Projet : siames

Synthèse d'Images, Animation, Modélisation et Simulation
Thème 3B
Rennes

15 novembre 2000

Table des matières

1 Composition de l'équipe

debug : Module de nom « en-tete » pour le projet Siames, section composition

Responsable scientifique
Bruno Arnaldi [Professeur Insa de Rennes]

Assistante de projet
Evelyne Livache [SAN Inria]

Personnel Inria
Guy André [CR]
Alain Chauffaut [IR]
Jean-Luc Nougaret [CR en détachement chez Sony Tokyo]
Thierry Duval [CR, détachement INRIA]

Personnel CNRS
Stéphane Donikian [CR]
Annick Leroy [IE]

Université de Rennes 1
Kadi Bouatouch [Professeur]
Rémi Cozot [Maître de conférences]
Éric Maisel [Maître de conférences]
Gwenola Thomas [ATER jusqu'au le 1er septembre 2000]

Ingénieur Expert
Richard Kulpa [Inria]

Chercheurs doctorants
Erwan Guillou [Bourse MENRT]
Nicolas Pazat [Bourse AMN]
David Margery [Bourse MENRT]
Frédéric Devillers [Bourse MENRT]
Stéphane Menardais [Bourse MENRT]
Mickael Pouliquen [Bourse MENRT]
Tangi Meyer [Bourse INRIA-Région]
Nicolas Courty [Bourse CNET]
Fabrice Lamarche [Bourse MENRT depuis le 1er octobre 2000]
Romain Thomas [Bourse INRIA depuis le 1er octobre 2000]

Post-doctorants

Jordi Regincos [Post-doc INRIA jusqu'en février 2000]
Zhongke Wu [Post-doc INRIA jusqu'en février 2000]
Guillermo Andrade [Post-doc CNRS depuis le 1er novembre 2000]

Poste d'accueil jeune

Jean-Michel Zelmeur [depuis le 15 octobre 2000]

Collaborateur Extérieur

Georges Dumont [Maître de conférence ENS Cachan]
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

debug : Module de nom « presentation » pour le projet Siames, section presentation
debug : corps du module.

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'éclairement 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 **plateforme logicielle de simulation** capable d'intégrer les différents composants de nos travaux dans un contexte de simulation temps réel distribuée. Cette plateforme 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

debug : Module de nom « fondements » pour le projet Siames, section fondements
debug : corps du module.

3.1 Panorama

debug : Module de nom « fondements-Panorama » pour le projet Siames, section fondements

Résumé : *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.*

debug : corps du module.

3.2 Simulation d'éclairage et synthèse d'image

debug : Module de nom « fondements-eclair » pour le projet Siames, section fondements

Mots clés : algorithmes parallèles, simulation d'eclairage, synthèse d'image, radiosité.

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

Photométrie : mesure des grandeurs physiques liees au rayonnement perçu par un observateur humain moyen caracterisé 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.

Résumé : *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.

debug : corps du module.

Simulation d'éclairage

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.

3.3 Modèles dynamiques de mouvement

debug : Module de nom « fondements-moddyn » pour le projet Siames, section fondements

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.

Résumé : *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 neuro-musculaire, planification et boucles d'asservissement, etc.).

debug : corps du module.

La création de mouvements d'objets ou de personnages synthétiques nécessite la mise en oeuvre 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 :

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

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 calibration 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 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 le 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

debug : Module de nom « fondements-comport » pour le projet Siames, section fondements

Mots clés : modélisation topologique, modélisation sémantique, systèmes multi-agents, animation comportementale, simulation évènementielle, automates, statecharts, réseaux de Pétri, objets réactifs, objets temps réels, scenario 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.

Résumé : *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.

debug : corps du module.

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 (mémorisation, prédiction, durée d'une action, séquencement 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, 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 modeleur 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 influer 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'applications

debug : Module de nom « domaine » pour le projet Siames, section domaine
debug : corps du module.

4.1 Panorama

debug : Module de nom « domaine-Panorama » pour le projet Siames, section domaine

Résumé : *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 Les environnements architecturaux et urbains

debug : Module de nom « domaine-urbain » pour le projet Siames, section domaine

Mots clés : simulation d'éclairage, environnements urbains, architecture, éclairage publique, éclairage naturel, éclairage artificiel.

Résumé : *Les travaux sur la simulation d'éclairage peuvent être étendus à des spectres de longueurs d'onde non visibles tels que l'infrarouge. De plus, nous avons exploité le résultat de nos travaux pour : (I) réaliser l'éclairage dans des environnements architecturaux, (II) éclairer et dimensionner les tunnels dans le but de permettre de rendre visible tout obstacle sur la chaussée, (III) modéliser à la fois des*

environnements architecturaux et urbains, (IV) simuler la conduite de véhicules en fonction des critères évalués par le logiciel de simulation, (V) simuler la propagation de rayonnement infrarouge. Nos travaux peuvent déboucher sur d'autres types d'applications liés à d'autres types de rayonnements : téléphone microcellulaire, réseau informatique sans fil, ...

debug : corps du module.

La principale application de nos travaux concerne la simulation d'éclairage et l'évaluation de critères de confort visuel et de visibilité. Un des logiciels que nous avons développés avec le CSTB (Centre Scientifique et Technique du Bâtiment) a permis de simuler l'éclairage pour différents types de scènes réelles, certaines ont été validées par des mesures expérimentales. Les exemples suivants représentent les dernières études réalisées par le CSTB en utilisant notre outil de simulation :

- Stade de France ;
- Tunnel de Tokyo ;
- Locaux de la Maif ;
- Stade de la Beaujoire ;
- Le Louvre.

En fait, le champ d'application de ces techniques de simulation d'éclairage est vaste : éclairage public et d'édifices historiques, confort visuel (bureau, atelier), tunnel, grandes surfaces, studio de cinéma. Elle peuvent jouer aussi un rôle important dans la vérification des normes d'éclairage naturel et artificiel, ou dans la simulation de luminaires avant fabrication.

Nous avons aussi appliqué notre logiciel d'éclairage et de dimensionnement des tunnels. Selon le type de tunnel, ce logiciel propose, en fonction de l'éclairage naturel extérieur, différentes solutions d'éclairage : type de sources, nombre de rangées de sources, réflectivités de la chaussée, murs et plafond.

4.3 La réalité virtuelle, la réalité augmentée et la téléopération

debug : Module de nom « domaine-RV » pour le projet Siames, section domaine

Mots clés : réalité virtuelle, réalité augmentée, téléopération.

Glossaire :

Réalité virtuelledésigne 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.

Résumé : *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 urbains virtuels ;*
- *contrôle du mouvement en temps réel de systèmes dynamiques.*

4.4 Les simulateurs pour la recherche dans le domaine des transports

debug : Module de nom « domaine-transport » pour le projet Siames, section domaine

Mots clés : simulation, transport, sécurité, traffic.

Résumé :

Le simulateur reproduisant finement le comportement du couple véhicule et conducteur en interaction avec son environnement est, à la fois, un outil d'entraînement à la conduite de véhicules et un outil d'étude et de recherche permettant une meilleure connaissance du système conducteur-véhicule-environnement. Ces dernières années ont vu une évolution très rapide des simulateurs haut de gamme dédiés à la recherche. La mobilisation sur le thème de la sécurité routière et les grands programmes de recherche sur les véhicules et l'assistance à la conduite n'ont fait qu'accentuer le développement de tels moyens d'essais. Ces simulateurs sont centrés sur un véhicule réel instrumenté dans lequel un conducteur va permettre de fermer la boucle de simulation en réagissant à chaque instant aux résultats du calcul qui lui seront rendus perceptibles. Pour ce faire, ces simulateurs sont composés de plusieurs sous-systèmes : dynamique du véhicule, retour d'efforts aux commandes, restitution visuelle, sonore et du mouvement, modèle d'environnement et de trafic. De tous ces sous-systèmes, le moins réaliste à l'heure actuelle est le modèle d'environnement et de trafic. Notre objectif est de pouvoir effectuer des simulations de conduite dans des environnements urbains réels et dotés d'un trafic dense et multimodal (poids-lourds, voitures, deux-roues, piétons). Les différents constituants de telles simulations sont le modèle mécanique des moyens de locomotion, le modèle de comportement des conducteurs et le modèle du réseau routier.

debug : corps du module.

À travers une longue expérience dans le domaine de la simulation de systèmes physiques et en particulier sur les simulations de véhicules (projet de simulateur avec l'Inrets, projet Praxitèle, projet européen DIATS), le projet SIAMES aborde le problème de la simulation dans le domaine du transport sous l'angle d'une conception intégrée de composants permettant d'offrir des possibilités aussi variées que :

- un modèle physique des modes de transport ;
- la définition d'un trafic réaliste (par spécification de comportement de pilotes virtuels) ;
- la simulation configurable en fonction du nombre d'entités ;
- la simulation temps réel interactive ou hors ligne ;
- un environnement 3D réaliste ;
- etc ...

Modèle mécanique des moyens de locomotion

La précision et le réalisme dans le comportement dynamique des entités simulées, comme des véhicules ne peuvent être obtenus que par la prise en compte d'un modèle mécanique plus ou moins détaillé. À travers la spécification d'un tel modèle (cf DREAM : **??**), s'intégrant dans notre plate-forme de simulation (cf GASP : 5.5), nous pouvons prendre en compte et spécialiser selon l'application les différents modèles dynamiques des entités évoluant dans la scène.

Modèle de comportement des conducteurs

En complément au modèle dynamique du mode de transport, il est nécessaire de pouvoir définir un comportement de pilote virtuel. En effet, le réalisme de la simulation dépendra pour beaucoup de la fidélité et de l'efficacité de ce pilote. Nous avons utilisé le modèle formel STPH (Système de Transition Parallèle Hiérarchisé) pour décrire le comportement d'un conducteur de véhicule.

