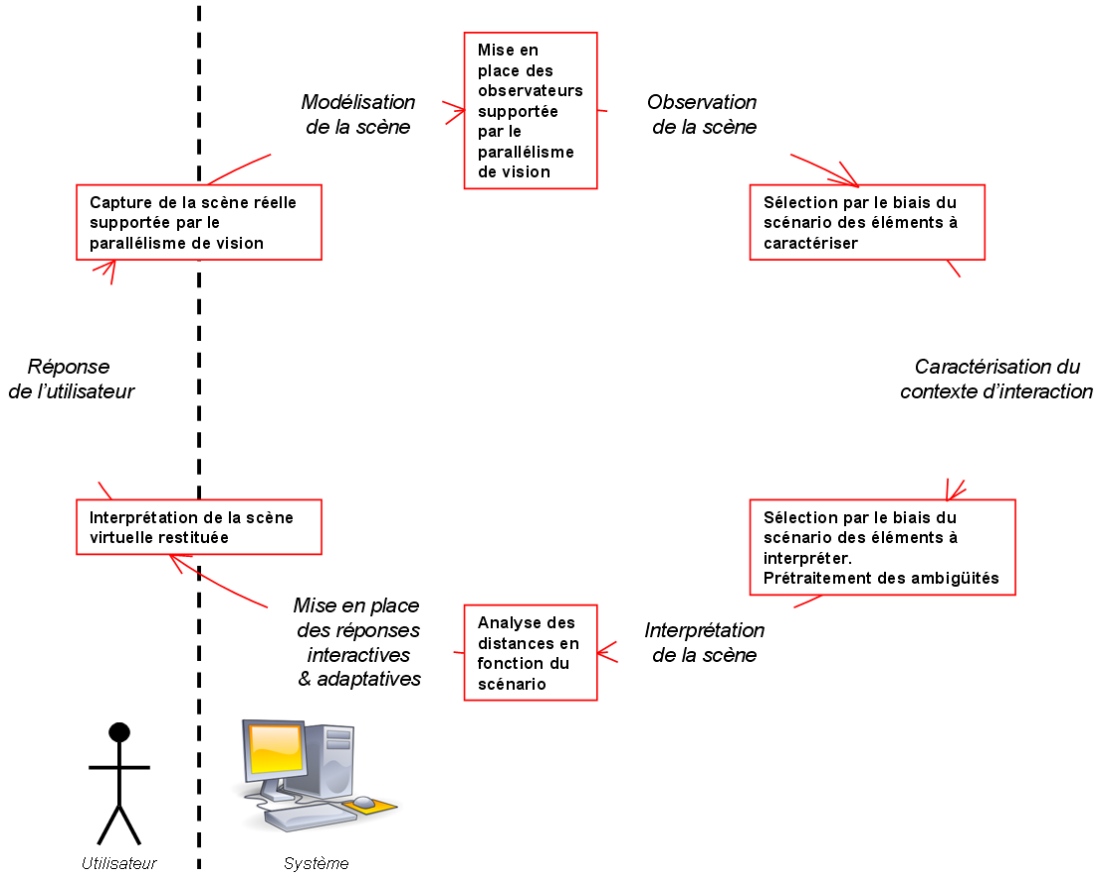


Figure 5.12. : Les boucles vertueuses au sein de notre système

Elle permet :

- **une meilleure modélisation de la scène**, notamment au niveau de la capture de la scène réelle ;
- **une meilleure observation de la scène** de la part du système, car une meilleure attente de ce qui doit être observé ;
- **une meilleure caractérisation du contexte d'interaction**, car des cadres d'études plus limités et ciblés ;
- **une meilleure interprétation de la scène** car plus rapide et simplifiée ;
- **une meilleure interaction et une meilleure adaptation**, ce qui permet une meilleure orientation de l'utilisateur vers le contexte d'interaction attendu par le scénario ;
- **une meilleure gestuelle** adoptée par l'utilisateur ;
- et ainsi de suite !

Dans ces travaux, nous parlons de **« boucles vertueuses »** (au pluriel), dans la mesure où il est possible de considérer cette boucle permanente à partir des différentes étapes d'une interaction. Le processus défini par une étape particulière est amélioré, au fur et à mesure de l'interactivité, par la boucle vertueuse s'effectuant à partir de cette étape. **Les 6 boucles vertueuses**, définies par les différentes étapes de l'interaction, sont totalement interdépendantes.
Les boucles vertueuses sont particulièrement présentes et observables, au cours des étapes au sein desquelles les mécanismes adaptatifs sont particulièrement élaborés (dans nos travaux : capture de la scène réelle, observation de la scène virtuel, réponse visuelle interactive). Les sections suivantes se proposent de présenter brièvement les différentes boucles vertueuses, ainsi que leurs effets sur l'ensemble des processus, à partir des différentes étapes de l'interaction :

- **Modélisation de la scène** (Section 5.4.3.1.).
- **Observation de la scène** (Section 5.4.3.2.).
- **Caractérisation du contexte d'interaction** (Section 5.4.3.3.).
- **Interprétation de la scène observée** (Section 5.4.3.4.).
- **Mise en place des réponses** (Section 5.4.3.5.).
- **Réponse de l'utilisateur** (Section 5.4.3.6.).

5.4.3.1. Modélisation de la scène

La boucle vertueuse mise en place au cours de cette étape concerne particulièrement **la capture de la scène réelle**, dans la mesure où la modélisation de la scène 3D n'est pas adaptée.

A l'instar des mécanismes adaptatifs présents à ce niveau, la boucle vertueuse est supportée par le parallélisme de vision mise en place par le processus de capture. La scène réelle est capturée de manière focalisée (gestion multiples-focus) mais le processus peut à tout instant considérer une capture plus globale de la scène réelle. Ce parallélisme permanent et possible à tout moment est important, car le processus doit pouvoir faire face à tout événement inattendu prenant place au sein de la scène observée. En fin de compte, le processus de capture doit être capable de prendre en compte n'importe quel résultat, à n'importe quel moment.

Plus les réponses de l'utilisateur seront cohérentes avec le scénario, plus il sera possible au système de considérer que l'utilisateur sait vers quels objectifs se diriger. Ainsi, le processus de capture focalise ses différents traitements : focalisation sur une partie du corps de l'utilisateur en particulier, sur les différentes parties en mouvement de son corps, etc. Au fur et à mesure que la boucle vertueuse se met en place, le processus de capture considère de moins en moins l'utilisateur dans sa globalité, se cantonnant aux gestuelles adoptées par certaines parties de sont corps bien particulière. De plus, les traitements s'effectuent à des instants particuliers, l'ensemble des *frames* acquises n'étant plus pris en compte. Le système considère uniquement les intervalles temporels mis en relief par le scénario de l'application.

Au niveau des traitements, cela se traduit par le fait que les régions d'intérêt considérées deviennent plus petites, définissant un cadre plus limité et plus ciblé pour les différents traitements. De plus, au cours de cette boucle vertueuse, la gestion des différentes ressources est optimisée, dans la mesure où le nombre de traitements et la portée de ceux-ci sont moins importants. Enfin, si les interactions se déroulent bien, les résultats deviennent plus en plus précis et les régions d'intérêt peuvent aller même jusqu'à influer sur l'acquisition vidéo de la scène réelle.

5.4.3.2. Observation de la scène

Le processus d'observation de la scène virtuelle est également supporté par le parallélisme de vision. Ces observations peuvent donc être ciblées ou globales, le changement de vision pouvant être décidé par le système à tout moment.

Au niveau de la scène réelle, n'est observé que ce qui est mis en relief par le scénario de l'application à un instant donné. Cette observation se fait par le biais des zones 2D, informations contextuelles comprises au sein du contexte « Observation ».

Au niveau de la scène virtuelle de manière générale, la boucle vertueuse se situe au niveau du moteur de jeu, plus particulièrement au niveau des gestionnaires des zones 3D et des écoutes d'événements générés par les interactions de l'utilisateur. En fonction de la situation d'interaction courante et des objectifs définis par celle-ci, le processus d'observation priorise de plus en plus ses traitements au niveau de relations et d'événements bien particuliers, activant les zones 3D et les ressources interactives correspondantes. Les autres zones et ressources, qui ne sont pas mises en relief par le scénario, ne sont pas considérées au cours de l'interaction en cours.

5.4.3.3. Caractérisation du contexte d'interaction

Au niveau du processus de **caractérisation du contexte d'interaction**, la boucle vertueuse permet une prise en compte de plus en plus focalisée des différentes connaissances extraites de la scène virtuelle. La caractérisation du contexte d'interaction est plus simple et plus rapide, tout en optimisant la gestion des ressources logicielles et matérielles. C'est le scénario qui définit ce qui doit être caractérisé, les informations n'étant alors pas utilisées.
L'amélioration au niveau de la caractérisation est possible de par le fait que le processus d'observation attende et observe de mieux en mieux certains événements bien particuliers.

5.4.3.4. Interprétation de la scène

La boucle vertueuse présente à ce niveau permet **une interprétation de plus en plus rapide**, précise et simplifiée de l'activité prenant place au sein de la scène observée. La distance pouvant exister entre le contexte d'interaction attendu et celui observé devient de moins en moins complexe et de plus en plus facile à analyser, au fur et à mesure que l'interactivité se déroule.
Le nombre de données exploitées est également de moins en moins important. Les ressources logicielles et matérielles nécessaires diminuent et la gestion de ces dernières est optimisée.

L'effet de la boucle vertueuse est particulièrement observable au niveau des ambigüités de sens que peut présenter l'activité au sein de la scène virtuelle. La boucle vertueuse permet une cohérence temporelle des différents résultats. Autrement dit, le processus d'interprétation résout une ambigüité de sens à un instant donné et que l'utilisateur adopte les gestuelles adéquates vis-à-vis du scénario, alors la résolution de cette ambigüité devient implicitement évidente. Le processus d'interprétation reste dans la 'vérité' et l'ambigüité ne se présente plus. Les différentes connaissances sont de moins en moins soumises à analyses, dans la mesure où elles suivent de plus en plus une évolution prévue par le scénario. Enfin, le fait qu'en amont de cette étape les résultats soient plus précis, entraine une diminution du nombre d'ambigüités de sens que peut présenter une scène observée.

5.4.3.5. Mise en place des réponses interactives et adaptatives

La boucle vertueuse, définie à partir de cette étape, permet **une orientation de plus en plus ciblée et directe de l'utilisateur**, vers le contexte d'interaction attendu par le scénario de l'application. Cette orientation se matérialise, du point de vue de l'utilisateur, au niveau de la **scène 3D immersive**.

Si l'utilisateur adopte de plus en plus la bonne gestuelle, alors le système peut se baser sur le fait que l'utilisateur comprend ses objectifs et se dirige vers ces derniers. Le système adapte alors l'ensemble de son fonctionnement à la gestuelle utilisateur, tel que cela est décrit au sein du scénario. Ses réponses **interactives** et **adaptatives** sont de plus en plus cohérentes et immédiates vis-à-vis des contextes d'interaction observés.
Si la gestuelle utilisateur n'est pas comprise ou si elle est ambigüe, le système met en place un ensemble de réponses, dans le but de replacer l'utilisateur au sein du contexte d'interaction attendu et de comprendre la gestuelle de ce dernier, ainsi que les objectifs vers lesquels il se dirige. Cela se traduit également par le fait que le système passe à une vision globale de la scène virtuelle pour mieux la capturer et l'observer.

5.4.3.6. Réponse de l'utilisateur

La dernière boucle vertueuse prend place à partir de **la réponse de l'utilisateur**. Plus la gestuelle qu'il adopte est cohérente du point de vue du scénario, plus le système peut considérer ce fait comme une vérité au cours de l'interactivité. Les distances, définies au cours de la Section 5.4.1., particulièrement celle pouvant exister entre les objectifs tels que se les ai définis l'utilisateur et ceux attendus par le scénario, se réduisent.

A partir de là, le système peut mieux adapter, et plus rapidement, ses réponses aux gestuelles utilisateur. L'interaction est de plus en plus immédiate et juste, dans le sens cohérent avec la gestuelle utilisateur.

6. Application : Entrainement virtuel de tennis

Nous présentons, dans cette section, l'application développée au cours de ces travaux. Le scénario de l'application, que nous rappelons ci-dessous, a été présenté au cours de la Section 1. de ce chapitre.

Une partie de jeu comprend 15 lancers de balles. Chaque partie est divisée en 3 phases de jeu de 5 lancers. L'utilisateur a pour objectif d'intercepter les balles, lancées par le canon se trouvant en face de lui, avec l'aide de ses mains. Dans la mesure où il s'agit d'un entrainement virtuel au tennis, l'utilisateur doit intercepter les balles lancées par le biais de coups droits (interception avec la main se situant du même côté que la balle lancée) et de revers (interception avec la main se situant du côté opposé à celui de la balle lancée). La partie se termine à l'issue des 15 lancers.

Au cours de cette section, nous adoptons notre approche, en mettant en valeur **les différentes contributions de ces travaux de thèse**. Nous détaillons **3 études de cas**, mis en place par ce scénario, que nous illustrons en explicitant le comportement du système vis-à-vis de l'activité capturée. Pour chaque étude abordée, nous identifions **les différentes informations contextuelles** permettant au système d'interpréter l'activité au sein de la scène, et **les mécanismes adaptatifs** mis en œuvre après cette interprétation.

La Section 6.0. rappelle **les différents fondements de notre approche**, concernant la modélisation d'un scénario sous la forme d'une structure **de situations et de contextes d'interaction**.
Les études de cas que nous présentons au cours de cette section nous amène à émettre **plusieurs hypothèses concernant le déroulement de l'interactivité**. Nous exposons ces hypothèses au cours de la Section 6.1.
La Section 6.2. décrit, sous la forme de situations et de contextes d'interaction, **les réactions de l'utilisateur**, **au cours d'un simple lancer de balle**, orchestré par le système.
L'étude de cas décrite par la Section 6.3. correspond à une partie plus haut niveau du scénario de l'application. Nous décrivons en effet, également sous la forme situations et de contextes d'interaction, **un lancer de balle** (au cours duquel les réactions de l'utilisateur sont observées), **une phase de jeu** et enfin **une partie de jeu**.
La Section 6.4. décrit une activité ne s'attachant pas à proprement parler à l'interaction avec l'utilisateur mais liée à **la présence de spectateurs au sein de la scène réelle**. Nous présentons au cours de cette section le comportement de notre système face à la présence éventuelle de spectateurs.

6.0. Rappel : Modélisation d'un scénario

Les deux premières études de cas sont décrites par un ensemble de **situations** et de **contextes** d'interaction. Notre contribution, présentée au cours de la Section 6. du Chapitre 2., expose notre modélisation d'un scénario sous cette forme. Nous en faisons ici un résumé simplifié.

Nous rappelons que, dans la suite de ce chapitre, nous ne détaillerons pas complètement les modélisations que nous décrivons, la modélisation de scénario sous la forme de situations et de contextes d'interaction n'étant pas à proprement parler de nos contributions.
Dans nos représentations, nous mettrons en avant les différents éléments, qui nous permettrons d'illustrer nos travaux et de valider notre approche. Les représentations que nous présentons ne sont bien sûr pas uniques et ont été élaborées de manière à mettre en relief, au mieux, nos contributions.

La Section 6.0.1. décrit brièvement **notre modélisation du système d'information global**, que nous appelons l'univers, **d'une situation d'interaction** et **d'un contexte d'interaction**.
La Section 6.0.2. souligne les grands lignes **d'une modélisation de scénario**, basée situations et contextes d'interaction.

6.0.1. Univers, situation d'interaction et contexte d'interaction

Notre système d'information global, l'univers, est représenté sous la forme d'un vecteur d'états, défini à un instant donné. Ce vecteur, à un instant donné, décrit les états relatifs des acteurs de l'interaction : l'utilisateur, le système et l'application, la scène réelle, l'interface et l'observation, i.e. l'ensemble des éléments perceptibles et interprétables par le système.
Dans les prochaines sections, nous ne ferons pas référence à ce mot d'état. En effet, se rapporter en permanence à celui-ci rendrait la suite de nos propos trop lourds. Nous mentionnerons les différents acteurs de l'interaction, prenant part à une situation ou un contexte, en explicitant certains de leurs états (le geste de l'utilisateur, la zone SASM de l'observation, etc.). Nous appellerons 'participant', un acteur de l'interaction prenant part à une situation ou un contexte d'interaction.

Une situation et un contexte d'interaction sont activés par la vérification d'un ensemble de pré-conditions et se terminent une fois qu'un ensemble de post-conditions sont remplies. Dans la suite de ce chapitre, nous ne parlerons pas des conditions visant à filtrer les participants, ou certains de leurs états. Les participants, ou les états de ceux-ci, nécessaires au bon déroulement du scénario, seront mentionnés dans nos propos, impliquant ainsi implicitement une sélection de ceux-ci. Les seules conditions que nous identifieront concerneront la valeur prise par certains états de certains participants.
Que cela soit pour une situation ou un contexte, les différentes pré et post-conditions sont organisés en un ou plusieurs sets, chaque set correspondant à un début ou une fin de situation ou de contexte.

Au cours du déroulement d'une situation ou d'un contexte, chaque participant agit en fonction du scénario qui lui est propre. L'utilisateur va agir en fonction de ce qu'il comprend vis-à-vis de la scène immersive, un processus va agir en fonction de différents algorithmes, etc. Ces scénarios peuvent être explicités sous la forme de situations et de contextes d'interaction, à un niveau sémantique plus bas que celui de la situation ou du contexte dans laquelle/lequel ils se déroulent.
A l'issue du déroulement d'une situation ou d'un contexte d'interaction, un gestionnaire de cohérence se charge de confronter les états des participants avec ceux attendus par le biais des post-conditions. Selon la nature de la situation ou du contexte, et des états des participants, le gestionnaire de cohérence peut décider que la situation ou le contexte est ou n'est pas terminé(e). Dans la suite de ce chapitre, nous ne représenterons pas les gestionnaires de

cohérence. Ceux-ci seront matérialisés par le biais des post-conditions à remplir pour terminer la situation ou le contexte d'interaction.

A la différence des situations d'interaction, les contextes d'interaction comprennent une étape de contextualisation des participants. Cette contextualisation est une sélection et une attribution de valeurs plus focalisées et plus orientées vers l'objectif à atteindre pour faire évoluer le scénario. Nous mentionnerons implicitement cette sélection et cette attribution en ne parlant que des informations contextuelles présentes au niveau du contexte. Par exemple, le fait de parler de la zone SASM du contexte « Observation » présuppose qu'au niveau de l'observation, une partie de l'image 2D, représentant un point de vue donné, a été sélectionnée et définie. De plus, nous n'énumérerons que les informations significatives quant à la compréhension et la cohérence de la nature du contexte d'interaction. Ces informations contextuelles, d'une part, engendreront la contextualisation des éléments de plus bas niveau se situant au sein du déroulement du contexte, d'autre part, permettront l'interprétation par le processus dédié de l'activité observée, et, enfin, le paramétrage des mécanismes adaptatifs mis en place à la fin du déroulement du contexte.
A la fin de son déroulement et parallèlement à la gestion de la cohérence des états des participants, le contexte d'interaction met en œuvre un ensemble de mesures adaptatives, paramétrées par les différentes informations contextuelles considérées. Chaque set de post-conditions peut correspondre à un set particulier de mécanismes adaptatifs, auquel s'ajoutent des mécanismes adaptatifs plus généraux, valables pour toutes fins de déroulement.

Les Figures 5.13. illustrent notre modélisation d'une situation et d'un contexte d'interaction, telle que nous la retenons dans la suite de ce chapitre.

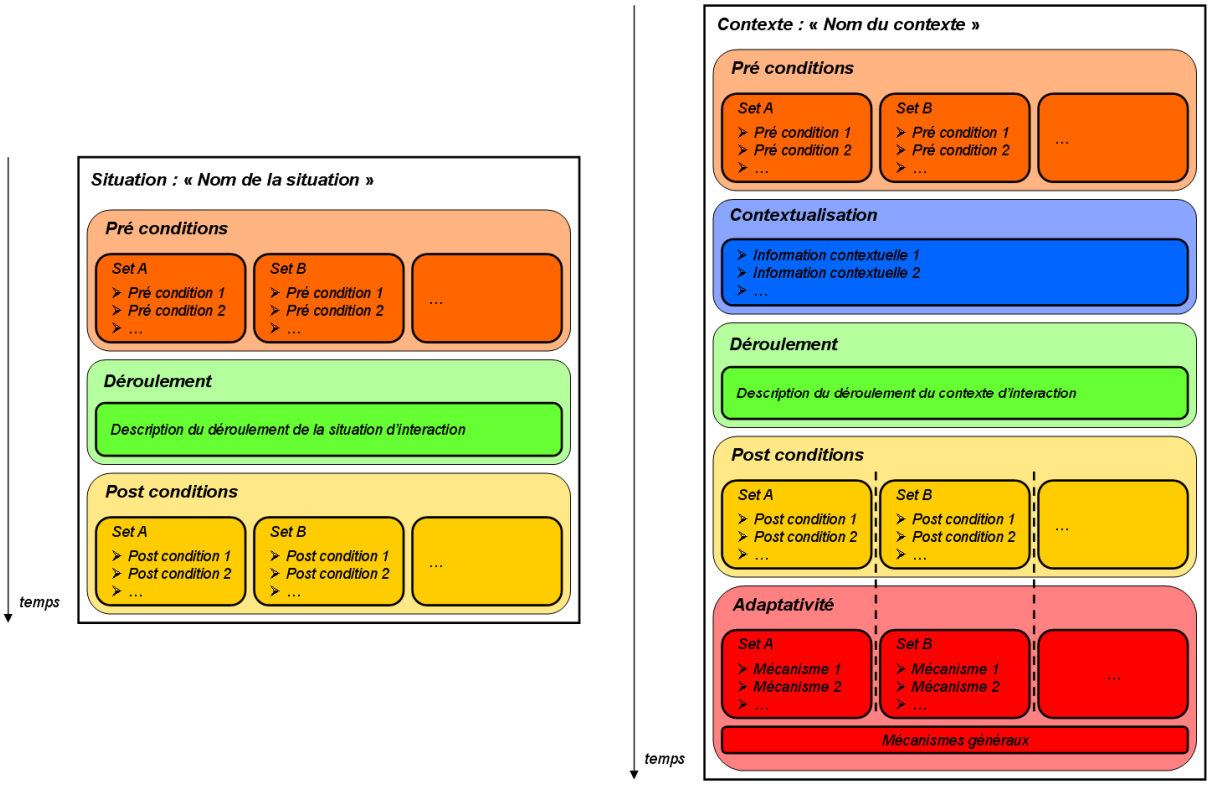


Figure 5.13. : Situations (à gauche) & contextes d'interaction (à droite)

6.0.2. Modélisation d'un scénario

Dans les prochaines sections, nous allons proposer plusieurs représentations de scénarios sous la forme d'une structure de situations et de contextes d'interaction.

Un scénario comprend plusieurs degrés sémantiques de représentation. Par exemple, au sein d'un contexte d'interaction pourront être modélisés différents situations et contextes d'interaction. Dans notre scénario, au sein d'une partie, nous avons plusieurs phases de jeu, au sein desquelles plusieurs lancers ont lieu. D'un point de vue modélisation, les éléments de plus bas niveau sémantique se situent au sein du déroulement de l'élément de plus haut niveau les englobant.

Chaque situation et chaque contexte, par le biais de pré-conditions, sélectionnent un certain nombre de participants. Ces participants, et leurs différents états, contextualisent implicitement tous les éléments de plus bas niveau englobés dans le déroulement de l'élément qui les ont sélectionnés. En effet, les éléments de plus bas niveau ne pourront considérer que ces différents participants, leur univers se limitant à eux. Ainsi, le déroulement de ces éléments de plus bas niveau sera, implicitement et automatiquement, orienté et cadré. C'est pourquoi, du point de vue d'un ensemble d'éléments d'interaction, il est possible de qualifier de « contexte » tout élément englobant ces derniers.
Le contexte d'interaction, à la différence de la situation, permettra une contextualisation explicite, plus centrée, plus orientée et plus focalisée, des éléments de plus bas niveau. De plus, les différentes informations contextuelles considérées permettent le paramétrage des mécanismes adaptatifs, mis en œuvre à la fin du déroulement du contexte d'interaction. C'est pour cela que nous différencions, dans notre modélisation, une situation et un contexte d'interaction. Et c'est également comme cela que nous déterminons la nature d'un élément au sein de notre scénario.

Nous allons représenter par la suite plusieurs scénarios, sous la forme de situations et de contextes. Par le biais des contextes seront mises en relief des informations contextuelles particulières. Conformément à nos propos précédents, ces informations contextuelles seront présentes à des niveaux sémantiques inférieurs. Dans un souci de clarté et pour ne pas alourdir les représentations, pour un niveau sémantique donné, nous n'expliciterons pas les informations contextuelles d'un niveau sémantique supérieur, celles-ci étant considérées implicitement. Pour un contexte d'interaction donné, nous ne parlerons que des informations contextuelles pertinentes quant à la nature et au degré sémantique de ce contexte.

Chaque scénario, décrit sous la forme de situations et de contextes d'interaction, s'accompagnent d'un agenda, permettant en partie la caractérisation du sous contexte « Temps ». Ces agendas n'auront pas forcément d'importance, pour illustrer nos études de cas. Toutefois, nous en proposerons une représentation pour chaque scénario modélisé.

6.1. Hypothèses sur le déroulement de l'interactivité

Nous présentons **les différentes hypothèses** que nous avons faites sur le déroulement général de l'interactivité. Il est nécessaire d'expliciter ces hypothèses, avant de présenter nos 3 études de cas.

Nous partons du principe que l'utilisateur cherchera à intercepter les balles avec ses mains. Dans la mesure où il sait qu'il est immergé dans un entrainement virtuel de tennis, il n'interceptera *a priori* pas les balles lancées avec d'autres parties de son corps que ses mains. Le fait de lui dire, avant de mettre en place l'interactivité avec le système, qu'il joue au tennis, entrainera chez lui, sur le plan cognitif, une mise en condition adéquate. Il se placera ainsi immédiatement dans un contexte d'entrainement de tennis et interceptera les balles avec les mains. C'est une manière d'orienter l'utilisateur vers les gestuelles adéquates à effectuer.

Concernant les lancers, ceux-ci sont effectués de manière à ce que la balle arrive à une extrémité (gauche ou droite) de la surface où évolue l'utilisateur. Ainsi, l'utilisateur, d'une part, se déplace du centre de la scène. Il est obligé de se déplacer dans une zone, qui ne se trouve pas à sa portée immédiate, s'il reste au centre de la scène. D'autre part, il effectue généralement un mouvement de la main droite, pour intercepter une balle arrivant sur sa droite, et un mouvement de la main gauche, pour intercepter une balle arrivant sur sa gauche. Autrement dit, il ne croise a priori pas les bras (un revers) pour intercepter les balles et ne génère ainsi pas d'ambigüités d'interprétation de la part du système, concernant les poses des différents éléments les composant. L'utilisateur ne cherchera normalement pas à intercepter une balle arrivant sur sa droite (ou sa gauche) avec sa main gauche (ou droite). Les gestuelles de type 'revers' sont cependant possibles, le système mettant en place les mécanismes adaptatifs adéquats.
Quoiqu'il en soit, dans le cadre de notre application, nous ne distinguons pas le fait que l'utilisateur intercepte une balle de la main gauche ou de la main droite. Nous considérons une interception comme bonne, à partir du moment où l'utilisateur intercepte la balle de la main, ou, à la rigueur, du bras. Cela dit, cette distinction serait facilement réalisable, dans la mesure où les différentes parties du corps de l'utilisateur sont sémantiquement identifiables.
Enfin, le canon envoie les balles à l'utilisateur, de manière alternée (droite-gauche) et selon des paramètres de lancer prédéfinis.

6.2. Réaction de l'utilisateur au cours d'un lancer de balle

Nous décrivons dans cette section le comportement adapté du système face aux réactions de l'utilisateur au cours d'un lancer de balle. Ces réactions sont observées entre l'instant où la balle est lancée et l'instant où elle entre en collision avec un élément de la scène (sol ou partie du corps de l'utilisateur). Cette partie du scénario est la description de plus bas niveau que nous présenterons dans ces travaux.

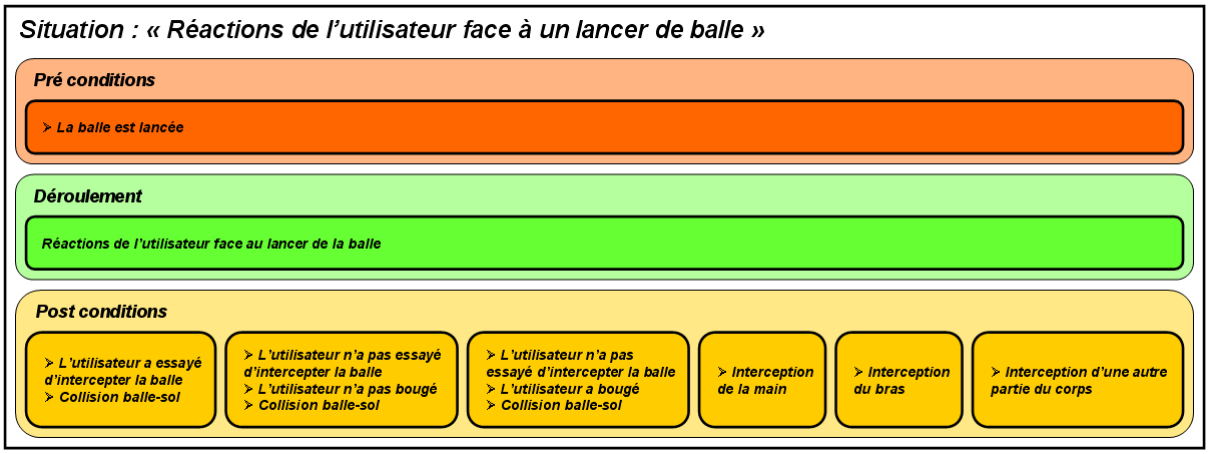


Figure 5.14. : Situation : Réactions de l'utilisateur face à un lancer de balle

Nous décrivons cette étude de cas sous la forme d'une situation d'interaction, comme le montre la Figure 5.14. Cette situation est activée sous la condition qu'une balle soit lancée par le canon se trouvant en face de l'utilisateur. Après le déroulement de la situation, cette dernière se termine de 6 façons possibles, chaque fin correspondant à un set de post-conditions bien particulier :

1. La balle touche le sol, l'utilisateur ne l'ayant pas interceptée. Cependant, ses réactions laissent à penser qu'il a essayé de le faire. Cette fin résulte en un loupé de la part de l'utilisateur.
2. La balle touche le sol, l'utilisateur ne l'ayant pas interceptée. Il n'a pas essayé de l'intercepter mais a cependant été mobile au cours du lancer. Cette fin résulte en un loupé de la part de l'utilisateur.
3. La balle touche le sol, l'utilisateur ne l'ayant pas interceptée. Il n'a pas essayé de l'intercepter et n'a pas été mobile au cours du lancer. Cette fin résulte en un loupé de la part de l'utilisateur.
4. L'utilisateur a intercepté la balle de la main, avant que cette dernière ne touche le sol. Cette fin résulte en une interception de la part de l'utilisateur.
5. L'utilisateur a intercepté la balle avec le bras, cet élément du modèle 3D représentant l'utilisateur étant composé de l'épaule, d'une partie supérieure, de l'avant bras, de la main et des articulations entre ces différents éléments (voir Section 3.2.1.). Cette fin résulte en une interception de la part de l'utilisateur.
6. L'utilisateur a intercepté, volontairement ou involontairement, la balle avec une partie de son corps, autre que son bras (donc que sa main). Cette fin résulte en un loupé de la part de l'utilisateur.

