

*Projet siames**Synthèse d'Images, Animation,  
Modélisation et Simulation**Rennes*

THÈME 3B

 *Rapport  
d'Activité*

2002



# Table des matières

<b>1. Composition de l'équipe</b>	<b>1</b>
<b>2. Présentation et objectifs généraux</b>	<b>2</b>
<b>3. Fondements scientifiques</b>	<b>3</b>
3.1. Panorama	3
3.2. Simulation d'éclairage et synthèse d'image	3
3.2.1. Simulation d'éclairage et rendu temps réel	4
3.3. Modèles dynamiques de mouvement	5
3.4. Modélisation et simulation comportementale	6
<b>4. Domaines d'application</b>	<b>8</b>
4.1. Panorama	8
4.2. La réalité virtuelle, la réalité augmentée et la téléopération	8
4.3. Humanoïdes virtuels et modèles biomécaniques	8
4.4. Le multimédia, l'audiovisuel et les jeux	9
4.5. Les modèles mécaniques et le prototypage virtuel	9
<b>5. Logiciels</b>	<b>10</b>
5.1. Panorama	10
5.2. Logiciel de simulation d'éclairage	10
5.3. OpenMASK : Plate-forme, OpenSource, de simulation en réalité virtuelle	11
5.4. Définition d'un outil pour l'entraîneur sportif	13
<b>6. Résultats nouveaux</b>	<b>14</b>
6.1. Navigation en temps réel dans des scènes complexes à travers un réseau bas débit	14
6.1.1. Architecture générale de la plate-forme :	15
6.1.2. Utilisation de la plate-forme :	16
6.2. Génération de villes	19
6.3. Modèles mécaniques et prototypage virtuel.	19
6.3.1. Présentation des résultats pour l'endoscope	20
6.4. Simulation avec retour d'efforts.	20
6.4.1. Contexte et Objectifs	20
6.4.2. Identification des besoins	20
6.4.3. Architecture	20
6.4.4. Généralisation des types des flux d'entrées/sorties	22
6.5. Interactions en Univers Virtuels 3D	22
6.6. Animation référencée vision et comportements	26
6.6.1. Cinématographie virtuelle	26
6.6.2. simulation de la perception visuelle d'un humanoïde de synthèse	27
6.7. Modélisation d'environnements urbains informés pour les agents autonomes	28
6.8. BCOOL : Behavioral and Cognitive Object Oriented Language	29
6.9. Navigation d'humanoïdes en environnement complexe	30
<b>7. Contrats industriels</b>	<b>32</b>
7.1. FranceTelecom : Modélisation d'Environnements Urbains en vue de Visualisation déportée, interactive et réaliste.	32
7.2. RNTL PERFRV : Plate-forme française de réalité virtuelle	32
7.3. RNTL Mouvement : Estimation du mouvement	35
7.4. RNTL Dramachina : scénario pour la fiction interactive	37
7.5. RNTL Dynamicité : ville virtuelle peuplée d'entités autonomes	40
7.6. Giat-Industrie : Environnement virtuel de formation à la maintenance.	42
7.7. ODL Plate-forme OpenMASK	42

7.8. RNRT VTHD++	43
<b>8. Actions régionales, nationales et internationales</b>	<b>43</b>
8.1. Actions nationales	43
8.2. Actions financées par la Commission Européenne	44
<b>9. Diffusion des résultats</b>	<b>45</b>
9.1. Animation de la communauté scientifique	45
9.2. Enseignement universitaire	45
<b>10. Bibliographie</b>	<b>46</b>

# 1. Composition de l'équipe

## Responsable scientifique

Bruno Arnaldi [Professeur Insa de Rennes]

## Assistante de projet

Evelyne Livache [SAN Inria jusqu'au 1er novembre 2002]

Catherine Godest [SAN Inria depuis le 1er novembre 2002]

## Personnel Inria

Alain Chauffaut [IR]

Jean-Luc Nougaret [CR jusq'au 1 avril 2002]

Georges Dumont [CR détachement depuis le 1 septembre 2002]

Anatole Lecuyer [CR depuis le 1 novembre 2002]

## Personnel CNRS

Stéphane Donikian [CR]

## Université de Rennes 1

Kadi Bouatouch [Professeur]

Yves Bekkers [Professeur]

Thierry Duval [Maître de conférences]

Nicolas Courty [ATER depuis le 1er octobre 2002]

## Personnel ENS Cachan

Georges Dumont [Maître de conférence ENS Cachan jusqu'au 31 aout 2002]

## Ingénieurs Experts

Guillaume Bataille [Inria]

Frantz Degrigny [Inria]

Olivier Filangi [Inria jusqu'au 30 juin 2002]

Claudie Fourn [Inria jusqu'au 30 juin 2002]

Armel Cretual [Inria jusqu'au 30 septembre 2002]

Mathilde Vandenberghe [Inria]

Guillermo Andrade [Inria]

Christian Le Tenier [Inria]

Frédéric Devillers [Inria]

Gérald Choqueux [Inria]

Thierry Jouin [Inria jusqu'au 30 juin 2002]

## Chercheurs doctorants

Stéphane Menardais [Bourse MENRT]

Tangi Meyer [Bourse INRIA-Région]

Nicolas Courty [Bourse CNET]

Fabrice Lamarche [Bourse MENRT]

Romain Thomas [Bourse INRIA]

Jean-Eudes Marvie [Bourse INRIA]

Caroline Larboulette [Bourse MENRT]

Nicolas Mollet [Bourse INRIA]

Chadi El Zammar [Bourse INRIA]

Julien Perret [Bourse MENRT depuis le 1er octobre 2002]

Marwan Badawi [Bourse Libanaise depuis le 1er novembre 2002]

## Post-doctorant

David Margery [Post-doctorant industriel Inria]

**Ingénieur associé**

Michael Rouillé

**Collaborateurs extérieurs**

Gérard Hégron [Professeur École des Mines de Nantes]

Franck Multon [Laboratoire de physiologie et de biomécanique de l'exercice musculaire Rennes II]

## 2. Présentation et objectifs généraux

Le contexte général de nos travaux de recherche concerne la **simulation de systèmes complexes**. En effet, nos axes de recherche traitent de simulation d'éclairage, de simulation de modèles mécaniques, de contrôle de systèmes dynamiques, de simulation temps réel et de modélisation d'environnements virtuels.

Nos études s'organisent principalement autour de trois axes :

- **l'informatique graphique** : l'essentiel de nos travaux consiste à élaborer et intégrer des *modèles*, à définir des *algorithmes* et à étudier les *complexités* des solutions proposées ;
- **la simulation** : notre objectif principal est de pouvoir confronter les résultats produits par nos algorithmes à des valeurs numériques mesurées sur site réel, ceci afin de *valider* expérimentalement les approches et les concepts étudiés ;
- **l'organisation « système »** : pour développer les deux points précédents, nous devons être à même de traiter des cas « grandeur nature » et de valider nos approches par des mises en œuvre.

Plus précisément, les études s'articulent autour de trois champs d'activités complémentaires mais de problématiques distinctes :

- **la simulation d'éclairage** : les algorithmes de synthèse d'image réaliste permettent d'obtenir des résultats de très haute qualité par l'introduction de modèles d'éclairage fondés sur la physique, permettant d'évaluer les interactions entre la lumière et les objets ;
- **la simulation de systèmes physiques** : nous abordons la simulation de systèmes physiques sous l'angle des schémas de calcul nécessaires à la production des équations régissant ces systèmes. Nous étudions également la résolution de ces équations (approche symbolique / numérique). De plus, nous travaillons particulièrement sur les techniques de contrôle du mouvement de systèmes dynamiques (animaux, humanoïdes). Cette approche nous permet d'aborder les problèmes de simulation ou d'animation par ordinateur.
- **la modélisation et la simulation comportementale** : Afin de simuler le comportement humain (ou animal) lors de tâches spécifiques, nous nous intéressons à la réalisation d'outils de spécification et de simulation du comportement d'entités dynamiques, autonomes mais néanmoins contrôlables, ainsi qu'à la modélisation de l'environnement dans lequel ces « acteurs » vont évoluer. Pour le comportement individuel ou collectif, nous devons prendre en compte les aspects continus (lien avec le système moteur) et discrets (lien avec le système cognitif) : il s'agit donc de systèmes hybrides. En ce qui concerne la modélisation de l'environnement, il s'agit de gérer non seulement les caractéristiques géométriques, mais aussi toutes les informations pertinentes pour les modèles comportementaux (nature topologique et sémantique de l'environnement géométrique).

Deux thèmes transversaux sont aussi activement explorés :

- **les environnements virtuels** : plus connu sous la dénomination de *Réalité Virtuelle*, ce secteur d'activité fait en effet intervenir les différents thèmes de recherche du projet (éclairage, animation, simulation, etc...). À travers des applications qui appartiennent aux domaines de la simulation ou de la téléopération, nous abordons ce champ applicatif en y incorporant pour une large part nos travaux de recherche.

- **les algorithmes parallèles** : la tendance actuelle est largement orientée vers l'utilisation de modèles de plus en plus complexes (forme, mouvement, rendu). Les conséquences directes en sont la forte augmentation des coûts de calcul liés à la production d'images fixes ou animées. Outre les recherches visant à réduire la complexité des algorithmes séquentiels, l'étude des schémas de parallélisation de ces algorithmes revêt un caractère fondamental. Ces travaux sont menés en étroite collaboration avec T. Priol du projet PARIS.

Notre activité de recherche nous a conduits à réaliser une **plate-forme logicielle de simulation** capable d'intégrer les différents composants de nos travaux dans un contexte de simulation temps réel distribuée. Cette plate-forme intègre les trois grandes familles de modèles de contrôle du mouvement :

les modèles descriptifs ou phénoménologiques qui sont utilisés pour reproduire uniquement les effets (mouvement, déformation), sans aucune connaissance a priori sur les causes qui pourraient les avoir produits. Ils décrivent la cinématique des phénomènes dynamiques.

les modèles générateurs ou fondés sur la physique qui décrivent les causes capables de produire un effet. Par exemple, les modèles utilisant la mécanique font partie des modèles générateurs.

les modèles comportementaux dont le but est de simuler le comportement d'individus vivants (plantes, animaux et êtres humains). Ces modèles définissent le comportement d'une entité, ses actions et ses réactions, de manière individuelle (animal, humain) ou collective (foule).

## 3. Fondements scientifiques

### 3.1. Panorama

Le projet s'intéresse de manière spécifique aux problèmes de simulation de systèmes dynamiques complexes incluant des besoins de restitution visuelle 3D des résultats, en temps réel ou différé selon la nature des phénomènes simulés. Dans le contexte général de l'informatique graphique, nos sujets d'étude portent essentiellement sur les couplages entre les calculs liés à des modèles de simulation complexes et la visualisation dynamique des résultats. Nos travaux portent sur les thèmes scientifiques suivants :

- l'étude des modèles dynamiques de mouvement pour l'animation et la simulation : ces travaux comprennent les problèmes liés à la modélisation des systèmes physiques, leur contrôle et les différents types d'interaction pouvant intervenir lors d'une simulation (guidage, collisions, contacts, ...);
- l'étude de la modélisation d'environnements dynamiques : la simulation d'entités comportementales s'appuie à la fois sur des modèles d'interaction entre entités, mais aussi sur la perception de l'environnement dans lequel elles vont évoluer. Les caractéristiques géométriques de l'environnement ne sont pas suffisantes pour rendre compte de l'interaction de l'entité comportementale avec son environnement. Il est nécessaire d'y adjoindre des informations sur l'organisation de l'espace et sur la caractérisation des objets qui le composent (niveaux topologiques et sémantiques).
- la simulation d'éclairage : il s'agit de simuler l'éclairage dans des environnements architecturaux complexes nécessitant des ressources de calcul et de mémoire importantes. L'objectif visé est de réaliser cette simulation à la fois avec une station de travail ordinaire ou avec un réseau de machines. Cette simulation doit aussi fournir un moyen d'apprécier le résultat obtenu en évaluant des critères de confort visuel. Ces critères doivent guider l'éclairagiste à améliorer l'éclairage.

### 3.2. Simulation d'éclairage et synthèse d'image

**Mots clés** : *algorithmes parallèles, simulation d'éclairage, synthèse d'image, radiosit .*

*Glossaire*

**Radiométrie** : mesure des grandeurs physiques liées au rayonnement.

**Photométrie :** mesure des grandeurs physiques liées au rayonnement perçu par un observateur humain moyen caractérisé par une fonction de sensibilité.

**Radiosité :** flux émis par unité de surface ; on l'appelle aussi émittance.

**Modèle d'illumination global :** modèle décrivant les différentes et multiples interactions entre lumière et matière.

**Flux :** énergie par unité de temps, exprimée en Watt ou Lumen.

**Luminance :** flux émis par unité de surface projetée et par unité d'angle solide.

**Réflectance :** rapport entre flux réfléchi et flux incident.

Depuis les années 1980, la synthèse d'image a abandonné la voie de l'empirisme pour prendre une direction réellement scientifique et rigoureuse en s'appuyant sur l'informatique, la physique, la radiométrie et la photométrie, les mathématiques, la perception visuelle... C'est en suivant cette direction qu'une simulation de la propagation de la lumière dans un environnement est possible. Cette approche de la simulation d'éclairage doit permettre à un éclairagiste d'évaluer avec précision les grandeurs radiométriques en tout point d'une scène afin d'évaluer différents critères de confort visuel et de visibilité. Cette approche physique de la simulation est celle suivie par notre équipe. Par ailleurs, il est connu que la simulation d'éclairage (par la méthode de radiosité par exemple) est une tâche nécessitant beaucoup de ressources mémoire et de calcul. Sans une algorithmique spécifique, la simulation dans un environnement complexe est impossible à réaliser avec un seul processeur.

Ce domaine de recherche touche à la fois le calcul de vues à partir de vues, la réalité virtuelle et la réalité augmentée.

### 3.2.1. Simulation d'éclairage et rendu temps réel

L'objectif de la simulation d'éclairage est de simuler à l'aide des lois de la physique, les échanges lumineux entre les sources de lumière naturelles (ciel et soleil) ou artificielles (luminaires) et les différents constituants matériels d'un environnement. Cette simulation de l'éclairage compte deux objectifs principaux, permettre de simuler de manière précise la quantité de lumière émise en tout point d'un environnement et évaluer le confort visuel à l'intérieur d'une pièce et modifier les éclairages en conséquence.

Pour effectuer cette simulation, il est indispensable d'utiliser les modèles d'illumination globaux (équation de luminance) et les techniques avancées de synthèse d'images réalistes, telles que la radiosité et le lancer de rayon. Ainsi, à l'aide de ces méthodes, il est possible de calculer l'ensemble des flux lumineux distribués dans tout l'environnement (le résultat de cette simulation est la distribution des luminances). Il suffit alors, à partir de la description d'un observateur (position, direction du regard, sensibilité, ...), de calculer l'image des luminances telles qu'elles seront perçues par cet observateur placé dans cet environnement. Cette simulation nécessite la connaissance des propriétés physiques des matériaux et des sources de lumière (telles que la réflectance...) constituant la scène pour laquelle on veut réaliser l'éclairage. Ces propriétés sont soit mesurées soit données par un modèle.

L'approche suivie par notre équipe s'appuie sur la méthode de radiosité et de lancer de rayon. Elle est fondée sur la physique de la propagation d'énergie électromagnétique, utilise des luminaires réels et l'éclairage naturel, et fait appel à des outils mathématiques tels que les techniques de projection, les ondelettes, la méthode de Monte Carlo, ... Les critères de confort visuel utilisés sont ceux définis par la Commission Internationale de l'Eclairage.

La simulation d'éclairage reste une tâche très coûteuse en temps de calcul et en capacité mémoire, même pour des environnements de complexité modérée. En effet, cette simulation consiste en fait à résoudre une équation intégrale que nous projetons dans un espace fonctionnel engendré par une base d'ondelettes. Cette projection implique le maillage des surfaces de la scène en plus petites surfaces appelées *éléments de surface*. De plus, de nombreuses structures de données sont mises en place pour représenter ce maillage et pour accélérer les calculs. Ceci nécessite une mémoire de stockage importante. Sans une algorithmique adaptée, ces techniques permettent de simuler l'éclairage dans des environnements de moyenne complexité (une ou deux pièces dans un bâtiment) mais non de traiter des bases de données plus conséquentes telles que des immeubles de plusieurs étages.

Une autre solution consiste à calculer des images d'une scène réelle ou virtuelle à partir de plusieurs images acquises à l'aide d'une caméra. Trois points sont étudiés. Le premier concerne la représentation de la scène à l'aide de primitives géométriques reconstruites ou de cartes de profondeur. Ce point concerne aussi l'extraction de textures indépendamment des ombres et des reflets. Le deuxième point traite du problème de rendu des scènes ainsi représentées : il s'agit de se déplacer dans ces scènes en temps réel et d'insérer des objets virtuels. L'objet du troisième point est d'estimer les conditions d'éclairage ainsi que les réflectances des objets afin de pouvoir recalculer la même scène éclairée de manière différente et d'homogénéiser l'éclairage entre les objets virtuels et réels.

Ces dernières années, l'augmentation de la puissance des ordinateurs grand public et des cartes graphiques 3D a permis le développement d'applications 3D temps réel de très bonne qualité visuelle. D'autre part, l'amélioration des débits réseaux pour les accès internet offre la possibilité de télécharger des fichiers VRML de petite taille permettant de décrire des scènes 3D. Une fois le fichier rattaché, un navigateur 3D permet de visualiser cette scène en temps réel. Cette technique pose cependant des contraintes sur la taille de la scène visitée. Il n'est en effet pas envisageable de rattachier un modèle de ville afin de permettre une navigation parce que le temps de téléchargement graphique 3D serait incompatible avec un tel volume de données.

### 3.3. Modèles dynamiques de mouvement

**Mots clés :** *mouvement, animation, simulation, identification, systèmes hybrides, niveaux de détail.*

*Glossaire*

**Animation :** modèles et algorithmes permettant de produire des mouvements conformes à la spécification de l'animateur.

**Animation par modèles physiques :** se dit des modèles d'animation qui prennent en compte les lois physiques, sur le plan structurel ou comportemental.

**Système hybride :** système dynamique mettant en interaction une partie différentielle continue et un système à événements discrets.

**Vecteur d'état :** vecteur de données représentant le système à un instant  $t$ , exemple : le couple position et vitesse.

A l'instar de l'approche adoptée en synthèse d'images photoréalistes, nous cherchons à baser nos algorithmes sur des modèles physiques. En outre, la synthèse de mouvements naturels nécessite la prise en compte de phénomènes complexes, au niveau mécanique, biomécanique ou neurophysiologique (activation neuromusculaire, planification et boucles d'asservissement, etc.).

La création de mouvements d'objets ou de personnages synthétiques nécessite la mise en œuvre de modèles dynamiques adaptés aux différents contextes applicatifs de la synthèse d'images : la simulation du mouvement naturel, l'animation pour la production audio-visuelle ou l'animation-simulation interactive.

La modélisation mathématique des processus de génération du mouvement et leur mise en œuvre algorithmique s'appuient sur la théorie des systèmes dynamiques et utilisent les outils de la mécanique, de l'automatique et du traitement du signal. La forme générale d'un modèle dynamique de mouvement est celle d'un système hybride, interaction d'une partie différentielle continue, avec un système à événements discrets :

$$\begin{aligned} \frac{dx}{dt} &= f(x(t), u(t), t) \\ x_{n+1} &= g(x_n, u_n, n) \end{aligned}$$

où le vecteur d'état  $x$  est une concaténation des variables d'état discrètes et continues,  $u$  est le vecteur de commande et  $t$  le temps.

Typiquement, le traitement physique des contacts (chocs et frottements) est effectué à l'aide d'un modèle hybride. Les collisions se manifestent par des sauts dans l'espace d'état (impulsions correspondant à des discontinuités des vitesses). Les procédés de contrôle des différentes phases de la locomotion se formalisent également à l'aide de modèles hybrides.

Nos études consistent à définir les méthodes et les algorithmes servant à construire un modèle dynamique hybride à partir des spécifications d'un utilisateur. L'objectif principal est de transférer la puissance des outils d'étude et de simulation de ces modèles, au domaine de l'animation par ordinateur.