Modélisation du réseau de transport

La définition de l'environnement dans lequel se déroule la simulation est aussi prépondérante quant à la qualité du simulateur. Dans le cas de la simulation de conduite, le comportement du pilote dépend de plusieurs types d'informations : les caractéristiques physiques de la route, la topologie du réseau routier, la signalisation routière et, si nous nous plaçons dans un cadre urbain, les éléments occultants. Toutes ces informations sont en fait de trois natures différentes mais complémentaires : géométriques, topologiques et sémantiques. Nous avons cherché à regrouper ces trois types de données à l'intérieur d'un modèle unique, avec comme objectif la reconstruction de toute l'infrastructure de réseaux de circulation urbains réalistes, à partir de données réelles fournies par la ville de Rennes.

4.5 Humanoïdes virtuels et modèles biomécaniques

debug : Module de nom « domaine-humano » pour le projet Siames, section domaine

Mots clés : humanoïde de synthèse.

Résumé : *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.

debug : corps du module.

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 expérimentions 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 inexistants). 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.

4.6 Le multimédia, l'audiovisuel et les jeux

debug : Module de nom « domaine-jeu » pour le projet Siames, section domaine

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.

Résumé : *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.*

debug : corps du module.

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. Dans ce domaine, nous entretenons un partenariat avec la société canadienne Alias-Wavefront ainsi qu'avec le CNET de Rennes.

5 Logiciels

debug : Module de nom « logiciels » pour le projet Siames, section logiciels
debug : corps du module.

5.1 Panorama

debug : Module de nom « logiciels-Panorama » pour le projet Siames, section logiciels

Résumé : *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

debug : Module de nom « logiciels-eclair » pour le projet Siames, section logiciels

Participants : Kadi Bouatouch [correspondant].

Mots clés : logiciel, simulation d'éclairage.

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

Résumé : *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.*

debug : corps du module.

- *GIN*: *GIN* est un logiciel de simulation d'éclairage développé en collaboration avec le CSTB. L'approche utilisée est spectrale, les longueurs d'ondes des échantillons sont déterminées automatiquement en fonction de la forme des spectres des sources de lumière et de celle des réflectances des matériaux. Le logiciel prend en compte l'éclairage naturel (utilisant des modèles de ciel et de soleil) et artificiel (utilisation des catalogues de luminaires). L'effet de la modification des propriétés de réflectance des matériaux sur l'éclairage est quasi instantané. La modification du type de source implique peu de calculs. Le logiciel s'appuie principalement sur la méthode de radiosité et utilise dans une deuxième passe le lancer de rayon pour les effets de réflexion spéculaire et de transparence.
- *SIM*: *SIM* permet d'effectuer une simulation d'éclairage dans des environnements complexes sur une station de travail courante. L'approche mise en œuvre est basée sur le découpage de l'environnement en plusieurs parties que nous appelons *cellules*. Chaque cellule ayant une liste d'ouvertures (portes ou fenêtres) à travers lesquelles elle voit d'autres cellules. Un calcul de visibilité détermine pour chaque cellule la liste de ses cellules visibles à travers des ouvertures. Ces opérations constituent la *structuration* de l'environnement. Le résultat de cette structuration est une liste de cellules et un graphe de visibilité.

  A partir des résultats de cette structuration, la simulation d'éclairage dans des environnements complexes est rendue possible. En effet, une surface appartenant à une cellule émet son énergie vers les surfaces de cette cellule mais aussi vers les surfaces des cellules qui lui sont visibles. Ce fait est exploité par *SIM* pour ne garder en mémoire que les cellules éclairées par la cellule émettrice courante, le reste des cellules étant stocké sur disque. Cela permet de réduire à tout instant l'espace mémoire utilisé et de rendre possible la simulation dans des environnements complexes.
- *SAS*: *SAS* est un logiciel interactif permettant de structurer un environnement complexe en cellules à des fins, soit de simulation d'éclairage, soit de déplacement virtuel en temps réel. Ce logiciel s'appuie sur les librairies *MOTIF* et *OpenGL*.
- *PAUS*: contrairement à *SAS*, ce logiciel est capable d'effectuer une structuration en cellues de scènes archtecturales 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 :
  1. 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. Ceci est d'un intérêt crucial pour le concepteur de tunnel qui pourra ainsi visualiser le tunnel avant sa construction effective, obtenir certains renseignements très utiles comme les niveaux de luminance ou une évaluation de l'éblouissement en certains points de la scène. Il disposera alors d'un outil permettant de détecter d'éventuels défauts et les qualités que pourrait présenter ce tunnel après construction.
  2. 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.

3. 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.

– *ModPh*: *ModPh* est un logiciel interactif de modélisation de sites réels à partir d'images. Son principe repose sur les méthodes décrites plus loin. Il permet à un utilisateur de reconstruire des sites réels avec une précision suffisante afin de pouvoir les explorer ("walk-through"). La reconstruction, interactive, est controlée aussi bien au moyen de plusieurs photographies du site que par une maquette virtuelle qui évolue au cours de la modélisation. Il est ainsi facile d'ajouter des objets virtuels dans ce site ainsi que de modéliser certains objets directement sur cette maquette lorsqu'on ne dispose pas de suffisamment d'informations dans les images. Le logiciel peut sauvegarder les modèles reconstruits dans différents formats (VRML, Iv, Pov, NFF, ...). Enfin, une grande partie de ce logiciel a été développée en Matlab, ce qui augmente ses capacités d'évolution.

5.3 Modeleur d'environnement urbain : VUEMS

debug : Module de nom « logiciels-vuems » pour le projet Siames, section logiciels

Participants : Stéphane Donikian [correspondant].

Mots clés : logiciel, environnements virtuels, géométrie, topologie, sémantique, transport.

Résumé : VUEMS *(Virtual Urban Environment Modeling System) est un modeleur d'environnements urbains qui a pour vocation de représenter le plus fidèlement possible l'espace urbain pour des applications liées à la simulation du comportement humain (conduite automobile, locomotion...). Il s'agit donc de représenter la voirie, la signalisation routière, mais aussi tous les éléments de l'espace urbain ayant un impact sur le comportement (bâtiments, éclairage public, arbres, ...).*

debug : corps du module.

La définition des comportements consiste à définir le raisonnement et les actions des acteurs en fonction de leur environnement et de la perception qu'ils en ont. L'acteur voit ce qui se passe autour de lui, décide de son action en fonction de sa perception et de ses connaissances, agit sur lui-même (en apprenant une nouvelle notion) et sur son environnement (en ouvrant une porte). Il est donc en interaction constante avec son environnement par le biais de capteurs (perception) et d'effecteurs agissant directement sur l'environnement. La simulation d'humanoïdes autonomes dotés de comportements sophistiqués passe par la prise en compte de l'environnement. Le comportement d'une entité dépend de son espace d'évolution et de son action : suivre un trottoir, traverser une route. 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. Nous avons proposé d'informer l'environnement urbain en caractérisant les espaces d'évolution, en prenant en compte leur configuration et les objets qu'ils contiennent. L'environnement contient alors les informations nécessaires aux prises de décision de l'acteur.

Un modeleur d'environnements urbains appelé VUEMS (Virtual Urban Environment Modelling System) a été développé dans le but de générer des bases de données complètes très réalistes, car construites à partir de données cartographiques (SIG ou plans détaillés numérisés) [DON97a, DON97b]. Une partie des informations est extraite automatiquement de la base de données, tandis que l'utilisateur du modeleur construit le réseau routier à l'aide des tronçons élémentaires, permettant ainsi de construire une base de données réaliste, intégrant les trois types d'informations (géométriques, topologiques et sémantiques). Deux types de données sont produites en sortie du modeleur : une base de données tactique utilisée par les entités comportementales et une base de données géométriques texturées aux formats Open Inventor ou VRML 2.0, utilisée pour la visualisation. Quatre familles d'objets sont gérées :

- Le réseau routier : une route est modélisée sous VUEMS par un squelette et son codage associé. Le squelette est dessiné grâce à une courbe spline, le codage décrit les voies (largeur et nature). La construction du réseau routier est automatique ; quand deux routes se croisent, le réseau routier est automatiquement mis à jour et la géométrie de la zone d'intersection est automatiquement calculée. Treize composants du réseau routier ont été répertoriés : croisements en X, en T, en Y, élargissement, etc.
- La signalisation et le mobilier urbain : la signalisation est associée aux routes. L'utilisateur crée un marqueur sur le squelette et lui associe un codage. Ce codage contient la nature de la signalisation, son emplacement (côté gauche ou droit de la route) et son orientation. Le mobilier urbain (lampadaires, boîtes aux lettres) est simplement positionné dans l'environnement. Des modèles géométriques préexistant sont utilisés pour la visualisation de ces objets (lampadaires, arbres, abribus, etc.). Dans la base d'informations, le décor et la signalisation sont associés à leurs espaces d'appartenance des informations symboliques (nom, fonction, ...) les caractérisent.
- Les bâtiments, les places et les parcs : ce sont des polygones au sol auxquels sont associés des attributs. Par exemple, une hauteur, une apparence, une fonction et un nom sont associés à un bâtiment. Modélisés en deux dimensions, les bâtiments sont extrudés et automatiquement texturés au moment de la génération de la scène en trois dimensions.
- Les objets linéaires (barrières, murs, haies, portes) : ils sont positionnés sur le contour des espaces et caractérisent ainsi les frontières de ces espaces : non franchissables, manipulables, franchissables. Au moment de la génération de la base d'informations, le graphe topologique est créé en connectant et en définissant les parties franchissables et manipulables des frontières des bâtiments, des espaces libres (parcs et places) et des espaces du réseau routier (voies, intersections, ...).

5.4 Modeleur de comportements : HPTS

debug : Module de nom « logiciels-hpts » pour le projet Siames, section logiciels

Participants : Stéphane Donikian [correspondant].

Mots clés : langage, comportement, logiciel, génération de code.

Résumé : *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.*

debug : corps du module.