Ces fins de situation ont été décidées au cours de l'écriture du scénario. Elles ont été déterminées dans le but d'apporter au scénario plusieurs variations intéressantes quant à la suite du déroulement de celui-ci.
La situation d'interaction se termine lorsque la balle entre, pour la première fois, en collision avec un élément de la scène, le sol ou une partie du corps de l'utilisateur. Cette collision est détectée par le moteur de physique, utilisé lors de l'exécution de l'application. En effet, ce moteur de physique gère les comportements des différents éléments de la scène (les parties du

corps de l'utilisateur, les balles, le sol, etc.), simulés selon les lois standards de la physique. Il permet la détection rapide des collisions entre ces différents éléments.

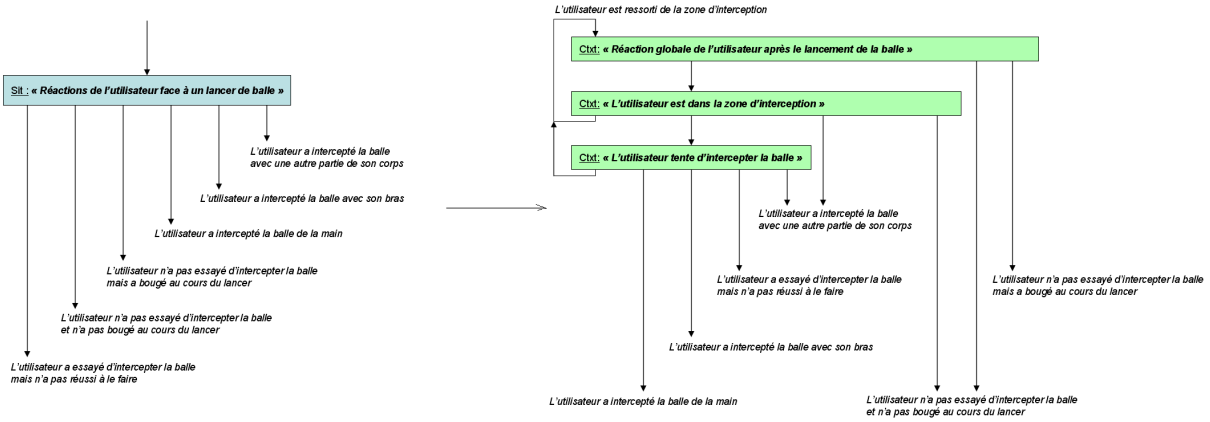


Figure 5.15. : Décomposition de la situation d'interaction en 3 contextes d'interaction

En descendant d'un niveau sémantique, il est possible de décomposer les réactions de l'utilisateur face à un lancer de balle. Nous décomposons, comme le montre la Figure 5.15., la situation d'interaction « Réactions de l'utilisateur face à un lancer de balle » en 3 contextes d'interaction, s'exécutant au cours du déroulement de celle-ci. Ces 3 contextes décrivent le déroulement du scénario :

- au cours de la « Réaction globale de l'utilisateur après le lancement de la balle » (Section 6.2.1.) ;
- lorsque « L'utilisateur entre dans la zone d'interception » de la balle (Section 6.2.2.) ;
- et lorsque « L'utilisateur tente d'intercepter la balle » (Section 6.2.3.).

Ces 3 contextes sont articulés comme présenté par la Figure 5.15. En accord avec la situation d'interaction de plus haut niveau, le déroulement de ces trois contextes se termine par les 6 fins précédemment décrites.
En revanche, le déroulement de la situation d'interaction de plus haut niveau n'implique pas nécessairement le déroulement des 3 contextes d'interaction la composant. Chaque contexte peut générer la fin de la situation d'interaction.
Parallèlement, le déroulement des 3 contextes d'interaction peut aboutir à une même fin de plusieurs manières différentes. Ainsi, la fin traduisant le fait que l'utilisateur intercepte la balle avec une partie de son corps, différente de son bras, est possible à partir du contexte « L'utilisateur est dans la zone d'interception » et du contexte « L'utilisateur tente d'intercepter la balle ».
Enfin, à partir des contextes « L'utilisateur est dans la zone d'interception » et « L'utilisateur tente d'intercepter la balle », l'utilisateur peut se retrouver au début de la situation, à savoir au sein du contexte « Réaction globale de l'utilisateur après le lancement de la balle ». Ce retour en arrière est du aux gestuelles de l'utilisateur, qui traduisent son abandon quant à sa volonté d'intercepter la balle. Ces retours en arrière sont gérés au niveau du gestionnaire de cohérence de la situation « Réactions de l'utilisateur face à un lancer de balle » (voir Section 6.0.1.).

Le scénario, ainsi décrit, nous permet d'élaborer également l'agenda associé, information contextuelle comprise dans le sous contexte « Temps » (voir Figure 5.16.).
Au niveau temporel, seule la collision de la balle, avec le sol ou une partie du corps avec l'utilisateur, nous intéresse. C'est pourquoi, les différentes sorties d'un même contexte peuvent être regroupées en accord avec la suite du déroulement du scénario ('x2' et 'x3' dans le schéma). Comme le montre l'agenda, la collision avec le sol est logiquement plus tardive que celle avec une partie du corps de l'utilisateur. Les retours en arrière ne sont pas mentionnés au niveau de l'agenda, car seules les instants des collisions, qui peuvent survenir à tout moment, sont importants dans le cadre de notre scénario.

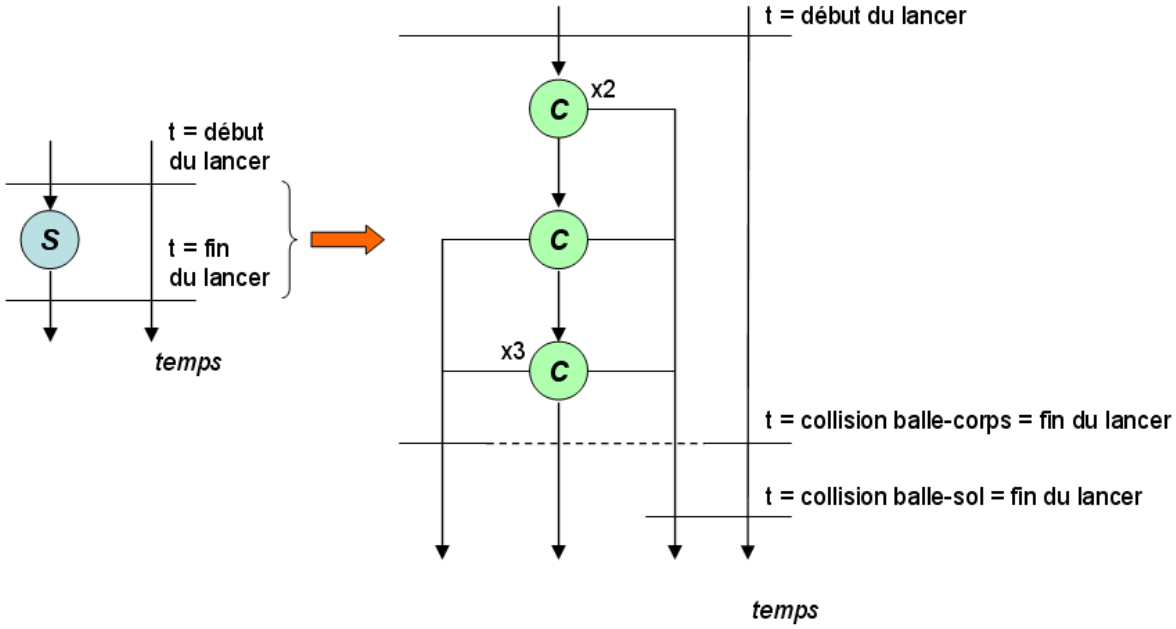


Figure 5.16. : Agenda associé à la situation « Réactions de l'utilisateur face à un lancer de balle »

6.2.1. Réaction globale de l'utilisateur après le lancement de la balle

C'est au sein du contexte « Réaction globale de l'utilisateur après le lancement de la balle » que le début de la gestuelle de l'utilisateur est observé. Le processus d'observation de la scène se focalise sur la gestuelle utilisateur globale, juste après le lancement de la balle.
La Figure 5.17. présente le contexte d'interaction, tel qu'il est modélisé au sein du scénario.

Au niveau des pré-conditions pour activer le contexte d'interaction, la balle de tennis à intercepter doit être lancée. Implicitement, celle-ci ne doit pas avoir encore rebondi sur un élément de la scène, tel que le sol ou une partie du corps de l'utilisateur.

A ce niveau du scénario, plusieurs informations contextuelles sont nécessaires pour interpréter l'activité observée au sein de la scène.
Le contexte « Utilisateur » comprend un ensemble de règles qualitatives vis-à-vis de la gestuelle utilisateur globale. Ces règles permettent ainsi de savoir si l'utilisateur bouge son corps vers la balle ou s'il se déplace dans le sens opposé. Etablies au niveau des bras et des mains, elles indiquent également si l'utilisateur a l'intention d'intercepter la balle avec son

bras situé du côté de la balle, ou avec son bras situé du côté opposé. Une fois la scène interprétée, le système mettra en place un ensemble de mécanismes adaptés pour traiter le bras opposé à la balle.
Le contexte « Interface » indique au système la caméra élue comme étant celle la plus en face de l'utilisateur et englobant l'ensemble de la scène réelle. Cette information contextuelle permet au système d'exécuter les mécanismes adéquats pour détecter la droite de la gauche de l'utilisateur. Cette connaissance permettra de qualifier (« droite » ou « gauche ») le bras en mouvement au cours du contexte d'interaction.
Au sein du contexte « Observation », le processus de capture de la scène réelle et le processus d'observation de la scène utilisent les zones SASM, pour se focaliser sur l'utilisateur au niveau de son corps, et SAMP, pour détecter le mouvement, notamment généré par l'utilisateur, au sein de la scène. Le processus d'interprétation peut ainsi décider si l'utilisateur est en mouvement après le lancement de la balle. Le contexte « Observation » fournit également les informations contextuelles permettant de caractériser une zone 3D d'interception supposée. Il est en effet possible d'estimer la trajectoire de la balle, et donc une zone approximative d'interception présumée, grâce aux paramètres connus de lancer. Enfin, comprise au sein de contexte « Observation », une dernière information permet de focaliser le processus d'observation sur la collision générée par la balle lancée.

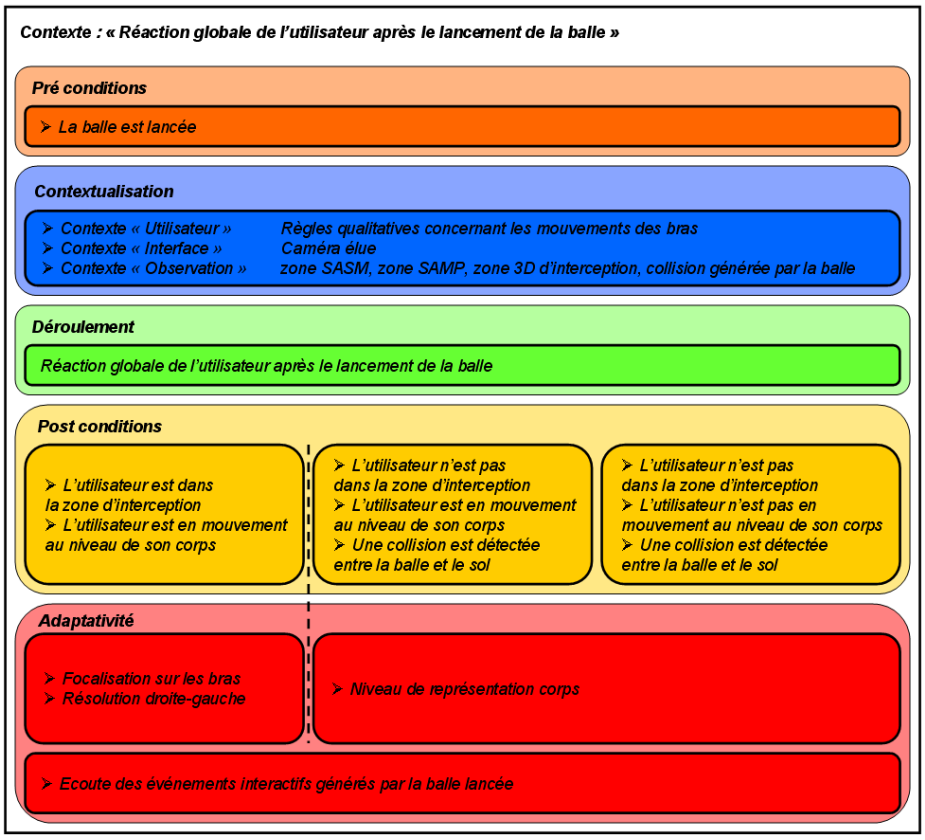


Figure 5.17. : Contexte : « Réaction globale de l'utilisateur après le lancement de la balle »

Le contexte d'interaction « Réaction globale de l'utilisateur après le lancement de la balle » comprend 3 sets de post-conditions.

Le premier set de post-conditions exprime l'intention de l'utilisateur d'intercepter la balle lancée. L'utilisateur, au niveau de son corps, est entré dans la zone d'interception. De plus, les mesures de mouvement, au niveau du corps de l'utilisateur, ne sont pas nulles. Ce set de post-conditions provoque une évolution du scénario vers le contexte « L'utilisateur entre dans la zone d'interception ».
Le second set traduit un mouvement de la part de l'utilisateur mais une non intention de se rapprocher de la balle lancée et ce, jusqu'à ce que la balle touche le sol. Ceci se traduit par des mesures de mouvement non nulles, au niveau du corps de l'utilisateur, et par une collision détectée entre la balle lancée et le sol. Le contexte d'interaction et la situation d'interaction de plus haut niveau « Réactions de l'utilisateur face au lancer de balle » sont alors terminés.
Enfin, le dernier set de post-conditions indique une immobilité du joueur au cours du lancer. Aucun mouvement n'a été enregistré, jusqu'à ce que la balle touche le sol. Le contexte d'interaction et la situation d'interaction de plus haut niveau « Réactions de l'utilisateur face au lancer de balle » sont alors terminés.

Le contexte d'interaction comprend un set de mécanismes adaptatifs relatifs au set de post-conditions indiquant une évolution de l'utilisateur vers la balle, et un set de mécanismes adaptatifs commun aux deux autres sets de post-conditions. De plus, il comprend un ensemble de mécanismes adaptatifs mis en place de manière générale.
Le premier set de mécanismes adaptatifs est mis en place avant de faire évoluer le scénario vers le contexte d'interaction suivant. Tout d'abord, le parallélisme de vision permet aux processus de capture de la scène réelle et d'observation de focaliser leurs traitements (segmentation, squelettisation, capture de mouvements, extraction de connaissances au niveau de la scène) sur les bras de l'utilisateur. Par rapport au modèle 3D représentant l'utilisateur, les traitements se font au niveau des « Membres », de type « Bras ». De plus, un ensemble de traitements est exécuté de manière à résoudre les ambigüités droite-gauche que le déplacement des bras peut générer (détection du visage de l'utilisateur par le biais de la caméra élue).
Les deux sets de post-conditions, entrainant une évolution du scénario vers la fin de la situation d'interaction de plus haut niveau, activent le même set de mécanismes adaptatifs. Par le biais de ceux, les processus de capture de la scène réelle et d'observation repassent à une vision globale et complète de la scène.
Enfin, de manière générale, la fin du contexte d'interaction entraine la mise en place de mécanismes adaptatifs, dédiés à la désactivation de la gestion physique des balles se trouvant sur le terrain. D'une manière ou d'une autre, seule la détection de la collision générée par la balle lancée est nécessaire au processus d'interprétation. Les balles se trouvant à terre ne doivent pas générer de traitements supplémentaires de la part du processus de gestion de la scène.

6.2.2. L'utilisateur entre dans la zone d'interception

Au cours de ce contexte d'interaction, l'utilisateur se trouve au sein de la zone 3D d'interception supposée de la balle. Une fois ce contexte activé, le processus d'interprétation suppose que l'utilisateur a l'intention d'intercepter volontairement la balle de tennis lancée. Le corps de l'utilisateur est positionné, les traitements s'effectuent au niveau des « Bras » (niveau « Membre »). Nous rappelons ici que, du point de vue de la collision avec la balle, nous nous intéressons aux bras de l'utilisateur de manière générale, sans se préoccuper s'il s'agit du bras gauche ou du bras droit. La Figure 5.18. illustre le contexte d'interaction présenté dans cette section, tel qu'il est modélisé au sein de notre scénario.

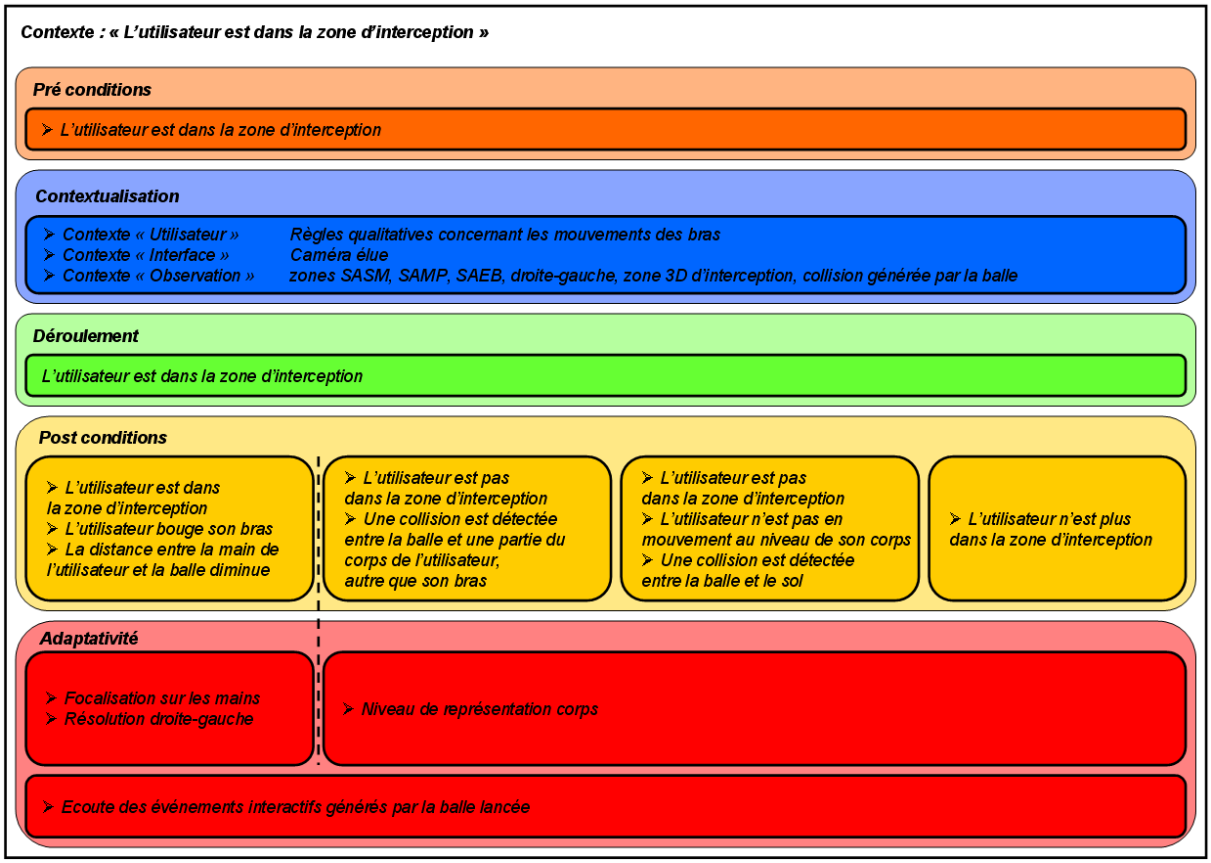


Figure 5.18. : Contexte : « L'utilisateur entre dans la zone d'interception »

Le contexte d'interaction est activé si l'utilisateur se trouve, au niveau de son corps, au sein de la zone d'interception 3D de la balle lancée. C'est la seule pré-condition nécessaire pour cette activation. Implicitement, la balle est lancée et n'a pas encore touchée le sol.

Les sous contextes nécessaires au cours de ce contexte d'interaction sont sensiblement les mêmes que ceux nécessaires au cours du contexte précédemment présenté.
Le contexte « Utilisateur » comprend un ensemble de règles qualitatives simples, concernant la position des bras par rapport au corps. Ces règles qualitatives permettent d'analyser la cohérence au cours du temps des gestuelles adoptées par l'utilisateur. Ces règles qualitatives sont supportées au cours du processus d'interprétation par les traitements dédiés à la résolution de l'ambigüité droite-gauche.
Le contexte « Interface » indique toujours la caméra permettant la détection du visage de l'utilisateur et donc la définition de la droite et de la gauche de l'image. Ces connaissances aident le processus d'interprétation à identifier et à suivre les bras de l'utilisateur.
Enfin, le contexte « Observation » fournit tout d'abord les différentes zones 2D nécessaires pour caractériser l'activité au sein de la scène. La zone SASM permet de focaliser les traitements sur l'utilisateur. La zone SAMP permet la mesure du mouvement au niveau de l'utilisateur, particulièrement au niveau de ses bras. Par le biais de cette zone, le processus d'interprétation peut également vérifier que l'utilisateur est toujours en mouvement au cours du lancer et qu'il bouge ses bras en direction de la balle. Enfin, le processus de capture de la

scène réelle et le processus d'observation se focalisent sur les bras de l'utilisateur, par l'intermédiaire des zones SAEB correspondantes. Le contexte « Observation » comprend également les informations contextuelles permettant de caractériser la zone 3D d'interception présumée de la balle, ce qui permet au processus d'interprétation de vérifier que l'utilisateur se trouve toujours au sein de celle-ci. La zone d'interception s'ajuste avec le mouvement de l'utilisateur. Enfin, les informations contextuelles comprises au sein du contexte « Observation » permettent au processus d'observation de se focaliser uniquement sur la collision générée par la balle et par les éléments composant le bras de l'utilisateur.

Le contexte d'interaction comprend 4 sets de post-conditions, le premier faisant évoluer le scénario vers le contexte d'interaction présenté dans la prochaine section, les deux suivants entrainant la fin de la situation d'interaction et le dernier faisant revenir le scénario au contexte d'interaction précédent.
Le premier set de post-conditions traduit le fait que l'utilisateur se trouve toujours dans la zone d'interception, qu'il bouge ses bras et que la distance entre la main du bras mobile et la balle diminue au cours du temps. Il concrétise donc une intention volontaire, de la part de l'utilisateur, de frapper la balle lancée avec sa main. Ce set de post-conditions termine le contexte d'interaction et fait évoluer le scénario vers le contexte « L'utilisateur tente d'intercepter la balle ».
Le deuxième set de post-conditions, une fois vérifié, indique que la balle lancée est entrée en collision avec une partie du corps de l'utilisateur autre que son bras. Le contexte d'interaction se termine, ainsi que la situation d'interaction de plus haut niveau.
Si les post-conditions du 3ème set sont vérifiées, le processus d'interprétation décide que l'utilisateur est resté immobile, et n'a donc pas essayé d'intercepter la balle, jusqu'à ce que cette dernière touche le sol. L'utilisateur est toujours dans la zone d'interception. Cela dit, aucune mesure de mouvement n'est enregistré jusqu'à la détection de la collision de la balle avec le sol. Le contexte et la situation de plus haut niveau se terminent.
Le dernier set de post-conditions indique que l'utilisateur est sorti de la zone 3D d'interception. Le gestionnaire de cohérence de la situation d'interaction de plus haut niveau permet alors au scénario un retour en arrière, au niveau du contexte « Réaction globale de l'utilisateur après le lancement de la balle ».
A noter que nous n'avons pas établi de set de post-conditions, traduisant le fait que l'utilisateur fasse tout autre chose qu'intercepter la balle ou que rester immobile. Nous sommes effectivement partis du principe que si l'utilisateur entre dans la zone d'interception, il a l'intention d'intercepter la balle, dans le but de participer au jeu.

En plus de mécanismes adaptatifs généraux, la fin du contexte génère la mise en place d'un premier set de mécanismes relatif au premier set de post-conditions, par le biais duquel le scénario évolue vers le contexte suivant, et d'un set commun aux 3 autres sets de post-conditions.
La vérification du premier set de post-conditions permet la focalisation des traitements, de la part des processus de capture de la scène réelle et d'observation de la scène, sur les mains de l'utilisateur. De plus, un ensemble de traitements adaptés est mis en place pour résoudre au mieux l'ambigüité droite-gauche (détection du visage de l'utilisateur par le biais de la caméra élue et analyse de la cohérence temporelle des poses des bras).
Si l'un des 3 autres sets de post-conditions est vérifié, les processus de capture de la scène réelle et d'observation de la scène refocalisent leurs traitements au niveau du corps global de l'utilisateur et de la scène complète.
Les mécanismes adaptatifs généraux orientent le processus d'observation vers une écoute unique des événements interactifs générés par la balle lancée.

6.2.3. L'utilisateur tente d'intercepter la balle

Le contexte d'interaction « L'utilisateur tente d'intercepter la balle » marque la fin de la situation d'interaction de plus haut niveau. Elle traduit le fait que l'utilisateur cherche volontairement à intercepter la balle avec l'une de ses mains. La Figure 5.19. illustre le contexte d'interaction défini dans cette section.

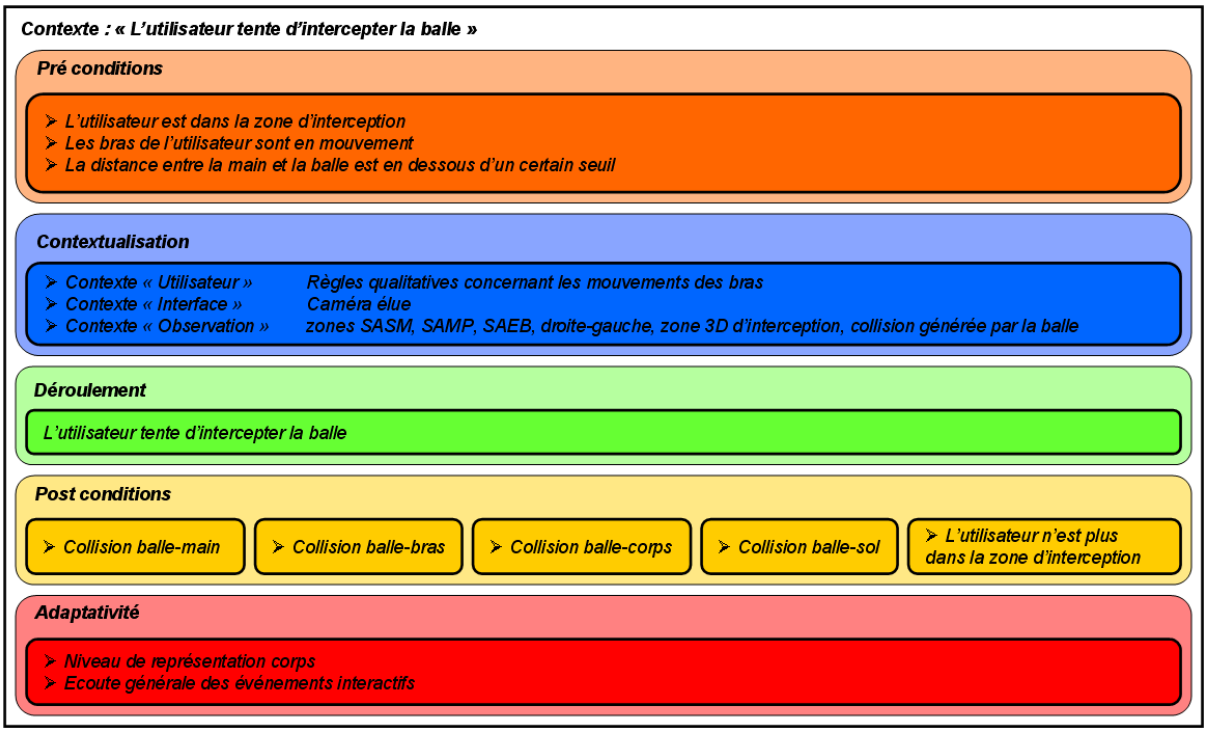


Figure 5.19. : Contexte : « L'utilisateur tente d'intercepter la balle »

Pour que ce contexte soit activé, il faut que l'utilisateur se trouve au sein de la zone présumée d'interception, que ses bras et donc ses mains soient en mouvement, et que la distance entre l'une de ses mains et la balle lancée soit en dessous d'un certain seuil.

Les informations contextuelles sont exactement les mêmes que celles nécessaires pour le contexte d'interaction « L'utilisateur entre dans la zone d'interception » (voir section précédente).

Le contexte d'interaction présenté ici comprend 5 sets de post-conditions. Un seul entraine un retour en arrière du scénario (au contexte « Réaction globale de l'utilisateur après le lancement de la balle »), les autres vérifient la détection de la balle avec un autre élément de la scène, terminant la situation d'interaction de plus haut niveau.
Si le processus d'interprétation du système détecte que l'utilisateur sort de la zone d'interception, le gestionnaire de cohérence de la situation d'interaction de plus haut niveau permet le retour en arrière du scénario.