Dans ce contexte, plusieurs thèmes de recherche émergent :

- la génération automatique des modèles : cela consiste, d'une part, à définir un langage de spécification des modèles de mouvement et d'autre part, à proposer des algorithmes optimisés pour des architectures spécialisées (comme des machines parallèles). Dans le cas d'objets inertes (sans actionneur), l'application des principes généraux de la mécanique garantit l'obtention du modèle à partir de ses caractéristiques géométriques et mécaniques. La synthèse des lois de commande peut faire appel à des méthodes systématiques comme la commande optimale, sous réserve que le problème posé puisse être mis sous la forme usuelle d'un schéma de commande.
- l'identification : la validité d'un modèle de mouvement passe par une phase de calibrage et d'identification. A partir de l'étude de trajectoires réelles, des techniques d'identification paramétriques et structurelles permettent d'inscrire la conception des modèles de mouvement dans le cadre d'une approche analyse-synthèse.
- les niveaux de détail : la notion de niveaux de détail est essentielle en animation. La restitution de scènes complexes nécessite une gestion adaptative des coûts de calculs associés aux différents objets animés de la scène. Ce problème comporte deux facettes complémentaires : la génération et la gestion en-ligne, à l'instar de ce qui est développé pour la géométrie et les textures.
- la commande interactive pour l'animation-simulation : la démarche générale consiste à adapter certaines fonctions ainsi que des méthodes, issues de la robotique ou de l'automatique, au domaine de l'informatique graphique interactive. L'objectif porte sur la production automatique du mouvement et la notion de contrôleur, destiné à être associé à chaque entité géométrique. Les recherches se situent ici comme des extensions par rapport aux techniques de contrôle au niveau cinématique ou dynamique (en particulier contrôle par contraintes). Nous cherchons à doter les entités de capacités sensorielles pour élaborer des fonctions de guidage adaptatif du mouvement. L'objectif est d'établir les fondements d'une structure de contrôle-commande en boucle fermée, pour la génération automatique de trajectoires dans un environnement complexe et/ou évolutif. Cette fonction de perception de l'environnement est fondamentale pour faire évoluer les techniques d'animation vers l'animation d'entités dotées de véritables propriétés comportementales.

### 3.4. Modélisation et simulation comportementale

**Mots clés :** *modélisation topologique, modélisation sémantique, systèmes multi-agents, animation comportementale, simulation événementielle, automates, statecharts, réseaux de Petri, objets réactifs, objets temps réel, scénario de simulation.*

*Glossaire*

- Modèle topologique :** ensemble des relations spatiales reliant les objets géométriques entre eux et formant un graphe.
- Modèle sémantique :** ensemble des propriétés fonctionnelles d'un objet liées à son usage.
- Perception :** analyse de l'environnement virtuel à travers un ensemble de capteurs représentant la ressource de perception de l'entité.
- Modèle décisionnel :** il évalue le nouvel état d'une entité en fonction de la perception qu'elle a de son environnement, de ses buts et de son état courant.
- Action :** l'entité réagit en agissant sur ces ressources propres (ex : actionneur) ou sur son environnement direct.
- Communication :** dialogue avec les autres entités de l'environnement.

L'animation comportementale vise à aborder une dimension nouvelle de l'animation par ordinateur, l'animation de scènes complexes en contexte multi-acteurs (ou multi-agents). Nous introduisons une représentation qui associe à chaque entité (acteur ou agent comportemental) d'une scène une *cellule* : perception / décision / action / communication. Les recherches actuelles visent à doter les entités d'une scène d'une certaine autonomie, puis à les gérer à partir de directives de haut niveau. La simulation est constituée d'un ensemble d'objets dynamiques, dont les évolutions dépendent des interactions qui peuvent se produire sous des formes très variées. Afin de rendre compte de la complexité décisionnelle, il est nécessaire de traiter conjointement les aspects continus et discrets, de coordonner les comportements concurrents et de gérer leur structure organisationnelle. En complément de la représentation géométrique de l'environnement, il est nécessaire de fournir pour chaque entité un modèle symbolique de son environnement, afin de pouvoir produire des comportements complexes. Un scénario peut également être spécifié afin de transmettre des directives visant à coordonner l'animation.

Plusieurs approches ont été étudiées dans différents laboratoires pour la définition du modèle décisionnel : stimulus / réponses, règles de comportement, environnements prédéfinis et automates. Ces modèles demeurent relativement simples, avec des champs de perception et d'action limités, et en outre ils ne prennent pas en compte l'aspect temporel qui est primordial (mémoire, prédiction, durée d'une action, séquençage de tâches). Nous cherchons donc à unifier ces différents modèles décisionnels et à les étendre à la prise en compte du temps. Le modèle d'une entité comportementale est composée de quatre parties : perception, décision, action, communication.

Le modèle décisionnel est quant à lui chargé de la définition de son nouvel état en fonction de la perception qu'il a de son environnement, à la fois au travers de la mémorisation de connaissances historiques (observation et interprétation), de l'évaluation de l'état courant et de l'anticipation du futur immédiat (modèle prédictif). A partir de ce nouvel état et des éléments fournis par l'utilisateur et/ou le scénario, il doit définir un schéma d'actions qui pourra être remis en question dans le futur (événements prioritaires comme par exemple une collision potentielle).

En règle générale, il est nécessaire de pouvoir définir le comportement d'une entité de manière modulaire et hiérarchique, chaque niveau d'abstraction ne manipulant pas les mêmes concepts et ne prenant pas en compte les mêmes éléments de l'environnement. Il est de même important de pouvoir gérer plusieurs fils d'activité en parallèle. Ainsi, à notre sens, un modèle permettant la spécification du comportement d'une entité doit intégrer les paradigmes suivants : hiérarchie, parallélisme, modularité et réactivité.

#### **Modélisation symbolique de l'environnement de simulation**

Nous nous plaçons dans le cadre de la modélisation non seulement de la géométrie de la scène, qui peut être effectuée par n'importe quel modèleur 3D, mais aussi de toutes les informations pertinentes pour les comportements à simuler. En effet, il est nécessaire de représenter les éléments symboliques importants de l'environnement dans lequel les entités vont évoluer, éléments qui vont influencer sur le comportement. Ces éléments seront des objets, des positions, des espaces, seront typés et décriront une certaine topologie et sémantique de l'environnement.

#### **Animation comportementale**

À un premier niveau, on considère le concept d'*entité comportementale* fondé sur les principes de *modularité* et de *réactivité*. Ce dernier point, indispensable, s'appuie sur la simulation événementielle (systèmes à événements discrets) incluant les modèles de type automates parallèles hiérarchisés, réseaux de Pétri et *statecharts*, ainsi que les aspects de langage et de génération de code. Le concept d'entité comportementale générique, vu comme un *objet réactif asynchrone*, conduit à approfondir plusieurs points : la conception même de l'entité, la génération et la modification des représentations états-événements, les processus d'interactions et de communications asynchrones. De plus, pour maîtriser le déroulement d'une animation, nous devons mettre l'accent sur les possibilités de contrôle des entités, tout en prenant en compte leur autonomie et les nombreuses interactions possibles au cours de l'évolution. C'est pourquoi nous nous intéressons à la notion de *scénario*, géré par l'utilisateur, et influant sur le comportement individuel ou collectif des entités.

## 4. Domaines d'application

### 4.1. Panorama

Les domaines d'application du projet concernent principalement les domaines pour lesquels un fort couplage entre simulation de systèmes physiques et restitution visuelle des résultats de calcul sont primordiaux. Nous nous attachons dans chacun de ces domaines, d'une part, à l'étude fine des modèles sous-jacents essentiellement liés à la physique du phénomène étudié et, d'autre part, à l'application directe de nos récents travaux de recherche. Les grands domaines applicatifs traités par le projet SIAMES concernent :

- les environnements architecturaux et urbains ;
- la propagation de rayonnement ;
- les humanoïdes virtuels et leurs applications en biomécanique ;
- la réalité virtuelle, la réalité augmentée et la téléopération ;
- les simulateurs pour la recherche dans le domaine des transports ;
- l'animation : multimédia, audiovisuel et jeux.

### 4.2. La réalité virtuelle, la réalité augmentée et la téléopération

**Mots clés :** *réalité virtuelle, réalité augmentée, téléopération.*

*Glossaire*

**Réalité virtuelle :** désigne tout système qui procure à l'opérateur humain la sensation d'immersion et la capacité d'interaction face à un environnement *virtuel*, c'est-à-dire basé sur un modèle de synthèse entièrement généré par ordinateur.

**Réalité augmentée :** caractérise tout système qui améliore la perception de l'opérateur vis à vis de l'environnement réel, généralement par superposition d'images de synthèse sur des images réelles ou vidéo.

**Téléopération :** désigne les principes et les techniques qui permettent à l'opérateur humain d'accomplir une tâche à distance, à l'aide d'un système robotique d'intervention, commandé à partir d'une station de contrôle, par l'intermédiaire d'un canal de télécommunication.

Le projet SIAMES est directement concerné par tous ces domaines, qui eux-mêmes sont interdépendants. En effet, la mise en œuvre du concept de *réalité virtuelle* s'appuie sur l'infographie pour la production d'images de synthèse, sur la simulation pour la génération d'environnements virtuels en temps réel et sur la téléopération pour les interfaces de communication homme-machine. À travers les différents sujets de recherche abordés dans le projet et les différentes applications contractuelles, nous sommes pour une large part impliqués dans des travaux ayant trait à la réalité virtuelle :

- simulation d'éclairage d'environnements virtuels et navigation interactive dans ces environnements ;
- simulation en temps réel et interactions en environnement urbain virtuel ;
- contrôle du mouvement en temps réel de systèmes dynamiques.

### 4.3. Humanoïdes virtuels et modèles biomécaniques

**Mots clés :** *humanoïde de synthèse.*

L'étude du mouvement humain est une science à part entière que médecins et biomécaniciens continuent à affiner. La représentation de cette connaissance sous forme de modèles paramétrables, le contrôle du niveau de détails, ainsi que la génération automatique d'un code de simulation numérique pour l'animation et la simulation, relèvent directement des compétences du projet SIAMES.

L'étude du mouvement humain en situation se heurte à des problèmes pratiques, d'ailleurs difficilement contournables : le prélèvement in situ des paramètres du mouvement passe par l'utilisation de capteurs.

L'acquisition des trajectoires 3D à l'aide de systèmes optiques ou magnétiques demeure superficielle et difficile à traiter. Cependant, tous ces systèmes ne fournissent qu'une observation partielle et imprécise. Par ailleurs, les équipements matériels et leur utilisation parfois invasive peuvent contraindre le sujet d'étude et obliger à restreindre les objectifs de l'analyse. La simulation de modèles humanoïdes offre la possibilité de mettre en place des expérimentations virtuelles complémentaires. L'utilisation de modèles dynamiques permet de suivre l'évolution de l'état interne lors de l'accomplissement du mouvement. En fonction du niveau de détails choisi, on pourra par exemple accéder aux couples articulaires mis en jeu ou aux efforts musculaires. La notion d'expérimentation virtuelle est d'autant plus intéressante lorsqu'il s'agit d'études difficiles, voire impossibles à mettre en œuvre dans la réalité (accidentologie, environnements à risques ou contextes inexistantes). Dans ce cadre, notre objectif est de fournir des outils logiciels pour la génération automatique de modèles d'humanoïdes, en y incorporant la connaissance des médecins et des biomécaniciens. Ces modèles permettent de prendre en compte le facteur humain dans les outils de CAO-CFAO dès la phase de conception de systèmes destinés à un opérateur humain.

#### 4.4. Le multimédia, l'audiovisuel et les jeux

**Mots clés :** *animation, multimédia, audiovisuel, jeux.*

*Glossaire*

**sprite :** ensemble de textures d'un personnage reproduisant par affichage successif un mouvement complexe.

**chipset graphique :** ensemble de composants matériels réalisant les fonctions évoluées de synthèse d'image 3D.

L'animation est, dans la pratique, une partie intégrante des logiciels à vocation éducative ou ludique. Domaines de l'interactivité par excellence, ils nécessitent de mettre en œuvre des modèles de géométrie et de mouvement, suffisamment peu coûteux pour pouvoir maintenir la fréquence d'affichage des images.

Les applications grand public, ludiques ou éducatives, utilisent de plus en plus la synthèse d'images et l'animation 3D. Logiquement, ce marché à fort potentiel suscite le développement de technologies nouvelles comme celui des cartes graphiques 3D pour ordinateurs personnels. L'animation interactive et le domaine des jeux vidéo en particulier, soulèvent des problèmes techniques spécifiques. Pendant longtemps, la puissance limitée des processeurs a contraint les concepteurs de ces applications aussi bien que les constructeurs des machines, à avoir recours à nombre d'artifices pour améliorer à moindre coût la qualité des animations (gestion des *sprites*, plaquage de textures, utilisation de la table des couleurs, notion de  $2D\frac{1}{2}$ ). Alors que le *chipset* embarqué sur les cartes graphiques ou les consoles de jeu a considérablement évolué, la maîtrise du coût calcul demeure une préoccupation constante, dans la mesure où les capacités sont rapidement utilisées pour gérer des environnements virtuels plus riches. Dans ce cadre, la notion de niveaux de détails est essentielle, qu'il s'agisse de géométrie, de textures, ou de modèles de mouvement. Des problèmes de compression et de codage de ces données apparaissent également, dans le contexte du multimédia et des applications distribuées sur un réseau.

#### 4.5. Les modèles mécaniques et le prototypage virtuel

**Mots clés :** *Modèles mécanique, prototypage virtuel, retour sur conception.*

La simulation numérique pour l'animation et la simulation, relèvent directement des compétences du projet SIAMES. La réalisation de simulateurs dans ce contexte est en cours, notamment celle d'un simulateur d'endoscopie. Par contre, la maîtrise des modèles mécaniques et leur adéquation à la réalité représentée n'est pas triviale.

L'idée du prototypage virtuel est d'utiliser les résultats du simulateur en cours de réalisation en vue de la conception de l'objet mécanique réalisant les fonctionnalités désirées. Pour ce faire, nous étudions les techniques d'optimisation, basées sur l'utilisation d'algorithmes de type évolutionnaires et l'intégration des

résultats obtenus dans le processus de conception de la CAO classique. Cette démarche est actuellement déclinée par la réalisation du simulateur d'apprentissage et l'étude parallèle de logiciels de CAO du commerce (SolidWorks) afin de définir correctement la boucle de retour.

## 5. Logiciels

### 5.1. Panorama

Du fait du caractère graphique de nos travaux de recherche, nous sommes amenés à valider nos travaux par la réalisation de prototypes logiciels capables de traiter des problèmes concrets. Les logiciels présentés dans cette section sont tous opérationnels et utilisés dans différents contextes contractuels (industriels ou projets européens).

### 5.2. Logiciel de simulation d'éclairage

**Participant :** Kadi Bouatouch [correspondant].

**Mots clés :** *logiciel, simulation d'éclairage.*

*Glossaire*

**simulation inverse :** technique permettant à partir d'un résultat d'éclairage désiré de trouver par calcul les positions et caractéristiques des sources lumineuses.

Différents types de logiciels ont été développés pour la simulation d'éclairage et le dimensionnement des tunnels : simulation d'éclairage naturel et artificiel, simulation en environnements complexes et simulateur parallèle.

- *PAUS* : contrairement à *SAS*, ce logiciel est capable d'effectuer une structuration en cellules de scènes architecturales ou urbaines. Il est écrit en Java et utilise Java3D.
- *SimP* : *SimP* est une version parallèle de *SIM* utilisant l'environnement de programmation parallèle *MPI*, ce qui lui permet de fonctionner sur n'importe quel réseau de machines séquentielles ou parallèles.
- *SimT* : Les caractéristiques du logiciel *SimT* sont :
  - i. Proposer un logiciel de simulation d'éclairage permettant d'obtenir une représentation virtuelle d'un tunnel et de son éclairage la plus proche possible de ce qu'elle devrait être en réalité.
  - ii. Proposer un outil d'aide à la conception. L'objectif est de déterminer, tout en respectant le niveau de luminance requis sur la chaussée, les meilleures configurations (choix du revêtement, type de sources, puissance d'émission, ...) relatives au tunnel. Conceptuellement, ce problème peut être vu comme un problème de simulation inverse.
  - iii. Proposer un outil de validation pour le concepteur de tunnel mais aussi par exemple pour les fabricants de luminaires qui pourront ainsi évaluer les performances de leurs luminaires vis à vis de la concurrence.
- *3Points* est un logiciel de calcul de radiosité par le méthode 3 points. Il permet de calculer une simulation d'éclairage sur des scènes contenant à la fois des objets diffus et spéculaires. Il permet aussi de d'effectuer un rendu diffus de scènes indépendamment du point de vue. Ces scènes pré-éclairées peuvent alors être exportées vers la plate-forme Magellan pour une navigation temps réel. Le logiciel permet également de calculer une image à partir d'un point de vue pour un rendu contenant à la fois les composantes diffuse et spéculaire de la scène. La composante spéculaire est

dépendante du point vue, il est donc nécessaire de la recalculer à chaque déplacement. Ce logiciel est interactif et possède plusieurs utilitaires tels que l’affichage de BRDF, de sources spectrales et de leur solide photométriques, des images de luminances, etc.

- *ModPh* : *ModPh* est un logiciel interactif de modélisation de sites réels à partir d’images. Il permet à un utilisateur de reconstruire des sites réels avec une précision suffisante afin de pouvoir les explorer (« walkthrough »). La reconstruction, interactive, est contrôlée aussi bien au moyen de plusieurs photographies du site que par une maquette virtuelle qui évolue au cours de la modélisation. Ce logiciel contient un module de traitement de textures. Il permet de supprimer d’une texture l’éclairage et les ombres.
- *Magellan* : *Magellan* est une plate-forme de navigation orientée client-serveur. Celle-ci permet la navigation au sein de scènes complexes exprimées en langage VRML97. Elle permet d’utiliser des extensions de ces langages afin d’utiliser des transferts de données progressifs. Cette technique de transfert progressif s’applique aux maillages, aux textures ainsi qu’au surfels (rendu d’objets à l’aide de nuages de points). En plus de cette approche, la plate-forme *Magellan* permet la navigation au sein de scènes structurées. Durant cette période, plusieurs améliorations et enrichissements ont été apportés à la plate-forme. En effet, les fichiers VRML97 représentant la scène et sa structuration ont été codés en format propriétaire binaire, ce qui réduit considérablement leur taille et par conséquent le temps de leur transfert à travers le réseau. Le calcul de visibilité a été amélioré. En effet, ce calcul est effectué par la carte graphique et permet de déterminer les relations de visibilité cellule-cellule et cellule-objets. Nous avons conçu et mis en œuvre un autre type d’imposteur qui permet de représenter soit des objets complexes soit les parties de la scène vues à travers des ouvertures telles que fenêtres, portes, etc. Ce type d’imposteur est modélisé par des points selon une approche multirésolution (niveaux de détail). La résolution est déterminée en fonction de la distance entre le point de vue courant et l’imposteur.
- *CityZen* : *CityZen* est un logiciel permettant la génération de modèles de villes destinés à la navigation temps réel. Les modèles générés sont directement exploités par la plate-forme *Magellan* pour la navigation. La ville est constituée, et ce quelque soit l’échelle à laquelle elle est considérée, de motifs répétitifs. Ces motifs peuvent par exemple être décrits par un mécanisme de réécriture tel que les L-systems, souvent utilisés pour la modélisation des plantes et de leur évolution, mais aussi récemment dans la création de réseaux routiers pour la modélisation urbaine. De plus, de par leur utilisation intuitive, les L-systems sont bien adaptés à la création urbaine, et fournissent un outil puissant de modélisation de scènes complexes pouvant être visualisées en temps réel grâce à un développement à la demande. *CityZen* est un logiciel basé sur un mécanisme de réécriture de type L-system qui permet la définition et l’utilisation de motifs géométriques tout en offrant la possibilité de navigation temps-réel. De plus, une fois le bâtiment décrit par motifs, une infinité de bâtiments similaire ou « de même style » peuvent être définis en modifiant les dimensions du bâtiments, données comme paramètres d’entrée du système de réécriture. Cette réutilisabilité joue un rôle primordial dans la réduction du temps de création d’un modèle de ville.

### 5.3. OpenMASK : Plate-forme, OpenSource, de simulation en réalité virtuelle

**Participant** : Alain Chauffaut [correspondant].

**Mots clés** : *plate-forme logicielle, simulateur distribué, simulateur temps réel, modularité, interactivité, réalité virtuelle.*

OPENMASK (Open Modular Animation and Simulation Kit) a pour vocation d’être la plate-forme d’accueil et d’expérimentation des différents travaux de recherche menés au sein du projet dans les domaines de l’animation et de la simulation. OpenMASK propose une architecture de programmation multicouches et modulaire. Notre objectif principal est de réduire autant que possible les contraintes pour le programmeur concernant l’intégration d’un module au sein de la plate-forme. La plate-forme offre avant tout les services de synchronisation et de communication entre modules ayant des fréquences propres différentes (fonctionnant à

des fréquences différentes afin de respecter leurs contraintes de simulation). OpenMASK est fourni avec des modules de visualisation et d'interactivité adaptés à l'immersion en réalité virtuelle.

Notre ambition, à travers ce projet d'équipe, est de réaliser une plate-forme logicielle de simulation et d'animation qui permette d'intégrer de manière homogène les différents travaux issus des thèmes de recherches développés dans le projet. Ce travail a été retenu comme opération de développement logiciel (odl) par l'Inria sur deux ans à partir d'octobre 2001.

OpenMASK est une plate-forme de développement et d'exécution d'applications modulaires dans les domaines de l'animation, de la simulation et de la réalité virtuelle.

Dans OpenMASK, l'unité de modularité est appelée objet de simulation. Il peut servir à décrire aussi bien le comportement et le contrôle du mouvement d'une entité ou d'un objet virtuel qu'un périphérique d'interaction.

Avec OpenMASK, concevoir un environnement virtuel revient à choisir (ou programmer) les objets de simulation à utiliser, à les paramétrer pour les adapter au contexte et à choisir un noyau d'exécution conforme aux objectifs de réalisation.