Un modèle d'entité comportementale se décompose classiquement en quatre parties: perception, décision, action, communication : l'humain voit ce qui se passe autour de lui, décide de son action en fonction de sa perception et de ses connaissances, et enfin agit sur lui-même et sur son environnement. La communication consiste en un dialogue avec d'autres objets de l'environnement. Nous avons proposé un modèle formel de la partie décisionnelle du modèle comportemental. Il s'agit d'un système réactif, de type systèmes de transition parallèles hiérarchisés (nommé HPTS [1]) et qui est composé d'une hiérarchie d'automates parallèles. Chaque état du système peut être vu comme une boîte noire recevant un flot de données en entrée, délivrant un flot de données en sortie et possédant un contrôle continu et discret. Un état du système est soit un état terminal, soit un état composite. À chaque état correspond un paramètre d'activité qui a trois valeurs possibles : actif, suspendu ou inactif. Les paramètres de contrôle permettent d'influer, par décision externe ou interne à l'entité, sur le comportement à adopter. Par exemple, un sous-état peut avertir l'état père qu'il s'est auto-terminé, tandis qu'un état peut avertir un de ses sous-états qu'il doit débuter ou se terminer. Un bloc de code peut être associé tant aux états qu'aux transitions (code C++ plus quelques instructions spécifiques). Une fonction de filtrage gère la cohérence des actions proposées par les états actifs des différents sous-automates parallèles et fait une proposition cohérente qui est renvoyée à l'état englobant. Il existe deux modes de franchissement des transitions : dans le mode normal, on teste les transitions dans l'ordre de spécification et la première ayant une condition de franchissement valide sera choisie, tandis que dans le mode aléatoire, l'ensemble des transitions franchissables seront prises en compte et un tirage aléatoire sera effectué, prenant en compte les pondérations associées aux transitions. D'autre part, deux modes d'exécution sont disponible : le mode normal ou une seule transition est franchissable par automate à chaque pas de temps et le mode TIMEGATE dans lequel les automates peuvent franchir plusieurs transitions lors du même pas de temps. Dans ce second mode, l'exécution s'arrête lorsque plus aucune transition n'est passable ou que la dernière transition venant d'être franchie possède l'option TIMEGATE.

Gecomp est un utilitaire permettant de compiler les automates écrits dans le langage HPTS en classes C++. Ces classes sont ensuite intégrées dans un module pour être exécutées par la plate-forme GASP. Gecomp se base sur un fichier de configuration défini par l'utilisateur pour générer les fichiers permettant l'exécution des automates. Un interpréteur de ce langage a aussi été développé au sein de la plate-forme GASP pour permettre une mise au point en ligne des comportements.

1. Hierarchical Parallel Transition System

5.5 Plate-forme de simulation

debug : Module de nom « logiciels-gasp » pour le projet Siames, section logiciels

Participants : Alain Chauffaut [correspondant],.

Mots clés : plate-forme logicielle, simulateur distribué, simulateur temps réel, modularité.

Résumé : GASP *(General Animation and Simulation Platform) 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. 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).*

debug : corps du module.

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.

Afin d'être générale, la plate-forme de simulation doit intégrer les différents modèles (générateur, descriptif et comportemental) de contrôle du mouvement et des interactions d'objets dans une scène. Un système de simulation est constitué d'un certain nombre d'entités, dont l'état évolue au cours du temps, chaque entité possédant une fréquence propre de calcul. La plate-forme doit donc permettre la simulation, sur une même échelle temporelle, de plusieurs entités dont la synchronisation et les communications sont gérées par un noyau temps-réel.

Ces entités communiquent soit de façon synchrone par flot de données (à chaque pas de simulation, un objet de simulation obtient auprès des autres objets toutes les informations qui lui sont nécessaires) ou bien de façon asynchrone par envoi d'événement ou de message.

Le formalisme adopté est particulièrement adapté à la distribution des calculs, et ce de façon transparente pour un utilisateur de la plate-forme, c'est-à-dire pour un programmeur de module.

Un des objectifs principaux consiste à rendre la conception d'un module d'animation complètement indépendant de l'architecture cible. Nous avons donc constitué un modèle orienté objet de la plate-forme dans lequel le concepteur d'un module doit respecter un canevas de programmation (le plus simplifié possible). Pour ce faire, il lui suffit de définir les entrées, les sorties, les paramètres de contrôle du module en utilisant des types prédéfinis, et de définir la fonction de transfert (initialisation, pas de simulation, terminaison) associée à l'objet. Une classe de service a aussi été définie : il s'agit de la visualisation 3D temps-réel des résultats de la simulation. Pour définir une simulation, il suffit ensuite de décrire à l'aide d'un fichier de configuration les différents éléments la constituant : l'environnement virtuel (par exemple, un quartier d'une ville), la configuration matérielle ainsi que le choix des entités dynamiques et de leur paramétrage.

Selon la configuration, des noyaux différents de la plate-forme pourront être utilisés pour effectuer une simulation :

- un noyau mono-processus,
- un noyau temps-réel mono-processus,
- un noyau temps-réel multi-processus mono-machine,
- un noyau multi-processus multi-machines.

Le noyau temps-réel multi-processus mono-machine a pour but l'obtention d'une plus grande puissance de calcul, il est plus particulièrement dédié aux machines SGI multi-processeurs. Le noyau multi-processus multi-machines est quant à lui voué à l'utilisation d'un parc hétérogène de machines, son objectif est double. D'une part, nous visons une augmentation supplémentaire de la puissance de calcul et, d'autre part, la coopération entre plusieurs utilisateurs humains au sein d'un environnement 3D virtuel.

Notons enfin qu'après une période de développement et de mise au point, cette plate-forme est maintenant arrivée à un stade tel qu'il devient possible de l'exploiter pleinement, et elle va ainsi être largement diffusée au sein de la communauté française d'informatique graphique.

Les évolutions récentes de la plate-forme sont les suivantes :

Fiabilité, évolutivité et nouvelles fonctionnalités Nous sommes dernièrement passés de la version 1 de GASP (première génération) à la version 2 (seconde génération). Ce changement de génération a vu une certaine évolution de l'architecture de cette plate-forme, qui a ainsi fait d'importants progrès sur le plan du génie logiciel, en particulier au niveau de la fiabilité du typage des attributs (entrées, sorties ou paramètres de contrôle des entités simulées) lui apportant ainsi plus de robustesse.

Un mécanisme d'envoi asynchrone d'événements et de messages a également été ajouté aux communications par flots de données qui étaient jusqu'alors les seules possibilités offertes par notre première génération de plate-forme.

Par ailleurs, les aspects multi-fréquentiels sont maintenant gérés de manière efficace et de façon homogène entre les différents noyaux de la plate-forme, lesquels ont également été fusionnés et cohabitent maintenant au sein de GASP.

Aspects distribués et temps-réel La version multi-processus multi-machines de GASP, reposant actuellement sur PVM, a été totalement intégrée, elle cohabite maintenant avec la version multi-processus mono-machine multi-processeurs qui repose quant à elle sur REACT-PRO.

Les communications entre processus et/ou machines se font aussi bien de façon synchrone, en flot de données entre entrées et sorties d'entités, que de façon asynchrone par envois d'événements ou de messages. Dans ce dernier cas, pour la version multi-machines, nous ne pouvons garantir une réception immédiate des messages, alors que nous sommes en mesure d'offrir une synchronisation entre toutes les machines, à un "clic d'horloge" près, c'est-à-dire que nous ne pouvons garantir au mieux que la propriété suivante : au temps t, les entités calculées sur un processus peuvent accéder aux valeurs exactes des sorties des entités calculées sur les autres processus à l'instant $t - dt$, où dt est l'écart de temps entre deux pas de simulation.

L'objectif des versions temps-réel est de garantir une équivalence entre le temps simulé et le temps physique, si elle est possible. Cette équivalence permet la réalisation de simulations interactives au cours desquelles le comportement des entités simulées peut être calibré de manière précise.

Aide à la création de modules Afin d'assister le programmeur de module dans sa tâche, des utilitaires ont été créés. Ils permettent de spécifier les interfaces des modules (en termes de paramètres d'entrée, de sortie et de contrôle) ainsi que la façon dont ils doivent être exécutés (expression d'éventuelles dépendances entre modules). Ensuite, des squelettes d'objets de simulation et de leurs objets de calculs associés sont automatiquement générés, ainsi qu'un makefile, il ne reste alors plus au programmeur qu'à écrire le code effectif de ces modules.

Visualisation stéréo Afin de profiter pleinement des performances du dispositif de visualisation immersive (Immersia) acquis cette année, la visualisation de la plate-forme est passée d'un mode mono-pipe monovision à un mode "tri-pipe" stéréovision.

En effet, le mode initial mono-pipe monovision ne permet pas d'exploiter au mieux les caractéristiques de notre dispositif immersif, à savoir un écran cylindrique de neuf mètres de développée, et la possibilité d'utiliser trois "pipes" graphiques afin de calculer les images.

Dans ce contexte, le passage en mode "tri-pipe" permet d'une part de répartir les calculs d'une image sur les trois pipelines graphiques disponibles, et d'autre part d'obtenir une meilleure qualité géométrique d'image, car l'image finale, calculée à partir de trois points de vues différents, est beaucoup moins déformée que lorsqu'elle résulte d'une seule prise de vue "grand angle".

Le passage à la stéréovision apporte quant à lui la notion de profondeur dans l'image, ce qui permet obtenir une réelle impression de relief et renforce encore le sentiment d'immersion au sein d'un univers 3D.

5.6 Définition d'un outil pour l'entraîneur sportif

debug : Module de nom « logiciels-sport » pour le projet Siames, section logiciels

Participants : Bruno Arnaldi [correspondant],.

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

Résumé : *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.*

debug : corps du module.

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. 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'analye 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.