Les autres sets sont vérifiés si la balle entre en collision avec un autre élément de la scène. Ainsi, nous distinguerons les collisions de la balle avec la main de l'utilisateur, avec son bras, avec une partie de son corps autre que son bras et avec le sol. Ces fins de contexte sont en cohérence avec les post-conditions de la situation d'interaction de plus haut niveau.

Les mécanismes d'adaptation, mis en place à la fin du contexte d'interaction, sont communs à l'ensemble des sets de post-conditions. Ils entrainent une refocalisation des traitements au niveau du corps de l'utilisateur et une écoute générale de l'ensemble des événements interactifs au sein de la scène.

6.3. Un lancer de balle, une phase de jeu, une partie

Nous allons, au cours de cette section, modéliser le scénario à un plus haut niveau sémantique, en représentant l'étape marquée par un lancer de balle, une phase de jeu et une partie.

Notre système s'adapte à l'activité observée en fonction du niveau de l'utilisateur, i.e. sa capacité à jouer au tennis, et de son intérêt pour le jeu. Le niveau et l'intérêt de l'utilisateur sont évalués au cours de l'interactivité. La Section 6.3.1. discute ces 2 notions et les définit dans le cadre de notre scénario.
C'est au cours de l'étape « lancer de balle » que les réactions de l'utilisateur sont observées et interprétées. Nous modélisons le lancer de balle comme une situation d'interaction, de plus haut niveau que la situation « Réactions de l'utilisateur face à un lancer de balle ». Nous décrivons cette situation au cours de la Section 6.3.2.
Le niveau sémantique supérieur décrit le contexte d'interaction « Phase de jeu ». Dans notre scénario, une phase de jeu comprend 5 lancers et une phase de temporisation. Le contexte d'interaction « Phase de jeu » est défini au cours de la Section 6.3.3.
Le dernier niveau sémantique décrit dans notre scénario concerne la partie en elle-même, que nous modélisons comme une situation d'interaction. Cette partie comprend 15 lancers et est donc divisée en 3 phases de jeu. La Section 6.3.4. présente notre modélisation.

6.3.1. Niveau et intérêt de l'utilisateur

Nous basons le paramétrage des mécanismes adaptatifs mis en place par le système, sur l'évaluation d'informations contextuelles de haut niveau : le niveau et l'intérêt de l'utilisateur. Le niveau de l'utilisateur donne une indication quantitative sur son aptitude à jouer au jeu. L'intérêt du joueur englobe son envie, son amusement, son découragement, etc.
Le niveau et l'intérêt de l'utilisateur sont déterminés au cours de l'interactivité, à partir d'informations contextuelles de plus bas niveau. La question se pose de savoir comment il est possible d'évaluer ces informations contextuelles. C'est ce dont nous discutons au cours des deux prochaines sections.

6.3.1.1. Niveau de l'utilisateur

Le niveau de l'utilisateur peut être évalué à partir de nombreuses informations contextuelles bas niveau. Dans le cadre de notre application, nous pouvons évaluer le niveau à partir :

- du nombre d'interceptions/de loupés au cours d'une phase de jeu/d'une partie ;
- du nombre de lancers loupés qu'il a fallu avant que l'utilisateur n'intercepte une balle, au cours d'une phase de jeu/d'une partie ;
- du nombre d'interceptions successives/de loupés successifs au cours d'une phase de jeu/d'une partie ;
- et bien d'autres !

De manière à valider rapidement notre approche, nous avons décidé de ne pas rendre le niveau de l'utilisateur trop complexe à construire, au cours de l'interactivité.

Le niveau de l'utilisateur est évalué à l'issue de chaque phase de jeu, impliquant alors une adaptation du système influant sur son comportement au cours de la phase suivante. Il est compris dans un intervalle de valeurs numériques, allant de -2 à 2, -2 traduisant un mauvais niveau, 2 un bon.
Le niveau initial de l'utilisateur est 0. A l'issue de chaque phase, si l'utilisateur a intercepté au moins 3 balles au cours de cette phase, son niveau augmente de 1 unité. S'il en a loupé au moins 3, son niveau diminue de 1 unité.

De manière générale, la réponse interactive du système, vis-à-vis de la gestuelle utilisateur, est immédiate et découle de l'exécution de l'application, par le biais des différents moteurs employés. Autrement dit, le rendu de la scène immersive et la simulation physique des objets au sein de celle-ci sont gérés de manière standard par le moteur de jeu. Cependant, chaque niveau de l'utilisateur se traduit par une adaptation visuellement remarquable, au niveau de la réponse interactive du système. Une fois établies, ces mesures adaptatives sont fixes (dans nos travaux) au cours d'une phase de jeu.

Niveau 0
C'est le niveau à partir duquel l'utilisateur commence le jeu. Il est possible, grâce au moteur de physique, de calculer et d'afficher les boites 3D englobant les objets physiques, associés aux balles et aux mains. A ce niveau, les boites englobant les mains de l'avatar et la balle lancée sont affichées. Ainsi, l'utilisateur a une meilleure idée de la portée de ses coups.

Niveau -1
En plus d'afficher les boites englobant les mains de l'avatar et la balle lancée, le gestionnaire de la scène 3D permet l'affichage de la trajectoire supposée de la balle avant son lancer. Les lancers de balle sont contrôlés par le gestionnaire de la scène. La balle, une fois lancée, effectue une simple parabole. Il est tout à fait possible, à partir des paramètres connus du lancer, de déterminer la trajectoire de la balle. L'affichage de la trajectoire de la balle permet à l'utilisateur de se positionner en avance et de mieux se préparer à intercepter la balle, ayant une idée de l'endroit où elle va tomber. A noter qu'il est également possible, à partir de cette trajectoire, d'afficher la zone 3D d'interception supposée de la balle lancée.

Niveau -2
Les boites englobantes et la trajectoire de la balle sont affichées. En plus de cela, la balle est ralentie au cours de son lancer. Il est effectivement possible de calculer l'équation paramétrique de la courbe suivie par la balle au cours du lancer. Cette équation est fonction du temps, cette valeur de temps étant contrôlable par le gestionnaire de la scène. Une variation du temps implique une variation de la vitesse de la balle. A ce niveau, la vitesse de la balle est donc diminuée, de manière à laisser le temps à l'utilisateur de mieux se préparer pour l'intercepter.

Niveau 1
A ce niveau, il n'y a aucune adaptation visuelle de la scène. Le lancer de balle se déroule de manière standard vis-à-vis du moteur de l'application.

Niveau 2
Par le biais de l'équation paramétrique de la trajectoire de la balle, la vitesse de cette dernière est augmentée. Les balles vont plus vite au cours de cette phase de jeu que pendant la précédente.

6.3.1.2. Intérêt de l'utilisateur

Plusieurs informations contextuelles bas niveau permettent le calcul de l'intérêt de l'utilisateur pour le jeu :

- Nombre/Durée d'immobilité de la part de l'utilisateur, au cours d'une phase de jeu/d'une partie ;
- Nombre/Durée d'immobilité de la part de l'utilisateur, sur plusieurs lancers successifs, au cours d'une phase de jeu/d'une partie ;
- Niveau de l'utilisateur. En effet, suivant son niveau, l'utilisateur peut montrer des signes de découragement (si son niveau est bas) ou de lassitude (si son niveau est haut) ;
- Et bien d'autres !

A l'instar du niveau de l'utilisateur, dans le cadre de notre scénario, nous n'utilisons pas l'ensemble des informations contextuelles bas niveau, qui nous permettraient de déterminer l'intérêt de l'utilisateur.

Dans nos travaux, l'intérêt de l'utilisateur est évalué à l'issue de chaque lancer. L'intérêt de l'utilisateur peut prendre la valeur 1, dénotant un intérêt de la part de l'utilisateur pour le jeu, et la valeur 0, traduisant un intérêt nul pour le jeu.
L'intérêt de l'utilisateur est initialement à 1. Sur l'ensemble de la partie, si 3 immobilités successives (i.e. l'utilisateur ne bouge pas au cours de 3 lancers successifs) sont enregistrées, alors l'intérêt de l'utilisateur devient nul. L'ensemble du jeu se met alors en pause. Cette pause se termine sur l'intervention d'un superviseur, au niveau du système, après ce dernier ait demandé à l'utilisateur s'il voulait continuer la partie. L'intérêt de l'utilisateur repasse à 1, à l'issue de cette pause.

6.3.2. Un lancer de balle

Dans le cadre de notre scénario, nous modélisons « un lancer de balle » comme une situation d'interaction (voir Figure 5.20.).
Le lancer de balle est l'étape de plus bas niveau de notre scénario. Au cours de cette situation d'interaction, une balle de tennis est lancée à l'utilisateur, projetée par un canon à balles se trouvant en face de lui. Les balles sont lancées à droite et gauche de l'utilisateur, de manière alternée et selon des paramètres de lancer prédéfinis. Quoiqu'il en soit, en accord avec nos hypothèses, l'utilisateur ne peut pas intercepter une balle s'il se trouve au centre de la scène. Il est donc obligé de se déplacer de celui-ci.

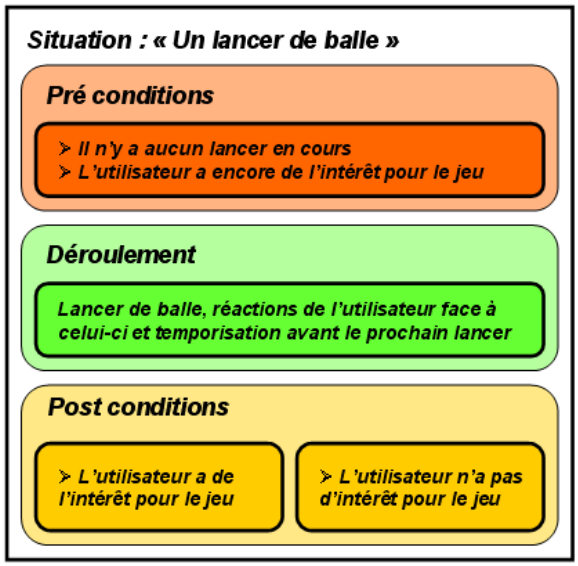


Figure 5.20. : Situation : « Un lancer de balle »

La situation d'interaction « Un lancer de balle » est relativement simple. Pour être activée, deux pré-conditions doivent être remplies : il ne doit pas y avoir de lancer en cours (soit aucune balle n'a encore été lancée au cours de la partie, soit la collision de la dernière balle lancée a bien été détectée) et l'utilisateur doit avoir encore de l'intérêt pour le jeu (intérêt = 1). La situation se termine, une fois que l'intérêt de l'utilisateur pour le jeu a été évalué. Soit l'utilisateur est encore intéressé, le jeu continue alors, soit il n'est plus intéressé, le jeu se met alors en pause.

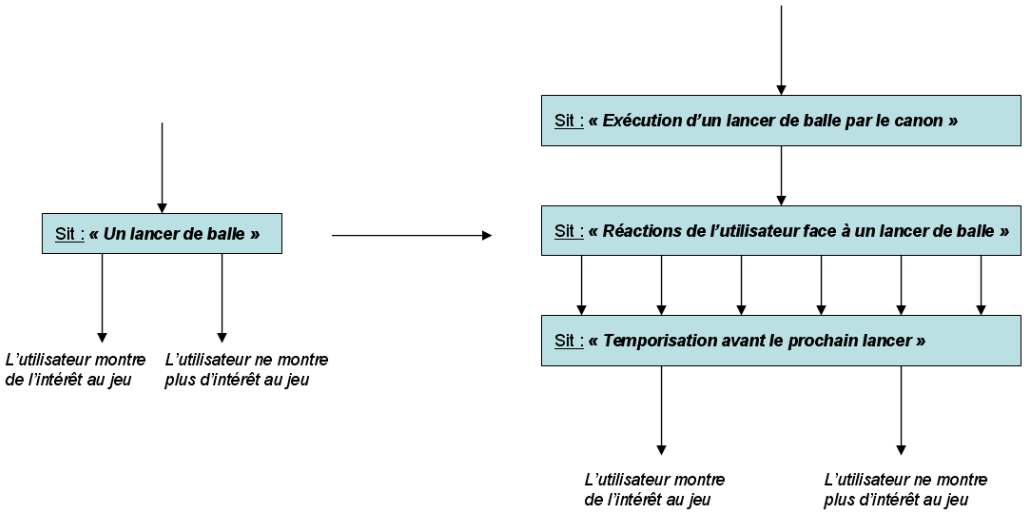


Figure 5.21. : Décomposition de la situation « Un lancer de balle »

Le déroulement de la situation d'interaction « Un lancer de balle » englobe trois situations d'interaction de plus bas niveau : la situation « Exécution d'un lancer de balle par le canon », la situation « Réactions de l'utilisateur face à un lancer de balle », décrite au cours de la Section 6.2., et la situation « Temporisation avant le prochain lancer » (voir Figure 5.21.). A partir de cette décomposition peut être établi l'agenda associé, illustré par la Figure 5.22.

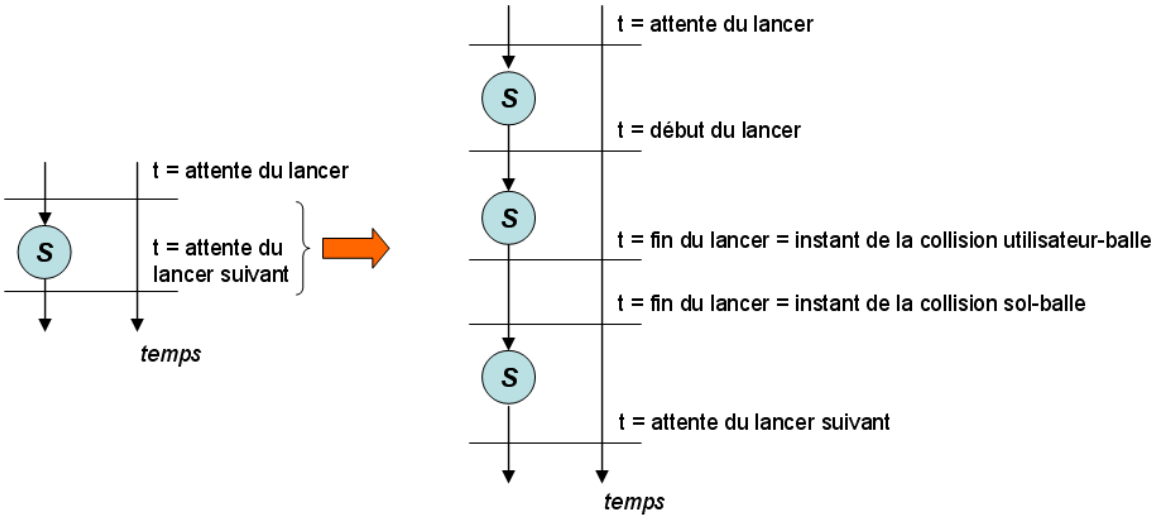


Figure 5.22. : Agenda associé à la situation « Un lancer de balle »

La Figure 5.23. donne notre modélisation de la situation d'interaction « Exécution d'un lancer de balle par le canon ». Cette simple situation ne comporte pas de pré-conditions. Dans le cadre de notre application, le canon a été implémenté de telle manière qu'il ne s'arrête d'envoyer des balles au cours de notre scénario. Il ne peut cependant pas envoyer de balle, si un lancer est en cours. Ainsi, dès la présence d'un temps mort dans notre scénario, si aucun lancer n'est en cours et si le scénario le permet, comme c'est le cas avec cette situation, le canon envoie une balle à l'utilisateur.
Nous rappelons que le canon envoie les balles à l'utilisateur de manière alternée (droite-gauche) et selon des paramètres de lancer prédéfinis.

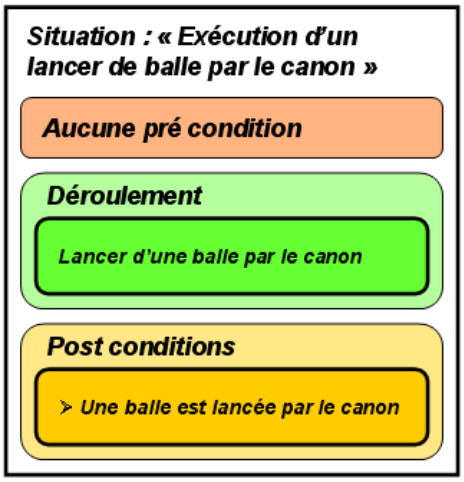


Figure 5.23. : Situation : « Exécution d'un lancer de balle par le canon »

Comme nous l'avons vu au cours de la Section 6.2., la situation « Réactions de l'utilisateur face à un lancer de balle » peut se terminer de 6 manières différentes :

1. La balle touche le sol, l'utilisateur ne l'a pas interceptée mais il a essayé.
2. La balle touche le sol, l'utilisateur ne l'a pas interceptée et n'a essayé. Il a cependant été mobile au cours du lancer.
3. La balle touche le sol, l'utilisateur ne l'a pas interceptée et n'a essayé. Il a été immobile au cours du lancer.
4. L'utilisateur a intercepté la balle de la main.
5. L'utilisateur a intercepté la balle avec le bras.
6. L'utilisateur a intercepté, volontairement ou involontairement, la balle avec une partie de son corps, autre que son bras (donc que sa main).

Ces différentes fins laissent un grand choix de possibilités, quant à l'écriture de notre scénario. Dans le cadre de nos travaux, cette variété de solutions n'est pas forcément nécessaire. De futurs travaux pourront cependant tout à fait l'exploiter. De plus, comme nous l'avons mentionné précédemment, il serait tout à fait possible de modéliser ce scénario d'une toute autre façon. Nous avons choisi cette représentation, de manière à valider au plus vite notre approche.

A la suite de la situation « Réactions de l'utilisateur face à un lancer de balle », nous avons modélisé une situation d'interaction appelée « Temporisation avant le prochain lancer ». La modélisation de cette situation a plusieurs objectifs.
Du point de vue scénaristique, tout d'abord, cette situation permet de marquer temporellement la fin d'un lancer, laissant ainsi le temps à l'utilisateur de se replacer et se repréparer. Au cours de cette situation, un *timer* spécifique (un chronomètre) s'active. La situation peut potentiellement se terminer une fois que ce *timer* a atteint une durée (en secondes ou en *frames*) particulière.
De plus, la mise en place de cette situation permet la concaténation des différentes sorties de la situation « Réactions de l'utilisateur face à un lancer de balle ». En effet, dans la mesure où nous n'utilisons pas totalement les différentes sorties possibles de cette dernière, la situation

« Temporisation avant le prochain lancer de balle » permet de regrouper les sorties résultant en une même évolution du scénario.
Au cours de cette situation, plusieurs informations contextuelles, nécessaires à l'évaluation du niveau et de l'intérêt de l'utilisateur, sont également mises à jour : nombre d'interceptions sur la phase de jeu courante/la partie ; nombre de loupés sur la phase de jeu courante/la partie ; nombre actuel d'immobilités successives. Ainsi, en prenant en compte les différentes fins possibles de la situation « Réactions de l'utilisateur face à un lancer de balle », nous pouvons établir les relations suivantes :

1. La balle touche le sol, l'utilisateur ne l'a pas interceptée mais il a essayé.
   - ⇨ *Nombre de loupés (sur phase et sur partie) + 1*
   - ⇨ *Nombre d'immobilités successives = 0*
2. La balle touche le sol, l'utilisateur ne l'a pas interceptée et n'a essayé. Il a cependant été mobile au cours du lancer.
   - ⇨ *Nombre de loupés (sur phase et sur partie) + 1*
   - ⇨ *Nombre d'immobilités successives + 1*
3. La balle touche le sol, l'utilisateur ne l'a pas interceptée et n'a essayé. Il a été immobile au cours du lancer.
   - ⇨ *Nombre de loupés (sur phase et sur partie) + 1*
   - ⇨ *Nombre d'immobilités successives + 1*
4. L'utilisateur a intercepté la balle de la main.
   - ⇨ *Nombre d'interceptions (sur phase et sur partie) + 1*
   - ⇨ *Nombre d'immobilités successives = 0*
5. L'utilisateur a intercepté la balle avec le bras.
   - ⇨ *Nombre d'interceptions (sur phase et sur partie) + 1*
   - ⇨ *Nombre d'immobilités successives = 0*
6. L'utilisateur a intercepté, volontairement ou involontairement, la balle avec une partie de son corps, autre que son bras (donc que sa main).
   - ⇨ *Nombre de loupés (sur phase et sur partie) + 1*
   - ⇨ *Nombre d'immobilités successives + 1*

Concernant la deuxième fin possible, même si l'utilisateur a été mobile au cours du lancer, nous incrémentons la valeur du nombre d'immobilités successives. En effet, cette fin traduit un intéressement nul de la part du joueur et l'intérêt du joueur n'est mis à jour qu'à partir de la valeur du nombre d'immobilités successives. De plus, les nombres d'interceptions et de loupés sont calculés sur la phase de jeu en cours et sur la partie de manière générale. Les valeurs de ces nombres permettront de mettre à jour le niveau de l'utilisateur à l'issue de la phase de jeu.
Enfin, c'est au cours du déroulement de cette situation que l'intérêt de l'utilisateur est finalement évalué. Dans notre scénario, le fait que l'utilisateur ait, ou non, de l'intérêt pour le jeu définit la fin de la situation d'interaction de plus haut niveau « Un lancer de balle ». L'intérêt de l'utilisateur est évalué à partir du nombre d'immobilités successives. Si le nombre d'immobilités successives est égal à 3, alors l'intérêt de l'utilisateur devient nul.
La Figure 5.24. modélise la situation « Temporisation avant le prochain lancer » et résume nos propos tenus au cours de cette section.

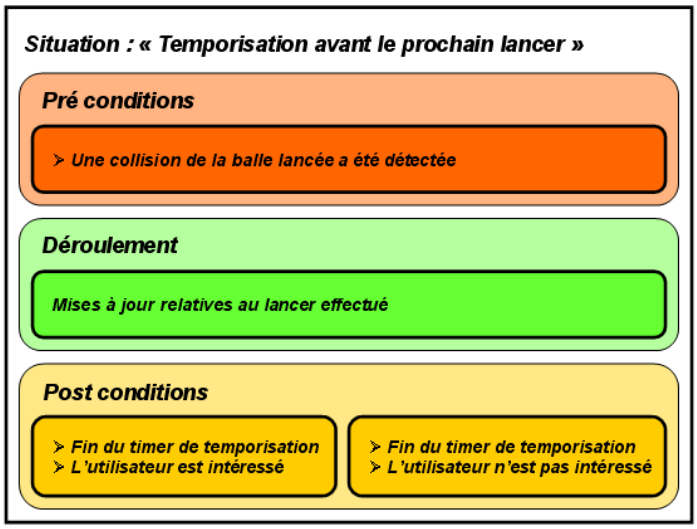
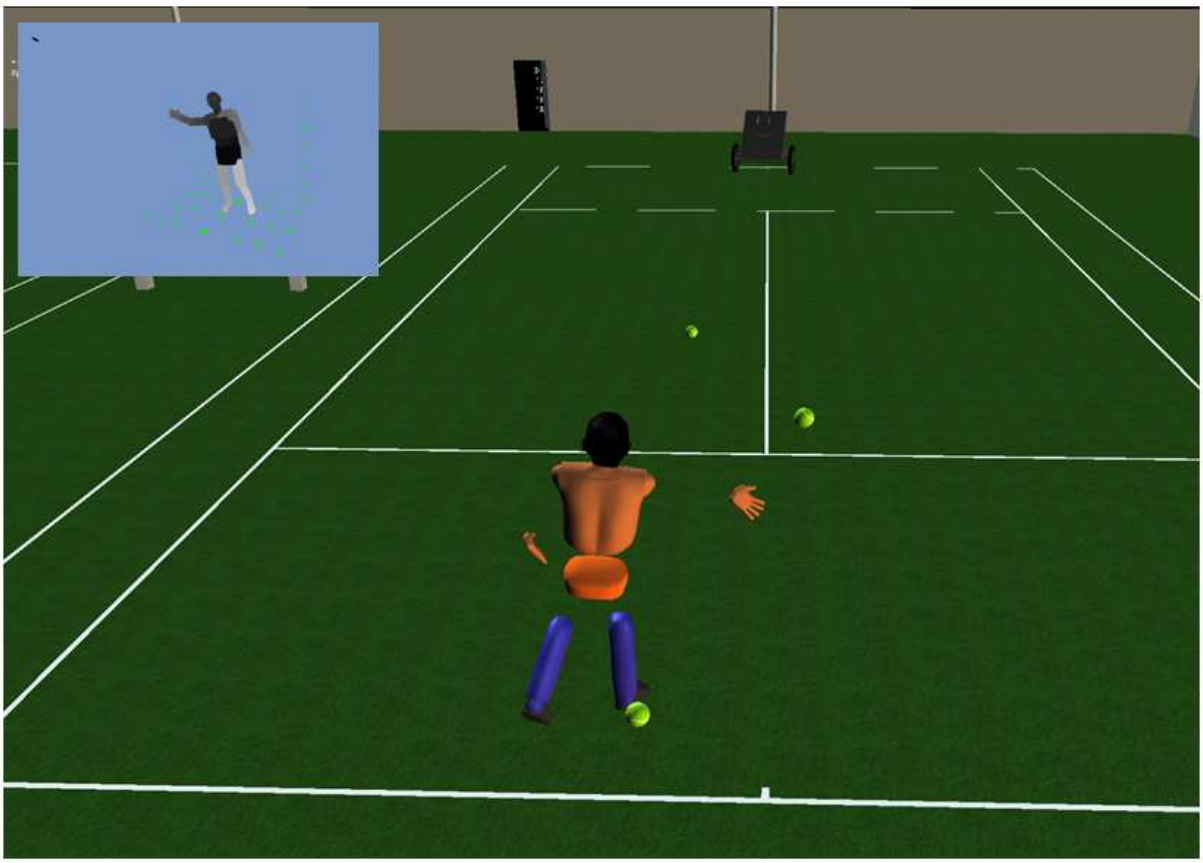


Figure 5.24. : Situation : « Temporisation avant le prochain lancer »

La Figure 5.25. illustre notre application au cours d'un lancer de balle.



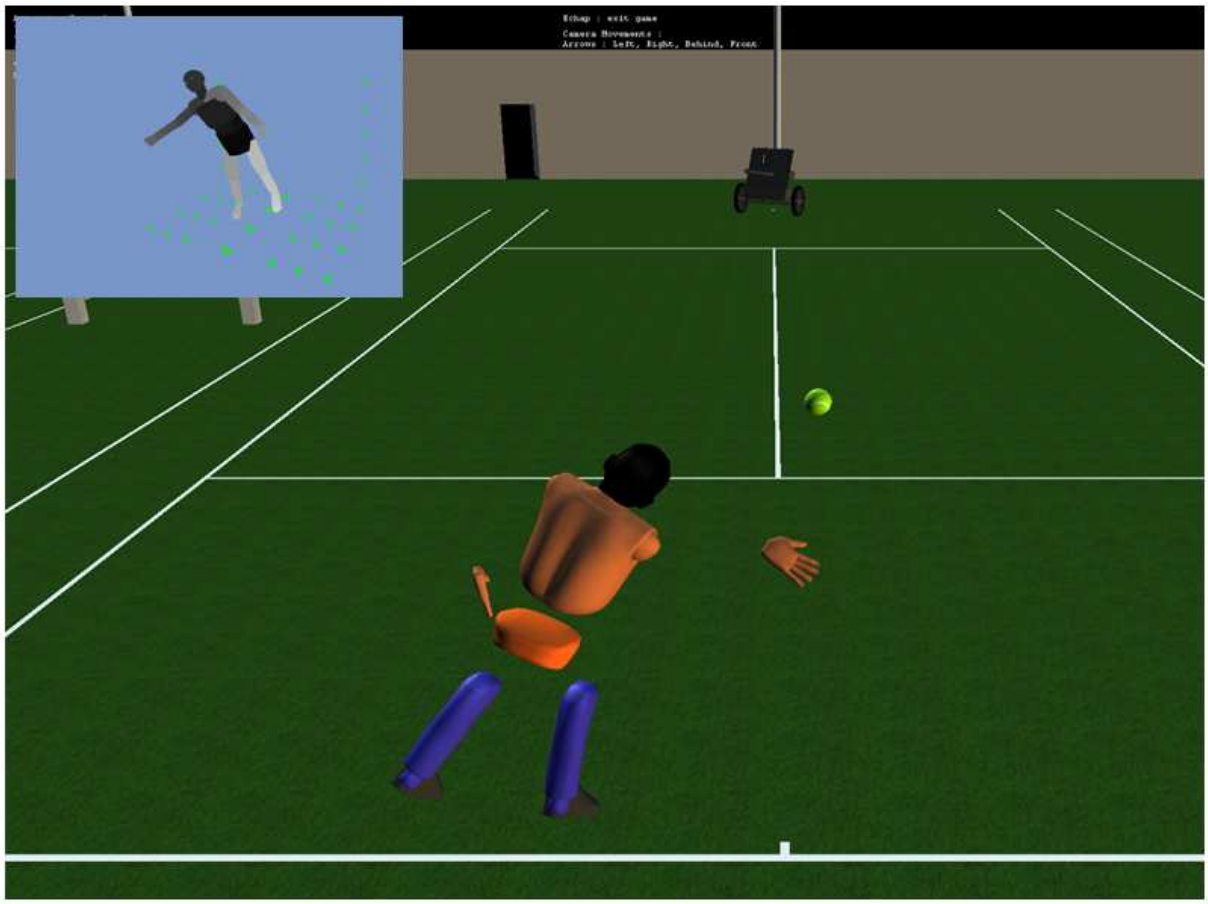


Figure 5.25. : Un lancer de balle

6.3.3. Une phase de jeu

Nous modélisons « une phase de jeu » comme un contexte d'interaction (voir Figure 5.26.). Une phase de jeu est une succession de 5 situations de type « un lancer de balle ».