OpenMASK intègre des noyaux d'exécution multi-sites (pour des applications distribuées : réalité virtuelle, simulation ...) et/ou multi-processeurs (pour un calcul parallèle de l'exécution). Ces noyaux permettent aussi bien la simulation hors ligne que l'animation interactive.

OpenMASK propose une API C++ ouverte pour le développement d'objets de simulation et pour l'adaptation des noyaux d'exécution.

OpenMASK-3DVis est le premier composant logiciel fourni avec OpenMASK pour la visualisation de l'animation d'environnements virtuels. Dans sa version actuelle, ce composant est basé sur la bibliothèque Performer de SGI.

Un kit d'interaction vient compléter le dispositif pour faciliter la gestion d'applications de réalité virtuelle, immersives et interactives.

OpenMASK et OpenMASK-3DVis sont disponibles avec une licence open-source (QPL) et fonctionnent sous linux et Irix.

Services offerts par les noyaux d'exécution :

Accueil : création et destruction des objets de simulation. Structuration en arbre d'objets.

Nommage : nommage des objets, des attributs et des classes. Recherches en fonction du nom, en utilisant les relations de filiation de l'arbre de simulation.

Activation : activation régulière (chaque objet peut avoir sa propre fréquence) et/ou occasionnelle (sur réception d'événements) pour les objets de simulation.

Communication :

- \* par chemin de données entre les objets de simulation
- \* par diffusion de signaux dans l'environnement
- \* par envoi d'événements
- \* en utilisant les types par défaut ou dédiés à l'application
- \* avec gestion de l'adaptation fréquentielle à l'aide d'interpolateur et d'extrapolateurs

Gestion du temps : datation automatique des données et unité de temps pour l'exécution.

Distribution : le fonctionnement multi-processus multi-machines de OPENMASK, repose actuellement sur PVM. La distribution est transparente au programmeur de module, mais contrôlée par l'opérateur d'une application.

Fonctionnalités d'OpenMASK-3DVis :

- Visualisation sur écrans et sur dispositifs d'immersion mono et multi-pipes : reality center et workbenchs. Visualisation en stéréovision et multi-vues.

- Chargement de tous les formats de fichiers supportés par Performer
- Possibilité d’extension du composant pour prendre en compte de nouvelles primitives d’animation (disponibles : quaternions, rotations, translations, matrices ...), ainsi que de nouveaux types de partenaires.
- Service de capture des événements X11 et de transmission aux objets abonnés.
- Service de désignation 2D ou 3D d’objets de la scène par l’utilisateur et communication des informations aux abonnés.

Fonctionnalités du kit d’interactivité : Cette bibliothèque de classes C++ propose des mécanismes pour automatiser la transformation d’un objet de simulation, qui s’y prête, en un objet interactif. La méthode consiste à faire apprendre automatiquement aux objets concernés, le protocole d’interaction grâce à deux classes d’outils :

- des adaptateurs génériques à réutiliser pour rendre interactif un objet de simulation
- des interacteurs génériques à réutiliser pour réaliser l’interface entre un driver de périphérique et un objet interactif

Nous fournissons certains drivers de périphériques, notamment pour des capteurs de positions 3D ou le dispositif clavier-souris.

Nous proposons aussi une classe de cameraman pour contrôler les points de vue sur la scène.

## 5.4. Définition d’un outil pour l’entraîneur sportif

**Participant :** Bruno Arnaldi [correspondant].

**Mots clés :** *Animation du mouvement sportif, suivi de l’entraînement.*

L’analyse du mouvement sportif concerne principalement l’identification des paramètres liés à la performance de l’athlète. L’amélioration des performances passe en effet par une meilleure compréhension des phénomènes mécaniques qui y sont liés : forces, énergies... Pour y parvenir, il est nécessaire de mettre en place des outils associant visualisation 3D, analyse et synthèse du mouvement. Le logiciel que nous avons développé répond totalement à ces besoins et permet donc aux entraîneurs sportifs d’effectuer un suivi de la performance de leurs athlètes.

L’analyse de la performance des athlètes intéresse depuis toujours les fédérations sportives nationales et internationales. Depuis le début du siècle, l’analyse à partir d’images est largement utilisée pour expliquer les raisons de la performance ou de la contre-performance d’un athlète. Toutefois, l’utilisation d’images 2D se révèle à l’heure actuelle insuffisante pour expliquer ces phénomènes. En conséquence, le nombre des systèmes d’analyse du mouvement en 3 dimensions a considérablement augmenté ces dernières années. La restitution des informations liées à la performance à partir de ces systèmes reste encore du domaine de la recherche. C’est pourquoi nous avons développé un logiciel permettant aux entraîneurs et aux sportifs de pouvoir exploiter simplement ces données (cf. figure 1). Ce système est développé en Java au sein d’une page Web associant une visualisation 3D (VRML) du mouvement et un ensemble d’outils d’analyse de la performance.

Ces outils sont composés :

- d’un prétraitement des données issues du système d’acquisition de mouvements (interpolation des points manquants, filtrage des données, calcul des angles à chaque articulation suivant le formalisme d’Euler),
- d’un calcul des vitesses et des accélérations des positions des segments et des angles aux articulations (ces valeurs sont affichées dans une fenêtre synchronisée avec la visualisation VRML),
- d’un affichage des trajectoires articulaires à divers échelles ainsi que d’un ensemble d’opérateurs de type *Motion Warping* permettant de modifier ces trajectoires et d’en visualiser les conséquences sur le geste.

- d'un affichage d'informations relatives au mouvement restitué. Ces données annexes sont généralement liées à la dynamique du mouvement telles que les énergies, les forces et les moments aux articulations.

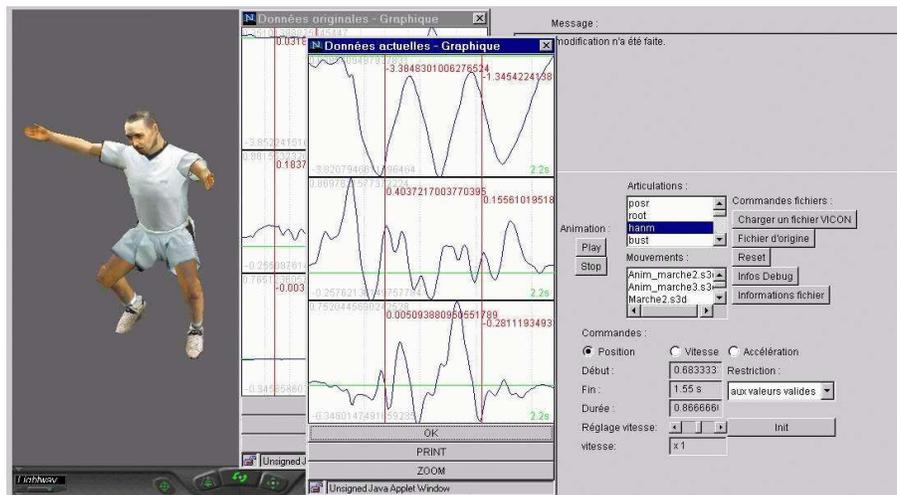


Figure 1. Interface du logiciel dédié aux entraîneurs et sportifs

La mise en place de ces outils dans une page Web permet aux entraîneurs de se connecter à un serveur de mouvements et de suivre l'évolution de leurs athlètes au cours du temps. De plus, cet outil peut servir de support de formation des athlètes par les entraîneurs en montrant les défauts les plus connus et en quoi les gestes peuvent être améliorés.

## 6. Résultats nouveaux

### 6.1. Navigation en temps réel dans des scènes complexes à travers un réseau bas débit

**Participants :** Kadi Bouatouch, Jean-Eudes Marvie, Gérald Choqueux.

**Mots clés :** *Navigation, temps réel, client-serveur, niveaux de détail, structuration de scènes, imposteurs, textures progressives.*

Dans le cadre de plusieurs projets (V2NET, OPENISE) nous nous intéressons à la navigation réaliste interactive dans des scènes complexes à travers un réseau bas débit (typiquement Internet). Le problème posé consiste à déterminer à chaque instant de la visite d'une scène quelle représentation de celle-ci transférer du serveur de scènes au terminal graphique (le client) de façon à réduire les temps de latence de l'affichage induits par la complexité géométrique et photométrique des scènes.

Pour résoudre ce problème, nous avons développé une plate-forme de navigation distribuée basée sur une architecture client/serveur. La difficulté principale de cette étude réside dans le choix de l'architecture logicielle permettant d'obtenir une fréquence d'affichage élevée tout en assurant une flexibilité importante. Cette architecture doit permettre de tester l'utilisation conjointe de représentations variées de scènes, chacune liée à une technique d'optimisation particulière.

La plate-forme de navigation Magellan que nous avons conçue et implémentée est une plate-forme distribuée basée sur un algorithme asynchrone. Une partie de l'algorithme est exécutée chez le serveur, l'autre partie est exécutée chez le client. C'est dans la partie asynchrone placée chez le client que la navigation et le

rendu sont effectués. Le schéma de connexion est du type 1 serveur,  $N$  clients. L'algorithme est implémenté en langage C++ sous forme d'une plate-forme de développement. L'architecture de la plate-forme est schématisée par la figure 2. La gestion des processus légers est basée sur la bibliothèque POSIX Threads. Afin d'en permettre une utilisation plus souple, nous avons encapsulé la gestion des processus dans des objets C++. La gestion du rendu des objets 3D est basée sur la bibliothèque OpenGL. La communication entre le serveur et les clients utilise quand à elle les sockets définis dans le standard POSIX.

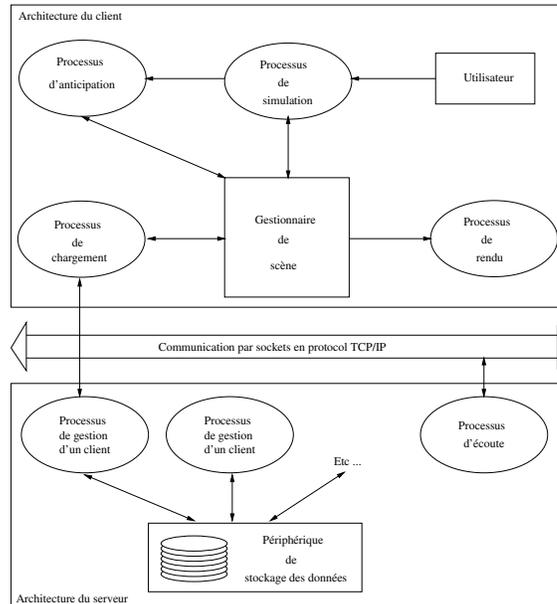


Figure 2. Cette figure représente l'architecture générale de la plate-forme.

Les parties client et serveur sont exécutées sur des machines différentes reliées par un réseau bas débit. La communication entre les deux parties de la plate-forme est effectuée à l'aide de sockets selon le protocole TCP/IP. La partie haute du schéma représente l'architecture de l'algorithme asynchrone du client. Les ellipses sont des processus concurrents. Ces différents processus travaillent sur une même représentation de la scène. Un processus est chargé du rendu de la scène. Un processus est chargé de la simulation de déplacement de la caméra en fonction des ordres de l'utilisateur. Un autre est chargé d'anticiper les mouvements de l'utilisateur afin de précharger différents objets via le réseau. Et le dernier s'occupe de la communication avec le serveur distant. La partie basse du schéma représente l'architecture du serveur. Celui-ci associe un processus de gestion à chaque client connecté. Ces processus de gestion se partagent l'accès à l'information stockée sur disque. Notre plate-forme propose un algorithme générique de gestion de scènes structurées dans un environnement distribué. Le développeur utilisateur de la plate-forme doit implémenter un certain nombre d'objets graphiques avant de pouvoir proposer un logiciel de navigation à un utilisateur quelconque. Celui-ci doit posséder une bonne compréhension du fonctionnement interne de la plate-forme mais ne nécessite en aucun cas de connaître les techniques utilisées dans l'implémentation de celle-ci. L'utilisation de la plate-forme dans le cadre d'un développement d'application nécessite une bonne connaissance du langage C++ ainsi que de la bibliothèque OpenGL.

#### 6.1.1. Architecture générale de la plate-forme :

La plate-forme propose un algorithme générique de gestion de scènes structurées. La plate-forme ne se soucie pas de savoir comment sont représentés les objets graphiques qu'elle manipule. Celle-ci sait uniquement qu'elle manipule quatre classes d'objets graphiques génériques :

1. des cellules,

2. des objets géométriques,
3. des matériaux,
4. des textures.

De plus, la plate-forme doit permettre l'utilisation de plusieurs *types* d'objets au sein de chaque *classe*. Par exemple l'utilisation de plusieurs formats de textures. Les seules contraintes posées sur ces classes d'objets graphiques résident dans le fait que chaque objet doit savoir :

1. Charger ses données à partir d'un fichier,
2. Sérialiser ses données dans un tampon mémoire afin de pouvoir les envoyer via un réseau,
3. Désérialiser ses données depuis un tampon mémoire afin de pouvoir les réceptionner via un réseau,
4. ainsi que :
  - S'afficher à l'aide de commandes OpenGL dans le cas des cellules et des objets géométriques.
  - S'activer et se désactiver à l'aide de commandes OpenGL dans le cas des matériaux et des textures.
  - Renvoyer la liste des identificateurs de ses cellules potentiellement visibles dans le cas des cellules.

#### 6.1.2. Utilisation de la plate-forme :

Parallèlement aux optimisations apportés à la plate-forme durant cette année, nous avons développés différentes applications utilisateur basées sur celle-ci :

1. Dans le but de pouvoir manipuler facilement tous types de bases 3D (de différentes provenances), nous avons implémentés une grande partie du langage VRML97 en tant qu'objets graphiques. Le langage VRML97 étant de nos jours un standard en tant qu'outil de description de scènes 3D, la plupart des logiciels de modélisation permettent l'exportation de leurs contenu dans ce format de fichier. Le module utilisateur que nous avons développés permet de créer facilement de nouveaux noeuds respectant les normes de ce langage. Ainsi, nous avons développés différent noeuds permettant la description de scènes structurées et en particulier des noeuds de description de cellules. Le format VRML97 étant un format de fichier ASCII (description en texte), celui-ci n'offre pas de performances suffisantes au niveau de la taille de stockage et du transfert réseau des fichier. Pour améliorer ces performances, nous avons développés un nouveau format de fichier binaire compressé offrant une représentation plus compacte de ce langage
2. Afin de structurer des scènes 3D complexes, nous avons développés deux modules de transformation :
  - Le premier permet de découper des scènes 3D, comportant beaucoup de plans d'occlusions (batiments, villes), en cellules convexes respectant au mieux la topologie de la scène (Fig 3).  
Ainsi, pour un bâtiment les cellules seront les pièces qui le constituent et pour une ville les cellules seront les routes et les parques. L'algorithme utilise massivement le matériel graphique de la machine pour effectuer le découpage, ce qui accélère le traitement de manière très importante.
  - Le second module récupère le résultat du découpage et effectue un calcul de visibilité entre les différentes cellules (Fig. 3). A l'issue de ce traitement, chaque cellule référence l'ensemble des cellules potentiellement visibles depuis tout point de son enveloppe convexe. De cette façon, lors de la navigation, si l'utilisateur se situe dans une cellule, il suffit de charger uniquement les cellules potentiellement visibles et de les afficher pour construire l'image présentée à l'utilisateur. Comme les scènes

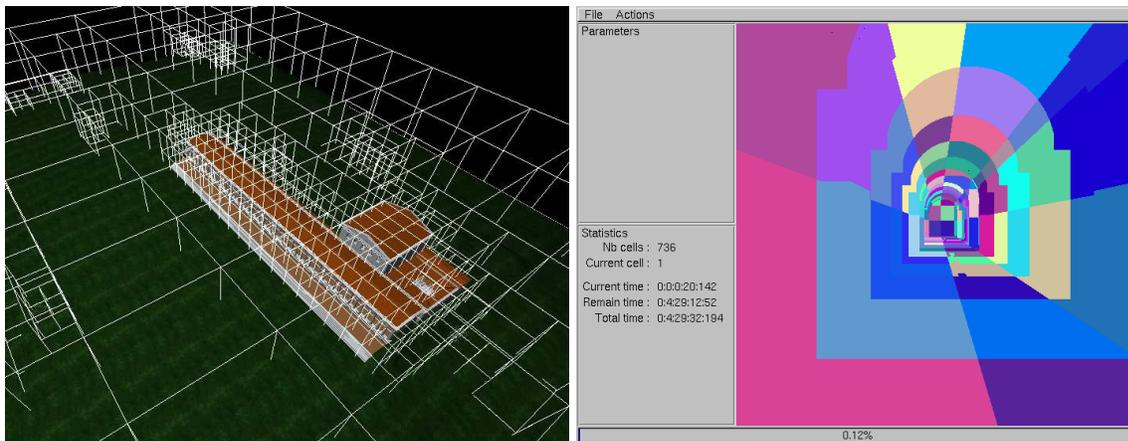


Figure 3. A gauche : subdivision spatiale d'une scène 3D représentant un temple Grec. A droite : calcul de visibilité sur une scène 3D représentant un musée.

comportent beaucoup de plans d'occlusion, l'ensemble des cellules potentiellement visibles est généralement petit, ce qui optimise nettement les transferts réseaux ainsi que le temps de rendu des images. L'algorithme de visibilité utilise lui aussi massivement le matériel graphique pour accélérer les calculs. Enfin, cet algorithme permet de calculer pour chaque cellule potentiellement visible une valeur d'importance visuelle qui permet aux modules de visualisation d'effectuer une gestion dynamique de la charge de la machine cliente. Cette gestion dynamique permet la navigation (de plus ou moins bonne qualité) quel que soit la puissance de la machine cliente.

3. L'ensemble des cellules potentiellement visibles depuis une même cellule est parfois trop volumineux. Pour réduire la taille de cet ensemble, nous avons développés un nouveau type d'imposteurs permettant de remplacer un ensemble de cellules lointaines. Ainsi, lors de la navigation, au lieu de charger l'ensemble des cellules lointaines et de les utiliser pour construire leur représentation, il suffit de charger et d'afficher un unique imposteur représentant cet ensemble. Le nouveau type d'imposteurs que nous avons développés est basé sur des textures dynamiques multi-resolutions. Ces textures peuvent donc être transférées de manière progressive via un réseau. L'aspect dynamique signifie que le contenu de ces textures varie suivant la position de l'utilisateur par rapport au polygones sur lesquels elles sont plaquées (Fig ; 4). Nous avons montrés, à l'aide de plusieurs jeux de tests, que ces textures peuvent être utilisées pour remplacer des ensembles de géométrie complexes de différentes manières. Par exemple, un polygone portant une telle texture peut être placé dans l'encadrement d'une fenêtre afin de représenter l'ensemble des objets situés derrière cette fenêtre. Il est aussi possible de plaquer de telles textures sur les faces de la boîte englobante d'un unique objet constitué d'un très grand nombre de polygones afin d'en produire une représentation de qualité.
4. Enfin, la plupart des scènes 3D complexes utilisent un grand nombre de textures pour améliorer la qualité visuelle des objets. Le transfert de textures de grande qualité via un réseau est un enjeu particulièrement intéressant puisque de telles textures consomment en général beaucoup de mémoire. Pour en optimiser les transferts, nous avons développés un format de textures progressives directement lié aux caractéristiques propres aux textures (dimensions en puissances de deux, ...). A l'aide de ce format et des informations d'importance visuelles générées par le module de visibilité, nous parvenons à déterminer la résolution de chaque texture nécessaire à l'affichage (Fig. 5). De cette façon, nous pouvons nous permettre de ne charger que les niveaux de détails de texture nécessaires à l'affichage. Précisons que cette technique permet aussi d'alléger le matériel graphique

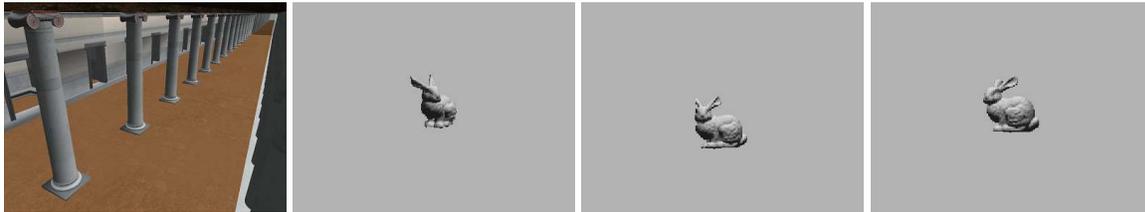


Figure 4. A gauche : les cellules du fond du temple Grec sont remplacées par un imposteur. A droite : représentation d'un objet en plaquant des imposteurs sur les faces de sa boîte englobante. La plate-forme affiche 120 images par secondes en utilisant les imposteurs au lieu de 40 en utilisant le maillage du lapin.



Figure 5. A gauche : image rendue à l'aide de l'allocation automatique de charge sur les textures. Les textures au fond du musée sont en niveau bas tandis que la plus proche est en très haute résolution. Les textures d'origine sont en 1024x1024x24bits. La plate-forme affiche 23 images par secondes. Sans l'optimisation, la plate-forme affiche 2 images par secondes. A droite : vue rapprochée du tableau, toujours à 23 images par secondes.

qui ne propose un général qu'une petite capacité mémoire.