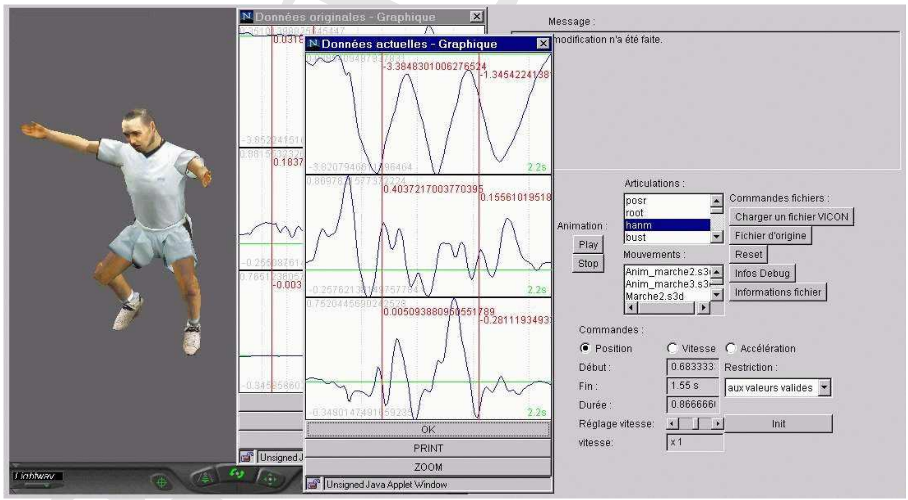

FIG. 1 – *Interface du logiciel dédié aux entraîneurs et sportifs*

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

debug : Module de nom « resultats » pour le projet Siames, section resultats
debug : corps du module.

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

debug : Module de nom « resultats-navtr » pour le projet Siames, section resultats

Participants : Kadi Bouatouch, Eric Maisel, Michaël Pouliquen.

Mots clés : Navigation, temps réel, client-serveur, niveaux de détail.

Résumé : *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 un prototype de navigation distribué basé sur une architecture client/serveur. La difficulté principale de cette étude résidant dans le choix de l'architecture logicielle permettant d'obtenir une fréquence d'affichage élevée tout en assurant une flexibilité importante afin de tester l'utilisation conjointe de représentations variées de scènes, chacune liée à une technique d'optimisation particulière.

debug : corps du module.

L'architecture de la plate-forme de visualisation est du type client/serveur, suffisante dans le cas d'une application de navigation dans une scène statique sans interaction entre les utilisateurs connectés au serveur. Le rôle de celui-ci est restreint puisqu'il se contente d'envoyer aux clients les données que ceux-ci demandent. Les clients ont donc deux rôles : ils assurent la visualisation des scènes et contrôlent les transferts de données géométriques et photométriques. Les choix de mise en oeuvre retenus satisfont les contraintes d'efficacité et de flexibilité.

L'efficacité est obtenue de deux façons :

- Le transfert des données est contrôlé par une description de la structuration de la scène que le client charge au préalable. Cette description indique au client quelle est la représentation de la scène nécessaire à la poursuite de la visite virtuelle en fonction de la position et de l'orientationé du clône virtuel de l'utilisateur.

- La façon dont les données graphiques sont représentées en mémoire. Les modèles tridimensionnels sont compilés (lorsque cela est jugé nécessaire) en des listes de polygones texturés, mises en oeuvre par des listes d'affichage dans la mémoire de la carte graphique du client.

La flexibilité est obtenue de deux façons :

- Par l'utilisation des capacités d'abstraction des langages orientés objets. La notion de classe virtuelle permet de considérer les objets à afficher à travers les fonctions qui les manipulent et cache la façon dont ils sont effcetivement représentés. Dans le cas de notre plate-forme de visualisation, il s'agit essentiellement de la fonction qui permet de transformer un objet graphique en une liste de polygones texturés. Ce mécanisme de transformation est étendu aux ensembles d'objets graphiques, en particulier aux ensembles structurés (représentations multi-résolution et graphe de visibilité).
- Par la conception modulaire de la plate-forme de visualisation. D'une part l'architecture client/serveur découple totalement la mise en oeuvre des parties client et serveur de la plate-forme. D'autre part le client est conçu sur un schéma producteur/consommateur : le producteur crée un ensemble de listes de polygones texturés à partir des données géométriques et photométriques transmises à travers le réseau. Ces listes sont parcourues périodiquement par le consommateur qui les transforme en images.

6.2 Structuration de scènes

debug : Module de nom « resultats-eclairext » pour le projet Siames, section resultats

Participants : Kadi Bouatouch, Eric Maisel, Reynald Dumont.

Mots clés : simulation d'éclairage, environnements complexes, décomposition de domaine.

Résumé : *Nous avons conçu des algorithmes et un logiciel permettant la subdision en cellules 3D (structuration) de scènes d'intérieur et urbaines, dans le but de rendre plus fluide la navigation dans ces types de mondes virtuels. La scène est subdivisée lors d'un prétraitement, et seules les partitions (ou cellules 3D) visibles par l'observateur sont affichée lors de la navigation, d'où une amélioration du taux de rafraichissement à l'écran.*

debug : corps du module.

Le principe de notre méthode de structuration consiste à découper totalement l'espace de la scène en cellules convexes. La structuration de la scène se fait dans les meilleurs cas en respectant la topologie, c'est à dire suivant les pièces dans un env ironnemnt d'intérieur, suivant les rues dans un environnement urbain. La recherche des cellules se fait par intersection des murs de la scène avec le plan horizontal (on obtient un ensemble de segments), puis en recherchant un ensemble de polygones convexes à partir de cet ensemble de segments incomplets (d û aux ouvertures des cellules). A partir de la projection des cellules sur le plan horizontal et de données additionnelles extraites de la scène, on découpe l'espace 3D dans sa totalité. Une scène de 350000 polygones a été structurée de manière quasi-automatique grâce à ce logiciel (figure 2).

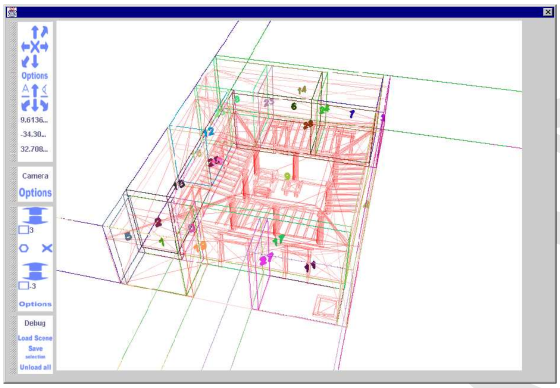

FIG. 2 – *Outil de partion de scène interactif*

6.3 Reconstruction 3D à partir d'une ou de plusieurs images

debug : Module de nom « resultats-renduimage » pour le projet Siames, section resultats

Kadi Bouatouch, Eric Maisel, Erwan Guillou

Mots clés : reconstruction 3D, calibration, texture.

Résumé : *Nous avons développé un logiciel interactif de reconstruction 3D d'un environnement réel à partir d'une ou plusieurs images. Les objets reconstruits sont approchés par un ensemble de primitives géométriques simples (telles que cubes, cônes, prismes, cylindres, etc.). Le processus de reconstruction est interactif et facilité par un ensemble procédures de traitements issues de la vision artificielle. Il utilise une nouvelle méthode de calibrage de caméra (ne nécessitant pas de mire réelle ajoutée à l'environnement) et d'estimation de mouvement rigide entre deux caméras.*

debug : corps du module.

Le logiciel que nous avons développé permet, de manière interactive, de calibrer une caméra à partir d'une seule image à l'aide de points de fuite. L'utilisateur sélectionne un ensemble de droites (parallèles dans l'espace 3D) déterminant deux points de fuite. Ceux-ci déterminent un repère 3D par rapport auquel est placée et orientée la caméra. En outre, L'utilisateur déplace, oriente et transforme des primitives géométriques (parallélépipèdes, cubes, cônes) devant englober les objets réels dans la scène. Ces primitives géométriques sont affichées sur toutes les images clés, ce qui permet une meilleure reconstruction. Cette affichage est rendu possible grâce à une nouvelle méthode d'estimation de mouvement entre deux caméras. Des textures (représentant des objets dans l'image) sont extraites, redressées, traitées (détection et remplissage de trous dus aux éventuelles occlusions) et plaquées sur les faces des primitives géométriques. Le résultat obtenu est un ensemble d'objets polyédriques texturés permettant à la fois un déplacement temps réel dans la scène et l'insertion d'objets virtuels (3, **??**,**??**).