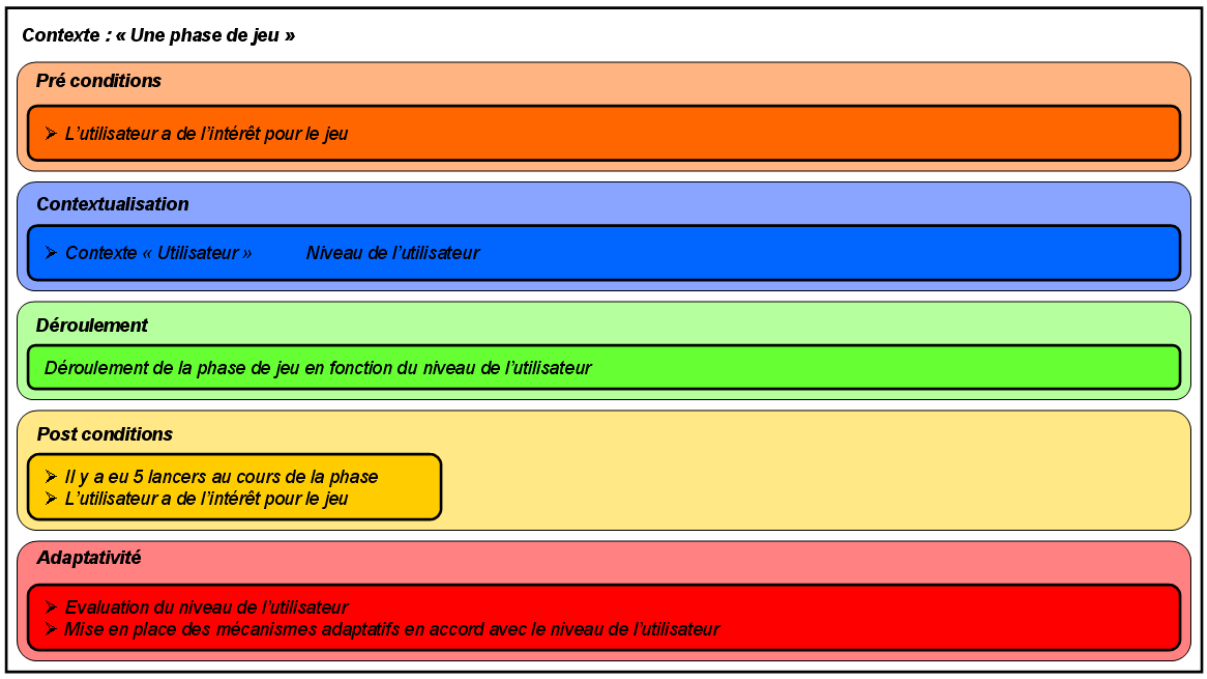


Figure 5.26. : Contexte : « Une phase de jeu »

Le contexte d'interaction est activé si l'utilisateur est encore intéressé par le jeu. La seule pré-condition requise pour le démarrage de ce contexte est donc que l'intérêt de l'utilisateur soit égal à 1.

La seule information contextuelle requise pour le bon déroulement du contexte « une phase de jeu » est le niveau de l'utilisateur. Cette information contextuelle appartient au contexte « Utilisateur ».

Le déroulement du contexte d'interaction se déroule en accord avec le niveau de l'utilisateur, comme nous l'avons présenté au cours de la Section 6.3.1.1. La Figure 5.27. illustre la décomposition de ce déroulement par le biais de situations de plus bas niveau.

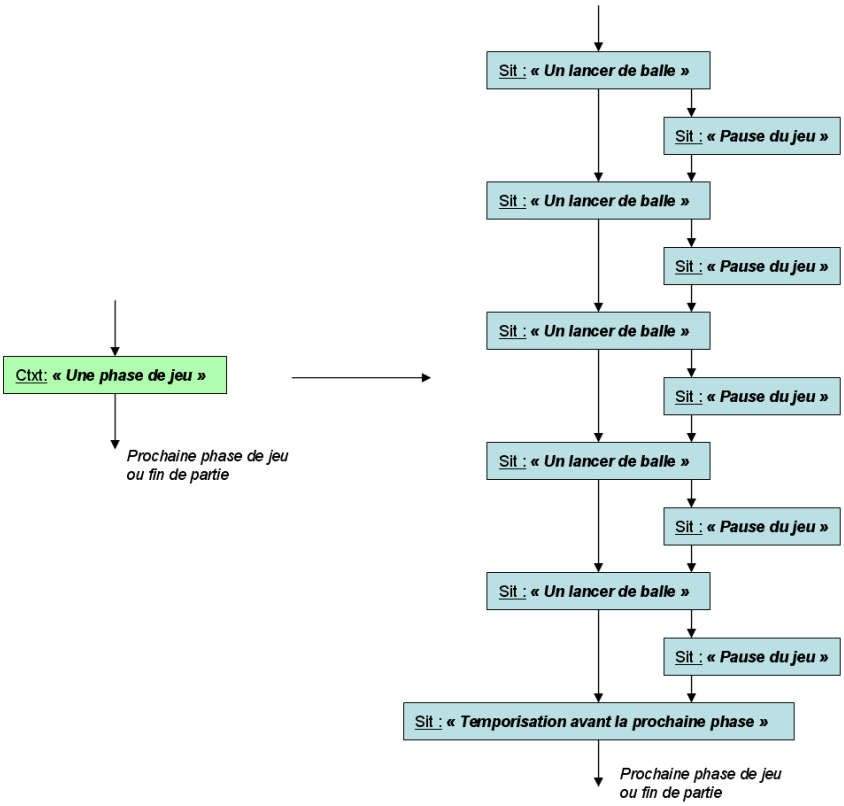


Figure 5.27. : Décomposition du contexte « Une phase de jeu »

Au cours de ce déroulement se succèdent 5 situations de type « Un lancer de balle » (voir section précédente).
5 situations de type « Pause de jeu » peuvent également avoir lieu, en accord avec l'évolution de l'intérêt de l'utilisateur. Après chaque lancer, l'intérêt de l'utilisateur est évalué. S'il reste immobile au cours de 3 lancers consécutifs (au niveau d'une partie), il est considéré comme n'ayant plus d'intérêt pour le jeu. Ce dernier se met alors automatiquement en état de pause. Le système atteint alors une intervention de la part d'un superviseur, qui est chargé d'interroger l'utilisateur sur sa volonté de continuer le jeu. Si le superviseur décide que la pause est finie, le jeu reprend son évolution normale et l'intérêt de l'utilisateur repasse à 1.
Enfin, le contexte se termine après le déroulement d'une dernière situation d'interaction appelée « Temporisation avant la prochaine phase », dont le rôle est similaire à celui de la situation « Temporisation avant le prochain lancer » (voir section précédente). Au cours de cette situation est donc déclenché un *timer*. Elle se termine une fois que le *timer* a atteint une durée spécifique, laissant ainsi le temps à l'utilisateur de se replacer. Cette situation permet également la concaténation des différentes fins possibles de la situation précédente. Enfin, les nombres de lancers interceptés sur la phase et la partie sont mis à jour, ainsi que les nombres de loupés sur la phase et la partie.

Le contexte « Une phase de jeu » ne comprend qu'un set de 2 post-conditions. Ainsi, le contexte d'interaction est terminé si il ya eu 5 lancers de balles et si l'utilisateur a de l'intérêt pour le jeu.

Le contexte d'interaction ne comprend également qu'un set de mécanismes adaptatifs.

En accord avec notre scénario (voir Section 6.3.1.1.), le niveau de l'utilisateur est tout d'abord évalué. Nous rappelons que le niveau de l'utilisateur est fonction des nombres d'interceptions et de loupés sur la phase. Si l'utilisateur intercepte au moins 3 balles, son niveau augmente de 1 unité. S'il en loupe au moins 3, son niveau diminue de 1 unité.
L'évaluation du niveau de l'utilisateur permet ensuite la mise en place des mécanismes adaptatifs, au niveau de la réponse interactive du système. Ces modifications au niveau de la réponse visuelle du système sont décrites au cours de la Section 6.3.1.1. Elles seront perceptibles par l'utilisateur au cours de la phase de jeu suivante. Il est à noter que les activations des différentes mesures adaptatives, qui ne sont finalement que des modifications d'affichage, résultent une modification des variables associées, comprises au sein du vecteur d'états, représentant le système d'information global (voir Section 6. du chapitre relatif à la gestion de contexte).
Une fois les mécanismes adaptatifs exécutés, en accord avec le niveau de l'utilisateur, les nombres d'interceptions et de loupés sur une phase sont réinitialisés à 0.

La Figure 5.28. présente l'agenda associé au contexte « une phase de jeu ». Au sein de cet agenda, nous ne représentons pas les pauses dues à un intérêt nul de la part de l'utilisateur. Ce sont juste les valeurs des instants qui sont modifiés. Au cours de la pause, le jeu s'arrête et les durées des différentes pauses n'influent pas sur l'évolution du scénario.

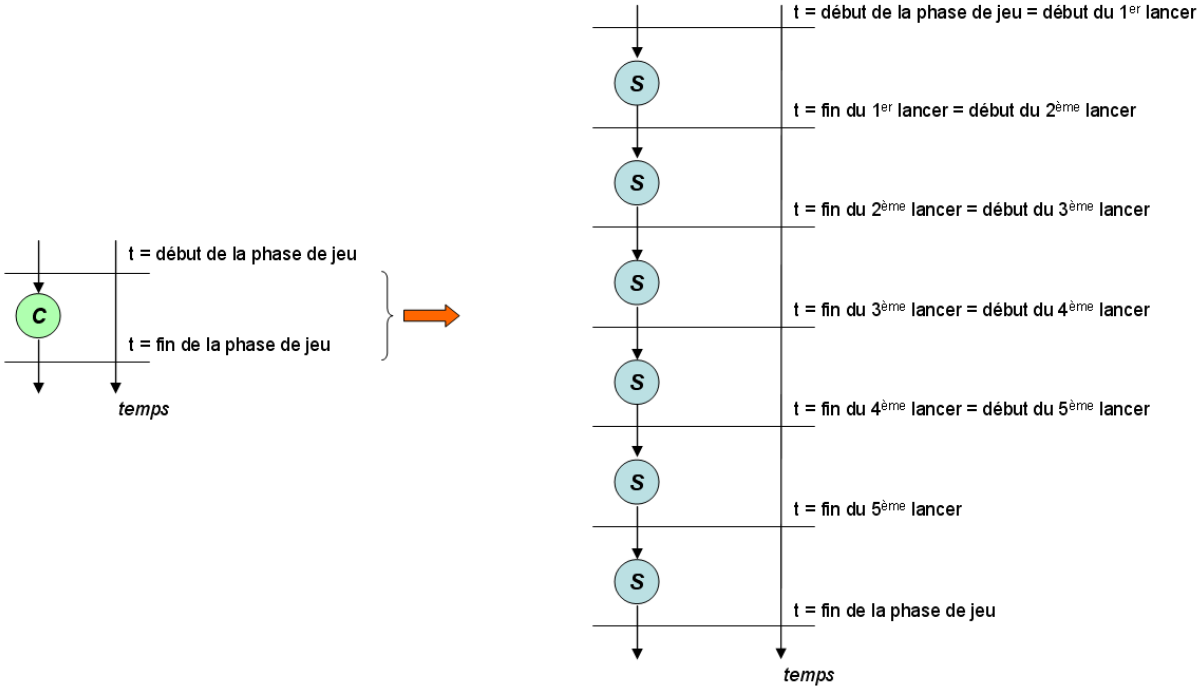


Figure 5.28. : Agenda associé au contexte « Une phase de jeu »

6.3.4. Une partie

« Une partie de jeu » est une situation, comprenant 3 contextes d'interaction de type « Une phase de jeu » (donc 5 situations d'interaction de type « Un lancer de balle »). Cette situation ne nécessite aucune pré ou post-condition.

La Figure 5.29. donne la modélisation de cette situation, la Figure 5.30. sa décomposition et la Figure 5.31. son agenda.

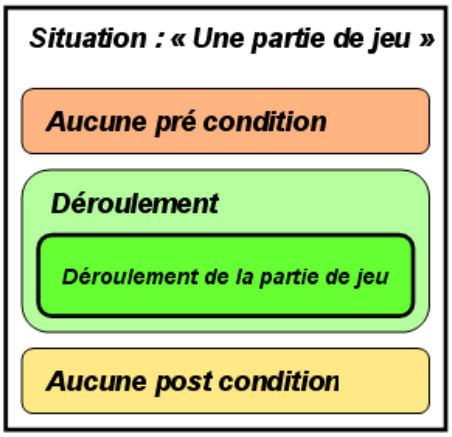


Figure 5.29. : Situation : « Une phase de jeu »

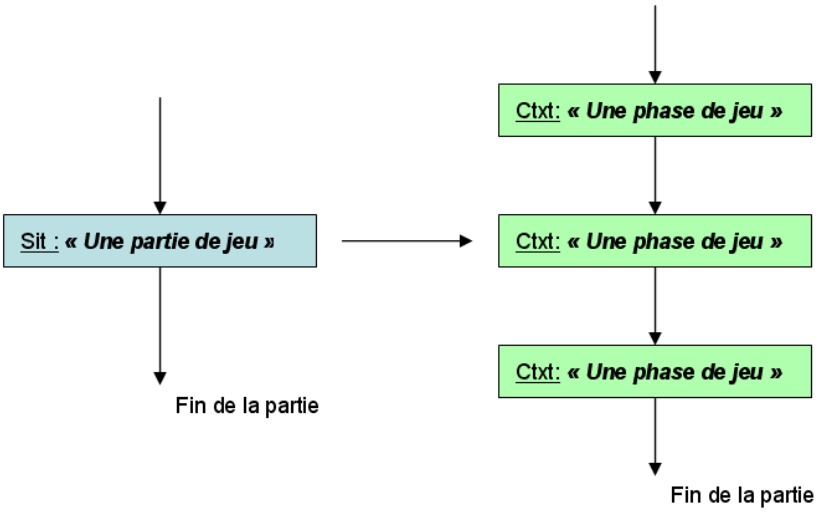


Figure 5.30. : Décomposition de la situation « Une phase de jeu »

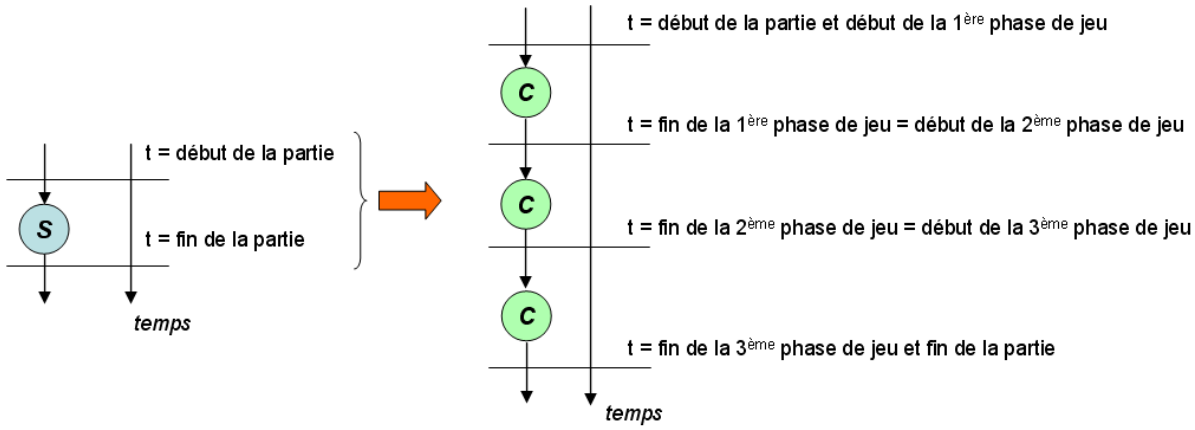


Figure 5.31. : Agenda associé à la situation « Une phase de jeu »

6.4. Présence de spectateurs au sein de la scène réelle

Nous proposons, au cours de cette section, plusieurs solutions pour faire face à la présence de spectateurs au sein de la scène réelle. Ces solutions se situent au niveau du processus de capture de la scène réelle.
Dans la mesure où il ne s'agit pas de l'interaction à proprement parler de l'utilisateur avec le système, nous ne modéliserons pas de scénarios par le biais de situations et de contextes d'observations.

Nous allons donc tout d'abord énumérer les différentes informations contextuelles, pertinentes pour le processus de capture de la scène réelle, qu'il peut utiliser de manière à paramétrer des mécanismes adaptatifs particuliers (voir Section 6.4.1.).
Puis, nous proposerons plusieurs solutions pour faire face à la présence de spectateurs dans le fond de la scène réelle, en prenant en compte les informations contextuelles précédemment identifiées (voir Section 6.4.2.).

Cette section est fortement corrélée avec la Section 3.5.1.3. du chapitre traitant des systèmes de capture de gestuelles, qui présente notre contribution en matière de modélisation dynamique de fond de scène.

6.4.1. Informations contextuelles

Nous identifions, dans cette, section, les différentes informations contextuelles pertinentes pour le processus de capture de la scène réelle, pour que ce dernier puisse faire face à la présence de spectateurs dans le fond de la scène.

Le contexte « Scène réelle » comprend les emplacements 3D, au sein de la scène réelle, où sont supposés se trouver l'utilisateur et les spectateurs. Sur certains points de vue, notamment sur celui défini par la caméra élue (contexte « Interface »), ces emplacements ne se recoupent pas. Ainsi, il est possible de traiter spécifiquement les emplacements où se trouvent les spectateurs, sans craindre de générer des artefacts dans la zone où est sensé évoluer l'utilisateur.

Le contexte « Interface » fournit, d'une part, l'identité de la caméra se trouvant la plus face de l'utilisateur, et dont le point englobe l'ensemble de la scène réelle. Cette information contextuelle peut permettre, au processus de capture de la scène réelle, de mettre en place des traitements spécifiques et plus précis, dans la mesure où ce point de vue présente moins d'ambigüités vis-à-vis du positionnement des spectateurs et de l'utilisateur.
D'autre part, le contexte « Interface » comprend les différentes matrices caméras, permettant de passer d'une coordonnée 3D, relative à la scène réelle, à une coordonnée 2D, relative au plan image de la caméra considérée, et réciproquement. Ainsi, en connaissant l'emplacement 3D supposé des spectateurs dans un point de vue donné (contexte « Scène réelle »), il est possible de projeter ce dernier sur le plan image de la caméra associée à ce point de vue. Ainsi, le processus de capture de la scène réelle peut mettre en œuvre des traitements spécifiques pour traiter ces zones 2D.

Le contexte « Observation » regroupe un ensemble de zones 2D permettant de dissocier l'utilisateur des spectateurs. Ces zones 2D peuvent être déterminées sur l'ensemble des points de vue, ou sur un point de vue particulier (contexte « Interface »).
Ainsi, la zone complémentaire à la zone SASM peut être traitée spécifiquement lors de la capture de la scène réelle, dans la mesure où elle ne comprend *a priori* pas l'utilisateur. La zone SAMP, fusionnée avec la zone précédente, peut donner également plusieurs indications sur les emplacements où d'éventuels spectateurs mobiles se trouveraient. Enfin, en projetant l'emplacement 3D supposé des spectateurs (contexte « Scène réelle »), les zones SAO peuvent être déterminées, permettant la focalisation de différents traitements au sein de celles-ci.

6.4.2. Mesures adaptatives

Nous proposons ici plusieurs solutions adaptatives, utilisant les informations contextuelles précédemment identifiées, permettant la gestion des spectateurs, au niveau du processus de capture de la scène réelle.

Dans le cadre de notre système, il existe une seule caméra, se trouvant en face de l'utilisateur et englobant complètement la scène (contexte « Interface »). Les autres points de vue présentent plusieurs occultations, ne permettant pas de dissocier rapidement l'utilisateur des spectateurs se trouvant derrière lui. C'est pourquoi la plupart des traitements, dont le but est de gérer la présence des spectateurs, peuvent s'effectuer uniquement sur ce point de vue donné (ou de manière plus importante).

Le processus de mise à jour dynamique du fond de la scène constitue un point de départ important, quant à la gestion de la présence de spectateurs. Ce processus a la capacité de mettre à jour plus ou moins rapidement le fond, en fonction du mouvement mesuré au sein de la scène. Ainsi, s'il mesure du mouvement dans le fond de scène, et qui n'est pas lié à l'utilisateur, il pourra alors mettre à jour le fond plus rapidement. S'il ne mesure pas de mouvement, il pourra alors mettre à jour le fond de manière plus progressive. Dans les deux cas, le processus de capture de la scène réelle, en utilisant ce fond, peut extraire la silhouette de l'utilisateur efficacement. Les informations contextuelles pertinentes sont donc l'ensemble des zones 2D comprises dans le contexte « Observation ».

Les zones 2D du contexte « Observation » peuvent également permettre l'acquisition partielle de la scène. Cette mesure est possible et valide si et seulement si l'emplacement de l'utilisateur ne recoupe pas les emplacements des spectateurs.

Dans tous les cas, si la scène présente la moindre ambigüité, le processus de capture peut repasser sur une vision globale et capturer l'ensemble de l'activité au sein de la scène réelle.

7. Conclusion

Nous avons présenté, dans ce chapitre, nos contributions, visant à développer un système interactif complet, sur lequel s'exécute une application d'entrainement au tennis, à exécution adaptative.
Notre système modélise tout d'abord la scène virtuelle au sein de laquelle l'utilisateur évolue. Cette scène est double et comprend la modélisation de la scène réelle capturée et celle d'une scène 3D immersive, de manière adéquate vis-à-vis du scénario de l'application. Par la suite, à partir des connaissances extraites de la scène virtuelle observée, le système caractérise le contexte d'interaction au sein duquel l'utilisateur évolue. Le système interprète ensuite l'activité au sein de la scène. Les processus de caractérisation et d'interprétation s'effectuent sous le support du scénario de l'application, à partir duquel sont extraits les différents contextes d'interaction attendus. Enfin, le système met en place ses réponses interactives et adaptatives, en accord avec la scène interprétée. La mise en place de ces réponses est également supportée par les informations extraites du scénario de l'application.

La première section a présenté le scénario de notre application. Notre application est basique et son scénario est simple, facilement analysable et développable. L'utilisateur doit effectuer des gestuelles simples, faciles à apprendre et à exécuter. Cependant, ce scénario nous a permis de valider notre approche, rapidement, et non forcément de manière complexe. Il a permis de mettre en avant les différentes problématiques, auxquelles nous avons fait face et d'illustrer l'ensemble de nos solutions de manières efficace et correcte.

La seconde section de ce chapitre a énuméré les différentes connaissances, extraites de la scène virtuelle, à partir desquelles le contexte d'interaction dans lequel évolue l'utilisateur, et précise leurs provenances : connaissances introduites *a priori*, lors de la phase de conception, lors de phases d'expérimentation et de calibrage, ou construites au cours de l'interactivité. Il est bien sûr tout fait possible d'extraire plus d'informations, également plus complexes, de la scène virtuelle. Ces informations et leur extraction feront l'objet de travaux futurs.
Ce besoin de nouvelles connaissances est nécessaire dans la mesure où nous travaillons à partir de points de vue 2D. Dans nos travaux, nous ne considérons que les silhouettes 2D de l'utilisateur. Ces silhouettes constituent le point de départ de tout le processus interactif, jusqu'à la mise en place des réponses de la part du système. La mauvaise qualité au niveau de l'extraction des silhouettes se répercute donc sur la qualité du reste de l'interactivité. Pour compléter l'information obtenue par le biais d'une silhouette, il serait tout à fait possible de d'autres informations, telles que des informations de couleur (travail sur la couleur de la peau par exemple) ou de texture (acquisition d'un modèle d'apparence par exemple).
De plus, notre application, de par son scénario, implique une scène virtuelle basique et comportant peu d'éléments. En complexifiant le scénario et la nature de la scène virtuelle, il serait donc possible de prendre en compte de nouvelles informations, plus complexes, nous permettant d'ajouter de nouvelles connaissances quant à la caractérisation efficace du contexte d'interaction observé.

La troisième section a présenté notre caractérisation d'un contexte d'interaction, à partir des différentes connaissances présentes au sein du système au cours de l'interactivité. Nous avons détaillé les différents sous-contextes, présentés au cours de la Section 6.4. du chapitre relatif à la gestion du contexte : contextes « Utilisateur », « Système/Application », « Scène réelle », « Interface », « Observation » et « Temps ». Nous nous sommes particulièrement arrêtés sur la caractérisation de la gestuelle utilisateur, comprise au sein du contexte « Utilisateur ».
Dans ces travaux, nous avons pris le parti de considérer qu'il était impossible de tout caractériser, et à tout moment. Dans notre contexte d'étude, nous avons donc cherché à caractériser, efficacement et de manière optimale, les différents contextes d'interaction, en fonction du scénario de notre application. Les contextes d'interaction, et les gestuelles utilisateur, sont décrits à partir d'événements clés, à des instants particuliers. Ces événements 'clés' se résument principalement à des événements facilement détectables et gérables, comme les collisions entre objets physiques au sein de la scène. N'étant pas absolue et permanente, cette caractérisation laisse le champ libre à l'utilisateur vis-à-vis de sa gestuelle et de la manière dont il l'adopte.
Cependant, notre caractérisation reste simple, relativement à notre contexte d'étude. Nous ne considérons que certains événements ponctuels et relativement basiques, dans le cadre d'un scénario limité et répétitif. La caractérisation d'un contexte d'interaction, et d'une gestuelle utilisateur, reste un processus difficile, source de problématiques qui sont l'objet de nombreuses études au sein de la littérature. De manière à valider rapidement notre approche, nous nous sommes dirigés vers une caractérisation relativement simple des différents contextes d'interaction, et notamment des gestuelles utilisateur. Cependant, il serait tout à fait envisageable par la suite de caractériser de manière plus complexe les différents contextes d'interaction et gestuelles utilisateur, dans le cadre de scénarios plus élaborés. Ceci ira de pair avec la complexité et le nombre d'informations extraites de la scène virtuelle. Il faudra également veiller à recouper différentes informations hétérogènes, précisément choisies en accord avec le scénario, permettant ainsi de diminuer le nombre d'ambigüités de sens que peut présenter une situation d'interaction donnée. Plusieurs travaux allant dans ce sens sont actuellement en cours, permettant ainsi la caractérisation de contextes d'interaction plus complexes.
La caractérisation, proposée au cours de cette section, a été analysée dans le cadre de plusieurs scénarios théoriques. Elle a montré sa validité et sa cohérence, permettant une description des contextes d'interaction sur plusieurs degrés de granularité (focalisation possible sur un événement particulier au sein de la scène virtuelle, sur une partie spécifique du corps de l'utilisateur, caractérisation plus globale de la scène virtuelle, etc.). La caractérisation que nous proposons pour décrire la gestuelle utilisateur est robuste face aux changements de morphologie de l'utilisateur et aux différences d'intervalles de temps d'observation. De plus, elle permet une caractérisation efficace de différents types de gestuelles et des différents éléments composant celles-ci (mouvements, gestes, actions et comportements).
Notre caractérisation est bien sûr fortement dépendante de la nature du scénario, et doit être repensée pour chaque nouvelle application, ce qui s'avérer laborieux dans le cas d'un scénario élaboré. De plus, l'établissement d'une caractérisation s'accompagne de l'identification des différentes informations à extraire de la scène et de la détermination de méthodes pour les extraire précisément et rapidement.

La problématique s'articulant autour de l'interprétation de l'activité au sein de la scène virtuelle a été discutée au cours de la Section 4. Cette interprétation est effectuée en calculant la différence, appelée distance, pouvant exister entre le contexte d'interaction construit à partir de la scène observée et le contexte d'interaction attendu par le scénario. En fonction de

cette distance, le système décide si le contexte d'interaction observé est en accord, ou non, avec le scénario de l'application, et/ou s'il présente des ambigüités de sens.
L'interprétation de l'activité au sein d'une scène observée reste un problème difficile. Selon le contexte d'interaction au sein duquel elle est observée, une activité peut présenter une ambigüité de sens, ne permettant pas toujours au système de l'identifier efficacement. Cette mauvaise interprétation influera alors, de manière indésirable, sur l'ensemble de l'interactivité, affectant le fonctionnement global du système.
La distance, que nous définissons entre deux contextes d'interaction, permet de concrétiser le résultat de l'interprétation de la scène observée. Lorsqu'une interprétation de scène est ambiguë, le processus dédié ne cherche pas forcément à résoudre cette dernière immédiatement. En revanche, il va progressivement diminuer le nombre de sens que peut présenter la scène virtuelle. Il ajuste alors sa vision de la scène, de manière à comprendre la signification véritable de l'activité au sein de celle-ci. Il est à noter que pour chaque nouveau scénario, donc pour chaque nouvelle caractérisation des contextes d'interaction, la distance est à redéfinir et à analyser précisément.
Nos contextes d'interaction sont simples, il en est bien sûr de même avec la distance qui peut exister entre le contexte d'interaction observée et celui attendu par le scénario de l'application. De plus, cette dernière accepte peu de compromis, dans notre contexte d'étude : l'activité au sein de la scène est reconnue si et seulement si toutes ses caractéristiques associées sont strictement observées. Notre distance ne prend également pas en compte les contextes d'interaction, qui pourraient présenter de trop nombreuses et importantes ambigüités de sens.
La complexité d'une distance est fortement variable. Elle dépend de la nature des caractéristiques comparées, du nombre de ces caractéristiques, etc. De futurs travaux devront donc s'intéresser à cette complexification. Il sera intéressant, par exemple, de considérer qu'une gestuelle est 'presque' observé, et d'adapter alors les réactions du système, de manière cohérente avec celle-ci. La complexification de la distance est évidement fortement liée à celle des contextes d'interaction, des informations extraites de la scène, etc.
Pour augmenter la robustesse et la précision du processus d'interprétation, il sera également possible de considérer des modèles prédictifs (***Filtres de Kalman***, ***HMM***, etc.) pour prévoir l'état de certaines caractéristiques à partir des états passés de celles-ci.