## 6.2. Génération de villes

**Participants :** Kadi Bouatouch, Julien Perret, Jean-Eudes Marvie.

**Mots clés :** *Génération de modèles de villes, L-systems, systèmes de réécriture.*

Nous avons conçu un logiciel permettant la création de villes à l'aide de modèles procéduraux basés sur un système de réécriture.

Nous avons conçu et mis en oeuvre un logiciel s'appuyant sur une méthode procédurale de modélisation de villes grâce à des L-systems dans lequel un ensemble de règles de réécriture est utilisé pour générer de façon statique et définitive un vaste et complexe modèle 3D. Notre approche utilise de façon différente le mécanisme de réécriture : en effet, celle-ci propose la représentation sous forme de système de réécriture non pas comme représentation initiale perdue après la génération du modèle, mais bien comme représentation de référence. C'est donc le système de réécriture qui est stocké, éventuellement transmis à travers un réseau puis utilisé lors de la navigation temps-réel. La navigation temps réel est alors effectuée à l'aide de la plate-forme de navigation Magellan (Fig. 6).

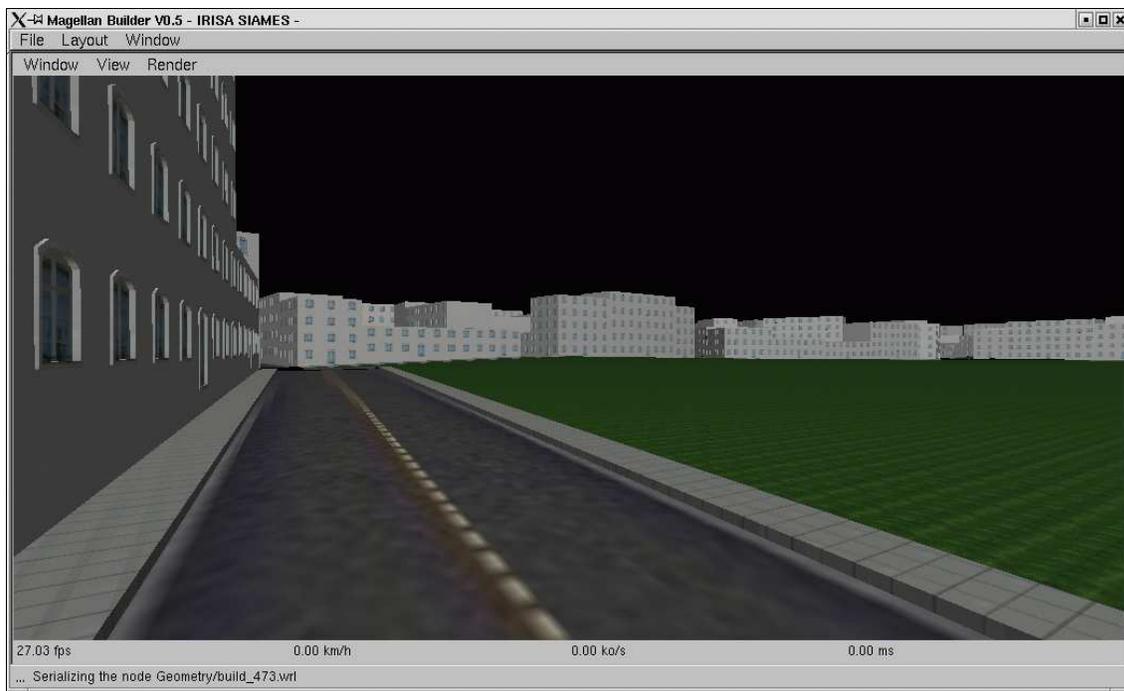


Figure 6. Modèle de ville visualisé sous Magellan. Les bâtiments sont générés par une même L-system. Les parcs, les routes et les carrefours sont aussi respectivement générés par trois L-systems différents.

## 6.3. Modèles mécaniques et prototypage virtuel.

**Participants :** Georges Dumont, Guillermo Andrade, Christofer Köhl.

**Mots clés :** *Modèles mécanique, prototypage virtuel, retour sur conception.*

Nous avons développé, ou organisé l'intégration, de deux modules de simulation dynamiques (SMR et Dynamo) dans la plate-forme logicielle GASP. Des travaux sont actuellement en cours sur l'optimisation des

ces modules ainsi que sur leur facilité d'utilisation. Les premiers résultats concernant la progression d'un modèle d'endoscope dans une base de données médicale sont présentés ici.

### 6.3.1. Présentation des résultats pour l'endoscope

Nous avons défini un modèle mécanique d'endoscope. Celui-ci est basé sur un enchaînement de modules articulés les uns par rapport aux autres. L'actionnement est assuré par des actionneurs en alliage à mémoire de forme dont nous avons modélisé le comportement. L'environnement extérieur est issu d'une base de données médicale acquise, de manière routinière, par scanner. Cette base est prétraîtée pour permettre l'interaction entre le modèle d'endoscope et celle-ci. En effet, le calcul des interactions de contact nécessite, dans une première phase, la détermination géométrique d'interpénétration au cours de la simulation et, dans une deuxième phase, le calcul des forces d'interactions. Ce dernier fait intervenir des calculs de distances, que nous avons choisis de précalculer pour viser une réponse du simulateur proche du temps réel. L'intérêt d'un endoscope actif est surtout le fait que les articulations sont commandables. Cette propriété est utilisée, sur la figure 7, où une commande, basée sur une approche multi-agents, est utilisée pour minimiser les contacts entre l'endoscope et le vaisseau inspecté. Ceci permet, en testant différentes stratégies de commande, de définir l'endoscope qui blesse le moins le patient, en minimisant les efforts de contact.

## 6.4. Simulation avec retour d'efforts.

**Participants :** Bruno Arnaldi, Tangi Meyer, Guillermo Andrade, Anatole Lecuyer.

**Mots clés :** *architecture logicielle, retour d'efforts, réalité virtuelle.*

L'objectif de ces travaux est le développement d'applications de réalité virtuelle avec retour d'efforts. Nous avons analysé les besoins en terme d'outils logiciels pour les simulations de ce type. Nous avons identifié un ensemble de composants nécessaires : détection de collisions et traitement, solveur mécanique, outil pour l'interaction bilatérale. Nous avons réalisé un travail de formalisation des échanges entre ces différents modules afin de proposer une architecture générique pour la simulation avec retour d'efforts exploitant OpenMASK. Un premier prototype a été validé et nous procédons actuellement à des évolutions afin de le rendre plus générique et plus robuste.

### 6.4.1. Contexte et Objectifs

Nous nous intéressons aux applications du retour d'efforts en réalité virtuelle. Les outils dont nous disposons sont : le système IMMERSIA (écran géant et calculateur haute performance), le périphérique Virtuose6D acquis dans le cadre de Perf-RV et la plate-forme de simulation OpenMASK. Notre objectif est de développer, en s'appuyant sur OpenMASK, des outils destinés à faciliter la mise en oeuvre d'applications de réalité virtuelle exploitant le retour d'efforts.

### 6.4.2. Identification des besoins

Une grande variété d'applications de réalité virtuelle utilise le retour d'efforts (chirurgie virtuelle, prototypage virtuel, apprentissage, ergonomie...etc). Pour couvrir les besoins issus de ces applications, différents outils spécialisés ont été développés. Dans ces conditions, une architecture générale pour le retour d'efforts en réalité virtuelle doit faciliter l'intégration de composants logiciels basés sur des technologies très variées. Ces composants seront contenus dans quatre grands types de modules :

- détection de collisions
- traitement de la collision
- résolution des équations de la mécanique
- interaction bilatérale

### 6.4.3. Architecture

Nous avons formalisé les échanges de données entre ces modules. Ce qui s'est traduit par une implémentation d'interfaces génériques sur la plate-forme OpenMASK. La figure 8 présente le schéma général de l'architecture que nous avons développée.

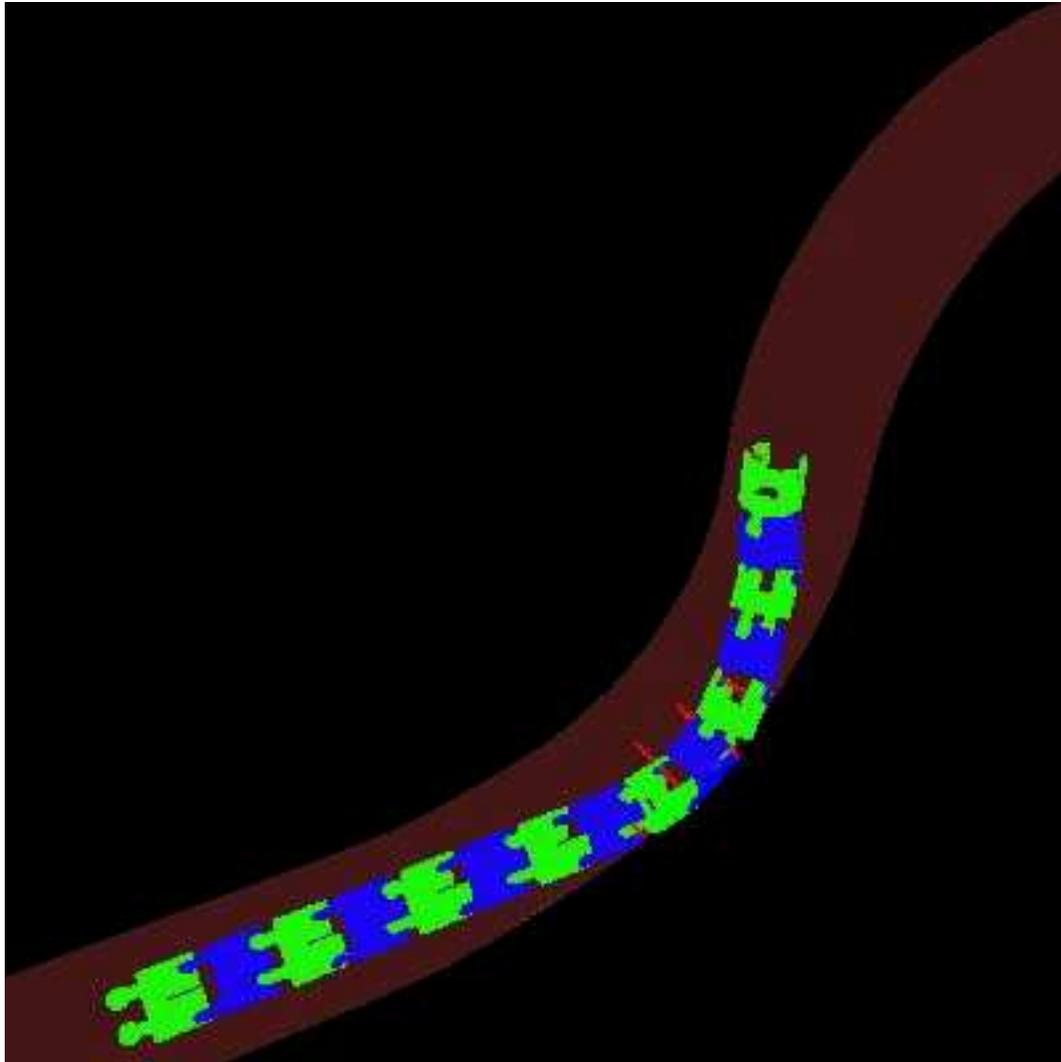


Figure 7. Minimisation du contact par une stratégie de commande multi-agents

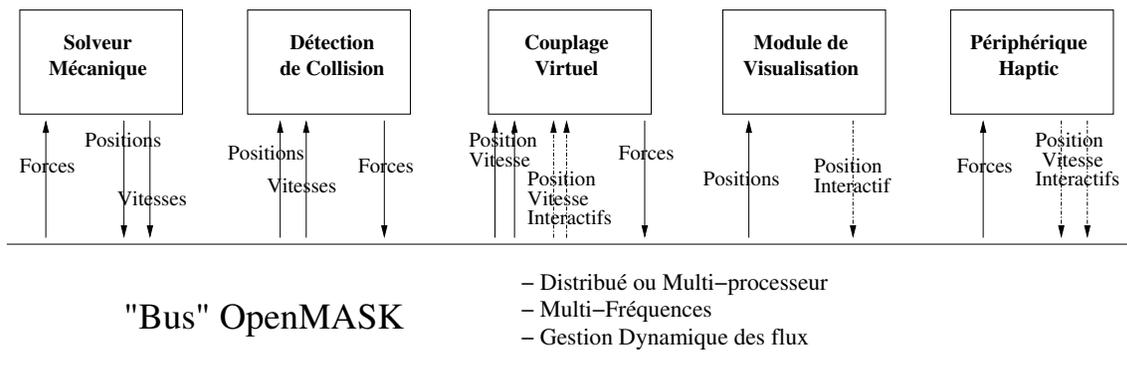


Figure 8. Schéma général de l'architecture

L'objectif premier de nos travaux, l'interchangeabilité des modules, a été validé. Nous avons en effet intégré différentes bibliothèques de simulations physiques sur la base de nos interfaces : dynamo, SMR et CONTACT. Des travaux du même type ont été réalisés pour les bibliothèques de détection de collisions comme SWIFT++ et VCollide. La manipulation interactive est réalisée au travers d'un couplage virtuel.

#### 6.4.4. Généralisation des types des flux d'entrées/sorties

Le couplage virtuel consiste à simuler un ressort et un amortisseur virtuel entre deux objets. Deux principes d'intégration ont été envisagés pour cet outil (figures 9 (a) et 9 (b) ).

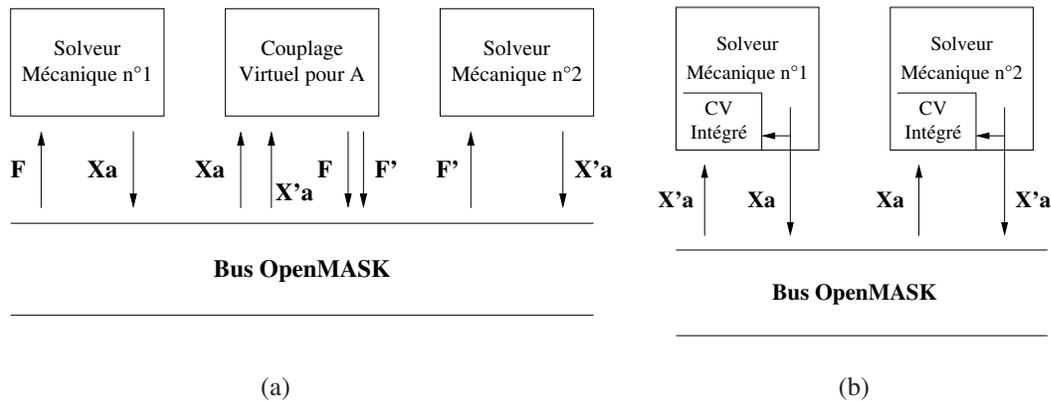


Figure 9. (a) Architecture générale pour le couplage virtuel. (b) Architecture avec couplage virtuel intégré.

Une étude formelle réalisée avec Simulink a mis en évidence une meilleure robustesse de notre architecture pour le modèle avec un couplage virtuel intégré. Nous avons donc généralisé les types des données échangées. Nous élargissons ainsi la gamme des modèles de traitement accessibles. Nous implémentons l'interface associée (figure 10).

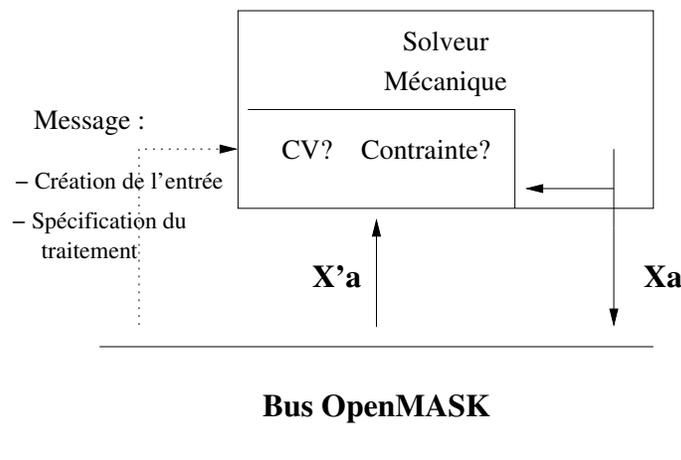


Figure 10. Modèle d'entrée généralisée.

## 6.5. Interactions en Univers Virtuels 3D

**Participants :** Thierry Duval, Alain Chauffaut, Michael Rouillé, Christian Le Tenier, Chadi Zammar.

**Mots clés :** *interactions immersives, réalité virtuelle, coopération, multimodalité.*

Nous continuons à améliorer les possibilités d'interaction offertes entre un utilisateur et les environnements virtuels 3D, puis d'une part à enrichir ce dialogue en explorant différentes possibilités d'interaction dans le domaine :

- de la coopération entre utilisateurs,
- des interactions multi-modales,

et d'autre part à automatiser les moyens à mettre en œuvre afin de pouvoir interagir avec des objets virtuels, en offrant :

- des moyens d'intégrer efficacement et de façon homogène des dispositifs physiques d'interaction,
- des moyens de rendre interactifs des objets virtuels.

Ces travaux de recherche utilisent dorénavant la plate-forme OpenMASK comme support de développement afin de prouver les concepts, de réaliser des démonstrateurs, et d'offrir ces moyens d'interaction aux autres membres du projet. Nous avons donc consacré un effort important au portage de nos travaux précédents de GASP vers OpenMASK. Les interactions à plusieurs utilisateurs et/ou multimodales s'appuient donc maintenant sur le paradigme OpenMASK de communication par flot de données entre objets de simulation via leurs entrées et leurs sorties, permettant ainsi une fusion aisée d'entrées en provenance de plusieurs canaux. La distribution des interactions sur des sites distants s'appuie quant à elle sur le mécanisme d'OpenMASK mettant en relation les référentiels et leurs miroirs.

Les travaux menés cette année concernent :

La coopération entre utilisateurs : Les coopérations entre utilisateurs distants et les environnements virtuels 3D étant possibles grâce à la mise au point de la version PVM et des travaux menés autour de la répartition des objets de simulation OpenMASK sur plusieurs processus, nous avons pu travailler sur plusieurs points :

- amélioration de la mise en évidence des interactions coopératives : toujours afin de permettre aux utilisateurs de mieux comprendre les actions des autres, pour plus d'efficacité en coopération. Nous proposons maintenant soit des boîtes englobantes, soit une version dilatée de la géométrie d'origine.



Figure 11.

Ces mise en évidence sont composites et peuvent être accompagnées d'un repère, des noms des utilisateurs ou encore des offres d'interaction associées (manipulation en translation seule, en rotation seule, en translation et en rotation) ;

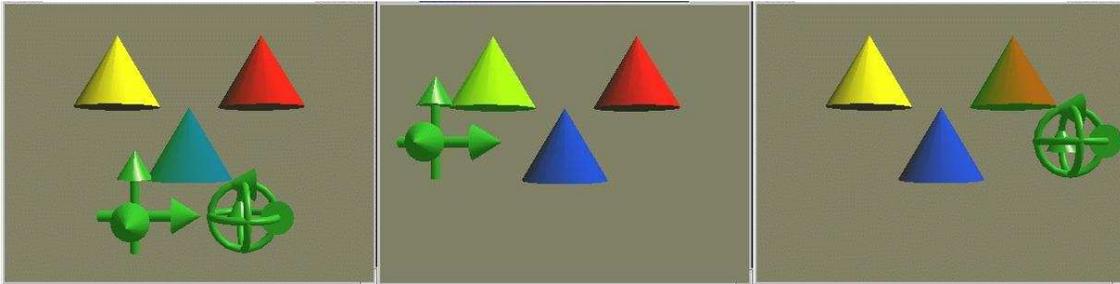


Figure 12.

- proposition de nouveaux paradigmes d'interaction permettant la saisie et la manipulation d'objets, construits sur nos pilotes de périphériques d'interaction. S'ajoutent donc maintenant au rayon virtuel, au rayon virtuel à taille variable et à la main virtuelle des dispositifs comme le curseur 3D et plusieurs rayons déformables. Ces derniers, développés sous GASP, n'ont pas encore été portés sous OpenMASK.

L'intégration d'un menu 3D pour l'interaction : Le concept de CCube (menu 3D permettant des choix en contexte immersif) développé par Jérôme Grosjean et Sabine Coquillart, a également été porté sous OpenMASK par Jérôme Grosjean. Nous avons à notre tour réimplémenté ce concept, en le rendant plus générique et plus facilement paramétrable.

De nouvelles métaphores d'interaction : Dans le cadre du stage de DEA d'Aurélien Fenals, nous avons également proposé et implémenté de nouvelles métaphores pour des outils virtuels permettant une coopération simultanée sur des objets partagés ou contraints, mettant en évidence que l'utilisateur ne peut pas manipuler l'objet exactement de la façon dont il le souhaiterait. Ces nouvelles métaphores ont été développées sous GASP et ne sont pas encore portées sous OpenMASK. Ces métaphores sont celles :

- du rayon avec élastique s'étirant entre l'endroit où l'objet devrait se trouver et l'endroit où il se trouve effectivement ;

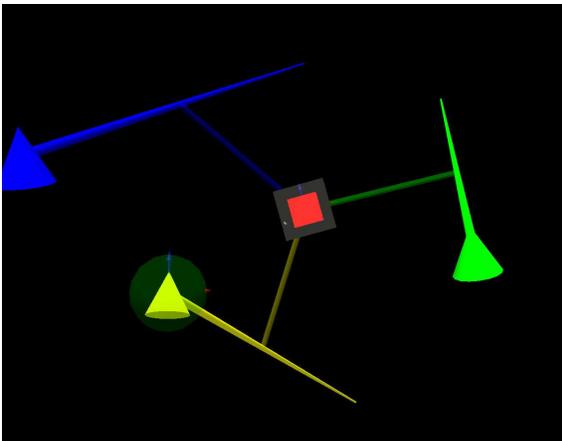


Figure 13.