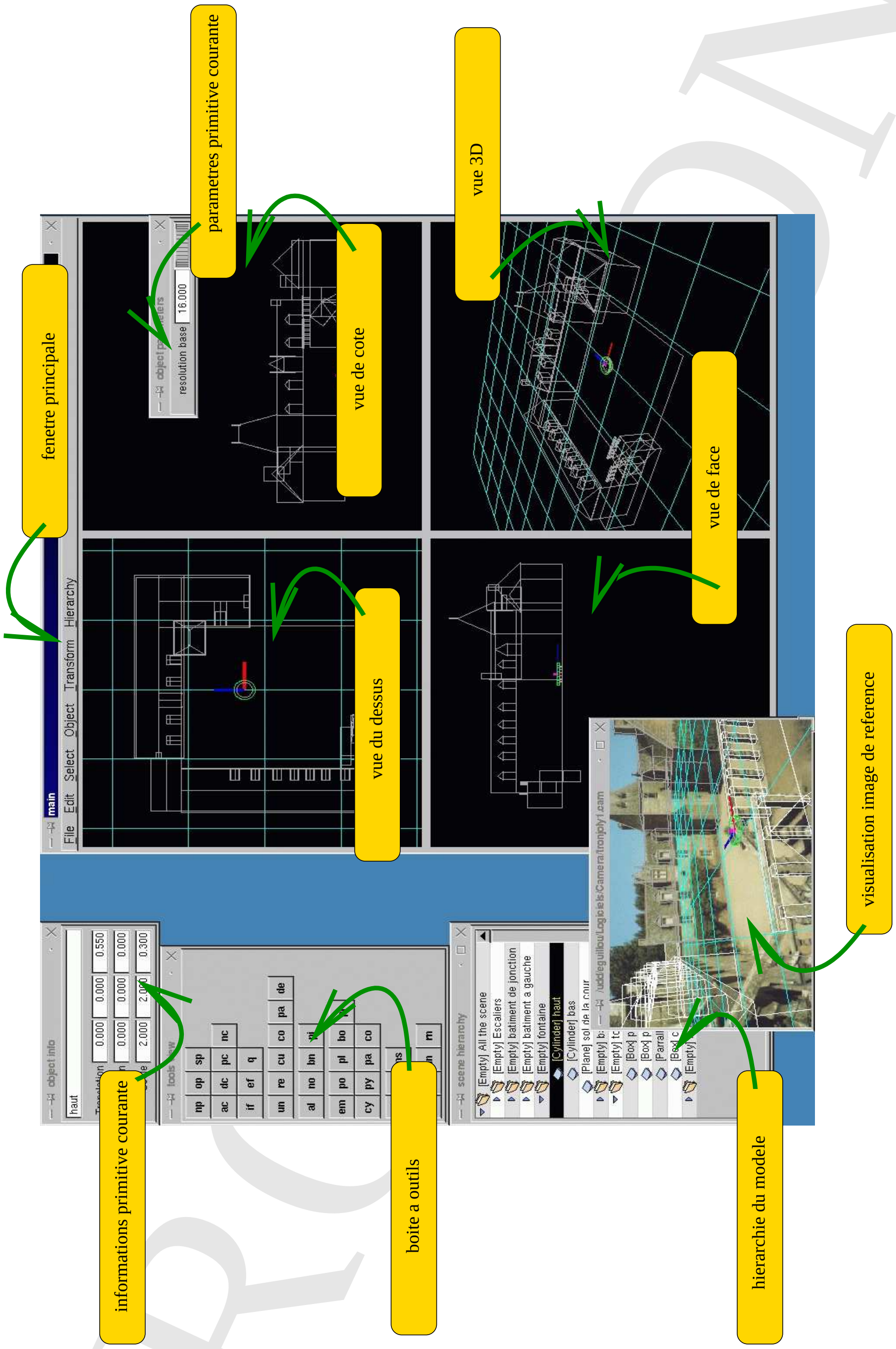

FIG. 3 – *Le logiciel de modélisation*

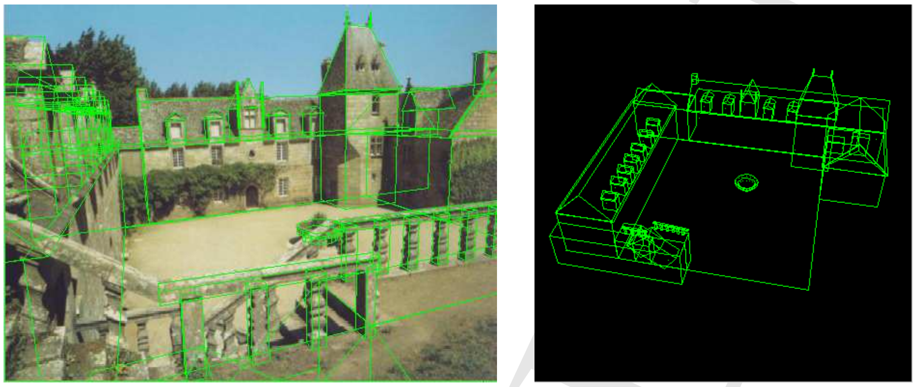

FIG. 4 – *Modélisation du chateau de Tronjoly (captures d'écran)*

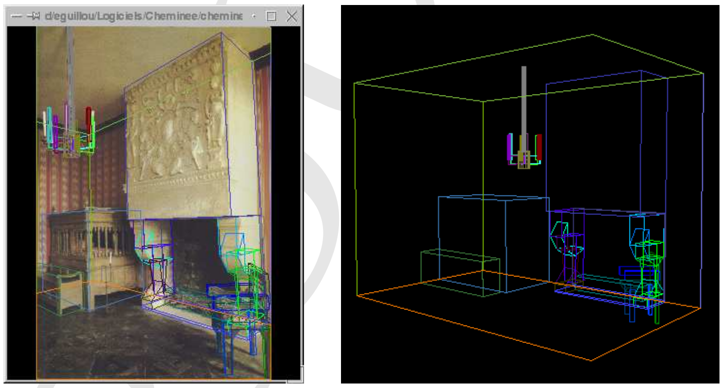

FIG. 5 – *Modélisation de la scène de la cheminée (captures d'écran)*

6.4 HPTS interprété

debug : Module de nom « resultats-interp » pour le projet Siames, section resultats

Participants : Frédéric Devillers, Stéphane Donikian.

Mots clés : animation comportementale.

Résumé : *Gecomp est un utilitaire permettant de compiler les automates écrits dans le langage HPTS en classes C++. Ces classes sont ensuite intégrées dans un module pour être exécutées par la plate-forme GASP. Gecomp se base sur un fichier de configuration défini par l'utilisateur pour générer les fichiers permettant l'exécution des automates. Dans ce fichier sont définis principalement le nom de la classe de base ainsi que les noms des fichiers décrivant les automates. L'emploi des différents modes d'exécution y est défini, comme le mode TIMEGATE et le mode ALEATOIRE. Chaque modification du modèle comportemental nécessite la génération des classes C++ ainsi que leur compilation. Lors de la programmation d'un comportement, la mise au point des automates demande de nombreuses générations et compilations, ce qui représente une perte de temps non négligeable. Un interpréteur du langage de spécification des comportements a été réalisé ; ainsi le programmeur n'a pas besoin de passer par la phase de génération et de compilation lors de la mise au point, ce qui lui fait gagner un temps précieux. Pour des raisons d'efficacité, l'utilisation de Gecomp sera tout de même nécessaire pour la version finale du programme.*

debug : corps du module.

L'interpréteur devant s'exécuter sur la plate-forme GASP, il est intégré dans un objet de calcul. Cet objet de calcul , qui sera par la suite appelée classe de calcul générée, est dérivé de l'objet de calcul écrit par le programmeur (la classe de base) dans lequel les méthodes appelées par les automates ont été définies. Ces méthodes pouvant être différentes pour chaque module, il a fallu trouver un moyen de les appeler à partir de l'interpréteur. La solution retenue est de créer pour chaque classe de base une classe Interface s'insérant entre l'interpréteur et la classe de calcul générée. C'est ensuite à cette dernière de faire appel aux méthodes de la classe de calcul de base décrite par le programmeur. Pour faciliter le travail de l'utilisateur, un programme de génération d'interpréteur a été écrit (GenInterp). Il se charge de générer la classe de calcul héritant de la classe de l'utilisateur et intégrant l'interpréteur ainsi que la classe d'interface. L'interprétation des automates HPTS se déroule en trois phases principales :

- construction des données à partir des fichiers de définition des automates et du fichier de configuration;
- vérification de l'intégrité des données;
- exécution des instances d'automates.

Le compilateur Gecomp laisse au compilateur C++ le soin de vérifier la plupart des données. Dans le cas de l'interprétation, toutes les vérifications doivent être faites par l'interpréteur avant le commencement de l'exécution. La version interprétée des automates HPTS étant utilisée pour la mise au point des comportements, il faut donc un moyen de voir l'évolution de l'état des automates au cours de l'exécution. Un affichage des attributs des automates peut ainsi être effectué à chaque pas de temps. L'utilisateur spécifie, dans le fichier de configuration de l'interpréteur, les attributs qui devront être affichés. Des mesure de performance ont été réalisée en comparant les temps d'exécution de la version compilée et de la version interprétée et la version interprétée prend en moyenne 110 % du temps de la version compilée.

Un gestionnaire permet d'exécuter des actions sur les interpréteurs et sur les automates au cours de l'exécution du programme de simulation. L'utilisateur peut ainsi suivre plus efficacement le déroulement de l'exécution de ses automates. Il peut aussi modifier leurs comportements et donc les mettre au point plus rapidement. Les actions sont les suivantes :

- suspension et reprise de l'exécution des automates d'un module interprété;
- modification du niveau de deboggage;
- modification de la valeur des variables des automates;
- modification du code des automates.

L'utilisateur peut naviguer à travers la hiérarchie d'automates afin d'accéder à l'instance qui doit être modifiée. Les modifications sont prises en compte dès le pas de temps suivant. La modification du code des automates au cours de l'exécution peut entraîner des problèmes de cohérence, ainsi la modification d'un automate est interdite si une instance de celui-ci est en cours d'exécution. La prise en compte de la modification nécessite seulement le contrôle de l'intégrité des données ; ce sont les mêmes vérifications que celles effectuées lors du chargement des automates.

6.5 Ressources et priorités dans HPTS

debug : Module de nom « resultats-ress » pour le projet Siames, section resultats

Participants : Fabrice Lamarche, Stéphane Donikian.

Mots clés : animation comportementale.

Résumé :

Pour illustrer la problématique de ce travail, prenons un exemple concret : une personne est assise devant une table, elle lit un livre en buvant un café et en fumant une cigarette. Cette phrase décrit un comportement relativement simple pour un être humain, cependant en terme d'animation comportementale, il pose divers problèmes. D'une part, il est constitué de trois comportements "indépendants" qui se déroulent en simultané : lire un livre, boire un café, fumer une cigarette. Dans le modèle HPTS tel qu'il a été décrit, simuler ce type de comportement est relativement complexe, il faut pouvoir les agencer de manière à respecter un certain nombre de contraintes telles que ne pas boire le café alors que l'on a une cigarette dans la bouche, ne pas tourner la page du livre avec la main tenant la tasse de café. D'autre part, lorsque l'on dispose des trois comportements codés séparément sous forme d'automates, il n'est pas utile de recoder un automate spécifique à leur réalisation en simultané. L'objectif est de pouvoir les lancer et les laisser s'agencer automatiquement en fonction des circonstances, des envies et des contraintes physiques de l'agent.

debug : corps du module.