La cinquième section de ce chapitre a traité de l'adaptativité du système, matérialisé par la double réponse (interactive et adaptative) de celui-ci, vis-à-vis de son interprétation de la scène observée. L'adaptativité du système se traduit par la mise en place de mécanismes adaptatifs, pilotés par le scénario de l'application et paramétrés par les différentes informations contextuelles. Ces mécanismes, au fur et à mesure de l'interactivité, mettent en place différentes boucles logicielles, appelées boucles vertueuses, améliorant en permanence l'interactivité avec l'utilisateur.
Encore une fois, de manière à valider notre approche au plus vite, nous avons élaboré des mécanismes adaptatifs simples. Par la suite, ces mécanismes seront complexifiés, de manière à réellement mettre en évidence les capacités adaptatives de notre système. Il sera intéressant, par exemple, de mettre en place des mécanismes adaptatifs au niveau du rendu de la scène 3D. En fonction de la gestuelle adoptée par l'utilisateur, le rendu de la scène 3D pourra s'adapter, la caméra virtuelle, etc.
Grâce aux boucles vertueuses, le parallélisme de vision de la scène virtuelle est en permanence assuré et amélioré. La vision de la scène est adaptée au cours de l'interactivité à l'interprétation de l'activité. Si l'activité au sein de la scène virtuelle n'est pas comprise, la vision de la scène est élargie : extraction de nouvelles connaissances de la scène virtuelle, gestion de l'ensemble des événements au sein de la scène virtuelle, caractérisation de la

gestuelle utilisateur globale, etc. Si, au contraire, le processus d'interprétation reconnaît l'activité, la vision de la scène se réduit et se focalise seulement sur certains événements. Ainsi, les résultats sont plus précis – traitements spécifiques dans des cadres d'étude limités – et plus fiables – diminution du nombre de sources possibles d'erreurs et d'incertitudes.
Parallèlement à cela, la mise en œuvre des boucles vertueuses a permis l'optimisation du système dans sa globalité et l'assurance du respect de la contrainte temps réel. Notre système est complexe et de nombreux processus et ressources sont nécessaires à son bon fonctionnement. Par le biais de nos mesures adaptatives et l'accumulation de leurs effets au cours du temps, seuls les processus et les ressources utiles et nécessaires au système, à un instant donné, sont considérés, actifs et utilisés. Le reste des processus et ressources sont en état de latence, en attendant une utilisation future potentielle. De plus, si l'interactivité se déroule bien, le système est en droit d'attendre une activité particulière au sein de la scène à un instant donné. Il peut donc anticiper ce qu'il doit observer, caractériser, interpréter et mettre en œuvre. La mise en œuvre des boucles vertueuses permet donc au système d'anticiper, dans une certaine mesure, ses réactions et ses besoins en ressources.

Enfin, nous avons appliquée notre méthodologie à 3 études de cas, relatives au scénario de notre application : étude des réactions de l'utilisateur face au lancement de la balle, étude des différentes étapes du scénario (un lancer, une phase et une partie) et étude des réactions du système face à la présence éventuelle de spectateurs dans le fond de la scène.
Nous modélisons nos 2 premières études de cas par le biais de situations et de contextes d'interaction. Notre scénario est simple. Il reste cependant difficile à modéliser par le biais de notre représentation, car cette dernière nécessite une analyse approfondie et exhaustive du scénario et permet l'écriture de ce dernier de différentes manières. Cette difficulté à modéliser un scénario s'accroit au fur et à mesure que le concepteur cherche à le modéliser à un niveau sémantique bas. De plus, la moindre modification entraine un effort supplémentaire de modélisation. Par exemple, si nous décidions d'ajuster les boites englobant les mains de l'avatar et la balle au fur et à mesure des lancers d'une phase, il faudrait repenser la situation d'interaction « Un lancer de balle » comme un contexte et mettre en œuvre les conséquences qui en découlent. En revanche, notre scénario, même simple, a permis la validation de cette représentation. Cette dernière est puissante, permettant de nombreuses possibilités de modélisation, et permet la mise en évidence de l'adaptativité de notre système, vis-à-vis des contextes d'interaction identifiés et caractérisés.
Concernant les différentes études de cas, de manière générale, nous sommes loin d'avoir utilisé l'ensemble des informations contextuelles que nous avons à notre disposition et qui sont facilement caractérisables. Encore une fois, même avec un scénario simple, il est difficile d'être exhaustif et de tout prendre en compte, la tâche s'avérant vite complexe.
Dans le cadre de notre première étude de cas, nous ne sommes pas allés jusque la situation élémentaire, notre modélisation étant assez représentative et significative pour illustrer nos travaux. Cela dit, une description à un niveau sémantique plus bas demanderait un effort important d'analyse et de modélisation.
La seconde étude de cas nous a montré que nous pouvions construire des informations contextuelles de plus haut niveau, tout en gardant un scénario cohérent, par le biais d'un nombre limité d'informations contextuelles de bas niveau. Le niveau et l'intérêt de l'utilisateur sont toutefois évalués de manière basique. Ces informations sont construites à partir de trop peu de paramètres pour caractériser tous les aspects qu'ils englobent. Nous avons également réussi à garder un scénario plausible en ne considérant pas toutes la variété de fins, de situations ou de contextes, que nous avions à notre disposition. Nous avons considéré uniquement les fins pertinentes, en restant dans le cadre scénaristique que nous nous étions assigné.

La 3ème étude de cas propose plusieurs solutions tout à fait envisageables pour prendre en compte un certain nombre de spectateurs au fond de la scène réelle et un certain degré de mouvements générés par ces derniers. Les informations contextuelles et les mécanismes adaptatifs sont facilement identifiés et permettent une capture de la scène réelle efficace. En revanche, nos méthodes peuvent vite montrer leurs limites si le nombre de spectateurs et le mouvement sont trop importants, impliquant des occultations non négligeables et une incapacité du processus de modélisation du fond à gérer le dynamisme de la scène réelle.

Conclusion

Ces travaux de thèse ont présenté **l'élaboration d'un système** avec lequel un unique utilisateur interagit, en accord avec **une application scénarisée** de type **jeu vidéo** (simulation virtuelle d'entrainement de tennis). Le système répond, **en temps réel**, à **l'activité observée** et **interprétée**, **interactivement** et de manière **adaptative**. Le système immerge donc, d'une part, l'utilisateur au sein d'un **environnement virtuel** qui traduit sa réponse **interactive**, mais, d'autre part, **adapte son fonctionnement global** en fonction de l'activité.

L'interactivité est alimentée par les **gestuelles** du corps de l'utilisateur, en **temps réel** et de manière **non invasive** (sans contraintes matérielles : pas de contrôle de l'environnement réel et non nécessité de porter des marqueurs de la part de l'utilisateur). Le système réagit également à **divers événements** prenant place au sein de la scène réelle, qui sont dus au **dynamisme** de cette dernière et dont l'utilisateur n'est pas directement responsable. **L'activité** à laquelle le système réagit comprend donc les gestuelles utilisateur et ces derniers événements (changement d'éclairage, présence de spectateurs, etc.).
L'activité est **capturée** et codifiée par **un système dédié et étendu** au cours de nos travaux, adopté en accord avec le contexte industriel de cette thèse (convention ***CIFRE*** avec la société XD Productions).

Nous avons émis l'hypothèse que **l'activité** pouvait être caractérisée par le biais de **la modélisation du contexte d'interaction** l'englobant. C'est à partir de ce contexte et des indices implicites et explicites qu'il contient (contextualisation de l'activité) que le système reconnaitra l'activité, par le biais d'un processus dédié.
Etant donné que la logique de l'activité évolue dans un cadre scénaristique prédéterminé, il est alors possible d'en extraire **un ensemble de connaissances caractérisant les contextes d'interaction attendus**, à tout instant. Le processus dédié du système interprète alors l'activité, en calculant **la distance** pouvant exister entre ces contextes **attendus** et ceux **observés** au cours de l'interactivité. L'évaluation de cette distance pilote ainsi l'évolution du scénario et donc **les réponses interactives et adaptatives** du système.
Naturellement, la notion de scénario ne doit pas être comprise comme une forme fortement restrictive de la logique d'évolution. Le scénario peut ne pas être simplement linéaire mais, au contraire, intégrer un niveau élevé de complexité. Cette complexité n'affecte en rien le mécanisme de gestion de contextes.

Résumé de nos contributions & Discussion

- ***Architecture de notre système***

Nous avons défini, au cours du Chapitre 3., **l'architecture de notre système interactif**, sur lequel s'exécute de manière **adaptative** une application **scénarisée**.
L'architecture de notre système est divisée en 4 couches sémantiques, chaque couche étant liée à un niveau sémantique de données manipulées. Les différents modules composant ces couches ont été identifiés, chacun traduisant une étape bien particulière de l'interaction se déroulant entre l'utilisateur et le système.

Cette architecture a été validée par le biais du démonstrateur fonctionnel que nous avons développé, présenté au cours du Chapitre 5. et qui nous permet d'illustrer nos différentes contributions

Il a particulièrement été nécessaire d'étudier deux aspects de cette architecture, antérieurement à son élaboration : les processus interactif et adaptatif et la modélisation et gestion de contexte au sein d'un système interactif.

- ***Etude des processus interactif & adaptatif***

Le Chapitre 1. a présenté deux études importantes concernant **l'interactivité** et **l'adaptativité** au sein d'un système. Nous nous sommes positionnés vis-à-vis de nombreux aspects et problématiques, englobés par ces deux notions, d'abord du point de vue de l'équipe ***ImagIN*** (cadre académique de cette thèse) puis relativement à ces travaux de thèse.

Ces études nous ont permis d'établir **un cahier des charges complet**, guidant et cadrant les concepteurs au sein de l'équipe ***ImagIN***. **L'architecture type** d'un système de l'équipe a alors été définie.

Il a ainsi été possible d'établir précisément **l'architecture de notre système**, sur plusieurs niveaux sémantiques, et de définir les modules les composant, chacun traduisant une étape particulière de l'interaction entre l'utilisateur et le système. L'architecture de notre système est donc une spécialisation de l'architecture générale des systèmes développés au sein de l'équipe ***ImagIN***, en accord avec nos différents objectifs de thèse.

- ***Modélisation et gestion de contexte***

Nous avons introduit, au sein des processus interactif et adaptif, la notion de **contexte**. Une de nos hypothèses de départ fut en effet de considérer que **l'activité** au sein de la scène pouvait être traduite par **le contexte d'interaction** qui l'englobe.
La modélisation et la gestion du contexte au sein de notre système nous a permis de mettre en place des processus de caractérisation et d'interprétation de l'activité, ainsi qu'un ensemble de paramétrages relatifs aux différents mécanismes adaptatifs, établis par le système au cours de sa réponse en adéquation avec l'activité observée.

1. Modèle de contexte et écriture de scénario

Nous avons donc proposé, au cours du Chapitre 2., **un modèle de contexte complet**, que nous avons utilisé pour modéliser les différentes études de cas, relatives à notre scénario applicatif et illustrant nos travaux au cours du Chapitre 5. Ces études nous ont permis de valider notre modélisation du contexte d'interaction. Ce modèle est cohérent d'une étude de cas à une autre et peut s'appliquer efficacement à d'autres cas théoriques. En effet, notre application est simple mais nous a permis de démontrer la complétude de notre modèle.

Notre modèle peut s'appliquer sur un scénario, ou sur une partie de celui-ci, comme nous l'avons démontré dans le cadre de nos études de cas. La description d'un scénario peut se

faire sur plusieurs niveaux de granularité, permettant ainsi aux concepteurs une plus grande souplesse d'écriture. L'agenda, en plus d'apporter un ensemble d'informations contextuelles, peut aider à déterminer les différents niveaux de granularité d'un scénario.

Cependant, comme nous avons pu le voir dans le cadre de nos études de cas, l'utilisation de notre modèle pour décrire un scénario implique un travail d'analyse lent et relativement fastidieux. Il est effectivement difficile de prendre en compte tous les événements possibles, même dans le cas d'un scénario simple. Plus le nombre de données est important et plus ces données sont diversifiées, plus une modélisation basée situation et contexte d'interaction devient compliquée (énumération des différentes informations contextuelles, construction d'informations de plus haut niveau, etc.).
De plus, les possibilités en matière de modélisation (situation ou contexte d'interaction) et les différents niveaux de granularité d'une description permettent effectivement une plus grande souplesse d'écriture. Mais, du fait que plusieurs modélisations soient possibles pour atteindre un même objectif, une description de scénario est difficile à élaborer et à maintenir.
Enfin, nous remarquons que notre modèle dépend véritablement de l'application que nous avons développée. Une nouvelle application implique donc une nouvelle analyse du scénario et de nouvelles modélisations. Il est donc possible d'affirmer qu'une bonne modélisation, dans le cadre d'un scénario donné, permettrait la réutilisation de différents éléments d'interaction (situation et contexte), dans le cadre d'autres scénarios.

2. Différents sous-contextes

Notre modèle de contexte **regroupe** et **structure les différentes informations contextuelles** en **6 sous-contextes** bien particuliers : **« Utilisateur »**, **« Système/Application »**, **« Scène réelle »**, **« Interface »**, **« Observation »** et **« Temps »**. La division de notre modèle en plusieurs sous-contextes nous a semblé pertinente, en tout cas dans le cadre de notre démonstrateur et en accord avec nos objectifs de thèse. Cette division implique une cohérence au niveau du regroupage et de la structuration des différentes informations contextuelles, accessibles avant ou au cours de l'interactivité. D'autres applications, donc d'autres scénarios, permettraient de parfaire notre modélisation.

Les différents contextes sont gérés à tous les niveaux de l'architecture de notre système, par le biais de gestionnaires dédiés, que nous avons développés au cours de ces travaux.

3. Caractérisation et interprétation de l'activité

Nous nous sommes servis de notre modèle pour modéliser le contexte d'interaction au sein duquel l'activité prend place, au cours de l'interactivité. Nous avons alors distingué le contexte d'interaction **observé**, caractérisé au cours de l'interactivité à partir des observations extraites de la scène au sein de laquelle l'activité prend place, et le contexte d'interaction **attendu**, qu'il est possible d'extraire du scénario de l'application. **La caractérisation du contexte d'interaction** se base principalement sur l'ensemble des silhouettes de l'utilisateur sur les différents points de vue. Le processus de caractérisation pourrait donc être plus élaboré, si d'autres informations étaient considérées, comme des informations de couleur ou des analyses temporelles.

Un processus d'**interprétation** de l'activité a été proposé et implémenté, basé sur l'évaluation de la distance pouvant exister entre le contexte d'interaction observé et celui attendu par le scénario de l'application. Ce processus est particulièrement simple, dans la mesure où il n'est

basé que sur la reconnaissance d'événements interactifs facilement détectables par le moteur de l'application (comme les collisions physiques par exemple). La distance pouvant exister entre les contextes d'interaction observés et attendus est donc basique et implique des décisions binaires concernant la suite du scénario et donc la réponse du système à mettre en place.

Dans le cadre de notre application, nous n'avons considéré qu'un ensemble de contextes d'interaction simples et faciles à interpréter. Cette caractérisation et cette interprétation s'accompagnent de l'identification et de la gestion d'un ensemble d'informations contextuelles, peu nombreuses, relativement diverses et de niveaux sémantiques peu élevés. Cela n'a pas empêché la tâche d'être ardue et longue. Nous avons ainsi modélisé et géré les différentes étapes clés de notre scénario par le biais de notre modèle. Nos processus de caractérisation et d'interprétation, malgré leur complexité peu importante, sont robustes et efficaces.

4. Adaptativité

L'interactivité entre l'utilisateur et le système a pu être améliorée et enrichie, grâce à la mise en place de **mécanismes adaptatifs** particuliers, traduisant la réponse **adaptée** du système à **l'activité observée** et **interprétée**. Les différentes informations contextuelles **paramètrent** ces mécanismes adaptatifs.
Dans le cadre de notre application, nous avons décrit ces mécanismes adaptatifs, en identifiant précisément les informations contextuelles qui paramètrent chaque mécanisme, dans le cadre de notre application. Ces mécanismes sont relativement simples, basés principalement sur des affichages adaptées de la scène 3D, au niveau de l'immersion.

L'adaptativité du système met en avant une gestion multiple-focus de la scène, notamment au niveau de la capture de la scène réelle et de l'observation de la scène virtuelle. Ainsi, un parallélisme de vision est assuré en permanence, permettant ainsi au système à la fois de traiter la scène dans sa globalité et de considérer des parties et événements particuliers au sein de celle-ci. Ce processus est bien sûr supporté par la gestion efficace des informations contextuelles, notamment celles comprises au sein du contexte « Observation ».
La gestion multiple-focus a permis l'optimisation du processus de capture de la scène réelle. Le parallélisme de vision a effectivement prouvé son efficacité et sa fiabilité au cours de nombreuses expérimentations pour traiter le dynamisme au sein de la scène réelle, impliquant notamment des changements de lumière ambiante et des spectateurs dans le fond. Grâce à la redirection du processus de modélisation dynamique du fond, les éléments mobiles, hors utilisateur, sont rapidement et efficacement mis à jour dans le nouveau fond.
Le parallélisme de vision a particulièrement été adopté au niveau de la redirection du processus de capture des gestuelles utilisateur. En effet, en focalisant le processus de capture sur des parties spécifiques du corps de l'utilisateur, les mauvais résultats dus à un état d'équilibre instable (tremblement constant du modèle représentant l'utilisateur) ou à des erreurs de segmentation (à cause d'ombres par exemple) sont évités. De plus, le processus de capture peut être exécuté uniquement sur les parties en mouvement du corps de l'utilisateur. Cela reste tout de même difficile à paramétrer car un humain est toujours mobile et un petit mouvement très localisé contribue la plupart du temps à faire bouger l'ensemble du corps. De plus, les zones d'intérêt au sein desquelles le processus de capture est redirigé sont généralement de petites dimensions. Ne pas exécuter le processus de capture hors de ces zones peut s'avérer être une décision trop extrême, générant un affaissement général du modèle représentant l'utilisateur.

Les boucles vertueuses constituent une contribution importante et permettent l'optimisation du système dans sa globalité, tout en obtenant des résultats plus précis et fiables. Elles permettent une gestion efficace des différentes ressources matérielles et logicielles, assurant ainsi une interactivité temps réel, malgré la complexité du système global.
De plus, les boucles vertueuses permettent une accélération des différents processus, notamment ceux de la capture de la scène réelle et de l'observation de la scène virtuelle. En effet, elles permettent une gestion de plus en plus minimale des différentes zones d'intérêt. Ainsi, les traitements liés à la capture de la scène réelle et à l'observation de la scène virtuelle, redirigés dans ces zones, sont cadrés et limités et s'exécutent rapidement. Les boucles vertueuses, au cours de l'interactivité, permettent donc de diminuer progressivement les temps et les coûts de calcul.

- ***Système étendu de capture de l'activité***

Nous avons présenté, au cours des Chapitres 4. et 5., un ensemble de solutions pour **étendre** le système commercial de capture de mouvements, le ***Cyberdôme***, que nous avons adopté initialement en accord avec le cadre industriel de cette thèse.

Les travaux concernant directement le système de capture ont été précédés de **l'étude des notions de 'capture', 'mouvement' et 'corps'**. Cette étude et les définitions précises qui en ont résulté nous ont permis de bien comprendre le processus que nous voulions implémenter, ainsi que les aspects et problématiques qu'il englobait.

- ***Etude des notions de 'capture', 'mouvement' et 'corps'***

Les notions de **'capture'**, **'mouvements'** et **'corps'** présentent de nombreuses ambigüités de sens. Nous avons cherché, au cours du Chapitre 3., à définir précisément ces termes et à décrire les différents aspects qu'ils englobent. Nous nous sommes alors positionnés vis-à-vis de ces définitions, nous permettant ainsi de définir précisément le processus que nous voulions implémenter et d'identifier les différentes problématiques auxquelles nous avons du faire face, en accord avec nos objectifs de thèse : capture temps réel de l'activité, de manière non invasive, en milieu non contrôlé.

La définition de la notion de 'mouvement' a impliqué les définitions de plusieurs aspects particuliers (mouvement, geste, action, comportement et gestuelle), établies en accord avec notre sujet de thèse. Dans nos travaux, seule la modalité liée à la gestuelle de l'utilisateur est considérée. Si d'autres modalités devaient être considérées dans le futur, ces définitions seraient à compléter et à préciser.

- ***Extension du système commercial de capture de mouvements***

Nos travaux ont permis d'**étendre le processus initial de capture de mouvements**, vers une capture plus **générale** de **l'activité globale** au sein de la scène, tout en permettant l'allégement des contraintes matérielles que présentait le système au départ : total contrôle de l'environnement et nécessité pour l'utilisateur de porter des marqueurs colorés.

Nous avons implémenté un algorithme de modélisation dynamique du fond de la scène, pour nous départir de la contrainte liée au contrôle de l'environnement de capture. Cet algorithme a permis, d'une part, de capturer plus efficacement les gestuelles utilisateur et, d'autre part, de détecter et codifier d'autres événements, liés principalement aux mouvements de spectateurs et au changement d'illumination. La toile du fond du système initial a pu être enlevée et la capture peut être assurée en présence de spectateurs. L'algorithme de modélisation dynamique du fond a démontré son efficacité au cours de plusieurs situations problématiques.
Malgré tout, le processus de segmentation des silhouettes reste basique et peut générer des artefacts conduisant à une mauvaise capture des gestuelles utilisateur. Autrement dit, l'introduction au sein du processus de capture d'un algorithme de modélisation dynamique du fond n'a pas entrainé de modifications au niveau du processus de segmentation des silhouettes utilisateur, qui peut parfois se révéler insuffisant. De plus, le processus de modélisation du fond nécessite plusieurs paramètres, qu'il faut parfois régler laborieusement, lors de plusieurs phases d'expérimentations. Enfin, de manière à réduire les artefacts liés aux ombres au sein de la scène et à augmenter les contrastes, une source de lumière diffuse a du être ajoutée au sein du ***Cyberdôme***.

La nécessité pour l'utilisateur de porter des marqueurs colorés a été soulagée grâce à l'introduction de l'algorithme de squelettisation des silhouettes au sein du processus de capture. En ne prenant en compte que la silhouette de l'utilisateur, l'algorithme génère le meilleur squelette que nous pouvions escompter visuellement.
Cependant, l'efficacité de l'algorithme de squelettisation est supportée par de fortes hypothèses, liées au scénario de notre application. En effet, au cours du scénario, l'utilisateur n'est pas sensé croiser ses bras, par exemple. Cette hypothèse assure de bons résultats au niveau du processus de capture. Sans ses hypothèses, l'algorithme de squelettisation peut s'avérer insuffisant dans certaines situations.

La gestion des différentes informations contextuelles au cours de l'interactivité a permis l'augmentation du processus initial de capture de mouvements, à une capture de l'activité plus générale au sein de la scène (gestuelles de l'utilisateur et événements liés au dynamisme de la scène réelle dont l'utilisateur n'est pas directement responsable). La gestion du dynamisme de scène, non lié à l'utilisateur, est particulièrement supportée par la gestion de ces informations contextuelles (contexte « Scène réelle » notamment).

De manière générale, nous avons passé beaucoup trop de temps sur le système de capture, notamment sur la maintenance du code relative à l'évolution permanente du logiciel de capture au cours des années. Cette maintenance s'est faite au détriment de travaux intéressants vis-à-vis des processus de caractérisation des contextes d'interaction et d'interprétation de l'activité au sein de la scène.
De plus, les contraintes matérielles du système ont été allégées, principalement en fonction de notre contexte applicatif (entrainement de tennis) et à partir d'hypothèses fortes sur la nature des gestuelles utilisateur (l'utilisateur ne doit *a priori* pas croiser les bras au cours d'une partie). Tous les problèmes relatifs à une capture non invasive temps réel n'ont donc pas été résolus.

- ***Développement d'un démonstrateur***

Nous avons développé, pour illustrer nos différentes contributions, un **démonstrateur**, sur lequel s'exécute une application de type jeu vidéo. Ce démonstrateur est présenté au cours du Chapitre 5.
Par le biais de ce démonstrateur, nous avons implémenté les processus **interactif** et **adaptatif** prenant place entre le système et l'utilisateur, dans le contexte d'une application particulière.

Le scénario de l'application immerge l'utilisateur au sein d'un entrainement virtuel de tennis. Nous avons extrait de ce scénario, 3 études de cas particulières, qui nous ont servi pour démontrer la pertinence de nos approches.
La première version de ce jeu vidéo a été développée. Ce dernier et son scénario sont basiques mais suffisants pour montrer la potentialité de notre approche. Notre scénario implique, en effet, une activité limitée, simplement caractérisable et interprétable, interprétation entrainant des réponses interactives et adaptatives rapides et faciles à implémenter. Cependant, notre application nous a permis de mettre en avant nos problématiques et d'illustrer nos approches et contributions.

Perspectives

- *Architecture de notre système*

Nous avons fait le choix dans ces travaux de ne pas traiter les problématiques liées à la logique concepteur du système.
Nous avons proposé un modèle de contexte que les différents concepteurs peuvent générer et gérer au niveau de la logique concepteur de l'architecture du système, mais nous ne nous occupons justement pas de cette modélisation et de cette gestion au cours de l'interactivité.
En fonction de l'étape du scénario, au sein de laquelle l'utilisateur évolue, nous avons également considéré que l'ensemble des informations contextuelles étaient disponibles et réparties en fonction des niveaux sémantiques de l'architecture du système. Autrement dit, la construction et la gestion des différentes informations contextuelles au niveau de la logique concepteur n'ont pas été considérées au cours de ces travaux.
Nous n'avons pas non plus abordé les mécanismes de décision, lors de l'établissement des réponses interactives et adaptatives, au niveau de la logique concepteur. Ces réponses sont supportées par le scénario de l'application et dépendent de l'activité observée et interprétée. Dans le cadre de nos travaux, chaque étape clé du scénario correspond à une réponse interactive et un ensemble de mécanismes adaptatifs que nous gérons directement au niveau de la couche et du module cibles de l'architecture de notre système.
Des travaux futurs sont donc à prévoir, concernant l'ensemble de ces problématiques au niveau de la logique concepteur.

Au niveau de l'architecture en elle-même, telle que nous l'avons développée, une meilleure indépendance des gestionnaires de connaissances serait certainement nécessaire, si cette architecture devait se complexifier. Les outils informatiques utilisés pour développer notre architecture ne permettent pas toujours de dissocier les modules de traitement des gestionnaires de connaissances.

De futurs travaux concernent la formalisation la modularité de notre architecture. En effet, le système doit être robuste vis-à-vis des changements de système de capture ou de moteurs

d'application. De plus, il doit rapidement et facilement s'adapter aux modifications et aux changements de scénario. Nous avons cherché, au cours de ces travaux, à identifier précisément les différentes entrées et sorties de chaque module de traitement, pour faciliter ce travail sur la modularité de l'architecture.

Enfin, notre système ne considère qu'une modalité d'interaction, de l'utilisateur au système (ses gestuelles). De futurs travaux concernent l'ajout de modalités au niveau de l'interface d'acquisition.

- *Scénario*

Tout d'abord, le scénario de notre application est simple. Pour mettre en place une application plus riche et divertissante, il faudrait complexifier son scénario. La complexification du scénario s'accompagnerait bien sûr de l'enrichissement de la scène virtuelle, au sein de laquelle l'utilisateur évolue.
De plus, dans le cadre de ces travaux, nous sommes partis du scénario de l'application pour ensuite élaborer notre approche. De futurs travaux sont en cours pour rendre notre méthodologie plus générique et indépendante du scénario.
Une nouvelle formalisation de notre modèle serait également nécessaire, de manière à travailler plus précisément sur la réutilisation de scénarios ou d'étapes scénarisées, basés situations et contextes, dans le cadre de nouveaux scénarios.

Nos travaux ont mis en relief le besoin d'un logiciel auteur de création et d'édition de scénarios représentés par le biais de situations et de contextes d'interaction. Ce logiciel offrirait un cadre automatisé pour son écriture, son édition et sa maintenance.
Ce logiciel permettrait la gestion de structures importantes de données hétérogènes, statiques ou dynamiques. Il supporterait la mise à jour automatique, et sur différents niveaux de granularité, d'un scénario. Un mécanisme de correction automatique assurerait la cohérence des scénarios, permettant ainsi la réutilisation libre des divers éléments les composant, dans le cadre de nouveaux scénarios.

- *Capture de l'activité*

De manière générale, de nouveaux algorithmes seraient nécessaires pour rendre plus robuste le processus de capture, basés sur le traitement et l'exploitation de nouvelles caractéristiques, 2D principalement, telles que la couleur, les contours, etc. La construction de modèles d'apparence nous semble une bonne perspective de travail.
De plus, un travail sur la cohérence temporelle d'un certain nombre de caractéristiques, ainsi que des forces dynamiques générées par le processus de capture, assurerait une meilleure robustesse de ce dernier.
Enfin, plusieurs algorithmes prédictifs, concernant l'évolution de différentes données, comme les poses des éléments du modèle représentant l'utilisateur, pourraient être envisagés.

Plusieurs approches permettraient d'améliorer le processus de modélisation dynamique du fond de la scène. Un travail sur les ombres nous semble notamment obligatoire.
De plus, ces améliorations pourraient s'accompagner de l'adoption d'un nouvel algorithme de segmentation des silhouettes de l'utilisateur, l'algorithme initial atteignant vite ses limites, en tout cas dans notre cadre d'études.

Le processus de squelettisation des silhouettes pourrait être complété par de nouveaux algorithmes, assurant une meilleure désambigüisation des silhouettes. Il peut par exemple être possible de distinguer et de localiser les bras et les jambes de l'utilisateur en exploitant les exosquelettes de Voronoï. Le recoupage des informations provenant des différents squelettes construits pourraient également supporter le processus. Ainsi, il serait possible de ne plus prendre en compte certaines hypothèses fortes, concernant les gestuelles utilisateur, comme le fait que l'utilisateur n'est pas sensé croiser les bras au cours du scénario.

- *Caractérisation du contexte d'interaction*

De futurs travaux impliquent forcément la complexification des contextes d'interaction. Nous pensons que le sous-contexte le plus intéressant est celui associé à l'utilisateur. En effet, de nouvelles applications pourraient prendre en compte un 'profil' utilisateur, entrainant l'adoption de mécanismes adaptatifs adéquats, en fonction de son niveau par exemple. Le vocabulaire de gestuelles devra également être complété, accompagné d'une caractérisation plus précise et plus élaborée de ces gestuelles.

- *Interprétation de l'activité*

L'adoption de nouvelles gestuelles, et donc la caractérisation plus complexe de ces dernières, entrainent également la mise en place de mécanismes d'interprétation plus élaborés. De futurs travaux auraient pour objectifs de formaliser la notion de distance, pouvant exister entre le contexte d'interaction observé, et celui attendu par le scénario. Les mécanismes de raisonnement qui en découlent devront alors être établis précisément, de manière à ce qu'ils permettent des compromis quant à l'interprétation de l'activité observée. Pour améliorer le processus d'interprétation, plusieurs modèles prédictifs (**Filtres de Kalman**, **HMM**, etc.) peuvent être considérés, pour prévoir l'état de certaines caractéristiques à partir des états passés de celles-ci.

- *Adaptativité du système*

Nos mécanismes adaptatifs, dans nos travaux, ne concernent que l'affichage adapté, au niveau de la scène 3D restituée à l'utilisateur, d'éléments permettant d'orienter ce dernier vers la bonne gestuelle à adapter.
De futurs travaux concerneront l'adaptation du système à différents niveaux, tels que l'optimisation et l'anticipation des ressources logicielles et matérielles, le rendu de la scène 3D, l'évolution et le contenu du scénario, etc.