- du rayon qui se plie en deux pour garder une extrémité sur l'objet manipulé tout en gardant l'autre cohérente avec les données (position et angle) de manipulation de l'utilisateur ;

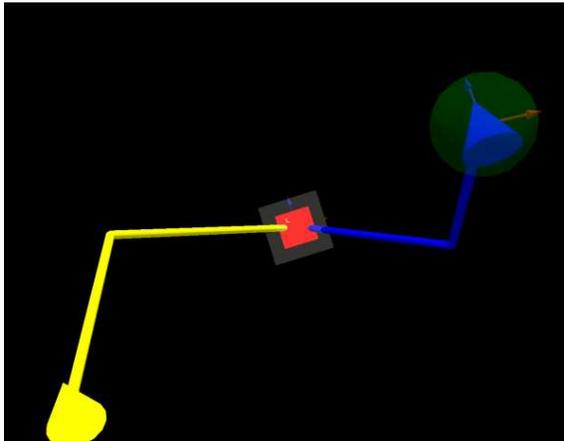


Figure 14.

- du rayon déformé qui se superpose au rayon initial, dont une extrémité se trouve sur l'objet manipulé et dont l'autre extrémité est tangente à ce rayon initial.

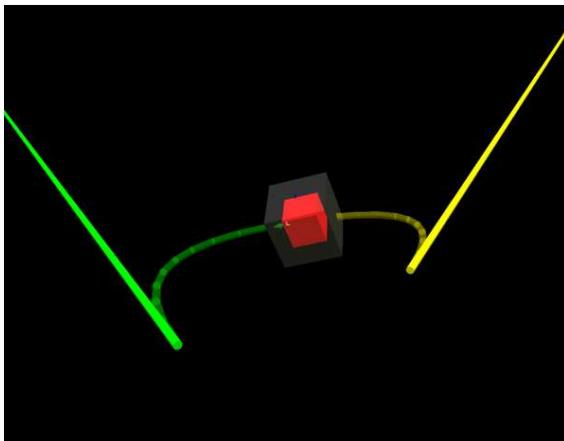


Figure 15.

Les adaptateurs pour objets interactifs : Nous continuons à travailler sur les méthodes d'adaptation efficace des modèles d'architecture logicielle issus du domaine des Interfaces Homme-Machine tels que le modèle PAC-Amodeus, et d'autre part à fournir des adaptateurs logiciels pour rendre interactifs *a posteriori* des objets de simulation qui n'avaient pas été conçus pour l'interaction. Ces adaptateurs permettent aux objets rendus ainsi interactifs d'interagir avec plusieurs types d'interacteurs, si ceux-ci implémentent le bon protocole d'interaction.

Les adaptateurs pour objets interactifs contraints : Nous fournissons maintenant des adaptateurs permettant d'obtenir des interactions contraintes sur les objets de nos univers. Ces contraintes sont des contraintes cinématiques, qu'il convient de préciser lors de l'assemblage des objets virtuels. La mise en évidence visuelle de ces contraintes (autre que le résultat correct de l'interaction) par ajout d'informations 3D n'est pas encore implémentée.

la visualisation de la latence en coopération : Toujours dans le but de fournir un maximum d'informations aux utilisateurs en cours d'interaction, nous nous attachons aussi à essayer de visualiser les écarts de valeurs entre les objets de simulation réels et leurs miroirs. Ceci pour faire comprendre à l'utilisateur situé sur le même noeud réseau qu'un objet que les autres utilisateurs peuvent avoir une perception différente de cet objet, due au retard des mises à jour entre le référentiel et ses miroirs.

Nous souhaitons offrir ce service à l'aide d'une légère désynchronisation entre les différents participants d'une session coopérative.

Ceci nous a amené dans un premier temps à travailler sur le noyau d'OpenMASK, afin de permettre à nos applications de supporter des coupures momentanées du réseau. Ces travaux nous offrent de ce fait une meilleure robustesse envers les coupures de réseau qui peuvent se produire de temps à autre.

Ensuite, nous devons travailler l'aspect visuel des objets de simulation afin de visualiser, via des partenaires privilégiés du module de visualisation interactive, les écarts entre valeurs réelles et valeurs estimées chez les miroirs.

## 6.6. Animation référencée vision et comportements

**Participants :** Nicolas Courty, Bruno Arnaldi.

En synthèse d'images, la qualité d'une animation dépend souvent de spécifications complexes et/ou peu intuitives. Dans le cadre de ce travail, nous nous sommes intéressés à l'étude de ce problème en appliquant une technique issue du domaine de la robotique : l'asservissement visuel. Cette technique permet de générer des mouvements 3D au travers d'une spécification 2D (dans l'image) d'une tâche visuelle. Deux grandes classes de problème ont été étudiés dans cette optique : le contrôle d'une caméra en environnement virtuel et la modélisation de l'attention visuelle pour les humanoïdes autonomes de synthèse. L'utilisation de l'asservissement permet dans les deux cas de spécifier simplement ce que l'on cherche à obtenir ; dans le cas de la caméra on spécifie ce que l'on cherche à observer, alors que dans le cas de l'humanoïde on spécifie ce qu'il doit observer. Grâce au principe de redondance, il nous a été permis de prendre en compte de manière dynamique et en temps réel des modifications de l'environnement, ou d'adresser des problèmes relatifs à l'animation. Ainsi dans le cadre de la caméra, les problèmes suivants ont été traités : suivi de trajectoires et planification, gestion des occultations et des collisions, problème de la « photo » ; dans le cadre de l'humanoïde : lien avec la locomotion, évitement des butées articulaires, gestion du réalisme. De plus, la validité de notre approche a été testée dans le cadre de l'animation comportementale, en modélisant à la fois des comportements type cinématographique pour la caméra, ainsi qu'un panel de comportements de perception pour l'humanoïde. Les résultats de cette étude ont été directement utilisés dans différents cadres applicatifs, dont l'animation d'un agent communicant réalisée par FranceTelecom R&D, ainsi qu'une démonstration de fiction interactive présentée lors d'Imagina 2002.

Ces recherches ont été menées en collaboration avec l'équipe Vista et portent sur plusieurs thèmes : l'animation d'une caméra et d'un humanoïde virtuel selon le principe d'animation référencée vision, et l'extension de ce domaine à la cinématographie virtuelle et la simulation de la perception visuelle d'un humanoïde de synthèse. Ces deux premiers thèmes ayant été déjà abordés dans les précédents rapports d'activité, nous ne nous concentrons ici que sur les deux derniers thèmes, tout en précisant que l'animation de caméra et l'animation d'humanoïdes sont deux thèmes qui continuent à être étudiés.

### 6.6.1. Cinématographie virtuelle

La cinématographie virtuelle désigne la capacité à pouvoir choisir les successions de points de vue depuis lesquels les environnements virtuels en 3D sont rendus. En cela, de nombreuses analogies avec le cinéma sont réalisables ; la manière de montrer les choses contient en elle une intention. De plus, cette manière sert souvent la narration, les mouvements de la caméra participent à créer une ambiance, à transmettre un contenu émotionnel. Depuis la création du septième art, les réalisateurs de films ont établi (de façon plus ou moins implicite) un ensemble de règles et de conventions permettant de transmettre les informations d'une manière

compréhensible et efficace. Cependant, le niveau de description de ces règles et conventions n'est pas assez détaillé pour définir une grammaire formelle du langage cinématographique. De plus, il existe une latitude nécessaire employée par les auteurs de film, sans quoi la dimension artistique ne serait pas présente. Une autre différence majeure entre la cinématographie réelle et virtuelle est la contrainte de temps réel. Un réalisateur de film connaît à l'avance les différents éléments des scènes, peut disposer les acteurs à sa manière, et réaliser une phase de montage post tournage. A l'inverse, dans un système temps réel, il faut pouvoir au mieux rendre le contenu d'une scène à chaque instant sans en connaître les modifications futures. Une telle contrainte peut nous amener à percevoir la caméra comme un agent (représentant à la fois le cameraman et le metteur en scène). Cet agent doit ainsi être doté de moyen de perception (lui permettant de quantifier la qualité de ce qu'elle voit, d'avoir des objectifs visuels), d'action (se déplacer et s'orienter dans l'environnement), et enfin de décision (que filmer ? dans quelles conditions ?). Si l'on considère simplement le système d'animation décrit dans la première partie de cette étude, seul le lien action-perception a été traité. L'objet de cette extension à l'animation référencée vision est de pouvoir rajouter un modèle décisionnel (HPTS) dans une optique de cinématographie virtuelle. Ainsi, nous avons établi comment réaliser des tâches élémentaires de positionnements cinématographiques, ainsi que les mouvements de caméra associés et connecter le module décisionnel au système d'animation, dans le but de définir un système réalisant automatiquement le tournage et le montage de **scènes dynamiques**.

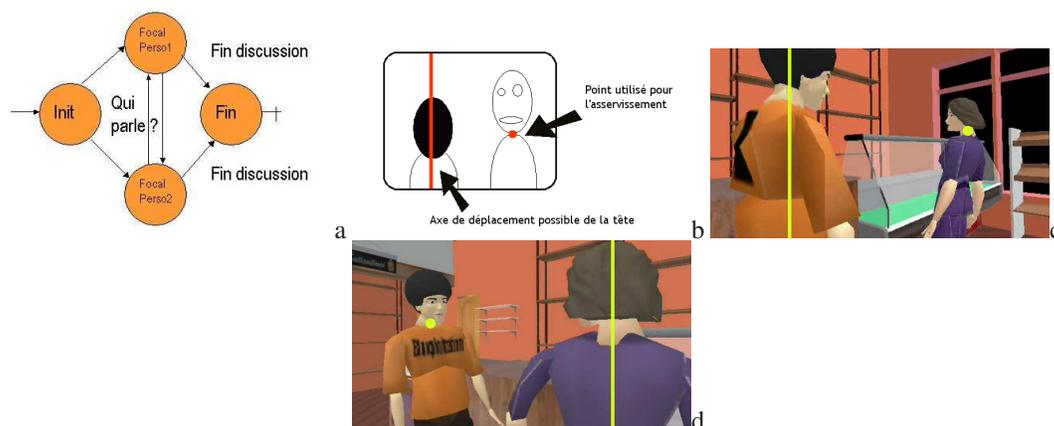


Figure 16. (a) Automate simple de gestion d'un dialogue entre deux personnages (b) Primitives d'asservissement pour la réalisation d'un champ contre-champ (c,d) exemples de plans associés aux deux états Focal perso ; les objectifs visuels sont en jaune dans l'image

### 6.6.2. simulation de la perception visuelle d'un humanoïde de synthèse

La modélisation du comportement lié à la perception visuelle d'un humanoïde de synthèse est un problème difficile ; ce comportement produit des indications subtiles et émergentes sur les intentions et les processus cognitifs régissant l'humanoïde. L'utilisation d'animations capturées ou basées sur des scripts se révèle inadaptée, car la variété des schémas possibles est trop importante pour produire des animations crédibles, et la nécessité d'adaptation à l'environnement implique un contrôle élevé ainsi qu'une adaptabilité forte. Notre objectif est donc de pouvoir simuler la perception visuelle de l'humanoïde pour :

- augmenter le réalisme visuel des animations,
- fournir une brique de base essentielle à la modélisation d'un humanoïde autonome.

Dans la première partie de cette étude, nous avons présenté un système d'animation répondant à une partie de ces exigences (réactivité, modalité de contrôle). Notre objectif est alors d'établir le lien entre ce modèle d'animation et un module d'animation comportementale établissant les différents comportements perceptifs

de l'humanoïde. nous avons développé un formalisme permettant de mettre en place une architecture pouvant simuler la perception visuelle d'un humanoïde de synthèse. Après avoir singularisé deux modalités perceptives distinctes : une perception endogène et une perception exogène, nous avons établi deux méthodes permettant de traiter ces deux aspects. Dans les deux cas, la cinématique associée à l'acte de perception visuelle a été réalisée grâce au système d'animation référencée vision. L'utilisation d'HPTS (protocole de définition des comportements) nous a permis de définir des comportements pouvant intégrer des paramètres de type cognitif, ainsi que des comportements de perception passive basés sur un modèle de cartes de saillance. L'utilisation de cette approche, nouvelle dans le domaine, constitue une solution intéressante au problème. A la différence des travaux réalisés dans le domaine de la perception visuelle d'un humanoïde de synthèse, nous combinons les deux aspects *environnements informés* et *acquisition d'informations à partir de l'image perçue par l'humanoïde*. La coexistence de ces deux aspects permet de se rapprocher de manière significative de la manière dont l'être humain perçoit les objets.

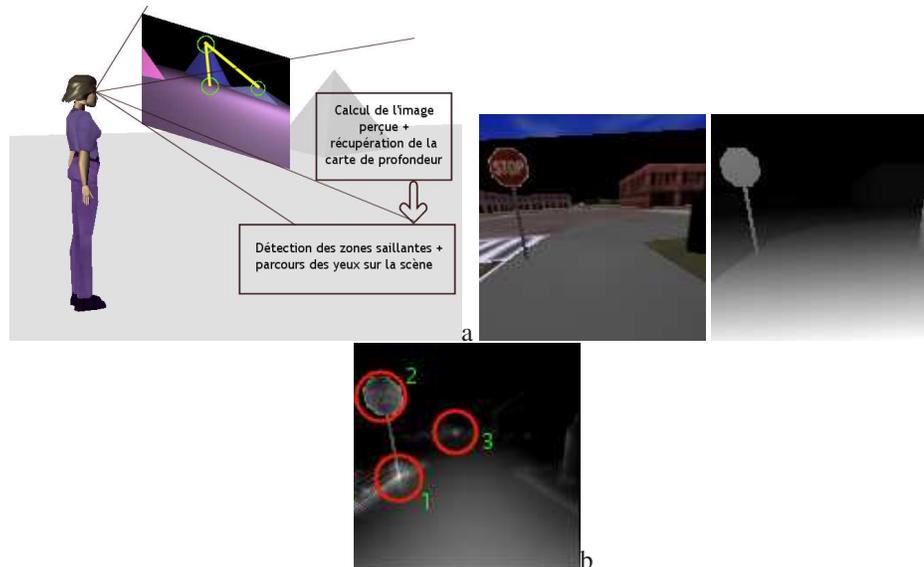


Figure 17. (a) Utilisation du principe de saillance (b) exemples de ce traitement sur des scènes virtuelles

## 6.7. Modélisation d'environnements urbains informés pour les agents autonomes

**Participants :** Romain Thomas, Stéphane Donikian.

Au vu des différents travaux effectués en Animation Comportementale, engageant des agents autonomes évoluant dans des environnements urbains, il apparaît que ces agents ont des capacités de locomotion et de perception de leur environnement mais pas d'interaction langagière sur ce même environnement. Le fait de leur permettre d'échanger des informations sur leur environnement apporte indéniablement plus de réalisme aux simulations. En environnement urbain, une tâche de communication intéressante est celle de pouvoir indiquer sa route à un visiteur perdu ou en quête d'information (agents autonomes n'ayant pas une bonne connaissance des lieux ou avatar d'une personne immergée dans l'environnement virtuel). Celle-ci a de plus le mérite d'enrichir les comportements possibles dans la simulation : il est nécessaire pour un agent désireux d'établir une communication, de chercher une interaction gestuelle et verbale avec les autres agents. De plus, une fois les informations pertinentes communiquées, il est possible d'observer la façon dont l'agent a intégré ces informations en étudiant la façon dont il les utilise pour naviguer dans l'environnement (et éventuellement

se perdre). Pour réaliser cet objectif, il est important de modifier les bases de données existantes décrivant des environnements urbains, de façon à y inclure des informations cognitives supplémentaires. Il est aussi nécessaire que l'agent soit doté d'un modèle de mémoire humaine que nous adapterons des modèles de psychologie cognitive. Enfin, il faut adapter les modèles d'échange d'informations sur les chemins, issus de la psychologie expérimentale, au modèle que nous voulons réaliser. Ces trois axes représentent actuellement l'essentiel de nos travaux.

La majorité des bases de données urbaines informées pour l'animation comportementale réalisées à l'heure actuelle, ne présentent pas de hiérarchie, c'est à dire, peuvent être vues comme un graphe topologique informé sans niveau d'abstraction. Ou, si abstraction il y a, elle ne s'effectue que selon des découpages spatiaux arbitraires, pauvres en informations liées au comportement. Nous avons repris la notion de graphe topologique hiérarchique pour l'élaboration de notre base de données urbaines, à laquelle nous avons mêlé la notion d'intégration spatiale, formalisation issue de la psychologie expérimentale des idées de K. Lynch. De plus nous avons conçu une application permettant de connecter automatiquement un ensemble de bâtiments virtuels au réseau routier de la base de données sur laquelle nous travaillons, cette connexité étant essentielle aux tâches de planification, nécessaires à la communication de chemins. Le reste des développements engage l'élaboration d'un modèle de carte cognitive, dont est doté chaque agent. Cette carte cognitive est une surcouche, propre à chaque agent, du graphe topologique modélisant l'environnement. Cette carte filtre les accès que peut réaliser l'agent à la base de données.

Il est à noter que la carte cognitive ainsi modélisée présente aussi une structure de graphe topologique, dont les sommets correspondent à des objets existant dans la base de données, mais dont les arcs, représentant la connexité entre chaque sommet, ne sont pas analogues à ceux remarquables dans le graphe modélisant la base de données. Cette double modélisation permet d'entretenir simultanément une représentation exacte de la connexité dans l'environnement, ainsi qu'une représentation plus floue et propre à chaque agent, destiné à une planification de chemins de plus haut niveau. Tous les arcs et tous les sommets de la carte cognitive sont dotés de paramètres de *Rappel* et de *Reconnaissance*, qui sont les deux tâches de recouvrement d'informations, propres à la mémoire humaine. De plus, chaque agent se voit doté d'un contrôleur dynamique de la mémoire, qui assure la gestion dynamique des paramètres mémoriels de la carte cognitive de l'agent en fonction de la navigation, du temps, et de son comportement, ainsi que les seuils d'activation propres à ces différents paramètres.

A terme, l'agent peut donc planifier un chemin, en ne tenant compte que des informations *rappelées* de sa carte cognitive. Si l'agent ne collecte pas suffisamment d'information pour pouvoir planifier un chemin cohérent, il bascule dans un mode de navigation réactive, jusqu'à rencontrer un élément *reconnu* de sa carte mentale, et basculer de nouveau dans le mode de navigation planifiée.

Le dernier axe concernant l'élaboration d'un système permettant l'échange d'information entre agents sera l'objet de nos développements pour l'année 2003.

## 6.8. BCOOL : Behavioral and Cognitive Object Oriented Language

**Participants :** Fabrice Lamarche, Stéphane Donikian.

**Mots clés :** *animation comportementale, raisonnement, planification, langage.*

La description d'un monde virtuel est un tâche fastidieuse surtout lorsque l'on doit offrir une base de connaissances à un humanoïde devant réfléchir sur celui-ci. En partant de cette constatation, nous avons spécifié et développé un langage, intitulé BCOOL, ayant pour but d'alléger cette phase de description. Ce langage s'appuie sur les concepts de la programmation objet, qui a de grandes qualités en terme de développement et de réutilisation. Les entités du monde sont donc décrites dans un premier temps de façon abstraite puis sont instanciées à volonté pour décrire un monde. Un moteur d'inférence se charge ensuite de déduire l'ensemble des informations pertinentes nécessaires au raisonnement des humanoïdes, permettant ainsi d'alléger le processus de description et de faire fonctionner un module de planification réactive.

Pour permettre à un humanoïde virtuel de raisonner sur son environnement, il faut offrir aux concepteurs des outils leur permettant de décrire ces environnements et les moyens d'actions sur ce dernier. Pour ce faire, le langage BCOOL a été développé.

Il s'agit d'un langage orienté objet permettant de décrire les différents types d'entités présentes dans un monde (notion de classe d'entité) ainsi que les différentes relations entre ces entités. Une entité abstraite (classe) est décrite de la manière suivante :

- Éventuellement une ou plusieurs classes mères (notion d'héritage).
- Des données membres qui sont des instances d'objets décrits par l'intermédiaire du langage.
- Des propriétés booléennes permettant de faire du *situation calculus*.
- Des actions paramétrées par des objets décrits dans le langage. Une action est décrite par :
  - Des contraintes portant sur les paramètres de l'action.
  - Une précondition qui est une expression booléenne portant sur des propriétés et des relations.
  - Un effet qui est une conjonction de propriétés et de relations.
  - La connaissance apportée par l'action. Ceci permet de décrire de la même manière les actions permettant l'acquisition de connaissances et les actions permettant de manipuler les objets (qui peuvent aussi fournir de la connaissance).