Il est utile d'offrir un mécanisme permettant de dérouler des comportements en parallèle sans que l'utilisateur ait à synchroniser "à la main" leur exécution, et sans qu'ils nécessitent, au niveau de leur description, d'avoir la connaissance des autres comportements en cours d'exécution. Les plus gros problèmes de synchronisation viennent de l'utilisation des ressources internes de l'agent. Par ressources internes, on entend la vue, les mains, les pieds, toute dépendance, physique ou non, limitant leur déroulement en parallèle. Il s'agit donc d'ajouter au formalisme d'HPTS la gestion des ressources internes de l'agent et un mécanisme de priorités dynamiques (variables au cours du temps). Ces priorités permettent de gérer l'importance relative des comportements à chaque instant. En combinant ces deux mécanismes, il devient possible d'obtenir un ordonnanceur tentant de respecter au mieux l'allocation des ressources en fonction des priorités associées aux automates. Pour inclure la gestion des ressources dans les automates du modèle HPTS, le choix a été de considérer un ensemble fini de ressources déclarées lors de la création des automates. Par la suite, il est possible d'associer à chaque état l'ensemble des ressources que ce dernier utilise. Les ressources sont prises et relâchées au niveau des transitions. Si une ressource est présente dans l'état d'origine et dans l'état d'extrémité de la transition, elle est considérée comme gardée lors du changement d'état. Tant que l'on reste dans un état, les ressources qui lui sont associées ne sont donc pas libérées. Les entités évoluant dans un environnement dynamique et imprédictible, aucune prévision n'est possible quand au franchissement des transitions. Le problème principal, se posant lors de l'exécution en parallèle des automates utilisant les ressources, concerne les interblocages possibles sur des prises de ressources. La solution retenue consiste à utiliser la connaissance que l'on a de la structure des automates pour extraire des informations suffisantes permettant d'assurer qu'il n'y aura pas de problèmes d'interblocages tout en essayant de ne pas trop contraindre le parallélisme. L'algorithme développé actuellement possède des limitations car il existe des cas où l'ordonnanceur va empêcher de paralléliser l'exécution de certains automates pour risque d'interblocage alors qu'il n'y a aucune chance que cette situation se présente. Cette méthode nous permet néanmoins de gérer les ressources automatiquement, sans que l'utilisateur d'HPTS ait besoin de se soucier des interblocages, des prises et libérations des ressources qu'il utilise (elles sont implicites). Une partie des traitements associés à l'évitement des interblocages peut être effectuée lors de la compilation puisque toutes les ressources sont déclarées et associées statiquement à un état. Le test de compatibilité, pour sa part est peu coûteux et peut donc être géré en cours d'exécution des automates, ce qui est nécessaire puisque l'on ne peut pas connaître à l'avance le nombre d'automates lancés en parallèle.

Pour influencer l'ordonnanceur dans le choix des automates à activer, une notion de priorité a été rajoutée afin de prendre en compte l'importance des comportements les uns vis à vis des autres au cours du temps, et ce en fonction de paramètres physiologiques (tels que la faim, la fatigue ou la soif). A chaque état valide de chacun des automates, on associe un poids, symbolisant l'importance, pour l'automate, d'aller dans cet état. Ce poids n'est autre que la priorité de l'automate multipliée par un coefficient lié à la transition; il représente le degré de préférence que l'automate associe au fait de choisir cette transition pour se trouver dans l'état extrémité. Une fonction de coût permet de choisir le n-uplet d'états qui la maximise, autrement dit, celui qui d'après la fonction satisfait au mieux les contraintes de ressources, les priorités et les préférences de transition des automates.

6.6 Scénario pour l'animation comportementale

debug : Module de nom « resultats-scenario » pour le projet Siames, section resultats

Participants : Frédéric Devillers, Stéphane Donikian.

Mots clés : scénario pour l'animation.

Résumé : *Afin de 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.*

debug : corps du module.

Dès lors que l'on parle de spécification de l'utilisateur en ce qui concerne les actions des acteurs autonomes se pose le problème du langage de spécification. Pour l'utilisateur, plus le langage sera proche du langage naturel, plus la tâche d'écriture du scénario sera simplifiée. Par contre, les acteurs autonomes n'ont aucune connaissance d'un tel langage naturel. Les premiers travaux sur la notion de script ou de scénario pour l'animation ont eu tendance à proposer des langages de programmation à l'utilisateur, ce qui n'offre pas un niveau d'abstraction assez élevé pour qu'un non informaticien puisse être à même d'écrire de tels scénarios. De tels scénarios sont composés de " triggers " (capteurs) et " beacons " (avertisseurs), utilisés pour produire des événements spécifiques, et de contrôleurs de comportements utilisés pour orchestrer les actions des entités dans le but de produire des situations particulières. Il s'agit ici d'un scénario vu comme une méta-entité dont la programmation se fait comme pour les entités à l'aide d'une bibliothèque d'objets C++.

Un des objectifs de ce travail est de s'approcher le plus possible du langage utilisé par les scénaristes. Il faudra ainsi être capable de passer d'une description dans un langage proche du langage naturel à un modèle opérationnel de cette description. Nous avons tout d'abord décrit un langage de représentation interne des scénarios qui servira de base pour la construction des langages de scénarisation plus proches du langage naturel. La sémantique de ce langage a été spécifiée en utilisant une extension du modèle HPTS pour le séquencement des instructions et la gestion des branchements et la logique d'Allen pour l'ensemble des contraintes temporelles sur les scénarios et les séquences d'actions.

Nous sommes entrain de travailler à l'automatisation du processus de compilation. Un scénario est décomposé en un ensemble de sous-scénarios et un scénario de base correspond à un ensemble de tâches ou d'actions. A tout niveau de la hiérarchie, il est possible de fournir des contraintes temporelles relatives ou absolues (un scénario doit débuter à telle date ou après que tel événement se soit produit ou encore terminer telle action trois secondes après le démarrage de telle autre). Afin de déterminer certaines situations, des observateurs doivent être positionnées dans la scène et toute action d'un acteur sur son environnement doit pouvoir être prise en compte (ouverture d'une porte par exemple). Les instructions d'un scénario peuvent modifier les caractéristiques, activités, ou motivations d'un acteur. Comme celui-ci peut avoir différentes activités en parallèle, il est nécessaire de prendre en compte des niveaux de priorités évoluant en fonction de ses motivations et de ses paramètres physiologiques (la faim ou la peur par exemple), ainsi que les ressources nécessaires pour réaliser telle ou telle action.

Nous sommes actuellement en phase de codage du système de scénarios. Les éléments suivants ont été implémentés et testés.

- Un analyseur lexico-syntaxique de la grammaire du langage de scénario, incluant les éléments de vérifications de types nécessaires pour lever les ambiguités du langage a été écrit.
- Un générateur du code pour les instructions exprimables via le langage d'automates HPTS a été écrit. Le code généré a permi des tests qui ont montré que la sémantique des instructions a bien été respectée.
- Le système de réservation d'acteur par le scénario a été écrit et testé. Il permet à un scénario de sélectionner parmi un ensemble d'acteurs ceux qui ne sont pas déjà occupés. Une priorité associée à la réservation permet, le cas échéant, à des scénarios prioritaires de "voler" des acteurs réservés par des scénarios secondaires.
- Le système de communication entre acteur et scénario a été défini et implémenté. Il permet au scénario de s'abstraire de la communication plus bas niveau de la plateforme GASP. L'accès par un scénario à un acteur se fait via un appel de méthodes d'une interface, qui produit une communication si nécessaire. Le retour d'information de l'acteur vers le scénario donne lieu, après communication, à l'appel d'une méthode du scénario. Un générateur permet au programmeur d'acteur d'automatiser au maximum l'écriture de cette interface.
- La structure globale du module GASP d'exécution des scénarios est maintenant définie et presque implémentée (figure 6). Il reste cependant à ajouter le système de contextes de scénarios, qui doit permettre les appels de sous-scénarios par un scénario père.

6.7 Outils de restitution et de contrôle de mouvements acquis

debug : Module de nom « resultats-mvtacquis » pour le projet Siames, section resultats

Participants : Bruno Arnaldi, Franck Multon, Stéphane Ménardais.

Mots clés : acquisition de mouvements, reconstruction, contrôle, déformation.

Résumé : *L'utilisation de systèmes d'acquisition optique est de plus en plus répandue pour restituer et analyser des mouvements. Contrairement aux modèles d'animation, ils offrent l'avantage de produire des séquences très personnalisées et complexes mais posent encore à l'heure actuelle un certain nombre de problèmes. Une chaîne logicielle permettant le traitement, la restitution et le contrôle de mouvements acquis pour l'animation d'acteurs virtuels a été mise en oeuvre.*

debug : corps du module.

L'objectif de cette chaîne logicielle est dans un premier temps de corriger au mieux les imperfections des acquisitions liées à l'utilisation d'un système optique afin de diminuer le temps de post-production et d'améliorer l'analyse du mouvement. En effet, lors d'une acquisition, deux caméras doivent voir un même marqueur au même instant pour obtenir sa