De plus, nos mécanismes adaptatifs sont supportés par des processus de contextualisation relativement simples, se basant majoritairement sur un ensemble de régions d'intérêt 2D. La complexification du scénario s'accompagnerait de celle de ces processus de contextualisation et des caractéristiques les supportant.

Annexe A. Terminologie

Plan

Au cours de ces travaux de thèse, nous présentons les versions, de plus en plus détaillées, de l'architecture du système interactif que nous développons (Chapitre 1., Chapitre 3.). Il nous semble donc important de nous positionner sur plusieurs points, dans le cadre de l'élaboration de notre système, pour bien comprendre ce que nous entendons par **'architecture'**, **'logique concepteur'** ou encore **'scénario'**.

Dans ces travaux, lorsque nous parlons de l'architecture, nous nous référons à l'**architecture logicielle** du système. C'est pourquoi la prochaine section traite de plusieurs **remarques** sur l'architecture matérielle du système, **justifiant notre positionnement**. La Section A.2. donnera une **définition** et une **description** de l'architecture logicielle du système. La Section A.3. introduira la notion de **logique concepteur**, définissant les logiques de conception de l'**architecte** du système et du **développeur** de l'application. Cette logique concepteur définit un ensemble de **scénarios** que nous distinguerons au cours de la Section A.4. Enfin, nous exposerons les différentes **propriétés** d'une bonne architecture de système (Section A.5.).

A.1. Remarques préalables sur l'architecture matérielle

Nous décidons de nous arrêter uniquement sur l'étude de l'**architecture logicielle** du système et non matérielle. En effet, nous avons fait le choix d'utiliser dans le cadre de cette thèse du matériel informatique standard et grand public. Nos travaux et contributions sont majoritairement logiciels et nous n'apportons aucune contribution au niveau de l'architecture matérielle d'un système.
Nous parlerons cependant dans cette thèse d'aspects matériels lorsque nous nous intéresserons aux **interfaces** utilisées, en particulier à l'interface d'acquisition (système commercial de capture de gestuelles du corps), autour duquel s'articulent ces travaux. C'est pourquoi l'architecture logicielle de notre système pourra comprendre des interconnections avec des interfaces matérielles, de manière à faire le lien entre ce dernier et l'utilisateur.
Enfin, nous nous référerons aux aspects matériels lorsque nous parlerons de **programmation** sur les *Graphics Processing Units* (GPU) des cartes graphiques standards. En effet, cette forme de programmation est très usitée dans les domaines de la synthèse et du traitement d'images, et sera largement utilisée dans le cadre de nos travaux.

A.2. Définition & description de l'architecture logicielle d'un système

L'**architecture logicielle d'un système définit** et **cadre la conception et l'implémentation** des différents **modules logiciels de traitements** d'un système interactif donné, ainsi que les **connections** qui peuvent exister entre ces composants, à différents niveaux d'abstraction. Dans ces travaux, nous pourrons nous y référer en parlant de l' « **architecture du système** ».

L'architecture du système comprend les différents **modules de traitements**, gérant l'interactivité avec l'utilisateur. A noter qu'un module peut être lui-même modélisé sous la forme d'une hiérarchie de sous-modules. Ces modules sont indépendants de l'application, dans le sens où **un changement d'application n'altère pas leurs rôles**.
Cependant, l'**application** doit être représentée au sein de cette architecture, car c'est par le biais de son scénario que l'interactivité avec l'utilisateur est dirigée. L'application s'exécute sur le système et un système peut supporter plusieurs applications.
Enfin, l'architecture du système comprend un ensemble de **connaissances spécifiques**, concernant l'utilisateur, une vérité terrain, la scène observée, etc. Ces connaissances peuvent être regroupées en une base de données, dont le but est de fournir aux différents modules du système celles dont ils peuvent avoir besoin.

Pratiquement, il n'est pas toujours évident de différencier explicitement l'application du système. En effet, il est courant, au niveau développement, qu'une application soit incluse et mélangée à un système ou à certains composants logiciels de celui-ci. Dans le cadre du développement de jeu vidéo par exemple, une application va nécessiter un moteur de rendu, un moteur de physique, etc., qu'il est possible de considérer comme des composants propres au système interactif. Cette application et ces moteurs seront alors en général imbriqués dans un seul et même exécutable. Une bonne modélisation du système permet de rendre modulaire son architecture, la rendant indépendante des composants propres à l'application. Cependant, les solutions logicielles permettent parfois difficilement une indépendance totale de tous les composants du système avec ceux de l'application.
Dans ces travaux, nous nous efforcerons de distinguer l'application, en particulier les connaissances scénaristiques qu'elle apporte, au sein de l'architecture logicielle du système. Cependant, les solutions logicielles que nous utilisons nous ont parfois contraints à développer des composants dépendants à la fois au système et à l'application.

A.3. Logique concepteur

L'élaboration d'un système sur lequel s'exécute une application nécessite donc plusieurs types de **concepteurs**. Nous parlerons dans ces travaux d'**architecte** lorsqu'il s'agira de l'élaboration du système (et donc de son architecture) et de **développeur** lorsqu'il sera question de l'application. De manière plus générale, englobant l'architecte et le développeur, nous parlerons de **concepteurs**.
Ainsi, le système est défini suivant la logique d'un architecte (**logique système**) et l'application est développée suivant la logique du développeur (**logique application**). La **logique concepteur** englobe ces deux logiques.

L'architecture d'un système peut être décrite sur plusieurs **niveaux de sémantique**. La logique concepteur constitue la couche sémantique la plus haute de cette description, la couche la plus basse étant celle des interfaces matérielles. Chaque couche de l'architecture

comprend un ensemble de connaissances, par le biais d'un gestionnaire spécifique, adaptées à son niveau de sémantique.

Ces travaux de thèse ne concernent pas la couche sémantique la plus haute correspondant à la logique concepteur. Cette logique ne sera d'ailleurs pas toujours présente dans les architectures que nous présenterons. De plus, lorsque nous en parlerons, nous ne ferons pas toujours la différence entre logique application et logique système.

A.4. Scénarios mis en œuvre au cours de l'interactivité

Au cours de l'interactivité, plusieurs **scénarios** sont mis en œuvre. Tout d'abord, le **scénario de l'application**, défini par la logique application, dirige l'interactivité. Ce scénario permet la mise en place de la réponse du système vis-à-vis de l'utilisateur. Dans le cadre de ce scénario, l'utilisateur adopte une **gestuelle** pour interagir avec le système.
La logique système définit les **scénarios que suivent les différents modules** compris dans l'architecture du système. Ces scénarios définissent les **traitements** à effectuer par les modules en fonction d'informations spécifiques paramétrant ces derniers. Il nous arrivera, dans ces travaux de thèse, de parler des comportements des modules ou des processus du système, nous référant alors aux traitements qu'ils effectuent.

La description de l'architecture sur plusieurs couches sémantiques doit permettre la compréhension du déroulement de l'interactivité entre l'utilisateur et le système. La réponse du système est déclenchée à partir de l'interprétation l'activité observée et particulièrement de celle de la gestuelle de l'utilisateur. C'est la logique concepteur, plus exactement la logique application, qui définit cette réponse. A partir de là, cette réponse est déclinée sur l'ensemble des couches sémantiques de l'architecture, jusqu'à la couche sémantique la plus basse. Les différents scénarios, définis par la logique système et guidant les traitements des différents modules de l'architecture, sont alors mis en œuvre, jusqu'à la restitution de la réponse du système à l'utilisateur.

Dans ces travaux de thèse, à l'instar des logiques, nous ne ferons pas toujours la différence entre le scénario de l'application et les scénarios définis par la logique système. Nous parlerons de 'scénario', nous référant majoritairement au scénario de l'application, qui est de toute façon le point de départ de la mise en place des scénarios des différents processus au sein du système. Lorsqu'il est question de « réponse du système à l'utilisateur », c'est bien sûr vis-à-vis du scénario de l'application.

A.5. Propriétés d'une architecture de système

L'architecture logicielle d'un système doit pouvoir évoluer au même rythme que les besoins des concepteurs. Selon leurs objectifs, certains modules de l'architecture devront être **réutilisés**, **modifiés**, ou **enrichis**. L'architecture logicielle d'un système interactif doit donc posséder certaines caractéristiques, qui orienteront les choix des concepteurs et qui ne les obligeront pas à la modifier dans sa totalité, dans le cas où un unique module de traitements serait modifié.

Tout d'abord, il est tout à fait possible que l'architecture d'un système soit composé de plusieurs modèles identiques et donc effectuant le même traitement. C'est pourquoi le

concepteur s'appliquera à rendre un module aussi générique que possible, pour pouvoir le réutiliser dans l'architecture sans le modifier. Ceci donne une mesure de la **réutilisabilité** d'un composant, ou d'une architecture de composants.
Deuxièmement, le concepteur sera désireux de comparer les performances et résultats de deux algorithmes effectuant le même traitement. Il doit donc pouvoir modifier facilement et rapidement le module dédié à ce traitement, dans ce cas, remplacer l'ancien algorithme par le nouveau. Cette capacité de l'architecture logicielle à pouvoir être modifiée facilement dénote sa **maintenabilité**.
De plus, certains modules sont développés en accord avec un paradigme d'interaction donné (voir Section 1.2. du Chapitre 1.), et doivent pouvoir être intégrés dans une architecture logicielle déjà existante, selon l'évolution des objectifs du concepteur (gestion d'un environnement mixte, de multimodalités, etc.). Et il en est de même pour l'ensemble des modules composant l'interface système-utilisateur (voir Section 1.4. du Chapitre 1.), qui dépendront des modalités prises en compte et donc des périphériques et des techniques d'interaction adoptés. L'architecture logicielle d'un système doit donc être **extensible**.

Annexe B. Historique de l'Interaction Homme-Machine

Plan

Cet historique (voir Fig. B.1.) est une synthèse faite à partir de [Myers 98, Beaudouin-Lafon 07, Booth 08, Renevier-Gonin 10] et peut être complété plus exhaustivement par [Sears & Jacko 08]. Il a pour but d'offrir au lecteur un aperçu de l'évolution du domaine de l'***IHM*** au cours du temps. L'objectif de cet historique est également de montrer que l'***IHM***, initialement l'étude des interfaces entre l'utilisateur et le système, est devenu au fil des années l'étude globale du phénomène interactif.

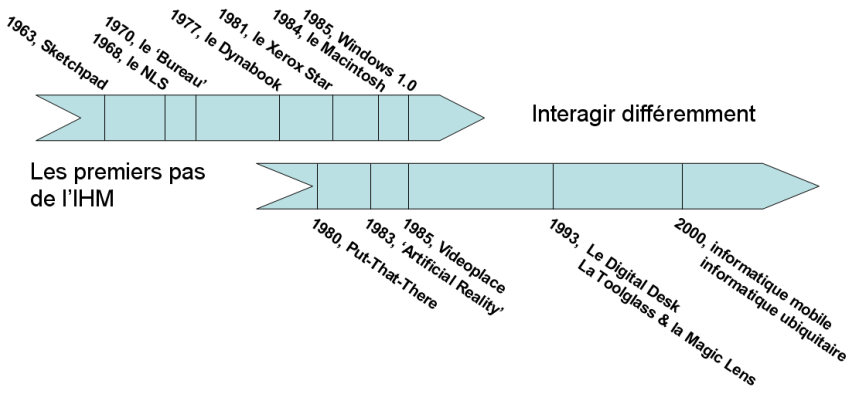
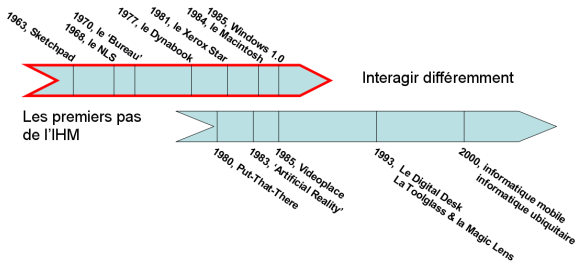


Figure B.1. : Historique de l'***IHM***

B.1. Les premiers pas de l'IHM



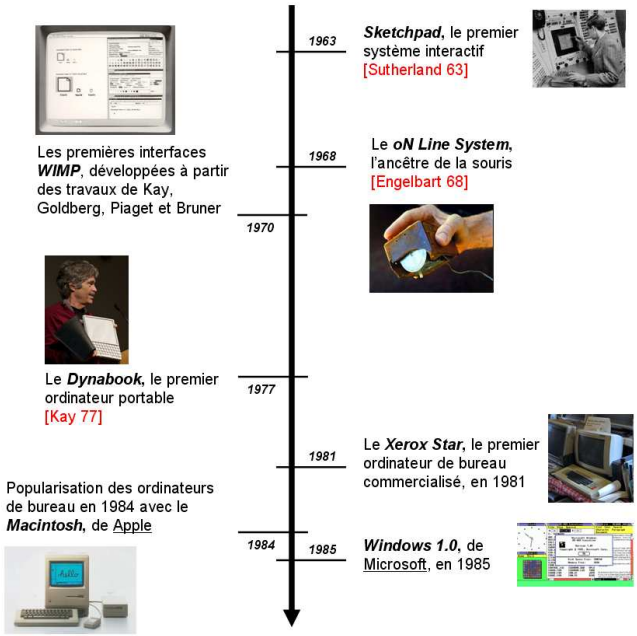


Figure B.2. : Les premiers pas de l'***IHM***

Comme le lecteur pourra le constater, l'***IHM*** était réduite depuis ses premiers pas à l'étude des **interfaces matérielles et logicielles**, avec l'avènement des premiers **périphériques standards** et des premières **interfaces *WIMP***. De 1960 à 1984 (voir Fig. B.2.), plusieurs systèmes interactifs ont vu le jour sans se soucier véritablement des autres facettes que pouvaient présenter le processus d'interaction.

Le système ***SketchPad*** [Sutherland 63] (Fig. B.3.), développé à partir des années 1960 au MIT Lincoln Laboratory, est considéré comme étant **le premier système interactif restituant des interfaces graphiques**. Ce système permettait l'édition graphique de dessins techniques par le biais d'un écran cathodique et d'un crayon optique. Avant le développement de ce système, des **langages de commandes** étaient systématiquement adoptés, obligeant l'utilisateur à mémoriser et à taper les noms des commandes et les éléments informatiques relatifs à la tâche qu'il devait accomplir.

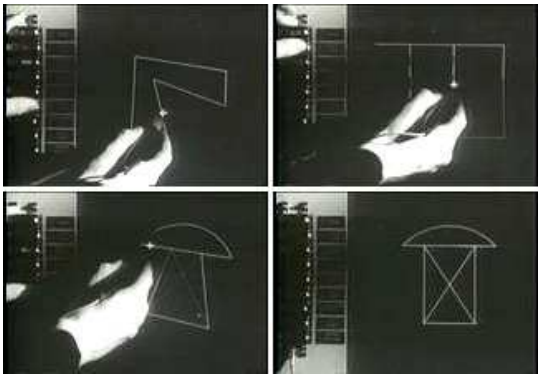


Figure B.3. : Système ***SketchPad*** [Sutherland 63]

Le ***oN Line System*** (***NLS***) [Engelbart 68] (Fig. B.4.), ancêtre de la **souris** actuelle, composé d'une boite en bois et de deux roues en métal, est présenté en 1968 et breveté en 1970. Et **le premier ordinateur portable**, le ***Dynabook*** [Kay 77] (Fig. B.4.), est conçu en 1972 dans les laboratoires PARC de Xerox.

Figure B.4. : Le ***oN Line System*** [Engelbart 68] (à gauche) & le ***Dynabook*** [Kay 77] (à droite)

C'est dans ce contexte et dans ces mêmes laboratoires qu'est développé et commercialisé, en 1981, le ***Xerox Star*** (nom commercial : ***Xerox 8010 Information System***, Fig. B.5.). Bien qu'il ne rencontre pas le succès commercial escompté (mauvaise étude de marché et encore trop de limites pour l'utilisateur), ce système peut être considéré comme **le premier ordinateur de bureau** : écran, clavier comportant 2 pavés et souris à 2 boutons.
L'interface graphique du ***Xerox Star***, présentée par [Smith & al. 82] (Fig. B.5.) avec laquelle l'utilisateur interagit est également révolutionnaire car elle utilise, pour la première fois au niveau commercial, la bien connue **métaphore du 'Bureau'**. Ce type d'interface a été développé depuis les années 70, particulièrement par Alan Kay et Adele Goldberg, en collaboration avec Jean Piaget et Jérôme Bruner, deux psychologues spécialistes du développement de l'enfant et de l'apprentissage. Par le biais de cette métaphore, l'utilisateur est immergé dans l'environnement connu qu'est son 'Bureau', composé de 'Dossiers', de 'Documents', d'une 'Corbeille', etc. Il dispose d'un 'Bloc-notes', premier éditeur de texte de type *What You See Is What You Get* (***WYSIWYG*** – « ce que vous voyez est ce que vous obtenez » [Smith 82]), d'un 'Calendrier', d'une 'Calculatrice'… Chaque élément du bureau est représenté par une icône spécifique, impliquant une large utilisation d'affordances[1] [Gibson 77], comme par exemple un document texte représenté par une feuille de papier. Ainsi, l'utilisateur a, pour la première fois, le sentiment de **manipuler directement** les éléments informatiques relatifs à la tâche qu'il doit accomplir et donc d'**interagir naturellement** avec le système.

[1] Eléments dont la représentation suggère leur usage, sans apprentissage au préalable

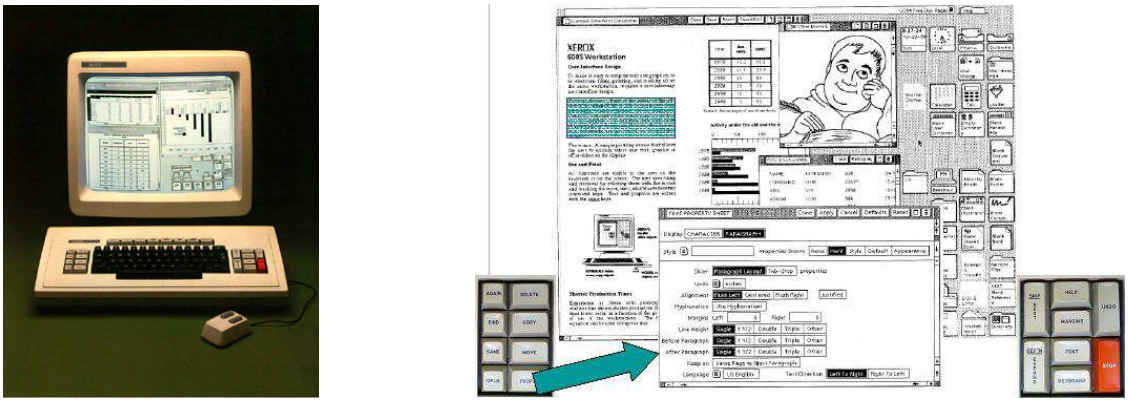


Figure B.5. : Le ***Xerox Star*** (à gauche) et son interface graphique [Smith & al. 82] (à droite)

Directement inspirés des ***Xerox Star***, **la popularisation des ordinateurs personnels** a lieu quelques années plus tard, en 1984, avec la commercialisation des systèmes ***Macintosh*** (Fig. B.6.), développés par Apple.

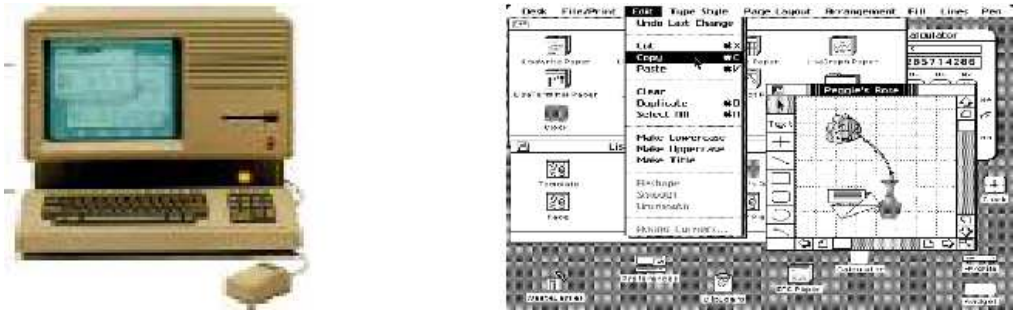


Figure B.6. : Le ***Macintosh*** (à gauche) et une interface de type ***WIMP*** (à droite)

Le succès commercial des ***Macintosh*** marqua non seulement l'adoption de la métaphore du 'Bureau' mais également l'avènement des **interfaces de type *WIMP*** (Section 1.2.1.3. du Chapitre 1.). L'origine du nom de ces interfaces est attribuée à Merzouga Wilberts en 1980 et ces interfaces ont été développées à partir des idées de Douglas Engelbart. A l'époque de la commercialisation du ***Xerox Star***, le concept d'interfaces ***WIMP*** était à peine défini et la société n'a pas désiré continuer dans cette voie… **Les interfaces de type *WIMP*** comportent des fenêtres superposables représentant un dossier au sein duquel différents documents peuvent organisés et modifiés ; des icônes représentatives, fonction des documents et fichiers que l'utilisateur manipule (utilisation des affordances [Gibson 77]) ; des barres de menus déroulants ; et un curseur, dirigé manuellement par le biais d'une souris et qui peut manipuler l'ensemble d'éléments informatiques. Elles ont été pensées et développées dans l'optique de rendre toujours plus naturelle et directe la manipulation des éléments informatiques relatifs à la tâche que l'utilisateur doit accomplir. A partir de 1984, les interfaces graphiques de type ***WIMP*** sont prédominantes sur le marché et seront adoptées par Microsoft pour le système d'exploitation ***Windows*** à partir de 1985 (***Windows 1.0***).

B.2. Interagir différemment

Il est évident aujourd'hui que l'utilisateur fait face quotidiennement aux mêmes interfaces qu'en 1984. Bien que les moyens techniques et les besoins de l'utilisateur deviennent plus importants, **aucune évolution notable** au niveau de ces interfaces n'a eu lieu depuis ces 40 dernières années. De nombreux travaux, comme [Beaudouin-Lafon 04] ont prouvé qu'actuellement, la manière d'interagir avec un système informatique, **l'interaction standard**, est loin d'être directe, adaptée et naturelle.
C'est pourquoi, une nouvelle interaction a pensée et envisagée. Toujours cantonnée à l'étude de l'interface avec l'utilisateur, de nouvelles façons d'**enrichir** celle-ci sont apparues dans le but de permettre une interaction plus directe avec les éléments relatifs à la tâche : **nouveaux périphériques** et **nouveaux outils logiciels**. Puis les concepteurs sont sortis du cadre restrictif de l'interface. L'étude du processus d'interaction entre l'utilisateur et le système ne se réduit plus à l'étude des interfaces mais tend à être pensé et élaboré en premier lieu. Le but est bien sûr de permettre à l'utilisateur d'**interagir naturellement** avec un système, comme il le ferait avec une autre personne, et de lui offrir les mêmes possibilités d'action, riches et nombreuses, qu'il peut réellement avoir. L'historique présenté dans cette section (Fig. B.7.) indique même que les fondements de cette nouvelle façon d'interagir remontent à plus de 40 ans.

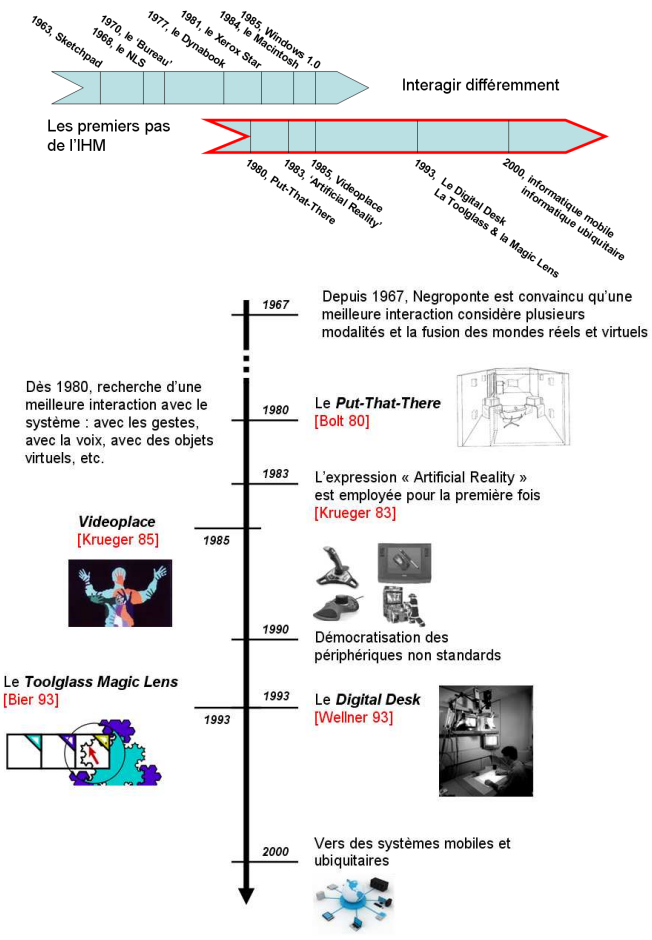


Figure B.7. : Interagir différemment

Tout d'abord, **l'interaction standard** (Section 1.2.1.3. du Chapitre 1.) a été enrichie et complexifiée grâce à la démocratisation de **périphériques non standards** (Fig. B.8., Section

1.4.3. du Chapitre 1.), plus adaptés à la tâche, à l'utilisateur ou à l'environnement : souris 3D, trackball, manettes de jeux vidéo, dispositif ayant la forme d'un instrument du monde réel (volant pour conduire, pistolet pour tirer, tablette graphique pour dessiner, etc.), bornes d'arcade, écran tactile, etc.
De plus, non seulement l'interface matérielle s'est diversifiée, mais de nouveaux outils logiciels, les outils semi-transparents, ont été développés, illustrant ainsi l'inconfort ressenti d'interagir par le biais d'interfaces ***WIMP***. Les outils semi-transparents les plus fréquemment cités sont la ***Toolglass*** et la ***Magic Lens*** (Section 1.4.2. du Chapitre 1.), définis par [Bier & al. 93].

Figure B.8. : Quelques périphériques non standards

Dorénavant, l'interaction doit aller au-delà du simple bureau virtuel et d'une illusion de manipuler directement les éléments relatifs à la tâche que doit accomplir l'utilisateur [Olsen & Klemmer 05]. L'utilisateur peut maintenant interagir avec le système avec ses mains, ses doigts, ses bras et jambes, sa tête, sa voix, ses expressions, des objets familiers… De plus, il est de moins en moins asservi par le système à un dialogue de type « contrôle-commande ». Le système lui laisse plus d'initiatives et réagit de manière adaptée à ses gestuelles implicites et explicites. Cette nouvelle forme d'interaction prend donc en compte des modèles efficaces d'interaction, une meilleure compréhension des niveaux sensori-moteur et cognitif de l'interaction au niveau de l'utilisateur et un cadrage pertinent et contextualisé des gestuelles de l'utilisateur, dans le but de mieux les observer et les interpréter.
C'est pourquoi, de **nouveaux paradigmes d'interaction** ont été étudiés (Section 1.2.2. du Chapitre 1.) : **interaction gestuelle**, **interaction parallèle** (interaction bimanuelle, interaction multidigitale, multimodalité), **réalités virtuelle & mixte**… Ces nouveaux paradigmes sont qualifiés de ***Post-WIMP*** [Beaudouin-Lafon 04], traduisant une rupture évidente vis-à-vis de l'interaction standard.
Les paradigmes ***Post-WIMP*** permettent aux concepteurs d'établir préalablement des cahiers des charges précis, des modèles d'interaction, avant d'élaborer un système interactif. L'interface devient une conséquence de cahier des charges et non pas une cause. En plus de cela, les technologies évoluant à un rythme effréné, de nouveaux outils logiciels, structurés et hiérarchisés, sont apparus, facilitant la modélisation, le développement et l'évaluation des architectures logicielles des systèmes interactifs.

L'étude d'une interaction plus naturelle entre l'utilisateur et le système ne date pas d'hier. Nicholas Negroponte [Negroponte 95], fondateur en 1967 de l'Architecture Machine Group

au Massachussets Institute of Technology (***MIT***) et en 1985 du MIT Media Lab, est très vite convaincu qu'une interaction plus naturelle avec le système est supportée par la considération de **plusieurs modalités** et la **fusion des mondes réel et virtuel**.

Le terme *Artificial Reality* (« Réalité Artificielle ») est utilisé pour la première fois par Myron Krueger [Krueger 83], qui élabore depuis 1969 des systèmes interagissant avec les gestes interprétés et en temps réel de l'utilisateur. Le système ***Videoplace*** ([Krueger 85], Fig. B.9.), conçu depuis 1975, en est un des exemples le plus concret.

Figure B.9. : ***Videoplace*** [Krueger 85] (à gauche) et ***Put-That-There*** [Bolt 80]

Le système multimodal *Put-That-There* (Fig. B.9.) est présenté en 1980 par Rich Bolt [Bolt 80]. Ce système reconnaît les gestes de la main de l'utilisateur, notamment la désignation d'un objet sur un grand écran, combinés à certaines commandes vocales.

Les réalités virtuelle et augmentée sont très étudiées à partir des années 1990 et l'impressionnant ***Digital Desk*** de Pierre Wellner en 1991 [Wellner & al. 93] est un des premiers systèmes utilisant ce type de réalités pour interagir avec l'utilisateur. L'utilisateur peut interagir avec une calculatrice et un éditeur de texte virtuels, projetés sur un véritable bureau, et leur transmettre des informations issues de documents réels (paragraphe d'un livre, figure d'une publication, nombre d'une note de frais, etc.). Seuls les mouvements de sa main sont filmés par le biais d'une caméra suspendue au dessus du bureau.

Aujourd'hui, peu de nouveaux paradigmes d'interaction sont adoptés sur le marché. De plus, les utilisateurs restent réticents à interagir avec de nouvelles interfaces, leurs habitudes étant bien ancrées avec les interfaces ***WIMP***. Cependant, avec l'avènement de **l'informatique mobile** et de **l'informatique ubiquitaire** depuis les années 2000, ces habitudes se modifient peu à peu et notablement vers une nouvelle façon d'interagir : interaction tactile de plus en plus omniprésente, disparition progressive d'interface matérielle, prise en compte du contexte d'interaction dans lequel évolue l'utilisateur (géo localisation, enregistrement automatique des habitudes interactives de l'utilisateur, etc.), réalité augmentée de plus en plus considérée, recherche sur les interfaces holographiques, etc. **L'étude de l'interface ne suffit plus.** Il est dorénavant nécessaire d'étudier les différents aspects du phénomène interactif, de manière à les implémenter efficacement et pertinemment au sein d'un système interactif.