Il est à noter que les actions d'une classe mère sont redéfinissables par une classe fille (notion de polymorphisme).

Les relations entre objets sont aussi des propriétés booléennes (externes aux objets) qui acceptent des contraintes sur les objets qu'elles mettent en relation.

Par la suite, la description du monde se fait en instanciant les différentes classes d'entités en fonction du monde que l'on veut décrire. Une fois cette description effectuée, un moteur d'inférence PROLOG calcule le monde instancié avec toutes les instances d'objets, de relations, d'actions et de propriétés en respect des différentes contraintes qui leur sont associées. Cette phase est critique car la combinatoire peut être très grande, cependant, dans la mesure où il s'agit d'une phase de précalcul, ceci reste acceptable. Enfin, une base de données correspondant au monde instancié est générée en offrant des structures permettant au moteur de raisonnement de fonctionner rapidement.

Pour la phase de raisonnement, une structure de graphe est générée, à partir d'une partie de la base de données, contenant des noeuds correspondant à des propriétés et des noeuds correspondant à des actions ainsi que des arêtes correspondant à des liens de précondition ou d'effet. Il devient alors possible en mettant ce graphe en couche de trouver des chemins d'actions connectant les faits vrais aux actions ainsi réalisables et permettant d'évoluer vers la réalisation d'un but fixé.

Cet environnement de description d'un monde virtuel offre toutes les qualités des langages orientés objets tout en étant dédié à la description de connaissances et de moyens d'actions. Lorsqu'un monde est décrit, il est possible de réutiliser à volonté les entités utilisées pour le décrire et d'en faire des spécialisations. Le moteur de raisonnement, par rapport aux moteurs classiques de planification est relativement peu coûteux en terme de calculs et accepte les modifications dynamiques de l'environnement ainsi que la remise en cause continue du plan actuellement en train de se dérouler, ce qui constitue une contrainte forte en animation comportementale.

## 6.9. Navigation d'humanoïdes en environnement complexe

**Participants :** Fabrice Lamarche, Stéphane Donikian.

**Mots clés :** *animation comportementale, triangulation de Delaunay, planification de chemin, navigation réactive.*

Pour naviguer dans un environnement, un humanoïde virtuel a besoin d'une représentation interne lui permettant de raisonner sur la topologie des lieux l'entourant et de planifier son chemin. D'autre part, il est nécessaire d'introduire lors de la navigation une phase de navigation réactive, permettant d'éviter les collisions entre entités. Pour représenter l'environnement, la base de données géométriques est traitée de manière à en extraire un plan sous forme de cellules convexes, puis pour la navigation réactive, une triangulation de Delaunay est calculée. Ceci permet d'obtenir une planification de chemin rapide ainsi que d'alléger le poids en calcul des contrôleurs de navigation réactive.

Pour naviguer dans un environnement, les humanoïdes virtuels ont besoin d'une représentation interne leur permettant de raisonner sur la topologie des lieux en vue de planifier leur chemin. D'autre part, ils ont aussi besoin d'avoir une connaissance sur les autres humanoïdes peuplant l'environnement pour pouvoir naviguer de manière réactive. Nous proposons donc deux approches complémentaires permettant de fournir les informations nécessaires à la réalisation des tâches de planification et de navigation réactive.

Dans un premier temps, la base de données géométriques de l'environnement est traitée de manière à en extraire automatiquement un plan. Ceci se fait par l'intermédiaire d'une triangulation de Delaunay contrainte de la projection de la géométrie sur le sol. Les triangles sont donc constitués de deux types de côtés : les côtés contraints correspondant aux limites des différents obstacles de l'environnement, les côtés non contraints ajoutés par la triangulation. Par la suite ces triangles sont réunis s'ils possèdent un côté commun non contraint et si leur union crée un polygone convexe. En fin de traitement, une carte constituée de polygones correspondant à la topologie est générée. Cette représentation offre plusieurs avantages :

- Cette carte représente de manière très précise l'environnement et permet de se déplacer dans l'environnement de manière très précise.
- Il est possible de déduire automatiquement un graphe de cheminement en connectant les cellules adjacentes possédant un côté non contraint en commun. Ce graphe par la suite peut être mis en couche pour permettre d'optimiser les calculs de position et de planification de chemin.
- Cette structure permet aussi une optimisation du lancer de rayon, utile pour gérer la visibilité entre entités par exemple.

Les entités sont désormais capables de planifier leur chemin dans un environnement complexe. Pour naviguer de manière effective, elles ont besoin de connaître les autres entités peuplant l'environnement pour être en mesure d'éviter les collisions. Pour ce faire, une triangulation de Delaunay est calculée sur les différentes entités de manière à obtenir une structure permettant d'accéder rapidement aux entités voisines. D'autre part, une entité ne peut avoir conscience d'une autre entité que lorsque cette dernière est potentiellement visible, la triangulation est donc filtrée via le lancer de rayon dans la structure de cellule pour ne garder que des liens de visibilité entre entités. Cette représentation offre plusieurs avantages :

- Ceci permet de concevoir un contrôleur de navigation réactive évitant les collisions avec les plus proches voisins visibles qui sont directement fournis par la triangulation de Delaunay filtrée.
- Le coût de calcul lié à la triangulation de Delaunay est raisonnable et la même structure est exploitée par toutes les entités.
- De part les propriétés de la triangulation de Delaunay, chaque entité est connectée à l'entité la plus proche. Cette structure peut donc être utilisée pour des algorithmes de classification permettant de détecter automatiquement des groupes d'entités et de calculer leur volume englobant, permettant ainsi d'éviter un groupe plutôt qu'une seule entité.

La combinaison des techniques de planification et de triangulation de Delaunay pour la navigation réactive offre de grandes opportunités quand à la gestion d'environnements complexes, confinés ou non, et permet de gérer un grand nombre de piétons de manière autonome tout en factorisant les calculs de voisinage entre entités.

## 7. Contrats industriels

### 7.1. FranceTelecom : Modélisation d'Environnements Urbains en vue de Visualisation déportée, interactive et réaliste.

**Participant :** Kadi Bouatouch.

**Mots clés :** *simulation, ondes radioélectriques.*

Un aspect important des systèmes informatiques concerne la représentation réaliste du monde qui nous entoure. Grâce à cette représentation, nous sommes en mesure de mieux comprendre, expliquer ou reproduire différents phénomènes physiques de plus en plus complexes.

Pour ce projet, nous nous intéressons à la représentation précise d'environnements urbains. Notre objectif est de reproduire des villes entières à des fins de visualisation réaliste et interactive. Nous souhaitons également que ces villes virtuelles puissent être visitées à partir d'un site distant. Ceci est important pour de nombreuses raisons liées au commerce, au tourisme, à l'information, ou encore à l'aménagement du territoire.

Pour cela, nous proposons de fournir un ensemble d'outils permettant de :

- Modéliser, à l'aide de cartes généralisées, une ville virtuelle complète, comportant de nombreuses informations à la fois géométriques, photométriques mais également topologiques ;
- Reconstituer une ville de manière automatique ou semi-automatique à partir de données réelles telles que les modèles numériques de terrain, les images acquises à partir de satellites, de plans cadastraux, le résultat d'un système de reconstruction 3D, etc.

Une attention particulière sera portée à tous les problèmes concernant la taille des données, à la fois pour résoudre le problème de stockage mais également pour permettre d'accélérer le transfert d'une partie de la scène, représentant la ville, entre deux ordinateurs distants.

Pour contourner les contraintes liées à la taille de la base de données, nous prendrons en compte d'autres types d'information :

- représentation hiérarchique de la géométrie,
- niveaux de détail,
- représentation hiérarchique de certaines images,
- utilisation d'imposteurs,
- techniques de rendu basées image,
- techniques de compression,
- calcul de relations de visibilité,

Par ailleurs, nous souhaitons permettre l'accès à certains types de données topologiques telles que la position de l'observateur en termes de rue ou de place à l'intérieur de la ville. Pour cela, il est nécessaire de répertorier le maximum d'informations relatives à l'environnement dans lequel l'utilisateur sera plongé au cours de la visualisation.

### 7.2. RNTL PERFRV : Plate-forme française de réalité virtuelle

**Participants :** Bruno Arnaldi, Alain Chauffaut, Thierry Duval, Tangi Meyer, Guillermo Andrade, Christian Le Tenier.

**Mots clés :** *Réalité Virtuelle, Immersion, Retour d'effort, travail coopératif.*

Les objectifs essentiels de la plate-forme PERF-RV concernent d'une part la mutualisation d'un ensemble de configurations matérielles pour la réalité virtuelle (Reality Center, Workbench, systèmes d'interaction haptiques, ...), et d'autre part la factorisation multidisciplinaire de l'étude de problèmes génériques en réalité virtuelle dans des secteurs applicatifs différents et complémentaires (automobile, défense, recherche pétrolière).

Les partenaires de PERF-RV sont : l'INRIA, le CEA, l'Ecole des Mines de Paris, l'ENSAM Chalon sur Saône, le Labri, le Laboratoire de Robotique de Paris, le LIMSI, l'ADEPA, EADS, l'IFP, le Centre Lavallois de Ressources Technologiques, Dassault Aviation, Giat-Industrie, PSA et Renault.

L'enjeu industriel des applications de réalité virtuelle est important. En effet, si actuellement les systèmes de visualisation immersifs de qualité permettant de démontrer des applications en réalité virtuelle restent des environnements haut de gamme, disponibles en petit nombre sur le territoire, la démocratisation annoncée de ces systèmes assure à court terme un déploiement industriel à grande échelle et à bas coût, qu'il est nécessaire de préparer maintenant.

Il est donc apparu nécessaire de créer en France une plate-forme d'expérimentation en réalité virtuelle sur la thématique du " Bureau d'Etude du Futur ", destinée à soutenir un programme scientifique sur la Modélisation et Simulation Numérique, les Interfaces et la Communication Homme-Machine ainsi que sur les Méthodes, Usages et Perceptions. Ces thèmes concourent à définir des méthodes, outils et interfaces communs au différents domaines d'application, et l'objectif de PERF-RV est d'offrir un cadre permettant d'y apporter des réponses au travers de la mise en commun des ressources humaines et expérimentales des différents laboratoires et partenaires industriels. Cette mise en commun concerne les équipements expérimentaux (Reality Center, Workbench, CAVE, dispositifs d'interaction, ...) et l'expérience acquise au travers de développements coordonnés et partageables permettant de capitaliser les ressources temporellement et géographiquement.

De plus, en cours de fonctionnement, cette plate-forme d'expérimentation, unique en France, sera ouverte à des partenaires académiques et industriels rejoignant le consortium initial et sera à même de nous positionner au niveau européen et mondial.

Dans le cadre de cette plate-forme, les activités du projet Siames concernent essentiellement trois points :

coordination de PERF-RV : B. Arnaldi est le responsable du projet PERF-RV. A ce titre, il organise le déroulement du projet :

- conduite et organisation des réunions plénières ou des groupes de travail ;
- synthèse et organisation de la rédaction des « deliverables » pour le Ministère de la Recherche.

Travail multimodal et coopératif : dans cette partie du travail, nous continuons à explorer les métaphores de navigation 3D adaptées à des dispositifs " semi " immersifs (comme les Reality Center et Workbench, par opposition à des dispositifs totalement immersifs comme des " Cave " ou des visiocasques), permettant d'embarquer des outils d'interaction 3D.

Nous avons également porté dans l'environnement logiciel OpenMASK les drivers de périphériques physiques d'acquisition 3D ou plus classiques dont nous avons fait l'acquisition (capteurs magnétiques 6 DoF, gants de données, joystick, souris 3D).

Les résultats des travaux de l'équipe sur les interactions en environnement 3D ont été intégrés dans nos développements liés à PerfRV, c'est ainsi qu'une démonstration permettant des interactions coopératives avec contraintes sur un modèle virtuel de Mégane Scénic, en environnement OpenMASK, a été installé au Technocentre Renault de Guyancourt le 7 novembre dernier.

Immersion et retour haptique : L'état de l'art réalisé présente le panorama complet et détaillé des périphériques de commande à retour d'effort, leur principe, leur architecture, leur classification, leurs caractéristiques. Les interactions haptiques quant à elles, orientent vers un état de l'art de modules logiciels tels que : logiciels de détection de collision et de simulation physique, adaptés pour les calculs en temps réel.

Suite aux différentes réflexions effectuées avec nos partenaires (Renault, CEA) , nous avons défini des scénarios représentatifs et dimensionnants du montage virtuel dans le domaine automobile. Les besoins identifiés concernent :



Figure 18. résultats d'interactions contraintes



Figure 19. Exemple d'interactions coopératives à 2 utilisateurs

- simuler un montage/démontage des pièces mécaniques sans système haptique, mais en utilisant des informations de contact et des modèles physiques des pièces (masses et inerties, glissements, etc.).
- simuler un montage/démontage avec système haptique qui restitue à l'opérateur des forces représentatives de interactions physiques entre l'objet manipulé et l'environnement proche (échelle de forces réduite).
- réaliser une revue de montage en locale ou distante entre différents participants : cas critiques, aide à la formation.

Ces différents aspects nous ont conduit à proposer une architecture logiciel modulaire et distribuée et adaptée aux exigences de la réalité virtuelle intégrant le retour d'effort. Celle-ci s'appuie sur la plus récente version de notre plate-forme logicielle OpenMASK (ex. GASP). Elle permet de créer un environnement commun de communication entre les divers composants logiciels nécessaires (calcul de collision, modeleurs physiques, périphériques, etc.).

Nous avons montré le bien fondé de notre approche, par l'intégration harmonieuse des technologies d'origine et d'action différentes au sens de notre démonstrateur. Ainsi le logiciel *CONTACT-Tolkit* de L'Inria Rocquencourt (projet I3D) communique avec le simulateur mécanique *SMR* du LAAS/CNRS et d l'ancien LRP. Bien que disposant d'une technologie complètement différente du logiciel *DYNAMO* de l'Université technique d'Eindhoven, *SMR* peut être inter-changé avec ce dernier en toute transparence<sup>1</sup>.

Ces caractéristiques assurent à notre architecture la souplesse nécessaire pour s'adapter à des contextes fortement spécialisés des applications industrielles (automobile, aéronautique).

Nos efforts portent aujourd'hui sur l'intégration du bras à retour d'effort *Virtuose 6D* de la société **Haption qui a été commandé par l'irisa (arrivée prévu fin décembre 2002). L'objectif est de définir un protocole robuste de communication basé sur les informations de contraintes mécaniques (thèse de T. Meyer). Ce protocole permettra d'accéder à la multi-coopération haptique en réalité virtuelle immergé.**

### 7.3. RNTL Mouvement : Estimation du mouvement

**Participants :** Bruno Arnaldi, Armel Cretual, Mathilde Vandenberghe, Franck Multon.

**Mots clés :** *Mouvement, Animation par modèles physiques, Acquisition de mouvement, Identification de modèles, Estimation, Fusion des informations capteurs.*

Le projet RNTL Mouvement associe les laboratoires industriels des sociétés Infogrames, Realviz et CEA-LETI, le laboratoire de physiologie et de biomécanique de l'exercice musculaire de l'Université de Rennes II et le projet SIAMES. Son objectif est d'exploiter la complémentarité de différentes sources de mesures et de tenir compte d'une connaissance a priori, afin de s'affranchir de certaines restrictions et limites des systèmes existants pour l'acquisition du mouvement.

Il s'agit avant tout d'un projet de développement logiciel basé sur une approche de prototypage rapide, justifiée par la mise au point itérative de l'interface utilisateur et des modes opératoires. Sur le plan scientifique, le modèle mathématique sur lequel se basent la conception générale du système et sa mise en œuvre algorithmique renvoie à des concepts fondamentaux de la théorie de l'automatique et du signal.

Le projet RNTL Mouvement a pour objectif de fournir un prototype logiciel innovant et performant sur le thème de l'acquisition du mouvement. Le logiciel que nous devons y développer apporte un ensemble de nouveaux services, complémentaires des outils habituels de la chaîne d'acquisition et de traitement du mouvement, et basés sur le principe d'une estimation du mouvement à partir de mesures réelles (observations). L'originalité du système d'estimation de mouvement est de permettre à l'utilisateur de spécifier *a priori* une connaissance sur le mouvement et sur le contexte de la mesure (nature et placement des capteurs). Dans

<sup>1</sup>*CONTACTTolkit* est un détecteur de collision « exact » en 4D et un solveur de contact, *SMR* est un simulateur de mécanismes poly-articulés arborescents et *DYNAMO* est un solveur/intégrateur de contraintes mécaniques non-hiérarchiques

le domaine des jeux vidéo, l'intérêt potentiel est de définir dès le départ un modèle procédural de synthèse du mouvement, et de la paramétrer à partir de mesures réelles. Dans le domaine de l'analyse biomécanique du geste sportif, cette approche autorise par exemple d'évaluer la capacité d'un modèle a priori à expliquer les trajectoires réelles. Le modèle fournit également un service d'assistance à la mesure lorsque l'absence, l'invalidité momentanée ou partielle des mesures justifie l'utilisation de trajectoires synthétisées pour la fusion des informations des capteurs.

La connaissance injectée par l'utilisateur sert de guide et conduit à reformuler le problème de l'acquisition de mouvement en un problème d'estimation des paramètres d'un modèle de mouvement que l'on cherche à identifier : il s'agit de retrouver parmi la classe de mouvement spécifiée, celui offrant la meilleure compatibilité avec les mesures observées. Cette approche permet en principe de palier aux difficultés inhérentes à la reconstruction des trajectoires à partir de la mesure (inversion de la fonction associée au processus de mesure), par un recours à la puissance de calcul de l'ordinateur. Il ne s'agit donc plus d'enregistrer les mesures et de reconstruire après-coup les trajectoires pour pouvoir ensuite remonter au modèle. Au contraire, le choix du modèle est ici antérieur au traitement des trajectoires réelles. Sur le plan de sa décomposition fonctionnelle représentée par la figure 20, ce système repose sur la notion de générateur de mouvement paramétré et sur un processus de simulation des capteurs. Ce dernier calcule un ensemble de mesures virtuelles à partir des trajectoires synthétisées. Un algorithme opérant en mode semi-automatique ajuste les paramètres du modèle de manière à faire coïncider au mieux mesures réelles et mesures virtuelles. L'ajustement terminé, les trajectoires synthétisées par le modèle générateur du mouvement peuvent être exploitées par l'utilisateur. On suppose que dans de nombreux cas, c'est l'obtention d'un modèle paramétré, davantage que les trajectoires qu'il produit, qui intéresse au final l'utilisateur.

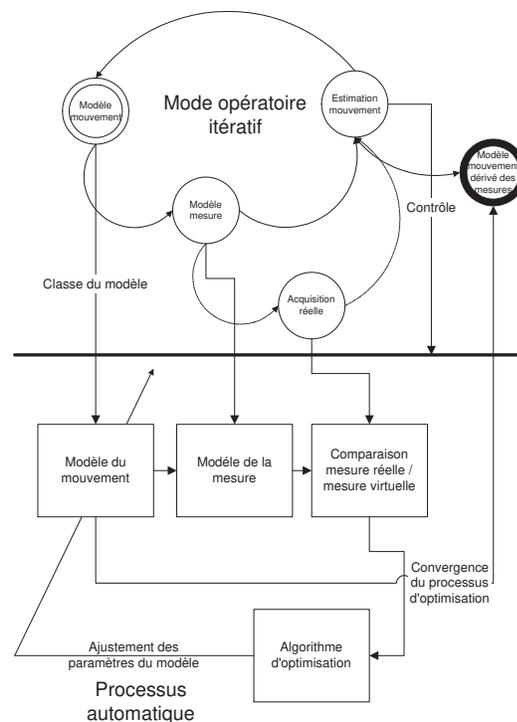


Figure 20. Diagramme représentant, en haut, les étapes successives de la spécification des modèles du mouvement et de la mesure par l'utilisateur. En bas, le schéma fonctionnel du processus d'optimisation comprenant le modèle de mouvement à identifier, la simulation des capteurs à partir des trajectoires à estimer. Ce processus vise à retrouver la « signature » du mouvement dans l'espace de la mesure.

Sur le plan pratique, la validité du concept proposé est essentiellement déterminée par l'aptitude du système à faire coopérer efficacement l'utilisateur avec la machine, de façon à tirer le meilleur parti de sa puissance de calcul. En plus de la conception d'algorithmes adaptés, le succès de cette approche semi-automatique est donc tributaire de modes opératoires et d'une interface utilisateur performants. Pour atteindre ces objectifs, nous avons choisi d'aborder la conception du prototype logiciel par une méthode de développement itératif basée sur une architecture ouverte, sur des outils de développement rapide et sur la génération automatique de code. A ce stade, le prototype visé prend davantage la forme d'une boîte à outils, i.e. d'une *toolkit*, dont l'intégration prototype est réalisée au sein du logiciel d'animation professionnel Maya (Alias-Wavefront, filiale de Silicon Graphics). A terme, l'élaboration d'un environnement de développement d'applications d'acquisition de mouvement, i.e. d'un véritable *framework*, peut être envisagée. L'architecture du prototype en cours de développement repose sur un ensemble de composants distribués (technologies COM), dont l'intégration au sein du logiciel d'accueil est réalisée à l'aide de scripts et de bibliothèques dynamiques basés sur l'API Maya. La figure 21 représente le diagramme de composants du prototype en cours de développement.