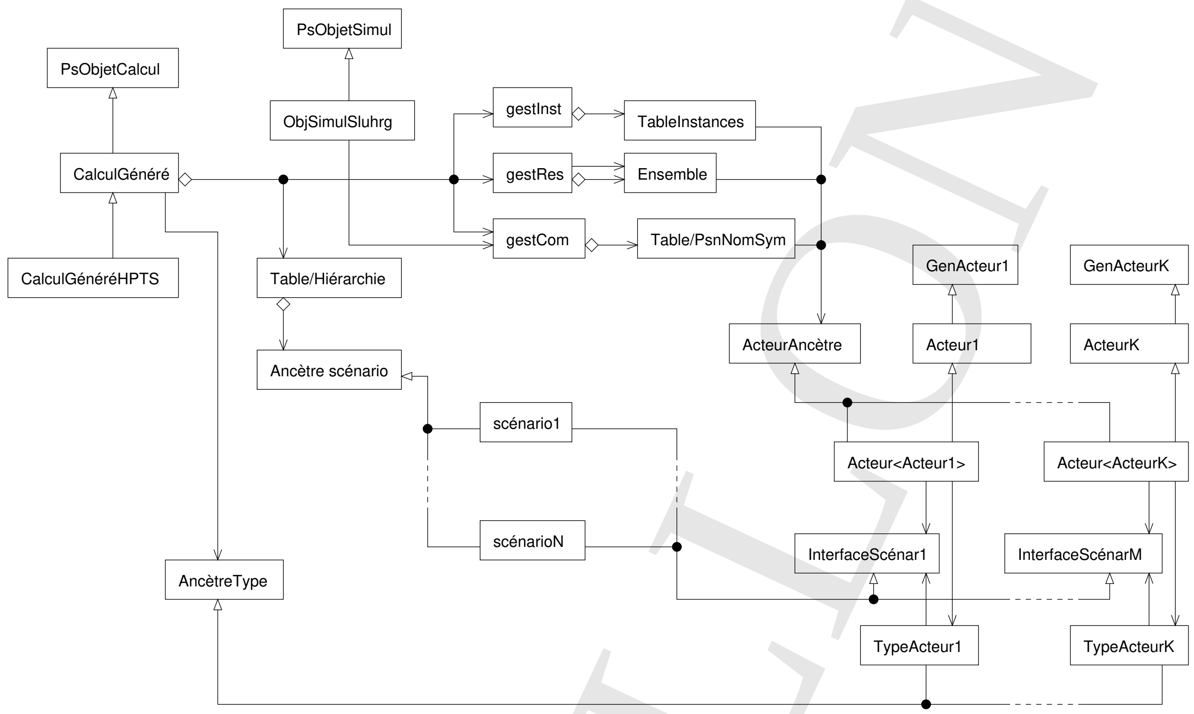

FIG. 6 – *schéma UML du module d'exécution.*

position spatiale. Or, le nombre de caméras étant limité, plus le mouvement est complexe (hautes fréquences, zone d'action importante) plus on observe des occlusions et des inversions de marqueurs. Notre chaîne corrige donc au mieux ces problèmes en tenant compte des données anthropomorphiques du squelette. A chaque instant, un graphe de voisinage est utilisé afin d'estimer l'erreur d'approximation engendrée par chaque reconstruction de marqueurs manquants. Les coefficients obtenus permettent d'ordonner au mieux une reconstruction récursive des positions manquantes et de minimiser ainsi l'imprécision des données calculées. Une fois le mouvement traités, les angles et quaternions sont automatiquement reconstruits et les propriétés anatomiques du squelette sont prises en compte pour mettre à l'échelle automatiquement l'acteur virtuel utilisé. La restitution des mouvements se fait à l'aide d'un pilote permettant de mixer toutes sortes d'action à n'importe quelles fréquences avec des niveaux de priorités variables et propres à chaque articulations. Plusieurs actions spécifiques offrent en outre la possibilité de contrôler le mouvement en appliquant des déformations par cinématique inverse, par des contraintes spatio-temporelles ou par "motion-warping". Ce pilote est actuellement utilisé pour la visite virtuelle d'un musée ainsi que pour un logiciel d'analyse du mouvement destiné à plusieurs fédérations sportives dans le cadre d'un projet au sein du pôle mouvement. La figure 7 illustre le traitement d'un mouvement de trampoline. La partie supèrieur représente les positions des marqueurs fournies par le système d'acquisition tandis que la partie infèrieure montre le même mouvement traité et restitué à l'aide d'un humanoïde virtuel

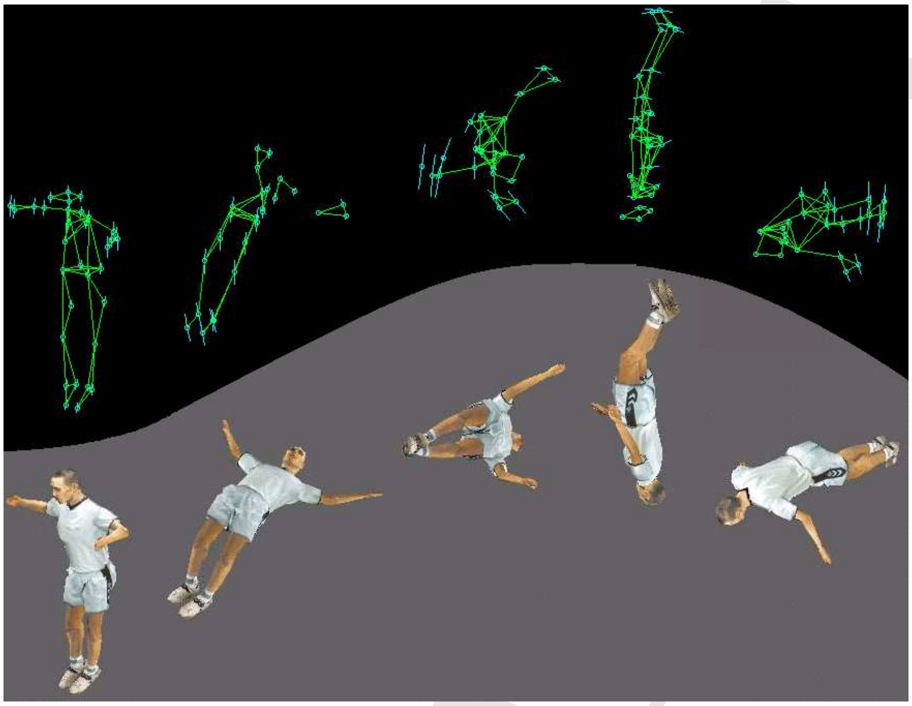

FIG. 7 – *Reconstruction et restitution de mouvement à l'aide d'un sytème d'acquisition optique*

6.8 Interactions en Univers Virtuels 3D

debug : Module de nom « resultats-interactions » pour le projet Siames, section resultats

Participants : Thierry Duval, Alain Chauffaut, Bruno Arnaldi, David Margery.

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

Résumé : *Nous avons tenu à offrir des possibilités d'interaction 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 la plateforme GASP 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. Les interactions à plusieurs utilisateurs et/ou multimodales sont rendues possibles par notre paradigme GASP

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 est quant à elle possible grâce au mécanisme de GASP mettant en relation les référentiels et leurs miroirs.

debug : corps du module.

6.8.1 Coopération entre utilisateurs

Les coopérations entre utilisateurs distants et les environnements virtuels 3D sont désormais possibles dans la version 2 de GASP, en particulier grâce à la mise au point de la version PVM et des travaux menés autour de la répartition des objets de simulation GASP sur plusieurs processus. En effet, le module de visualisation étant un objet de simulation à part entière, il est aisé d'en instancier un sur chacun des postes utilisateurs afin de permettre aux utilisateurs de partager un même univers virtuel 3D, comme nous l'avons décrit dans [14]. Un tel exemple est illustré figure 8.

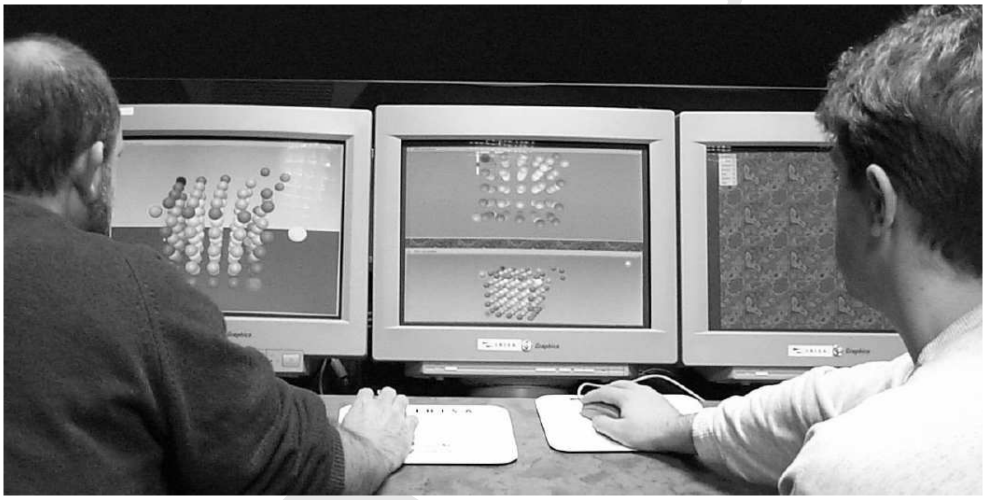

FIG. 8 – *Coopération entre 2 utilisateurs sur machines distinctes*

Nous avons également souhaité explorer de nouvelles possibilités d'interactions à plusieurs utilisateurs, et ce dans le cadre particulier d'utilisateurs présents physiquement devant la même machine : notre dispositif immersif de visualisation. Nous avons ainsi réalisé l'intégration de nouveaux dispositifs d'interaction (suivi de position de diodes électro-luminescentes) permettant à plusieurs utilisateurs d'interagir simultanément dans ce contexte, comme présenté dans [17] et illustré figure 9.

Durant nos tests, nous avons d'ailleurs pu faire coopérer un groupe de 4 utilisateurs utilisant les diodes au sein du dispositif immersif avec 3 autres personnes utilisant des stations de travail, permettant ainsi de valider techniquement la possibilité de travail coopératif à 7 personnes. Pour cet essai coopératif, nous n'étions limités que par le nombre de machines disponibles : pour faciliter la démonstration nous nous étions restreints à la seule utilisation des stations de

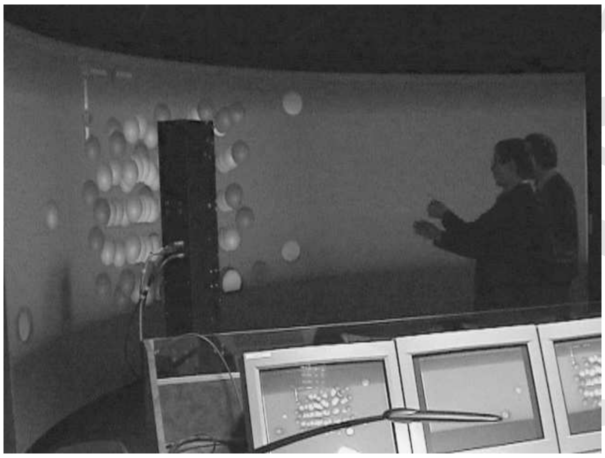

FIG. 9 – *Coopération locale entre 2 utilisateurs*

travail présentes dans notre local immersif. Nous pourrions de la même façon relier entre eux plusieurs dispositifs immersifs pour permettre à différents groupes d'utilisateurs de coopérer tous ensembles.