Annexe C. Règles générales de conception de système

Nous rappelons, au cours de cette annexe, les 7 Règles Générales de travail, composant notre cahier des charges, au sein de l'équipe ***ImagIN***. Ce cahier des charges pilote et cadre les différents concepteurs, lors de l'élaboration d'un système interactif, sur lequel s'exécutent, de manière adaptée, des applications scénarisées.
Ce cahier des charges est expliqué au cours de la Section 3.1. du Chapitre 1.

RG 1. Le processus d'interaction scénarisé mis en œuvre par le système peut s'établir à la fois avec une personne observée par ce dernier, l'**Observé**, et avec une personne le commandant, l'**Utilisateur**.

RG 2. L'application s'exécutant sur le système comprend **deux scénarios** : un scénario destiné à la commander, suivi par l'Utilisateur ; et un scénario l'animant, immergeant l'Observé et le faisant évoluer au sein d'une narration.

RG 3. Un **dialogue interactif**, implicatif, constant et temps réel doit s'établir **entre l'Utilisateur et le système** et **entre l'Observé et le système**. Ce dialogue est piloté par le scénario que suit chacun des interactants.

RG 4. L'application répond par le biais d'une **interface de restitution** aux **commandes** explicites de l'**Utilisateur**, reconnus par le biais du scénario qu'il suit, acquises par le biais d'une **interface d'acquisition**. Elle répond par le biais d'une nouvelle **interface de restitution** à l'activité, relative à l'**Observé**, reconnue par le biais du scénario, observée et capturée par une **interface d'acquisition**.

RG 5. L'**activité**, relative à l'Observé, est **observée**, **caractérisée** et **interprétée**. Ces trois processus sont **pilotés** par le **scénario** de l'application.

RG 6. L'activité, relative à l'Observé, si elle est **attendue** par le **scénario** de l'application, **déclenche le processus d'adaptativité scénarisée** au sein du système.

RG 7. Guidé par les connaissances scénaristiques, le système s'adapte **en temps réel** à l'activité, relative à l'Observé, **aussi bien au niveau de sa réponse vis-à-vis de ce dernier qu'au niveau de son fonctionnement global**.

Bibliographie

[ACCORD 02]
Projet ACCORD, « Etat de l'art sur les Langages de Description d'Architecture (ADLs) », Assemblage de Composants par Contrats en Environnement Ouvert et Réparti, 2002.

[Accot & Zhai 97]
Accot, J. & Zhai, S., "Beyond Fitts' law: Models for trajectory-based HCI tasks", In Proc. of ACM Human Factors in Computing Systems, CHI'97, 1997.

[Ahlberg & Schneiderman 94]
Ahlberg, C. and Shneiderman, B., "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays", ACM conference on Computer-Humain Interaction (CHI), pp. 313-317, 1994.

[Appert 03]
Appert, C., « CIS : Evaluer les Techniques d'Interaction en Contexte », In Annexes des Actes d'IHM 2003. Caen, pages 23-25, 2003.

[Appert & al. 04]
Appert, C., Beaudouin-Lafon, M. & Mackay, W.E., « Context matters: Evaluating Interaction Techniques with the CIS Model », Technical report 1372, Laboratoire de Recherche en Informatique, Université Paris-Sud, France, 2004.

[Araki & Tachibana 06]
Araki, M., Tachibana, K., "Multimodal Dialog Description Language for Rapid System Development", In Proc. of the 7th SIGdial Workshop on Discourse and Dialogue, 2006.

[Arcelli & Sanniti di Baja 85]
Arcelli, C. and Sanniti di Baja, G., "A width-independent fast thinning algorithm", IEEE Transactions on Pattern Recognition and Machine Intelligence 7(4), 463–474, 1985.

[Aurenhammer 91]
Aurenhammer, F., "Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure", ACM Computing Surveys, 23:345-405, 1991.

[Balint 95]
Balint, L., "Adaptive interfaces for human-computer interaction: a colourful spectrum of present and future options", Systems, Man and Cybernetics, Intelligent Systems for the 21st Century, IEEE International Conference on Volume 1, Issue , 22-(pp. 292 - 297), vol.1, 1995.

[Balme 08]

Balme, L., « Interfaces Homme-Machine plastiques : Une approche par composants dynamiques », Thèse de doctorat, Université Joseph Fourier – Grenoble 1, 2008.

[Baron & al. 06]
Baron, M., Lucquiaud, V., Autard, D., & Scapin, DL, « K-MADe : un environnement pour le noyau du modèle de description de l'activité », In Proc. of the 18th French-speaking conference on Human-computer interaction, 2006. http://kmade.sourceforge.net/index.php

[Bartlett & al. 03]
Bartlett, M.S., Littlewort, G., Fasel, I. and Movellan, R., "Real Time Face Detection and Facial Expression Recognition: Development and Application to Human Computer Interaction", Proc. CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction, 2003.

[Baudel 95]
Baudel, T., « Aspects Morphologiques de l'Interaction Humain-Ordinateur : Étude de Modèles d'Interaction Gestuels », Thèse de doctorat, http://thomas.baudel.name/Morphologie/These/index.html, 1995.

[Beaudouin-Lafon 97]
Beaudouin-Lafon, M., « Interaction Instrumentale : de la Manipulation Directe à la Réalité Augmentée », in Actes des 9èmes Journées IHM, Poitiers, 1997.

[Beaudouin-Lafon 00a]
Beaudoui-Lafon, M., « Intrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces », In Proc. Of ACM Human Factors in Computing Systems (CHI'2000), 2000.

[Beaudouin-Lafon 00b]
Beaudouin-Lafon, M., « Ceci n'est pas un ordinateur. Perspectives sur l'Interaction Homme-Machine », Technique et Science Informatique vol. 19, 2000.

[Beaudouin-Lafon & al. 01]
Beaudouin-Lafon, M., Mackay, W. E., Andersen, P., Janecek, P., Jensen, M., Lassen, M., Lund, K., Mortensen, K., Munck, S., Ravn, K., Ratzer, A., Christensen S. and Jensen, K., "CPN/tools: revisiting the desktop metaphor with post-WIMP interaction techniques", In CHI '01 Extended Abstracts on Human Factors in Computing Systems, pages 11-12. New York, NY, 2001.

[Beaudouin-Lafon 04]
Beaudouin-Lafon, M., "Designing interaction, not interfaces", Proceedings of the working conference on Advanced visual interfaces, Gallipoli, Italy May 25-28, 2004.

[Beaudouin-Lafon 07]

Beaudouin-Lafon, M., « 40 ans d'interaction homme-machine : points de repère et perspectives »,)i(Interstices, http://interstices.info/histoire-ihm, 2007.

[Bérard & al. 99]
Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L. & Schnoebelen, P., « Vérification de Logiciels. Techniques et Outils du Model-Checking », Vuibert, 1999.

[Bérard 99]
Bérard, F., « Vision par ordinateur pour l'interaction homme-machine fortement couplée », PhD Thesis, Université de Joseph-Fourier-Grenoble I, 1999.

[Bier & al. 93]
Bier, E., Stone, M., Pier, K., Buxton, W., and DeRose, T., "Toolglass and magic lenses: The see-through interface", In Proc. of SIGGRAPH'93, 1993.

[Billon & al. 08]
Billon, R. & al., "Gesture Recognition In Flow based on PCA Analysis using Multiagent System", In Proc of ACE 2008, 2008.

[Blanch & al. 04]
Blanch, R., Guiard, Y., & Beaudouin-Lafon, M., « Semantic pointing: Improving target acquisition with control-display ratio adaptation", In Proc. of ACM Human Factors in Computing Systems, CHI '04, 2004.

[Blum 67]
Blum, H., "A transformation for extracting new descriptors of shape", in Models for the Perception of Speech and Visual Form (W. Wathen-Dunn, ed.), Cambridge MA: MIT Press, 1967.

[Bobick & al. 99]
Bobick, A. F., Intille, S. S., Davis, J. W., Baird, F., Pinhanez, C. S., Campbell, L. W., Ivanov, Y. A., Schütte, A., and Wilson, A. "The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment", Presence: Teleoper. Virtual Environ, 1999.

[Bobrow 77]
Bobrow, D., "An Overview of KRL", Cognitive Science 1(1), 1977.

[Bolchini & al. 07]
Bolchini, C., Curino, C. A., Quintarelli, E., Schreiber, F. A. and Tanca, L., "A Data-oriented Survey of Context Models", in ACM SIGMOD Record, vol. 36, no. 4, pp. 19-26, Dec 2007.

[Bolt 80]
Bolt, R.A, "Put-That-There: Voice and Gesture at the Graphics Interface", ACM SIGRAPH Computer Graphics 14::3 262-270, 1980.

[Booch & al. 00]
Booche, G., Jacobson, I. & Rumbaugh, J., “OMG Unified Modeling Language Specification”, http://www.uml.org/, 2000.

[Booth 08]
Booth, C., “Alan Kay and the graphical User Interface”, 2008.

[Bortaloso & al. 09]
Bortolaso, C., Dubois, E., Bach, C., Nigay, L., Coutrix, C., « Conception de systèmes interactifs mixtes : articulation d’une méthode informelle et d’un modèle d’interaction », Dans Interaction Homme-Machine (IHM 2009), 2009.

[Bouchard & al. 08]
Bouchard, B., Roy, P. C., Bouzouane, A., Giroux, S. & Mihailidis, A., "An activity recognition model for Alzheimer’s patients: Extension of the COACH task guidance system", In Proc. of the 18th European Conference on Artificial Intelligence (ECAI'08), 2008.

[Bouchet & al. 07]
Bouchet, J., Madani, L., Nigay, L., Oriat, C., & Parissis, I., « Formal Testing of Multimodal Interactive Systems », In Engineering interactive Systems: EIS 2007 Joint Working Conferences, EHCI 2007, DSV-IS 2007, HCSE 2007, 2007.

[Bouwmans & al. 08]
Bouwmans, T., El Baf, F. and Vachon, B., “Background Modeling using Mixture of Gaussians for Foreground Detection – A Survey”, In Recent Patents on Computer Science, 2008.

[Bray 01]
Bray, J., “Markerless Based Human Motion Capture: A Survey”, Department Systems Engineering Brunel University, 2001.

[Brdiczka & al. 06a]
Brdiczka, O., Reignier, P., Crowley, J. L., Vaufreydaz, D. and Maisonnasse, J. “Deterministic and probabilistic implementation of context”, In Proc. Of the 4th IEEE Int. Conf. On Pervasive Comput. Commun. Workshops, 2006.

[Brdiczka & al. 06b]
Brdiczka, O., Yuen, P.C., Zaidenberg, S., Reignier, P. and Crowley, J.L., “Automatic Acquisition of Context Models and its Application to Video Surveillance”, In Proc. of the 18th International Conference on Pattern Recognition (ICPR), 2006.

[Brdiczka & al. 07]
Brdiczka, O., Crowley, J. L. and Reignier, P., “Learning Situation Models for Providing Context-Aware Services”, In Proc. of HCI International, 2007.

[Brdiczka & al. 09]

Brdiczka, O., Langet, M., Maisonnasse, J. and Crowley, J. L., “Detecting human behavior models from multimodal observation in a smart home”, IEEE Transactions on Automation Science and Engineering, 2009.

[Bremond 97]
Bremond, F., « Environnement de résolution de problemes pour l'interprétation de séquences d'images ». PhD. Université de Nice-Sophia Antipolis, 1997.

[Bremond & Thonnat 96]
Bremond, F. and Thonnat, M. "A context representation for surveillance systems", Proc. ECCV Worshop on Conceptual Descriptions from Images, 1996.

[Buttussi & al. 07]
Buttussi, F., Chittaro, L., Ranon, R., and Verona, A., “Adaptation of Graphics and Gameplay in Fitness Games by Exploiting Motion and Physiological Sensors”, In Proc. of the 8th international Symposium on Smart Graphics, 2007.

[Buxton & Myers 86]
Buxton W. and Myers, B. A., "A Study in Two Handed Input", Proceedings of CHI ’86 (Boston, MA, April 1317), ACM, New York, pp. 321326, 1986.

[Buxton 09]
Buxton, B., “Multi-Touch Systems that I have Known and Loved”, http://www.billbuxton.com/multitouchOverview.html, 2009.

[Cadoz 94]
Cadoz, C., « Le geste canal de communication homme-machine. La communication "instrumentale" », Techniques et Sciences Informatiques. Vol. 13, 1 p. 31-61. 1994.

[Calvary & al. 08]
Calvary, G., Coutaz, J., Balme, L., Demeure, A. and Sottet, J.S., « The Many Faces of Plastic User Interfaces », Workshop on User Interface Description Languages for Next Generation User Interfaces (CHI 2008), 2008.

[Card & al. 83]
Card, S.K., Newell, A., & Moran, T.P., « The Psychology of Human-Computer Interaction », Lawrence Erlbaum Associates, Inc., 1983.

[Card & al. 91]
Card, S.K., Mackinlay, J.D. & Robertson, G.G., “A Morphological Analysis of the Design Space of Input Devices”, In Proc. of ACM Transactions on Information Systems, 1991.

[Calvillo-Gámez & al. 03]
Calvillo-Gámez, E.H., Leland, N., Shaer, O., & Jacob, R.J.K., “The TAC paradigm: unified conceptual framework to represent Tangible User

Interfaces", In Latin American conference on Human computer interaction, 2003.

[Chalidabhongse & al. 03]
Chalidabhongse, T. H., Kim, K., Harwood, D., Davis, L., "A Perturbation Method for Evaluating Background Subtraction Algorithms", In Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 2003), 2003.

[Chalmers & Galani 04]
Chalmers, M, Galani, A., "Seamful Interweaving: Heterogeneity in the Theory and Design of Interactive Systems", In Proc. of ACM DIS 2004, 2004.

[Champagnat 11]
Champagnat, R., « Pilotage de Systèmes : des Systèmes de Production Hybrides aux Appliccations Interactives Scénarisées », Habilitation à Diriger des Recherches, Université de La Rochelle, 2011.

[Chau & Betke 05]
Chau, M. and Betke, M., "Real time eye tracking and blink detection with USB cameras", Tech. Rep. 2005--12, Boston University Computer Science, 2005.

[Chen & Kotz 00]
Chen, G., & Kotz, D., "A survey of context-aware mobile computing research", Technical report, Dept. of Computer Science, Dartmouth College, Hanover, NH, USA, 2000.

[Clarke & al. 99]
Clarke, E.M., Grumberg, O. & Peled, D.A., "Model Checking", MIT Press, 1999.

[Cockton 87]
Cockton, G., « Some Critical Remarks on Abstractions for Adaptable Dialogue Managers », In Proc. of the Third Browne - 62 - Conference of the British Computer Society, Human Computer Interaction Specialist Group, 1987.

[Cohen & al. 98]
Cohen, J. D., Braver, T. S., and O'Reilly, R. C., "A computational approach to prefrontal cortex, cognitive control, and schizophrenia: Recent developments and current challenges", In A. C. Roberts, T. W. Robbins & L. Weiskrantz (Eds.), The prefrontal cortex: Executive and cognitive functions (pp. 195-220). Oxford: Oxford University Press, 1998.

[Collins & al. 00]
Collins, R.T., Lipton, A.J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P., Wixson, L., "A System for Video Surveillance and Monitoring", VSAM Final Report, Technical report CMU-RITR-00-12. Robotics Institute, Carnegie Mellon University, 2000.

[Corlouer & al. 08]

Corlouer, F., Bellamine, E., Gabet, S. and Dumartin, T., « Analyse de mouvement humain 3D à base d'extraction de squelette géométrique », Projet de Programmation, Mémoire Final, Université Bordeaux 1, 2008.

[Coutaz 88]
Coutaz, J., « Interface Homme-Ordinateur : Conception et Réalisation », Thèse de doctorat, Laboratoire de Génie Informatique (IMAG), Université Joseph Fourier, 1988.

[Coutaz & al. 95]
Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J. Young, R., "Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE properties", In Proc. of INTERACT'95, 1995.

[Coutaz & al. 05]
Coutaz, J., Crowley, J. L., Dobson, S. and Garlan, D., "Context is Key", Communications of the ACM, Special issue on the Disappearing Computer, Vol 48, No 3, pp 49-53, March 2005.

[Coutrix & Nigay 08]
Coutrix, C. et Nigay, L., « Balancing Physical and Digital Properties in Mixed Objects », AVI'08, 2008.

[Coutrix 09]
Coutrix, C., « Interfaces de Réalité Mixte : Conception et Prototypage », Thèse de doctorat, Laboratoire d'Informatique de Grenoble (LIG), Université Joseph Fourier, 2009.

[Coutrix & Nigay 09]
Coutrix, C. & Nigay, L., "An Integrating Framework for Mixed Systems", In The Engineering of Mixed Reality Systems, Chapitre 1, Springer-Verlag. 2009.

[Crowley & al. 00]
Crowley, J., Coutaz, J., and Bérard, F., "Things that see: Machine perception for human computer interaction", Communications of the ACM, 43(3):54–64, 2000.

[Crowley & al. 02]
Crowley, J.L., Coutaz, J., Rey, Gaeten and Reignier, P., "Perceptual Components for Context Aware Computing", UBICOMP 2002, International Conference on Ubiquitous Computing, 2002.

[Crowley & Reignier 03]
Crowley, J.L. and Reignier, P., "Dynamic Composition of Process Federations for Context Aware Perception of Human Activity", In International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2003.

[Crowley & al. 06]

Crowley, J.L., Brdiczka, O. and Reignier, P., “Learning Situation Models for Understanding Activity”, In International Conference on Development and Learning, ICDL '06, 2006.

[Crowley & al. 07]
Crowley, J.L., Hall, D. and Emonet R., “Autonomic Computer Vision Systems”, In International Conference on Computer Vision Systems, ICVS'07, 2007.

[Cucchiara & al. 03]
Cucchiara, R., Grana, C., Piccardi, M. and Prati, A., “Detecting moving objects, ghosts, and shadows in video streams”, In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003.

[Davies & Plummer 81]
Davies, E. R. and Plummer, A. P. N., “Thinning algorithms: a critique and a new methodology”, Pattern Recognition 14, 53–63, 1981.

[de la Gorce & Paragios 09]
de la Gorce, M., Paragios, N., "A Variational Approach to Monocular Hand-pose Estimation", Computer Vision and Image Understanding (CVIU), 2009.

[Delamarre 03]
Delamarre, Q., "Suivi du mouvement d’objets articulés dans des séquences d’images video", PhD thesis, University of Nice - Sophia Antipolis, France, 2003.

[Delmas 09]
Delmas, G., « Pilotage de récits interactifs et mise en œuvre de formes narratives dans le contexte du jeu vidéo », Thèse de doctorat, Laboratoire Informatique, Image, Interaction, Université de La Rochelle, La Rochelle, 2009.

[Demko 92]
Demko, C., « Contribution à la Gestion du Contexte pour un Système de Compréhension Automatique de la Langue », Thèse de Doctorat, Université de Technologie de Compiègne, Compiègne, 1992.

[Dey 01]
Dey, A., « Understanding and Using Context », Personal Ubiquitous Comput. 5, 1, 2001.

[Dietrich & al. 93]
Dieterich, H., Malinowski, U., Kühme, T. and Schneider-Hufschmidt, M., „State of the Art in Adaptive User Interfaces”, In M. Schneider-Hufschmidt, T. Kühme, U. Mallinowski (Ed.), Adaptive User Interfaces, Principles and Practice (pp.13-48). Included in series Human Factors in Information Technology, 10, 1987.

[Dill & al. 87]

Dill, A. R., Levine, M. D. and Noble, P. B., “Multiple resolution skeletons”, IEEE Transactions on Pattern Recognition and Machine Intelligence 9(4), 495–504, 1987.

[Dirichlet 50]
Dirichlet, G.L., "Uber die Reduktion der Positiven Quadratischen Formen mit Drei Unbestimmten Ganzen Zahlen", J. Reine Angew. Math., 40:209-27, 1850.

[Di Verdi & Höllerer 07]
Di Verdi, S. & Höllerer, T., “GroundCam: A Tracking Modality for Mobile Mixed Reality”, In IEEE Virtual Reality, 2007.

[Dourish 04]
Dourish, P., “What we talk about when we talk about context”, Pers. Ubiquitous Comput., vol. 8, pp. 19–30, 2004.

[Dragicevic 04]
Dragicevic, P., « Un modèle d'interaction en entrée pour des systèmes interactifs multi-dispositifs hautement configurables », PhD Thesis, Université de Nantes, 2004.

[Druin & Perlin 94]
Druin, A. & Perlin, K., “Immersive Environments: a Physical Approach to the Computer Interface”, In Human Factors in Computing Systems (CHI), 1994.

[Dubois & Gray 08]
Dubois, E., Gray, P., “A Design-Oriented Information-Flow Refinement of the ASUR Interaction Model”, In EIS‘08, 2008.

[Dumas & al. 09]
Dumas, B., Lalanne, D., and Oviatt, S. “Multimodal Interfaces: A Survey of Principles, Models and Frameworks”, In Human Machine interaction: Research Results of the MMI Program, 2009.

[El Baf & al. 08a]
El Baf, F., Bouwmans, T. and Vachon, B, “A Fuzzy Approach for Background Subtraction”, In International Conference on Image Processing (ICIP), 2008.

[El Baf & al. 08b]
El Baf, F., Bouwmans, T. and Vachon, B, “Foreground Detection using the Choquet Integral”, In the 9th International Workshop on Image analysis for Multimedia Interactive Services, 2008.

[El Baf & al. 08c]
El Baf, F., Bouwmans, T. and Vachon, B, “Fuzzy Foreground Detection for Infrared Videos”, In the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.

[Elhabian & al. 08]

Elhabian, S. Y., El-Sayed K. M. and Ahmed, S. H., “Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art”, In Recent Patents on Computer Science, 2008.

[Elgammal & al. 00]
Elgammal A., Harwood, D. and Davis, L., “Non-parametric model for background subtraction”, In European Conference on Computer Vision, 2000.

[Elgammal & al. 02]
Elgammal, A., Duraiswami, R., Harwood, D. and Davis, L. S., “Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance”, In Proc. of IEEE, 2002.

[Engelbart 68]
Engelbart, D. C., “A Research Center for Augmenting Human Intellect”, http://sloan.stanford.edu/MouseSite/1968Demo.html, 1968.

[Fabbri & al. 08]
Fabbri, R., Da F. Costa, L., Torelli, J. C. and Bruno, O. M., « 2D Euclidean Distance Transform Algorithms: A Comparative Survey », In ACM Computing Surveys, 2008.

[Feldman 92]
Feldman, T., "Generating isovalue contours from a pixmap", The Graphics Gems, 3, 29–35, 1992.

[Fikkert & al. 09]
Fikkert, W., Hakvoort, M., van der Vet, P., Nijholt, A., "FeelSound: Interactive Acoustic Music Making", In Proc. of ACE2009, 2009.

[Fisher & al. 96]
Fisher, B., Perkins, S., Walker, A. and Wolfart, E., “Hypermedia Image Processing Reference”, Department of Artificial Intelligence, University of Edinburgh, UK, 1996.

[Fishkin 04]
Fishkin, K.P., “A Taxonomy for and Analysis of Tangible Interfaces” In Journal of Personal and Ubiquitous Computing, 2004.

[Fitzmaurice & al. 95]
Fitzmaurice, G. W., Ishii, H., and Buxton, W., “Bricks: Laying the foundations for graspable user interfaces”, In Katz, I. R., Mack, R., Marks, L., Rosson, M. B., and Nielsen, J., editors, Proceedings of the Conference on Human Factors in Computing Systems (CHI'95), 1995.

[Fitts 54]
Fitts, P. M., “The information capacity of the human motor system in controlling the amplitude of movement”, Journal of Experimental Psychology, 47, 381-391, 1954.

[Fitts & Posner 67]
Fitts, P.M. & Posner, M.L., “Human Performance”, 1967.

[Fortune 86]
Fortune, S., « A Sweepline Algorithm for Voronoi Diagrams », In Proc. 2nd Annual ACM Symp. on Comp. Geom., pages 313-322, 1986.

[Gaines 91]
Gaines, B.R., “Modeling and Forecasting the Information Sciences” In Information Sciences, 1991.

[Ghahramani 98]
Ghahramani, Z. “Learning dynamic Bayesian networks”, Lecture Notes in Computer Science, 1387:168–197, 1998.

[Gibson 77]
Gibson, J. J., “The Theory of Affordances”, In Perceiving, Acting, and Knowing, Eds. Robert Shaw and John Bransford, 1977.

[Gibson 79]
Gibson, J., “Ecological Approach to Visual Perception”, Lawrence Erlbaum. 1979.

[Girard 08]
Girard. N., « Développement d’une méthode d’évaluation de la qualité en image », Rapport de Master 2. Université de La Rochelle (L3i), 2008.

[Gold & al. 99]
Gold, C.M., Thibault, D. and Liu, Z., “Map Generalization by Skeleton Retraction,” Proc. ICA Workshop Map Generalization, Aug. 1999.

[Greenberg 01]
Greenberg, S., “Context as a Dynamic Construct”, Human-Computer Interaction, 16, 2001.

[Grimson & al. 98]
Grimson, W.E.L, Stauffer, C., Romano, R., Lee, L., “Using adaptive tracking to classify and monitor activities in a site”, In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998.

[Guiard 87]
Guiard, Y., “Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model”, The Journal of Motor Behavior, 19(4):486-517, 1987.

[Guiard & al. 01]
Guiard, Y., Bourgeois, F., Mottet, D., & Beaudouin-Lafon, M., “Beyond the 10-bit barrier: Fitts’ law in multiscale electronic worlds”, People and Computer

XV - Interaction without frontier (Joint proceedings of HCI 2001 and IHM 2001), 2001.

[Gutiérrez Alonso 05]
Gutiérrez Alonso, M.A., “Semantic Virtual Environments”, Thèse de doctorat, Faculté Informatique et Communications, EPFL, Lausanne, 2005.

[Hall & al. 05]
Hall, D., Nascimento, J., Ribeiro, P., Andrade, E., Moreno, P., Pesnel, S., List, T., Emonet, R., Fisher, R. B., Santos-Victor J. and Crowley, J. L., "Comparison of target detection algorithms using adaptive background models", In of Proc. the 2nd Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, (VS-PETS), 2005.

[Hall & al. 06]
Hall, D., Emonet, R. and Crowley J.L., “An Automatic Approach for Parameter Selection in Self-Adaptive Tracking”, In International Conference on Computer Vision Theory and Applications (VISAPP), 2006.

[Hämäläinen & al. 05]
Hämäläinen, P., Ilmonen, T., Höysniemi, J. Lindholm, M. and Nykänen, A. “Martial arts in artificial reality”, In CHI ’05: Proceedings of the conference on Human factors in computing systems, 2005.

[Haritaoglu & al. 98]
Haritaoglu, I., Harwood, D. and Davis, L.S., « W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People », In International Conference on Automatic Face and Gesture Recognition, 1998.

[Haritaoglu & al. 00]
Haritaoglu, I., Harwood, D. and Davis, L.S., “W4: real-time surveillance of people and their activities”, In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

[Hasenfratz & al. 04]
Hasenfratz, J.M., Lapierre, M. and Sillion, F., “A Real-Time System for Full Body Interaction with Virtual Worlds”, In Eurographics Symposium on Virtual Environments, 2004.

[Heguy & al. 01]
Heguy, O., Rodriguez, N., Luga, H., Jessel, J.P., & Duthen, Y., "Virtual Environment for cooperative Assistance in Teleoperation", In WSCG 2001, p. 9-13, 2001.

[Hick 52]
Hick, W. E., “On the Rate of Gain of Information”, Quarterly Journal of Experimental Psychology, 1952.

[Ho & Dyer 86]

Ho, S.-B. and Dyer, C., “Shape smoothing using medial axis properties”, IEEE Transactions on Pattern Recognition and Machine Intelligence 8(4), 512–520, 1986.

[Hoff III & al. 00]
Hoff III, K. E., Culver, T., Keyser, J., Lin M., and Manocha D., "Fast computation of generalized voronoi diagrams using graphics hardware" Proc. of ACM Symposium on Computational geometry, 375–376, 2000.

[Hohl & al. 04]
Hohl, F., Mehrmann, L. and Hamdan A., « A Context System for a Mobile Service Platform », In Proc. of the International Conference on Architecture of Computing System, 2004.

[Hommel & al. 00]
Hommel, B., Pösse, B. and Waszak, F., “Contextualization in perception and action”, Psychologica Belgica, 40, 227-245, 2000.

[Hopper,99]
Hopper, A., “Sentient Computing”, The Clifford Paterson Lecture, Phil. Trans. R. Soc. Lond. A, 1999.

[Horprasert & al. 99]
Horprasert, T., Harwood, D. and Davis, L.S., “A statistical approach for real-time robust background subtraction and shadow detection”, In IEEE ICCV’99 Frame-Rate Workshop, 1999.

[Horprasert & al. 00]
Horprasert, T., Harwood, D. and Davis, L.S., “A Robust Background Subtraction and Shadow Detection, In Proc. of the Asian Conference Computer Vision, 2000.

[Hull & al. 97]
Hull, R., Neaves, P., and Bedford-Roberts, J., “Towards situated computing”, In Proc. of the First International Symposium on Wearable Computers, Cambridge, Massachusetts, October 1997.

[Hutchins & al. 86]
Hutchins, E. L., Hollan, J. D. and Norman, D. A., “Direct manipulation interfaces”, In Norman, D.A. and Draper, S. W., editors, User Centered System Design: New Perspectives on Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, NJ and London, pp. 87-124, 1986.

[Inagaki & al. 92]
Inagaki, H., Sugihara, K. and Sugie, N. “Numerically Robust Incremental Algorithm for Constructing Three-dimensional Voronoi Diagrams”, In Proc. 4th Canad. Conf. Comp. Geom., pgs 334-339, 1992.

[Ishii & Ullmer 97]

Ishii, H. & Ullmer, B., “Tangible bits: Towards seamless interfaces between people, bits and atoms”, In Proc. of ACM CHI 97 Conference on Human Factors in Computing Systems, 1997.

[Ivanov & al. 00]
Ivanov, D., Kuzmin, E. and Burtsev, S., « An Efficient Integer-based Skeletonization Algorithm », In Computers & Graphics, 2000.