Sur le plan théorique, l'approche étudiée renvoie à des concepts fondamentaux du domaine de l'automatique et du traitement du signal : ceux de l'*estimation* et l'*identification*. L'objectif scientifique corollaire de ce projet de développement est d'explorer le potentiel et les limites de ces outils théoriques, en évaluant notamment l'apport des modèles statistiques : ainsi différentes politiques d'estimation sont envisageables, selon que les lois de probabilité a priori des paramètres, voire du bruit de mesure, sont connues ou non (hypothèse Bayésienne ou maximum de vraisemblance), mais aussi selon que l'on recherche un estimateur optimal (meilleur au sens de l'espérance statistique) ou un estimateur robuste (meilleur au sens du pire cas). Plus généralement, l'approche proposée se base sur un processus d'optimisation dont la formulation mathématique et la mise en œuvre algorithmique dépendent des hypothèses disponibles et de la politique d'estimation choisie.

## 7.4. RNTL Dramachina : scénario pour la fiction interactive

**Participants :** Guillaume Bataille, Frantz Degrigny, Stéphane Donikian.

**Mots clés :** *jeux vidéo, fiction interactive, langage de scénario, outil auteur.*

Dans la fiction interactive, le spectateur n'est pas passif comme dans les œuvres littéraires, cinématographiques, théâtrales ou chorégraphiques classiques : il peut influencer directement sur l'évolution de l'histoire. Cela apporte un niveau de complexité supplémentaire pour les scénaristes, mais les moyens dont ceux-ci disposent sont encore assez rudimentaires (essentiellement des outils de traitement de texte et de représentation graphique d'un scénario décomposé en une arborescence de scènes) au regard de l'évolution des techniques. Le programme DraMachina, objet d'un partenariat entre le projet Siames, de l'Irisa et la société Daesign (anciennement Dramaera), contribuera à corriger cette situation. Ce programme soutenu par le RNTL (Réseau national de recherche et d'innovation en technologies logicielles), démarré à la fin de 2000 et devant s'achever en mars 2003, a pour but la mise au point d'un outil permettant aux auteurs de concevoir des environnements narratifs en manipulant directement les éléments clefs du récit (lieux, personnages, rôles, relations entre personnages, actions dramatiques, etc.). Une analyse de la méthodologie d'écriture qu'avait développée Daesign a été effectuée, ce qui a permis de proposer un modèle de représentation plus ambitieux, capable de prendre en compte les divers types de fictions interactives ou de trames dramatiques définis par les théoriciens de la littérature. Sur ces bases, le programme a développé un outil d'écriture qui incorpore un éditeur de texte adapté (avec liens hypertextes), un éditeur de graphes pour la réalisation des dialogues et le traitement logique de l'architecture de la fiction, un module d'extraction de données (actions, lieux, objets, acteurs) à partir du langage naturel, une interface graphique ergonomique, etc. Il sera lui-même intégré dans un " moteur de fiction interactive " développé par la société Daesign. Destiné à l'équipe de production de la fiction, ce moteur permettra, grâce à l'outil DraMachina et à la base de données que celui-ci génère, de réduire considérablement l'étape d'analyse et de synthèse des documents produits par le scénariste.

Débuté au 1er décembre 2000 et d'une durée de 2 ans, DraMachina a pour objectif final la réalisation d'un outil d'édition de fictions interactives devant intégrer une représentation interne de la description, ainsi que la génération d'une base de connaissance équivalente pouvant être utilisée par l'équipe de production de la fiction

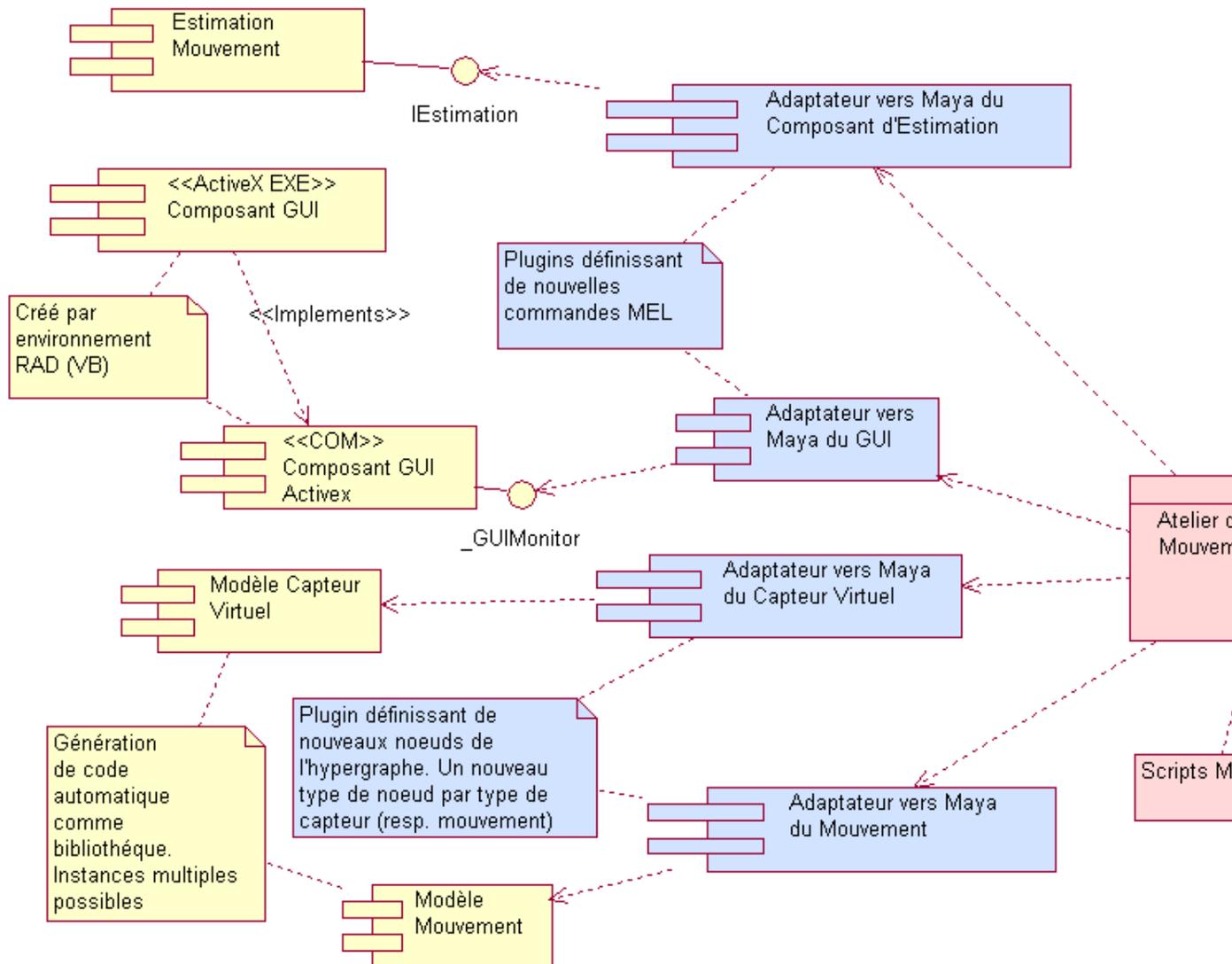


Figure 21. Diagramme de composants du prototype logiciel que nous développons dans le cadre du projet RNTL Mouvement.

interactive. Ce projet est le fruit d'un partenariat entre la société Dramaera et le projet SIAMES de l'IRISA. La société Dramaera (devenue Daesign, début 2002 après un changement d'actionnaire) consacre depuis plusieurs années d'importantes ressources à une recherche approfondie sur le thème de la fiction interactive. Le projet de l'entreprise est de développer les technologies, mais aussi les méthodes d'écriture, permettant de réaliser des " Drames Interactifs " au plein sens du terme, c'est-à-dire qui combinent de fortes qualités narratives et un véritable espace d'interaction et de jeu. Au delà du jeu et de la fiction interactive, l'entreprise souhaite déboucher sur la création d'Environnements Narratifs, c'est-à-dire une structure d'espace et de temps qui réunit les conditions nécessaires pour faire naître un " bouquet d'histoires " :

- des acteurs virtuels autonomes (AVA), doués d'une intelligence comportementale,
- les moyens d'actions fournis à ces personnages pour jouer leur rôle,
- des objets physiques et des situations types qui procurent aux acteurs un champ de jeu,
- une situation initiale qui donne son sens au rôle de chacun des personnages,
- un ou plusieurs personnage(s) semi-autonome(s), qui constitue(nt) l'interface manipulée par l'utilisateur.

Ces composants scénarisés font de l'Environnement Narratif un espace d'interaction riche, souple et dramatiquement fort, dont l'interface est totalement nouvelle et accessible à tout public. Ce travail a été mis en œuvre au cours de la production d'une première fiction interactive sur cédérom, intitulée "The Insider" (interrompue à l'automne 2001). Ils ont élaboré à cette occasion une méthode d'écriture qu'ils commencent à enseigner à des auteurs pour la création de nouveaux scénarios. L'objectif de ce projet est de créer un outil interactif qui permette aux auteurs de concevoir et développer des environnements narratifs en manipulant directement les modèles, objets et outils que notre méthode intègre, le tout via une interface adaptée.

Les produits d'aide à l'écriture existant actuellement sont de deux ordres :

- les outils de "découpage", qui proposent au scénariste de structurer son scénario sous forme de scènes, et/ou d'écrans, de lieux... Ces outils sont des aides précieuses à la représentation graphique d'une arborescence complexe, mais leur apport créatif se limite à ça.
- Les simples éditeurs de scénario (Final Draft, Scenario Pro), qui, même s'ils proposent une saisie du texte dans des masques appropriés (scène, dialogue,...), n'en sont pas moins de simples éditeurs de texte.

Les outils et techniques aujourd'hui utilisés pour scénariser des programmes interactifs, qu'ils soient ludiques ou de formation, n'ont pas suivi l'évolution des techniques, des supports, et des interfaces... L'outil que nous proposons de créer n'a donc pas d'équivalent actuellement.

La méthode d'écriture que nous avons mise au point permet à l'auteur de manipuler des objets logiques (acteurs, rôles, unités dramatiques,...) ayant chacun une fonction bien précise dans le scénario. Ces objets sont bien évidemment étroitement liés les uns aux autres. Nous proposons à l'auteur de manipuler directement ces objets via l'interface, d'en visualiser les liens, de repérer visuellement les conséquences de l'ajout ou du retrait d'un objet. Actuellement, ce travail est effectué sur papier, via une série de documents à produire à l'aide d'outils différents (traitement de texte, tableur, éditeur de graphes) dont nous avons spécifié la forme (base de données, textes, carte des unités dramatiques), mais dont la manipulation reste lourde, et surtout ne permet pas une véritable vue d'ensemble de l'environnement, encore moins une simulation de son fonctionnement. L'outil que nous sommes entrain de réaliser deviendra pour son utilisateur le moyen de créer plus facilement et plus rapidement un environnement narratif, et d'en vérifier la cohérence au fur et à mesure de sa création. Si l'outil est avant tout utile à l'auteur, il sera également un maillon essentiel dans la chaîne de production. En effet, le travail à effectuer actuellement pour passer du scénario à une spécification de logiciel est exclusivement " manuel " : lecture du scénario, analyse des données, puis codage du contenu. Notre outil sera aussi un outil de production automatique d'un modèle équivalent à ce qui a été décrit dans le scénario. Le gain est double : productivité d'une part, respect de la cohérence entre scénario pensé et produit fini d'autre part. L'auteur construit avant tout la protohistoire, c'est-à-dire une continuité narrative qui raconte l'une des histoires

possibles pouvant se dérouler dans l'environnement. A partir de là, il en extrait des éléments particuliers comme les personnages, leur rôle, ou encore les protocènes. Chaque personnage possède un rôle (fonction motrice dans l'histoire, par exemple "le méchant"), un objectif (état de l'histoire dans lequel il considérera sa fonction, comme accomplie), un système de valeurs (des valeurs, des interdits, les règles de transgression de ces interdits). Chaque personnage peut accomplir un certain nombre d'actions dramatiques, en cohérence avec son rôle (le méchant peut tuer). Les personnages sont également décrits dans leur relation avec les autres : ce sont les couples relationnels. On décrit la façon dont peut évoluer la relation entre deux personnages en fonction des actions qui se déroulent. Les briques de base de l'histoire sont appelées unités dramatiques. Elles sont représentées dans une carte, qui permet de repérer les différents chemins que peut prendre le récit, et les éventuels points critiques. Enfin, le drama manager représente l'auteur lors de l'exécution. Il arbitre les différentes demandes d'actions dramatiques, de façon à rester dans un arc dramatique fort et cohérent.

L'application est fortement interactive. L'utilisateur a la possibilité d'ouvrir des vues multiples sur les éléments. Il faut donc assurer la cohérence et la synchronisation des données. Pour cela l'architecture de l'application suit le modèle d'architecture PAC-amodeus. Nous avons développé pour DraMachina un éditeur de texte de type WYSIWYG adapté aux besoins du projet, en particulier l'intégration de liens hypertextes afin de simplifier l'accès aux différentes informations manipulées par l'auteur : personnages, lieux, objets, actions dramatiques, unités dramatiques. Un éditeur de graphes est apparu nécessaire pour la réalisation des dialogues et le traitement logique de la cohérence de l'architecture de la fiction. En complément de l'édition de graphes, l'édition de tableaux est nécessaire pour la saisie des dialogues. En effet, une des possibilités du joueur est de pouvoir décider du choix que fera un personnage en ce qui concerne les réponses à des questions posées, influant ainsi directement sur l'évolution de l'histoire. Le module d'extraction de données à partir du langage naturel permet de trouver automatiquement les actions, localisations et déplacements décrits dans un texte. Notamment, concernant, les actions, nous pouvons extraire différentes informations : qui fait l'action, de quelle manière, dans quel lieu et quels objets ou personnages sont concernés par cette action. Il est ainsi possible de fournir un document de synthèse à l'usage de l'équipe de production, permettant de réduire considérablement la phase fastidieuse d'analyse/synthèse des documents produits par le scénariste.

L'interface graphique est un point important de ce projet, car l'outil s'adresse à des utilisateurs de logiciels de bureautique. Ceci exige donc une ergonomie d'utilisation en adéquation avec les standards du marché. Pour ces raisons et après l'étude des solutions du marché, nous avons choisi d'utiliser Cordial de Synapse pour le traitement du langage naturel, Qt de Trolltech pour l'interface graphique et les différents éditeurs, Rational Rose comme atelier de génie logiciel pour l'étape de conception, XML pour le stockage des données et Visual C++ comme environnement de développement sous Windows.

## 7.5. RNTL Dynamicit  : ville virtuelle peupl e d'entit s autonomes

**Participants :** Olivier Filangi, Claudie Fourn, St phane Donikian.

**Mots cl s :** *Environnements virtuels, maquettes num riques, animation comportementale.*

La soci t  IWI a d velopp  une technologie innovante et unique qui lui permet une quasi-automatisation du processus de mod lisation de villes num riques d'un tr s grand niveau de r alisme,   partir d'un ensemble de donn es g or f renc es. L'objectif du projet est de mettre en commun nos outils de mod lisation et de simulation du comportement humain ainsi que le savoir faire en mod lisation d'IWI afin de pouvoir proposer,   l'issue du projet, des maquettes virtuelles de villes r elles, dans lesquelles l'utilisateur pourra se d placer et interagir avec des acteurs autonomes. Les probl matiques scientifiques abord es sont la mod lisation la plus automatique possible d'environnements virtuels urbains de grande taille, et l'animation d'une quantit  importante d'entit s autonomes au sein de ces maquettes num riques.

La soci t  IWI, notre partenaire dans ce projet RNTL de type pr comp titif qui s'est d roul  de d cembre 2000   juin 2002, a d velopp  une technologie innovante et unique qui lui permet une quasi-automatisation du processus de mod lisation de villes num riques d'un tr s grand niveau de r alisme,   partir d'un ensemble de donn es g or f renc es. L'objectif du projet DynamiCit   tait de mettre en commun nos outils de mod lisation de comportement humain et le savoir-faire en mod lisation d'IWI afin de proposer des maquettes virtuelles

de villes réelles, dans lesquelles l'utilisateur puisse se déplacer et interagir avec des acteurs autonomes. Les problématiques scientifiques abordées sont la modélisation la plus automatique possible d'environnements virtuels urbains de grande taille, et l'animation d'une quantité importante d'entités autonomes au sein de ces maquettes numériques.

Le premier travail de ce projet a été d'étudier en détail les deux approches de gestion de bases de données urbaine utilisées par la société iWi et le projet SIAMES de l'IRISA. Il s'agissait dans les deux cas de maquettes numériques de ville mais avec des objectifs assez différents. Le logiciel FastBuilder développé par la société iWi vise à automatiser le plus possible le processus de génération de la maquette après analyse et fusion des données issues des différentes sources (photo aérienne, système d'information géographique, modèle numérique de terrain, plan cadastral, ...), tandis que le modéleur VUEMS s'intéresse à la création manuelle d'informations topologiques et sémantiques en s'appuyant sur la lecture et la visualisation de deux types de sources (système d'information géographique et plans de circulation numérisés). Les données produites sont aussi de nature différente, car les objectifs diffèrent : pour iWi, il s'agit de permettre la navigation en temps réel au sein de maquettes de villes grande nature alors que pour le projet SIAMES, il s'agit de simuler le comportement d'acteurs autonomes (conducteur de véhicule, piéton) évoluant au sein de ces maquettes. Les avantages de l'approche d'iWi concernent la quasi-automatisation du processus de reconstruction et la fusion multi-sources en entrée de la base de données, tandis que ceux de l'approche IRISA concernent la gestion homogène de plusieurs types d'information (géométrie, topologie et sémantique) et ce pour plusieurs types d'acteurs autonomes (piéton, conducteur de véhicule, conducteur de tramway). D'autre part, l'approche d'iWi se focalisait sur le bâti tandis que celle de l'IRISA sur le réseau routier. L'objectif a donc été de mettre en commun les savoir-faire et ainsi de bénéficier des avantages des deux approches. Un nouveau format de représentation d'une scène urbaine a ainsi été spécifié en commun et réalisé au sein d'une base de donnée relationnelle. Un certain nombre de modules d'importation et d'exportation ont aussi été développés afin d'enrichir la base de donnée et d'extraire des sous-parties de celle-ci. Un des formats d'exportation en XML sert ainsi à alimenter la base de données orientée objet qui fournit les données sur leur environnement aux acteurs autonomes au cours des simulations. L'import XML au sein de la plate-forme de simulation et la construction de la base de connaissances équivalentes ont aussi été réalisés. Deux types d'acteurs autonomes ont été développés dans ce projet : il s'agit du piéton et du conducteur de véhicule individuel. Ces modèles reprennent les résultats des travaux de thèse de Guillaume Moreau et Gwenola Thomas.

D'autre part, le processus de reconstruction quasi-automatique développé par iWi dans sa technologie FastBuilder a été étendu à la génération du réseau routier, tant pour sa représentation géométrique que pour les informations topologiques complexes nécessaires à la simulation d'entités autonomes dans OpenMask. Un module d'édition graphique, avec visualisation 2D ou 3D du modèle de ville, basé sur Interactive City d'iWi, permet la vérification et les corrections de la génération automatique du modèle, tout en laissant le choix de la modification manuelle de tous les paramètres topologiques, sémantiques ou géométriques de la scène.

Dans un principe d'optimisation tant du nombre d'acteurs autonomes que de la vitesse de rafraîchissement lors de la visualisation, nous avons réalisé un nouveau service de la plate-forme de simulation OpenMASK qui permet de rejouer visuellement une simulation avec la possibilité pour l'utilisateur d'interagir à sa guise tant au niveau spatial (navigation dans la maquette 3D) que temporel (avance rapide, pause, retour en arrière). Il s'agit donc d'enregistrer au cours de la simulation l'ensemble des données nécessaires à ce rejeu. Les principes de ce module de rejeu sont à la fois la générique (tout nouvel objet de simulation introduit dans la plate-forme doit pouvoir être rejoué par ce mécanisme) et l'optimisation (seul ce qui est visualisable sera enregistré et rejoué).

Intégrer des figures de l'Editeur de maquettes numériques de villes et d'une simulation.

Nous n'avons pas pu atteindre la barrière des 10 000 personnages animés en temps-réel dans la maquette numérique de la ville. Les points bloquants sont la complexité de la base de données en nombre de polygones et textures, mais surtout le nombre de données dynamiques devant être mise à jour à chaque pas de temps (déplacement des piétons et véhicules). Nous étudions actuellement plusieurs solutions pour dépasser cette limite, notamment de n'envoyer au module de visualisation que les données des objets se trouvant dans le champ de vision.