[JeuxVideo.com 06]
JeuxVideo.com, « GamePlay émergent », http://www.jeuxvideo.com/dossiers/00006203/le-gameplay-emergent.htm, 2006.

[John & Gray 95]
John, B. E. & Gray, W. D., “CPM-GOMS: An Analysis Method for Tasks with Parallel Activities”, In Proc. of CHI’95 Mosaic Of Creavity, 1995.

[John & Kieras 96a]
John, B. E. & Kieras, D. E, “The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast”, In Proc. of ACM Transactions on Computer-Human Interaction, 1996.

[John & Kieras 96b]
John, B. E. & Kieras, D. E, “Using GOMS for User Interface Design and Evaluation: Which Technique ?”, In Proc. of ACM Transactions on Computer-Human Interaction, 1996.

[John & al. 02]
John, B., Vera, A., Matessa, M., Freed, M., & Remington, R., “Automating CPM-GOMS”, In Proc. of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves, 2002.

[Kalman 60]
Kalman, R. E., “A New Approach to Linear Filtering and Prediction Problems”, Transactions of the ASME--Journal of Basic Engineering, 1960.

[Katsurada & al. 03]
Katsurada, K., Nakamura, Y., Yamada, H., Nitta, T., “XISL: a language for describing multimodal interaction scenarios”, In Proc. of ICMI’03, 2003.

[KaewTraKulPong & Bowden 01]
KaewTraKulPong, P., Bowden, R., “An improved adaptive background mixture model for real-time tracking with shadow detection”, In Proceedings of Second European Workshop on Advanced Video-based Surveillance Systems, 2001.

[KaewTraKulPong & Bowden 03]
KaewTraKulPong, P., Bowden, R., “A Real Time Adaptive Visual Surveillance System for Tracking Low-resolution Colour Targets in Dynamically Changing Scenes”, In Image and Vision Computing, 2003.

[Kay 77]
Kay, A., “Microelectronics and the Personnal Computer”, 1977.

[Kieras & Polson 85]
Kieras, D.E. & Polson P.G., “An approach to the formal analysis of the user complexity”, International Journal of Man Machine Studies, 1985.

[Kieras & Meyer 96]
Kieras, D.E. & Meyer, D.E., “The EPIC Architecture: Principles of Operation”, 1996.

[Kieras 97]
Kieras, D. E., “A Guide to GOMS model usability evaluation using NGOMSL”. In M. Helander, T. Landauer, and P. Prabhu (Eds.), Handbook of human-computer interaction (Second Edition), 1997.

[Kim & al. 05]
Kim, K., Chalidabhongse, T.H., Harwood, D. and Davis, L., “Real-time foreground-background segmentation using codebook model”, In Real-Time Imaging, 2005.

[Kontio & al. 04]
Kontio, J., Lehtola, L., & Bragge, J., “Using the focus group method in software engineering: obtaining practitioner and user experiences”, ISESE 2004, 2004.

[Kristensen & al. 06]
Kristensen, F., Nilsson, P. and O’Wall, V., “Background segmentation beyond RGB”, In Asian Conference on Computer Vision, 2006.

[Krueger 83]
Krueger, M., “Artificial Reality”, Addison-Wesley, 1983.

[Krueger 85]
Krueger, M., “VIDEOPLACE: A Report from the Artificial Reality Laboratory”, Leonardo, 18(3):145-151, 1985.

[Krzywinski 09]
Krzywinski, A., "RoboTable: A Tabletop Framework for Tangible Interaction with Robots in a Mixed Reality", In Proc. of ACE 2009, 2009.

[Javed & al. 02]
Javed, O., Shafique, K., and Shah, M., **“**A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information”, In Proc. of the Workshop on Motion and Video Computing, 2002.

[Lavender & al. 92]

Lavender, D., Bowyer, A., Davenport, J., Wallis, A. and Woodwark, J. “Voronoi Diagrams of Set-theoretic Solid Models” IEEE Computer Graphics and Applications, 12(5):69-77, Sept 1992.

[Lee 05]
Lee, D.S., “Effective Gaussian mixture learning for video background substraction”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005.

[Loke 05]
Loke, S. W., “Representing and reasoning with situations for context-aware pervasive computing: A logic programming perspective”, Knowledge Eng. Rev., vol. 19, no. 3, pp. 213–233, 2005.

[Lucquiaud 05]
Lucquiaud, V., « Proposition d'un noyau et d'une structure pour les modèles de tâches orientés utilisateurs », In Proc. of the 17th French-speaking conference on Human-computer interaction, 2005.

[Lyytinen & Yoo 02]
Lyytinen, K. and Yoo, Y., “Research Commentary: The Next Wave of Nomadic Computing”, Info. Sys. Research, 2002.

[Maass 97]
Maass, H., “Location-aware mobile applications based on directory services”, In Proc. of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 23-33, 1997.

[Mackay 96]
Mackay, W. E., « Réalité augmentée : le meilleur des deux mondes », La Recherche, numéro spécial 'L'ordinateur au doigt et à l'œil', 284, 1996.

[Mackay 98]
Mackay, W.E., “Augmented Reality: linking real and virtual worlds”, In Proc. of ACM AVI '98, Conference on Advanced Visual Interfaces, 1998.

[Maes & al. 94]
Maes, P., Darrell, T., Blumberg, B. and Pentland, A., “The ALIVE System: Wireless, Full-body Interaction with Autonomous Agents”, M.I.T. Media Laboratory Perceptual Computing Technical Report No. 257, 1994.

[Mann 96]
Mann, S., 'Smart Clothing': Wearable multimedia computing and 'personal imaging' to restore the technological balance between people and their environments.”, In Proc. of the Fourth ACM Multimedia Conference (MULTIMEDIA'96), pages 163-174, New York, NY, USA. ACM Press, 1996.

[Martins & al. 09]
Martins, T. & al., "Headbang Hero", In Proc of ACE 2009, 2009.

[McGuffin & Balakrishnan 02]
McGuffin, M. & Balakrishnan, R., “Acquisition of Expanding Targets”, In Proc. of ACM Human Factors in Computing Systems, CHI '02, 2002.

[McKenna & al. 00]
McKenna, S.J., Jabri, S., Duric, Z. and Wechsler, H., “Tracking Interacting People”, In International Conference on Automatic Face and Gesture Recognition, 2000.

[Michaud 08]
Michaud, L., “Serious games – Advergaming, edugaming, training…”, Etude IDATE Consulting & Research, 2008.

[Micheli 06]
Micheli, R., « Contexte et contextualisation en analyse du discours : regard sur les travaux de T. Van Dijk », Semen [En ligne], 21 | 2006. URL : http://semen.revues.org/1971.

[Michoud & al. 07]
Michoud, B., Guillou, E. & Bouakaz, S., “Real-Time and Markerless 3D Human Motion Capture Using Multiple Views”, In Workshop on Human Motion, 2007.

[Michoud 09]
Michoud, B., “Reconstruction 3D à partir de séquences vidéo pour la capture de mouvements de personnages en temps réel et sans marqueur”, PhD Thesis, LIRIS, France, 2009.

[Milgram & Kishino 94]
Milgram, P. & Kishino, F., “A taxonomy of mixed reality visual displays”, IEICE Transactions on Information Systems, 1994.

[Min & al. 04]
Min, J., Powell, M., and Bowyer, K. “Automated performance evaluation of range image segmentation algorithms” In IEEE Trans. on Systems Man and Cybernetics, 2004.

[Minsky 75]
Minsky, M., "A Framework for Representing Knowledge", In The Psychology of Computer Vision, P. Winston, Ed., McGraw Hill, New York, 1975.

[Moeslund & Granum 01]
Moeslund, T. B., & Granum E., “A Survey of Computer Vision-Based Human Motion Capture”, Comput. Vis. Image Underst, 2001.

[Moeslund & al. 06]
Moeslund, T. B., Hilton, A., and Krüger, V., “A survey of advances in vision-based human motion capture and analysis”, Comput. Vis. Image Underst, 2006.

[Montanari 69]
Montanari, U., “Continuous skeletons from digitized images”, J. Assoc. Comput. Machinery 16(4), 534–549, 1969.

[Mori & al. 08]
Mori, H., Fujieda, T., Shiratori, K., and Hoshino, J., “Kangaroo: the trampoline entertainment system for aiding exercise”. Proc. Advances in Computer Entertainment Technology, vol. 352, 414-414, 2008.

[Morris 78]
Morris, D., « La clé des gestes », Bernard Grasset, 1978.

[Myers 98]
Myers, B.A., "A Brief History of Human Computer Interaction Technology", ACM Interactions, Vol. 5 (2), pp. 44-54, http://www.cs.cmu.edu/~amulet/papers/uihistory.tr.html, 1998.

[Negroponte 95]
Negroponte, N., “Being Digital”, Paperback edition, 1995.

[Newell 90]
Newell, A., “Unified Theories of Cognition”, Cambridge MA: Harvard University Press, 1990.

[Nigay & Coutaz 93]
Nigay, L. & Coutaz, J., “A design space for multimodal systems: Concurrent processing and data fusion”, In Proc. of ACM INTERCHI'93 Conference on Human Factors in Computing Systems, Voices and Faces, 1993.

[Nigay & Coutaz 95]
Nigay, L. & Coutaz, J., “A Generic Platform for Adressing the Multimodal Challenge”, In Proc. of the SIGCHI Conference on Human Factors in Computing Systems, 1995.

[Nigay & Gray 06]
Nigay, L. & Gray, P., “Interactive systems & new interface technologies: Engineering for Multimodal Human-Computer Interaction”, In European Workshop on Interactive systems & new interface technologies, 2006.

[Norman & Draper 86]
Norman, D.A. et Draper, S.W, “User Centered System Design; New Perspectives on Human-Computer Interaction”, Lawrence Erlbaum Associates, Inc., 1986.

[Norman 88]
Norman, D. N., “The design of everyday things”, New York: Doubleday, 1988.

[Norman,98]
Norman, A. D., “The Invisible Computer”, Cambridge, Massachusetts; MIT Press. 1998.

[Noury & al. 09]
Noury, N., Fleury, A., Nocua, R., Poujaud, J., Gehin, C., Dittmar, A., Delhomme, G., Demongeot, J. & MacAdams, E., « Capteurs pour la télésurveillance médicale - Capteurs, algorithmes et réseaux », ITBM RBM, 2009.

[Ogniewicz & Ilg 92]
Ogniewicz, R. and Ilg, M., "Voronoi skeletons: Theory and applications" In Proc. Conf. on Computer Vision and Pattern Recognition, 63–69, 1992.

[Ogniewicz & Kübler 95]
Ogniewicz, R. and Kübler, O., "Hierarchic Voronoï Skeletons" In Pattern Recognition, Vol. 28, 3, 1995.

[Oh & Jung 09]
Oh, Y. & Jung, K., "Vision-Based Korean Manual Alphabet Recognition Game for Beginners", In Proc. Of the International Conference on Advances in Computer Entertainment Technology, 2009.

[Okabe & al. 00]
Okabe, A., Boots, B. and Sugihara, K. "Spatial Tessellations: Concepts and Applications of Voronoi Diagrams", John Wiley & Sons, Chichester, UK, 2000.

[Olsen & Klemmer 05]
Olsen, D. R. & Klemmer, S. R., "The Future of User Interface Design Tools", ACM CHI 2005 Workshop, 2005.

[Oreizy & al. 99]
Oreizy, P., Gorlick, M.M., Taylor, R., N, Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, David, A., Rosenblum, S. and Wolf. A., L., « An Architecture-Based Approach to Self-Adaptive Software », IEEE Intelligent Systems, 1999.

[O'Sullivan 03]
O'Sullivan, S., "An Empirical Evaluation Of Map Building Methodologies in Mobile Robotics Using The Feature Prediction Sonar Noise Filter And Metric Grid Map Benchmarking Suite", Master Thesis. University of Limerick, November, 2003.

[Pascoe 98]
Pascoe, J., "Adding Generic Contextual Capabilities to Wearable Computers", In Proc. of 2nd International Symposium on Wearable Computers, 1998.

[Pascoe & al. 99]
Pascoe J., Ryan N.S., Morse D.R., "Issues in developing context-aware computing", In Handheld and Ubiquitous Computing, 1999.

[Paterno 99]

Paterno, F., “Model-Based Design and Evaluation of Interactive Applications”, Springer Verlag, 1999.

[Perng & al. 99]
Perng, J. K., Fisher, B., Hollar, S., Pister, K. S. J., "Acceleration Sensing Glove", In Proc. of Third International Symposium on Wearable Computers (ISWC'99), 1999.

[Perreira Da Silva & al. 08a]
Perreira Da Silva, M., Courboulay, V., Prigent, A., Estraillier, P., « Real-time face tracking for attention aware adaptive games », ICVS 2008, 2008.

[Perreira Da Silva & al. 08b]
Perreira Da Silva, M., Courboulay, V., Prigent, A., Estraillier, P., « Adaptivité et Interactivité : Vers un système de vision comportemental », Actes de la Manifestation des Jeunes Chercheurs en Sciences et Technologies de l’Information et de la Communication, 2008.

[Perreira Da Silva & al. 09]
Perreira Da Silva, M., Courboulay, V., Prigent, A. and Estraillier, P., « Fast, low resource, head detection and tracking for interactive applications », In Psychology Journal, Volume 7, Number 3, 243-264, 2009.

[Petri 73]
Petri, C.A., “Concepts of Net Theory”, In Proc. of Symposium and Summer School, High Tatras, Sep. 3--8, 1973.

[Picard & Estraillier 08a]
Picard, F., Estraillier, P., “Extraction contextualisée de silhouettes”, Actes de la Manifestation des Jeunes Chercheurs en Sciences et Technologies de l’Information et de la Communication, 2008.

[Picard & Estraillier 08b]
Picard, F., Estraillier, P., “Motion Capture System Contextualization - Application to game development”, Proc. of Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games, 2008.

[Picard & Estraillier 08c]
Picard, F., Estraillier, P., “Motion Capture System Contextualization”, Proc. of Advances in Computer Entertainment Technology, 2008.

[Picard & Estraillier 09]
Picard, F., Estraillier, P., “Enhancing a Motion Capture Interface by Introducing Context Management”, Proc. of Advances in Computer Entertainment Technology, 2009.

[Picard & Estraillier 10]
Picard, F., Estraillier, P., “Context-dependent Player’s Movement Interpretation – Application to Adaptive Game Development”, Proc. of 3D Image Processing & Applications, 2010.

[Pizer & al. 87]
Pizer, S. M., Oliver, W. R. and Bloomberg, S. H., “Hierarchical shape description via the multiresolution symmetric axis transform”, IEEE Transactions on Pattern Recognition and Machine Intelligence 9(4), 505–511, 1987.

[Plesca & al. 08]
Plesca, C., Charvillat, V., Grigoras, R., "Adapting Content Delivery to Limited Resources and Inferred User Interest", In International Journal of Digital Multimedia Broadcasting, 2008.

[Porikli 05]
Porikli, F., “Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images”, In Proc. of IEEE Motion Multi-Workshop, 2005.

[Porikli & Tuzel 05]
Porikli F. & Tuzel, O., “Bayesian background modeling for foreground detection”, In Proc. of ACM Visual Surveillance and Sensor Network, 2005.

[Porikli & Wren 05]
Porikli, F. & Wren, C., “Change detection by frequency decomposition: Wave-back,” In Proc. of Workshop on Image Analysis for Multimedia Interactive Services, 2005.

[Porikli 06]
Porikli, F., « Achieving real-time object detection and tracking under extreme conditions », In the Journal or Real-Time Image Processing, 2006.

[Preece & al. 94]
Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T., “Human-Computer Interaction”, Addison-Wesley Publishing, Reading, Mass, OCLC 35598754, 1994.

[Quillian 68]
Quillian, M. R., "Semantic Memory", In Semantic Information Processing, MIT Press, Cambridge, 1968.

[Rabiner 89]
Rabiner, L.R. “A tutorial on hidden markov models and selected applications in speech recognition”, Proceedings of the IEEE, vol.7, n°2, 1989.

[Ranganathan & al. 04]
Ranganathan A., Al-Muhtadi J., and Campbell R. H., “Reasoning about Uncertain Contexts in Pervasive Computing Environments”, In IEEE Pervasive Computing, Vol.3, No. 2, pp. 62-70, 2004.

[Rempulski & al. 08]

Rempulski, N., Prigent, A., Estraillier, P., "Adaptive Storytelling based on model-checking approaches", Proceedings of CGames'2008, 13th international conference on Computer Games, 2008.

[Renevier-Gonin 10]
Renevier-Gonin, P., Cours sur les Nouveaux Moyens d'Interaction (NMI), Master IFI – Polytech Nice-Sophia Antipolis, Université de Nice-Sophia Antipolis, 2010.

[Robertson & Brady 99]
Robertson, P. & Brady, J., « Adaptive image analysis for aerial surveillance », IEEE Intelligent Systems, 1999.

[Roques 09]
Roques, P., « UML 2 par la pratique : Etudes de cas et exercices corrigés », Edition Eyrolles, 2009.

[Rosenfeld & Pfaltz 68]
Rosenfeld, A., and Pfaltz, J., "Distance functions on Digital Pictures", Pattern Recognition, Vol. 1, pp 33-61, 1968.

[Rouillard 08]
Rouillard, J., "Adaptation en contexte : contribution aux interfaces multimodales et multicanal", Habilitation à Diriger des Recherches, Laboratoire d'Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, 2008.

[Roussarie 06]
Roussarie, L. "Contexte", Sémanticlopédie: dictionnaire de sémantique, 2006.

[Sage 77]
Sage, G. H., "Introduction to Motor Behavior, A Neuropsychological Approach (2nd ed)", Addison-Wesley Publishing Company, 1977.

[Sakoe & Chiba 78]
Sakoe, H. & Chiba, S., "Dynamic programming algorithm optimization for spoken word recognition", IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(1):43–49, 1978.

[Salber & al. 99]
Salber, D., Dey, A.K., and Abowd, G.D., "The context toolkit: Aiding the development of context-enabled applications", In Proc. of the CHI 99 Conference on Human Factors in Computing Systems, 1999.

[Samaan 06]
Samaan, K., « Prise en compte du modèle d'interaction dans le processus de construction et d'adaptation d'applications interactives », Thèse de doctorat d'informatique, Ecole Centrale de Lyon, France, 2006.

[Saponas & al. 09]

Saponas, T. S., Tan, D. S., Morris, D, Balakrishnan, R., Landay, J.A., and Turner, J., “Enabling Always-available Input with Muscle-Computer Interfaces”, Proc. of ACM UIST, 2009.

[Schank & Abelson 77]
Schank, R. C. & Abelson, R. P., “Scripts, Plans, Goals and Understanding”, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[Schank 99]
Schank, R. C., Dynamic memory revisited, Cambridge University Press, New York, NY, USA, 1999.

[Schilit & al. 93]
Schilit, B. N., Theimer, M. M. and Welch, B. B, “Customizing Mobile Application”, In USENIX Symposium on Mobile and Location-independent Computing, p, 129-138, 1993.

[Schilit & al. 94]
Schilit, B. N., Adams N. I. and Want, R., “Context-Aware Computing Applications”, In Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, pp 85-90, IEEE Computer Society, December 1994.

[Schlienger & al. 06]
Schlienger, C., Dragicevic, P., Ollagnon, C. & Chatty, S., « Les transitions visuelles différenciées : principes et applications », Actes d'IHM 2006, 2006.

[Schmidt & al. 99a]
Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Laerhoven, K. V., and Velde, W. V., “Advanced Interaction in Context”, In Proceedings of the 1st international Symposium on Handheld and Ubiquitous Computing, 1999.

[Schmidt & al. 99b]
Schmidt A., Beigl M. and Gellersen H.-W., “There is More to Context than Location”, Computers and Graphics, Volume 23, Number 6, pp. 893-901(9), December 1999.

[Schmidt 02]
Schmidt A., “Ubiquitous Computing – Computing in Context”, PhD Thesis, Lancaster University, Lancaster, UK, 2002.

[Schwartz & al. 04]
Schwartz, J-L., Berthommier, F. and Savariaux, C., "Seeing to hear better: evidence for early audio-visual interactions in speech identification", Cognition 93, B69–B78, 2004.

[Sears & Jacko 08]
Sears, A. & Jacko, J.A., “The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications”, CRC, 2009.

[Sears & Jacko 09]
Sears, A. & Jacko, J.A., "Human-Computer Interaction Fundamentals", CRC, 2009.

[Shaer & al. 04]
Shaer, O., Leland, N., Calvillo-Gamez, E. H., and Jacob, R. J., "The TAC paradigm: specifying tangible user interfaces", Personal Ubiquitous Comput. 8, 5, 359-369, 2004.

[Shamos & Hoey 75]
Shamos M.I. and Hoey, D. "Closest-point Problems", In Proc. 16th Annual IEEE Symposium on Foundations of Comp. Sci., pages 151-162, 1975.

[Sheehy & al. 95]
Sheehy, D.J., Armstrong, C.G. and Robinson, D.J. "Computing the Medial Surface of a Solid from a Domain Delaunay Triangulation", In Proc. ACM/IEEE Symp. on Solid Modeling and Applications, May 1995.

[Shneiderman 83]
Shneiderman, B., "Direct Manipulation: A step beyond programming languages", IEEE Computer, Vol. 8, p. 57-69, 1983.

[Shneiderman 98]
Shneiderman, B., "Designing the User Interface", Addison Wesley Longman, 3e edition, 1998.

[Sire & Chatty 02]
Sire, S., Chatty, C., "The Markup Way to Multimodal Toolkits", In W3C Multimodal Interaction Workshop, 2002.

[Sminchisescu 02]
Sminchisescu, C., "Estimation Algorithms for Ambiguous Visual Models", PhD. Thesis, Institut National Polytechnique de Grenoble, Laboratoire GRAVIR, 2002.

[Smith 82 & al.]
Smith, D.C., Irby, C., Kimball, R. and Verplank, B., "Designing the Star User Interface", Byte, 7(4), pp. 242-282, April 1982.

[Smith 87]
Smith, R.W., "Computer processing of line images: A survey", Pattern Recognition, vol. 20, n°1, pp. 7-15, 1987.

[Stanciulescu & al. 05]
Stanciulescu, A., Limbourg, Q., Vanderdonckt, J., Michotte, B., Montero, F., "A transformational approach for multimodal web user interfaces based on UsiXML", In Proc. of ICMI 2005, 2005.

[Stauffer & Grimson 99]

Stauffer, C. & Grimson, W.E.L., "Adaptive background mixture models for real-time tracking", In Computer Vision and Pattern Recognition, 1999.

[Stauffer & Grimson 00]
Stauffer, C., Eric, W., Grimson, L., "Learning patterns of activity using real-time tracking", In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

[Strang & Linnhoff-Popien 04]
Strang T. and Linnhoff-Popien C., "A Context Modelling Survey", Workshop on Advanced Context Modelling, Reasoning and Management, In Ubicomp'04, 2004.

[Suchman 87]
Suchman, L., "Plans and Situated Actions: The Problem of Human-Machine Communication", Cambridge University Press, 1987.

[Sugihara & Iri 94]
Sugihara K. and Iri, M., "A Robust Topology-oriented Incremental Algorithm for Voronoi Diagrams", International Journal of Comp. Geom. Appl., 4:179-228, 1994.

[Sutherland 63]
Sutherland, I.E., "SketchPad: A Man-Machine Graphical Communication System", In AFIPS Spring Joint Computer Conference, 23. pp. 329-346, 1963.

[Tabary & Barthet 01]
Tabary, J.C, Barthet, M.F, « Analyse et modélisation des tâches dans la conception des systèmes d'information : la méthode Diane+ », Analyse et Conception de l'IHM, Interaction homme-machine pour les SI1, 2001.

[Tarpin-Bernard 06]
Tarpin-Bernard, F., « Interaction Homme-Machine Adaptative », HDR, ICTT, Lyon, France, 2006.

[Teichmann & Teller 97]
Teichmann, M. and Teller, S., "Polygonal Approximation of Voronoi Diagrams of a Set of Triangles in Three Dimensions" Tech Rep 766, Lab of Comp. Sci., MIT, 1997.

[Thevenin & Coutaz 99]
Thevenin, D., Coutaz, J., "Plasticity of User-Interfaces: Framework and Research Agenda", In Proc. Of the 7th IFIP Conference on Human-Computer Interaction (INTERACT 1999), 1999.

[Thevenin 01]
Thevenin, D., « Adaptation en Interaction Homme-Machine : le cas de la Plasticité », Thèse de Doctorat, Université Joseph Fournier, Grenoble, 2001.

[Thomas & Calder 01]

Thomas, B.H. & Calder, P., “Applying Cartoon Animation Techniques to Graphical User Interfaces”, ACM Transactions on Computer-Human Interaction, 2001.

[Totterdell & Rautenbach 90]
Totterdell, P., & Rautenbach, P., “Adaptation as a Problem Design”, In Adaptive User Interfaces, 1990.

[Toyama & al. 99]
Toyama, K., Krumm, J., Brumitt, B., Meyers, B., “Wallflower: Principles and practice of background maintenance”, In Proceedings of IEEE International Conference on Computer Vision, 1999.

[Trigg & al. 87]
Trigg, R., Moran, T. and Halasz, F., “Adaptability and Tailorability in NoteCards”, in Bullinger and Shackel (Eds.) Proc. INTERACT `87, 1987.

[Turk 00]
Turk, M., “Perceptive Media: Machine Perception and Human Computer Interaction”, In Chinese Journal of Computers, Vol. 23, No. 12, pp. 1235-1244, 2000.

[Ullmer & Ishii 00]
Ullmer, B., Ishii, H., “Emerging frameworks for tangible user interfaces”, IBM Syst. J., 2000.

[Urtasun & al. 05]
Urtasun, R., Fleet, D. and Fua, P., “Monocular 3D Tracking of the Golf Swing”, In Conference on Computer Vision and Pattern Recognition, 2005.

[Vacchetti 04]
Vacchetti, L., “Multi modal tracking in complex environments for augmented reality applications”, Thèse de doctorat, CvLab, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Suisse, 2004.

[Van Oers 98]
Van Oers, B., “From Context to Contextualizing”, In Learning and Instruction, Volume 8, Number 6, pp. 473-488(16), 1998.

[van Welie & al. 98]
van Welie, M., van der Veer, G.C., Eliëns, A., « Euterpe - Tool support for analyzing cooperative environments », In Proc of the Ninth European Conference on Cognitive Ergonomics, 1998.

[Vernier & Nigay 00]
Vernier, F. & Nigay, L., “A Framework for the Combination and Characterization of Output Modalities”, In Proc. of DSV-IS2000, 2000.

[Vlasic & al. 07]
Vlasic, D. & al., « Practical Motion Capture in everyday Surroundings », 2007.

[Voronoï 08]
Voronoi, G.M., « Nouvelles Applications des Paramètres Continus à la Théorie des Formes Quadratiques », Deuxième Mémoire : Recherches sur les Paralléllоèdres Primitifs. J. Reine Angew. Math., 134:198-287, 1908.

[W3C MIA 02]
World Wide Web Consortium (W3C) Multimodal Interaction Activity, http://www.w3.org/2002/mmi/, 2002.

[Ware & Balakrishnan 94]
Ware, C. & Balakrishnan, R. "Reaching for Objects in VR Displays: Lag and Frame Rate", ACM Transactions on Computer-Human Interaction (TOCHI), Vol. 1, No. 4, pages 331-356, December 1994.

[Weinland 08]
Weinland, D., "Action Representation and Recognition", Thése de doctorat, Institut National Polytechnique de Grenoble, Grenoble, 2008.

[Weiser 91]
Weiser, M., "The Computer for the 21st Century", Scientific American Special Issue on Communications, Computers and Networks, 1991.

[Weiser 93]
Weiser, M., « Some computer science issues in ubiquitous computing », In Communications of the ACM, 36, 7, 1993.

[Weissberg 00]
Weissberg, J.L., "Présences à distance: Déplacement virtuel et réseaux numériques : pourquoi nous ne croyons plus à la télévision", Edition L'Harmattan, 2000.

[Wejchert 00]
Wejchert, J. "The Disappearing Computer", Information Document, IST Call for proposals, European Commission, Future and Emerging Technologies, February 2000.

[Wellner & al. 93]
Wellner, P., Gold, R. & Mackay, W., Special issue on computer-augmented environments, Communications of the ACM, 36(7), July 1993.

[Wilson 06]
Wilson, C.E., « Brainstorming pitfalls and best practices », In Interactions 13, 5, 2006.

[Wren & al. 97]
Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A. P. "Pfinder - Real-time tracking of the human body" In IEEE Transactions on pattern analysis and machine intelligence, 1997.

[Wu & al. 02]
Wu H., Siegel M., Ablay S., “Sensor Fusion for Context Understanding”, IEEE Instrumentation and Measurement, Technology Conference, 2002.

[Xia 89]
Xia, Y., “Skeletonization via the realization of the fire front’s propagation and extinction in digital binary shapes”, IEEE PAMI, vol. 11, n° 10, pp. 1076-1086, 1989.

[Yang & al. 04a]
Yang, T., Li, S., Pan, Q. and Li, J., "Real time and accurate segmentation of moving objects in dynamic scene", ACM Video surveillance and sensor networks, 2004.

[Yang & al. 04b]
Yang, T., Li, S.Z., Pan, Q., Li, J. “Multiple layer based background maintenance in complex environment” In Proceedings of the third International Conference on Image and Graphics, 2004.

[Ye & al. 03]
Ye, G., Corso, J., Burschka, D. and Hager, G.D., “VICs: A Modular Vision-Based HCI Framework”, In Lecture Notes in Computer Science, 2003.

[Yim & Graham 07]
Yim, J., and Graham, T.C.N., “Using games to increase exercise motivation”. Proc. Future Play, 166-173, 2007.

[Zaidenberg & al. 06]
Zaidenberg, S., Brdiczka, O., Reignier and P., Crowley, J.L., “Learning context models for the recognition of scenarios”, In 3rd IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI), 2006.

[Zaidenberg & al. 09]
Zaidenberg, S., Reignier, P. and Crowley, J.L., “An Architecture for Ubiquitous Applications”, In Ubiquitous Computing and Communication Journal (UBiCC), Volume 4, Number 2, 2009.

[Zhang & al. 09]
Zhang, D., Cai, Z., Chen, K. and Nebel, B., “A Game Controller Based on Multiple Sensors”, Proc. Advances in Computer Entertainment Technology, 2009.

[Zivkovic04]
Zivkovic, Z., “Improved adaptive Gaussian mixture model for background subtraction”, In International Conference on Pattern Recognition, 2004.