## 7.6. Giat-Industrie : Environnement virtuel de formation à la maintenance.

**Participants :** Bruno Arnaldi, Stéphane Donikian, Thierry Duval, Frédéric Devillers, Nicolas Mollet.

**Mots clés :** *Réalité Virtuelle, Formation, Scénarisation, Interaction.*

Ce projet intitulé GVT (pour GIAT Virtual Training) est financé en totalité par la société GIAT Industries et a débuté en juin 2002. Il s'agit d'un partenariat entre l'IRISA, l'ENIB et GIAT, qui a pour objectif de développer une application en réalité virtuelle permettant d'aider à la formation à la maintenance de dispositifs industriels. Un prototype doit être développé et livré au bout de dix-huit mois tandis que des aspects complémentaires plus long termes sont étudiés dans le cadre de deux thèses, celle de Nicolas Mollet à l'IRISA, ainsi qu'une autre personne à l'ENIB. Nous sommes actuellement dans la phase d'analyse détaillée du besoin et de spécification du cahier des charges de l'application.

La formation à la maintenance est un domaine en plein essor sur la place de la réalité virtuelle, représentant en 1995 environ 5% du marché des applications de la RV, et plus de 20% aujourd'hui. Le but de ce projet est l'étude des apports de la RV pour les formations à la maintenance spécifiques aux produits Giat-Industrie. Apports évalués en termes de coûts (humains, matériels), et de pédagogie. Un intérêt plus fondamental se dessine dans ce projet, avec l'étude des méthodes de conception de ces environnements de formation à la maintenance en RV.

## 7.7. ODL Plate-forme OpenMASK

**Participants :** Bruno Arnaldi, Alain Chauffaut, Stéphane Donikian, Thierry Duval, David Margery, Mickaël Rouillé.

**Mots clés :** *Middleware, Plate-forme logicielle d'animation, Réalité Virtuelle, Interaction 3D, Exécution distribuée, temps-réel, modularité.*

OPENMASK (Open Modular Animation and Simulation Kit) a pour vocation d'être la plate-forme d'accueil et d'expérimentation des différents travaux de recherche menés au sein du projet dans les domaines de l'animation, de la simulation et de la réalité virtuelle. Ceci nous a conduit à développer une architecture de programmation multicouches et modulaire, avec comme souci principal de réduire autant que possible les contraintes pour le programmeur concernant l'intégration d'un module au sein de la plate-forme. La plate-forme offre avant tout les services de synchronisation et de communication entre modules ayant des fréquences propres différentes (fonctionnant à des fréquences différentes afin de respecter leurs contraintes de simulation).

Le projet SIAMES a répondu et a été retenue par la campagne 2001 d'opérations de développement logiciel (ODL) de l'INRIA, à propos de la diffusion en OpenSource du logiciel GASP. La plate-forme GASP dans sa version 3 a été renommée OpenMASK à la fois pour indiquer dans le nom le fait que le code soit diffusé en open source (<http://www.openmask.org>) et aussi parce que le nom GASP était déjà utilisé par une société américaine. Un dépôt de marque est en cours.

David Margery, actuellement post-doctorant dans l'équipe, est l'auteur de la dernière version du noyau de la plate-forme, cœur de cette distribution open-source. Il travaille actuellement sur une version optimisée du noyau dans le cas d'une exécution distribuée sur un réseau de machines, et fonctionnant notamment sur un cluster de PC.

Dans le cadre de l'ODL, Michael Rouillé a été recruté comme ingénieur associé pour travailler sur :

- module de visualisation immersive et interactive temp-réel ;
- robustification et anglicisation du code ;
- documentation en ligne et hors ligne ;
- développement d'exercices d'apprentissage des concepts de la plate-forme.

## 7.8. RNRT VTHD++

**Participants :** Thierry Duval, Chadi El Zammar.

**Mots clés :** *interactions coopératives en univers virtuels, interactions 3D, coopération en environnements 3D, prise de conscience des latences en interaction.*

Dans le cadre de ce contrat, qui est la suite du contrat VTHD, nous travaillons sur la robustification de notre noyau distribué OpenMASK ainsi que sur la mise en évidence des retards visuels de l'état des objets distants en cours d'interaction. Le but est ici de fournir des outils permettant d'évaluer les capacités offertes par VTHD en termes de bande passante (validation du provisioning dynamique), de temps de latence, et de capacité à se remettre en route suite à une panne (reroutage rapide).

Notre but est ici de nous servir de notre noyau OpenMASK coopératif afin de déployer des applications coopératives multi-sites et multi-utilisateurs, afin de tester les capacités offertes par le réseau VTHD.

Nous avons pour but de mettre en évidence la latence du réseau offrant par exemple à des utilisateurs d'une session OpenMASK coopérative une visualisation de l'écart entre l'endroit où se trouve un objet 3D et l'endroit où d'autres utilisateurs pensent qu'il se trouve effectivement. Ce genre de visualisations doit permettre de mettre en évidence la présence ou non de fluctuations dans les temps de latence du réseau. Ceci devrait également permettre de valider la qualité du service de provisioning dynamique, en montrant la qualité instantanée obtenue.

Nous avons également pour but de permettre à une simulation coopérative de pouvoir continuer à se dérouler même lors de coupures momentanées du réseau, grâce aux modifications apportées à notre noyau OpenMASK coopératif. Le but est ici aussi de faire comprendre aux utilisateurs qu'une coupure s'est produite, et que le travail coopératif reste possible dans certaines limites, en attendant une reprise des conditions de travail normales. Ceci devrait permettre de valider le bon fonctionnement du service de reroutage rapide qui doit prochainement être offert par VTHD.

## 8. Actions régionales, nationales et internationales

### 8.1. Actions nationales

#### Autres actions

- Pôles MESI :

Le projet MESI repose sur un groupement de laboratoires qui ont mis des moyens en commun pour mener un projet de recherche sur l'endoscopie intelligente. Cette activité s'inscrit dans le cadre de l'axe « machines intelligentes » du CNRS. Ces laboratoires associés sont :

- le Laboratoire de Robotique de Paris (LRP, Université de Versailles-Saint Quentin et Paris VI) ;
- l'Irisa ;
- l'Ecole Normale Supérieure de Cachan dans sa composante rennaise.

Les endoscopes actuels utilisés pour l'inspection de réseaux de tubes dans l'industrie ou pour l'inspection du corps humain sont encore bien rudimentaires. L'idée portée par le projet est d'intégrer une stratégie de commande reposant sur l'utilisation de matériaux actifs (alliage à mémoire de forme) en guise d'actionneurs.

L'ENS et l'Irisa travaillent plus particulièrement sur la modélisation d'un tel système en utilisant des techniques de prototypage virtuel. Il s'agit de réaliser le modèle, puis la simulation, d'un endoscope dans un réseau représentant au mieux la réalité.

Une thèse de l'ENS Cachan, encadrée par Georges Dumont, a débuté dans le cadre de cette collaboration en septembre 2000. Il s'agit d'utiliser le prototypage virtuel pour optimiser, en vue de sa réalisation, un prototype d'endoscope. Elle se décline en deux phases :

- La réalisation, dans l'environnement Gasp d'un simulateur d'étude d'endoscopie. Ce simulateur permettra de tester le comportement de l'endoscope dans un environnement géométrique représentant les vaisseaux du corps humain à explorer ;
- L'optimisation, à partir des résultats du simulateur, des prototypes mécaniques d'endoscopes. Cette optimisation servira de base au retour sur conception de l'objet endoscope en vue de sa réalisation effective. Cette partie explore des thèmes nouveaux qui visent à intégrer les résultats du calcul dans la phase de conception. Ils pourront trouver des échos dans les problématiques de la Plate-Forme Française de Réalité Virtuelle (PERF-RV).

Le financement de ce projet est arrivé à échéance cette année. Néanmoins, les actions initiées dans ce cadre se poursuivent et la collaboration est active, comme en témoignent les publications communes en 2002.

- V2NET V2NET est un projet RNRT (Réseau National pour la Recherche en Télécommunication). Les différents partenaires de ce projet sont : THOMSON multimedia, Le LABRI (laboratoire de recherche en informatique dépendant de l'université de Bordeaux 1, de l'ENSERB et du CNRS), Le GROUPE RDC (société d'ingénierie informatique), ARCHIVIDEO (société spécialisée dans la production de simulations 3D fixes et animées pour les métiers de la construction), le laboratoire " Hyperlangages et Dialogues Multimédia " de la Direction des Interactions Humaines de Francetelecom R&D et enfin l'IRISA. L'objectif du projet V2NET est de développer et intégrer de nouveaux outils pour une navigation optimale dans des mondes 3D réels et virtuels dans un contexte d'applications client-serveur. L'optimalité étant ici définie comme la recherche d'un meilleur compromis qualité d'image/fluidité de navigation. En effet, le développement des réseaux et de la technologie 3D vont permettre l'émergence de nouveaux services de consultation en ligne, qui permettront d'accéder à de grandes bases de données contenant des scènes réelles/virtuelles. Or, les outils existants ne sont pas adaptés à ces futurs services. Il faut développer les outils algorithmiques appropriés en analysant l'intégralité des processus mis en jeu dans une application client-serveur. Ceci couvre les aspects de modélisation et de représentation compressée des objets, de structuration de scène, de dialogue client-serveur et de visualisation des scènes. L'intérêt de ce projet est donc de combiner les connaissances scientifiques et les savoir-faire des partenaires pour traiter ces problèmes dans un contexte de services de télécommunications.

## 8.2. Actions financées par la Commission Européenne

### IST : OPENISE

OPENISE est un projet européen IST réunissant différents partenaires : français, allemands, italiens, grecs et israéliens. Ce projet a pour objectif de définir, développer, intégrer et expérimenter une plate-forme ouverte et échelonnée pour fournir des services interactives de qualité à travers un réseau. Les différents objectifs visés sont :

- plate-forme d'application : en évaluant et recommandant de nouvelles technologies multimédia émergentes (MPEG4, VRML et ses extensions) le projet développe un ensemble de technologies clés essentielles pour supporter de nouvelles applications interactives ; deux exemples d'application seront développées, couvrant deux scénarios de service : les jeux et l'héritage culturel.
- plate-forme de contenu : le projet étudie, définit et intègre une plate-forme de haute performance modulaire et ouverte pour le codage et la transmission (à travers un réseau) de contenu tout en prenant en compte les exigences de QoS (Qualité de Service) dans des environnements de réseau IP et ATM.
- plate-forme de réseau : pour supporter efficacement les différentes applications prévues, le projet spécifie et intègre une plate-forme de diffusion de données (contenu) avec des possibilités de QoS, Multicast et accès rapide (ADSL, ADSL Lite, Radio large bande).

- expérimentations : les composantes contenu, application et réseau seront intégrées dans une plate-forme de laboratoire pour réaliser des expérimentations visant à évaluer les performances de la plate-forme OPENISE compte tenu des objectifs du projet.

## 9. Diffusion des résultats

### 9.1. Animation de la communauté scientifique

- Membre du CCSTIC (Comité de Coordination STIC) : B. Arnaldi.
- Membre du jury de thèse SPECIF : B. Arnaldi.
- Membre du Comité d'Orientation du RNTL : B. Arnaldi.
- Participation aux travaux de l'association française d'informatique graphique AFIG - trésorier de l'association : S. Donikian.
- Membre du conseil scientifique du GDR ALP (Stéphane Donikian).
- Co-animation du groupe de travail Animation-Simulation de l'AFIG et du GDR ALP : S. Donikian.
- Membre du comité scientifique du Programme Interdisciplinaire de Recherche du CNRS, ROBEA (Robotique et Entités Artificielles) : S. Donikian.
- Thierry Duval a animé un workshop sur les architectures logicielles pour les applications interactives 3D lors de la conférence IHM-HCI'2001, à Lille, le 11 septembre 2001.
- Participation aux travaux du réseau thématique pluridisciplinaire microrobotique (RTP numéro 44 du département STIC du CNRS). Nous avons organisé les premières journées du RTP les 6 et 7 novembre 2002 : G. Dumont.
- Participation au RTP numero 7 du département STIC : Réalité virtuelle, synthèse et visualisation (B. Arnaldi).
- Participation à l'AS du département STIC : Détection de collision et calcul de réponses (G Dumont).
- Participation au RTP 15 du département STIC : Interfaces médiatisées et réalité virtuelle (Anatole Lecuyer)
- Participation au AS 30 et AS 86 du département STIC : Réalité virtuelle et cognition (B. Arnaldi, S. Donikian, T. Duval)
- Participation au AS 44 du département STIC : Algorithme pour la simulation et la visualisation de phénomènes physiques et naturels en réalité virtuelle (K. Bouatouch).

### 9.2. Enseignement universitaire

- responsabilité de la filière *Image* du DEA d'informatique (K. Bouatouch).
- DEA d'informatique Ifsic : synthèse d'image (K. Bouatouch).
- DEA d'informatique Ifsic : Mouvement temps réel (B. Arnaldi).
- DESS MITIC (IFSIC), cours sur la réalité virtuelle et l'interaction (Thierry Duval et Stéphane Donikian).
- DEA Calais, cours sur l'animation (Stéphane Donikian).
- Ecole Supérieure des Beaux-Arts de Paris, cours sur la fiction interactive (Stéphane Donikian).
- DIIC LSI & INC, DESS ISA Ifsic : cours sur les interfaces homme-machine et la conception d'applications interactives (T. Duval).
- DEA d'informatique : participation au cours sur le raisonnement spatial et temporel (S. Donikian).
- DIIC-INC : Ifsic, Rennes 1, (K. Bouatouch) option image.
- Responsabilité du module Réalité Virtuelle et IHM du dess Mitic (T. Duval)
- DIIC LSI & INC, DESS ISA, DESS MITIC Ifsic : cours sur les interfaces homme-machine et la conception d'applications interactives (T. Duval).

- DESS CCI Ifsic : option informatique graphique (K. Bouatouch, T. Duval).
- MAÎTRISE D'INFORMATIQUE Ifsic : partie synthèse d'image de l'option images numériques (T. Duval).
- Mastère Réalité Virtuelle Distribuée de l'ENI de Brest : cours sur la coopération en univers 3D et sur la multimodalité (T. Duval).
- Module Interfaces de l'ENI de Brest : cours sur la conception, les modèles d'architecture et l'ergonomie des interfaces homme-machine (T. Duval).
- DIIC LSI, DESS MITIC, DESS ISA option Génie Logiciel, Ifsic : cours sur les interfaces homme-machine (lien avec les Design Patterns) (T. Duval).
- DEA RESIN ENS Cachan, Paris 6 : Prototypage virtuel (G. Dumont).
- DESS ITIHMRV Laval, université d'Angers : Modèles physiques en réalité virtuelle (G. Dumont).
- Ecole d'été CEA/INRIA/EDF sur la réalité virtuelle (B. Arnaldi, T. Duval et A. Chauffaut)

## 10. Bibliographie

### Bibliographie de référence

- [1] B. ARNALDI. *Animation de systèmes physiques*. mémoire d'habilitation à diriger les recherches, université de Rennes I, février, 1994.
- [2] B. ARNALDI, R. COZOT, G. DUMONT. *A unified Model for Physically based Animation and Simulation*. in « Applied Modelization, Simulation and Optimization », Cancun, Mexique, juin, 1995.
- [3] B. ARNALDI, T. PRIOL, L. RENAMBOT, X. PUEYO. *Visibility Masks for Solving Complex Radiosity Computations on Multiprocessors*. in « Parallel Computing (Elsevier) », numéro 7, volume 23, juillet, 1997, pages 887-897.
- [4] D. BADOUEL, K. BOUATOUCH, T. PRIOL. *Distributed data and control for ray tracing in parallel*. in « IEEE Computer Graphics and Applications », numéro 4, volume 14, juillet, 1994, pages 69-77.
- [5] K. BOUATOUCH, P. GUITTON, B. PÉROCHE, F. SILLION. *Simulation de la lumière en synthèse d'images : aspects algorithmiques*. in « TSI », numéro 10, volume 14, novembre, 1996.
- [6] S. DONIKIAN. *Les modèles comportementaux pour la génération du mouvement d'objets dans une scène*. in « Revue Internationale de CFAO et d'Infographie, numéro spécial AFIG-GROPLAN », numéro 6, volume 9, 1994.
- [7] G. HÉGRON, B. ARNALDI, C. LECERF. *Computer Animation*. Prentice Hall, juillet, 1995, chapitre Dynamic Simulation and Animation.
- [8] J. NOUGARET, B. ARNALDI. *Spécification cinématique et simulation dynamique : quelle combinaison optimale pour l'animation ?*. in « Numéro Spécial « Journées d'informatique graphique », Revue Internationale de CFAO et d'Informatique Graphique », numéro 4, volume 10, octobre, 1995.

### Thèses et habilitations à diriger des recherche

- [9] N. COURTY. *Animation référencée vision : de la tâche au comportement*. thèse de doctorat, université de Rennes I, novembre, 2002.

## Articles et chapitres de livre

- [10] K. BOUATOUCH. *Practical Parallel rendering, Chapitre de livre*. 2002.
- [11] F. DEVILLERS, S. DONIKIAN, F. LAMARCHE, J. TAILLE. *A programming environment for behavioural animation*. in « Journal of Visualization and Computer Animation », 2002.
- [12] G. ROUGERON, F. GAUDAIRE, Y. GABILLET, K. BOUATOUCH. *Simulation of the indoor propagation of a 60 GHz electromagnetic wave with a time-dependent radiosity algorithm*. in « Computer And Graphics », numéro 1, volume 26, 2002, pages 125-141.

## Communications à des congrès, colloques, etc.

- [13] P. BIDAUD, F. CHAPELLE, G. DUMONT. *Evolutionary Optimization Of Mechanical And Control Design : Application to Active Endoscopes*. in « Proceedings of RO.MAN.SY 2002 », éditeurs IFTOMM., Udine, Italy, 1-4 july, 2002.
- [14] B. BIDEAU, L. FRADET, F. MULTON, S. MENARDAIS, R. KULPA, B. ARNALDI. *Validation d'un processus de traitement allant de la capture du mouvement à l'immersion de sujets en réalité virtuelle : application au tir au handball*. in « Quinzièmes journées de l'Association Francaise d'Informatique Graphique », AFIG, Lyon, France, Décembre, 2002.
- [15] G. DUMONT, C. KÜHL, G. ANDRADE. *A Dynamical Simulator For Designing Active Endoscopes*. in « Proceedings of the Fifth World Congress on Computational Mechanics (WCCMB V) », Vienna University of Technology, Austria, ISBNB 3-9501554-0-6,, éditeurs H. MANG, F. RAMMERSTORFER, J. EBERHARD-STEINER., Vienna, Austria, July, 2002, <http://wccm.tuwien.ac.at>.
- [16] G. DUMONT, C. KÜHL, P. BIDAUD. *Simulating And Optimizing Active Endoscope Prototypes*. in « Proceedings of ISR2002 (International Symposium on Robotics) », IFR (International Federation of Robotics), Stockholm, Sweden, october, 2002, 7-11 october 2002.
- [17] C. KÜHL, G. DUMONT. *Réalité virtuelle : de la simulation à l'optimisation d'un endoscope actif*. in « Journées des Jeunes Chercheurs en Robotique », jjcr15-AJCR, LSIIT Strasbourg, janvier, 2002, 31 january and 1 february 2002.
- [18] C. KÜHL, G. DUMONT. *Virtual Endoscopy : From Simulation To Optimization Of An Active Endoscope*. in « Proceedings of Modélisation et Simulation pour la Médecine et la Chirurgie Assistée par Ordinateur », INRIA, 12-15 November, 2002.
- [19] C. KÜHL, G. DUMONT, P. MOGNOL, S. GOULEAU, B. FURET. *Active Catheter Prototyping : From Virtual To Real*. in « Proceedings of 4th International Conference On Integrated Design And Manufacturing In Mechanical Engineering », Primeca, IFMA, Clermont-Ferrand, France, 2002, 14-16 may 2002.
- [20] F. LAMARCHE, S. DONIKIAN. *Automatic Orchestration of Behaviours through the management of Resources and Priority Levels*. in « Autonomous Agents & Multi-Agent Systems (AAMAS 2002) », ACM, éditeurs W. L. JOHNSON, C. CASTELFRANCHI., Bologna, Italy, juillet, 2002.

- 
- [21] D. MARGERY, B. ARNALDI, A. CHAUFFAUT, S. DONIKIAN, T. DUVAL.. *OpenMASK : Multi-Threaded or Modular Animation and Simulation Kernel or Kit : a General Introduction*. in « VRIC 2002 », éditeurs S. RICHIR, P. RICHARD, B. TARAVEL., Laval, France, juin, 2002.
- [22] D. MENEVEAUX, K. BOAUTOUCH, G. SUBRENAT, P. BLASI. *Efficient clustering and visibility calculation for global illumination*. in « AFRIGRAPH », February, 2003.
- [23] T. MEYER, G. ANDRADE-BARROSO, B. ARNALDI. *Une architecture pour le Retour d'Efforts*. in « Quinzièmes journées de l'Association Française d'Informatique Graphique », AFIG, Lyon, France, Décembre, 2002.
- [24] T. DUVAL, A. FENALS. *Faciliter la perception de l'interaction lors de manipulations coopératives simultanées en environnements virtuels 3D*. in « communication informelle à IHM 2002 », Poitiers, France, 2002.
- [25] T. DUVAL, C. LE TENIER. *Interactions 3D coopératives en environnements virtuels avec OpenMASK pour l'exploitation d'objets techniques*. in « Proceedings of Virtual Concept 2002 », pages 116-121, Biarritz, France, 2002.