6.8.2 Interactions multimodales

Ces interactions coopératives utilisant plusieurs canaux d'entrées de même type peuvent également être considérées comme multimodales ainsi que nous l'avons décrit dans [12]. Un tel exemple d'interaction est illustré figure 10.

En effet, ce qui nous semble important est que nous avons à fusionner des données en provenance de plusieurs canaux d'entrées qui fournissent ici des positions 3D de diodes, et non pas le fait que ces canaux offrent des données de même type (ce qui est le cas ici) ou non (ce qui est le cas classique en multimodalité). De ce point de vue, GASP peut probablement être considéré comme un outil permettant de concevoir aisément un moteur de fusion primitif. Dans ce cadre, il apparait clairement que nous traitons la multimodalité comme une coopération de modes, et que nous faisons de ce fait de nombreux rapprochements entre interactions multimodales (coopération de modes) et collaboratives (coopération de personnes).

6.8.3 Intégration homogène de dispositifs d'interaction

L'intégration d'un dispositif d'acquisition de positions 3D (le suivi de diodes) nous a amené à proposer une décomposition logicielle en interacteurs de bas niveau ou "pilotes" (*drivers*) et interacteurs de plus haut niveau. Ces pilotes sont des modules assurant l'intégration des dispositifs physiques et ne fournissent que des données brutes, alors que les interacteurs de haut niveau sont purement logiciels et font des interprétations logicielles des données brutes fournies par les pilotes. Les mécanismes mis en œuvre sont détaillés dans [16]. En particulier, ces mécanismes permettent de créer de nouveaux interacteurs totalement reconfigurables de façon logicielle. Un exemple de modélisation d'un tel interacteur (une souris 3D créée à partir des positions de 2 diodes) est présenté figure 11 avant interaction, et figure 12 en cours d'interaction.

FIG. 10 – *Interaction avec 2 diodes pour obtenir une souris 3D*

Avant interaction, cet interacteur est en fait branché sur deux sorties du pilote des diodes, ainsi que sur les sorties de position des objets avec lesquels il est susceptible d'interagir (pour un service de "picking").

Suite à une interprétation particulière des sorties du pilote des diodes (rapprochement des positions des 2 diodes suivies), un message est envoyé à l'objet de simulation le plus proche, afin de lui proposer un asservissement en position.

Cet objet crée alors dynamiquement une entrée qu'il vient brancher sur la sortie de la visualisation, puis change de module de calcul afin de réaliser l'asservissement en position.

6.8.4 Adaptateurs pour objets interactifs

Enfin, nous avons mené des réflexions ayant pour but de trouver le meilleur moyen de rendre interactifs des objets de simulation. La problématique est ici du domaine du génie logiciel, elle consiste d'une part à trouver une méthode d'adaptation efficace des modèles d'architecture logicielle issus du domaine des Interfaces Homme-Machine tels que le modèle PAC-Amodeus, comme nous l'avons décrit dans [15], 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'in-

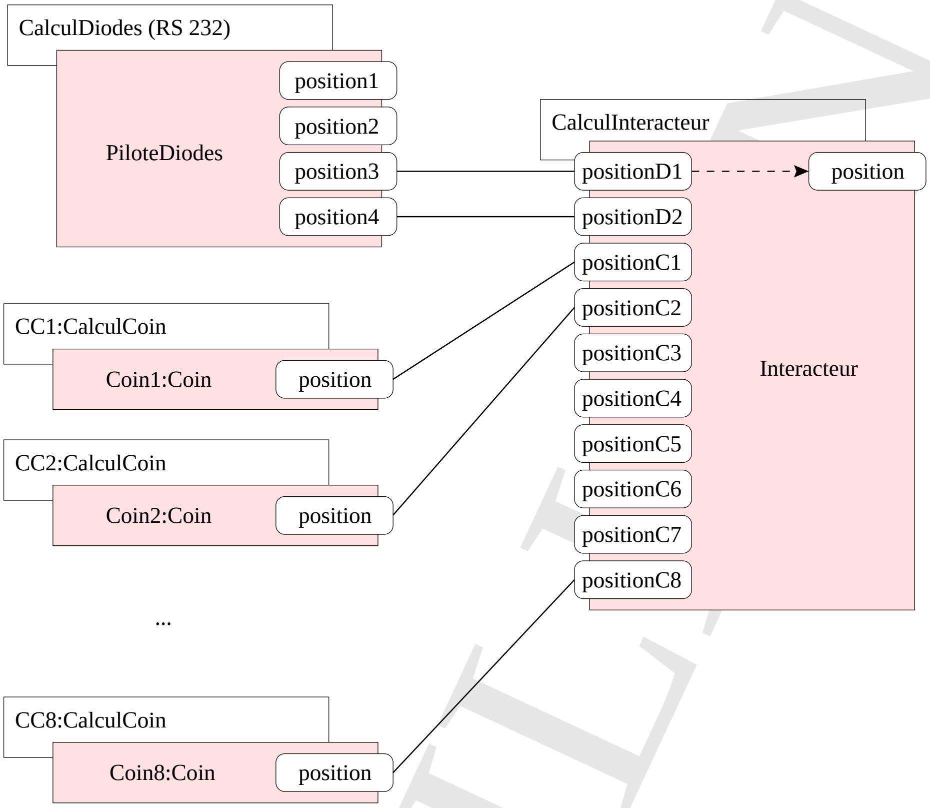

FIG. 11 – *Stucture modulaire de l'interacteur souris 3D*

teraction, comme présenté dans [13].

C'est en utilisant de tels adpatateurs qu'en particulier les objets de simulation observables figures 8, 9, et 10 ont été rendus interactifs, afin de pouvoir répondre de façon adequate aux messages en provenance des interacteurs pour passer de l'état de la figure 11 à celui de la figure 12.

7 Contrats industriels (nationaux, européens et internationaux)

debug : Module de nom « contrats » pour le projet Siames, section contrats
debug : corps du module.

7.1 Cité des sciences et de l'industrie :images animées autonomes

debug : Module de nom « contrats-cite » pour le projet Siames, section contrats

Participants : Frédéric Devillers, Stéphane Donikian, Jean-Michel Zelmeur.

Mots clés : animation comportementale, environnement virtuel, scénario pour l'animation.

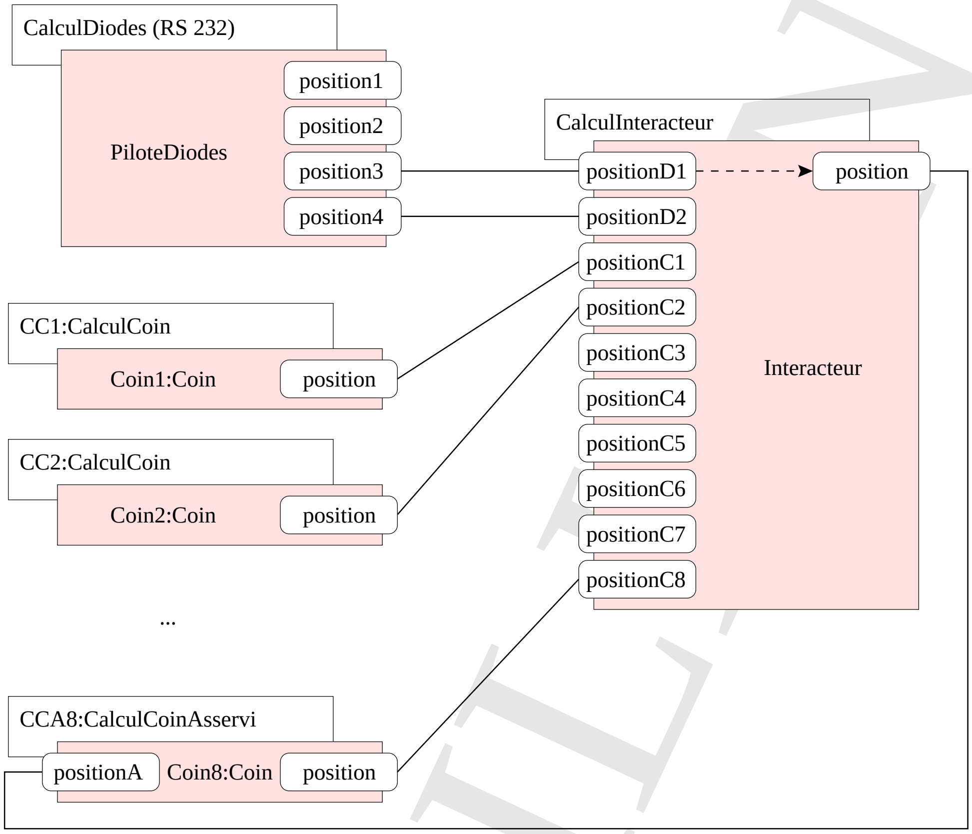

FIG. 12 – *L'interacteur souris 3D en cours d'utilisation*

debug : corps du module.

Un prototype de musée virtuel a été développé pour illustrer les travaux sur la scénarisation d'animations comportementales. Il faut noter que cette application intègre différents travaux des membres du projet SIAMES, relatifs au contrôle du mouvement d'humanoïdes, à l'interaction utilisateur et à l'animation comportementale. Cette factorisation a eu lieu grâce à l'utilisation de la plate-forme GASP. Ce musée virtuel a intéressé la cité des sciences et de l'industrie, et il fera partie de la nouvelle exposition permanente sur l'image qui ouvrira en juin 2001.

Deux types d'acteurs comportementaux ont été réalisés pour le moment.

– Les visiteurs, qui possèdent deux modes de comportements principaux. Le premier leur permet de parcourir le musée en évitant les collisions et en suivant la visite (cf figure 13). Le second, provoqué par l'utilisateur, permet à ce dernier de prendre le contrôle du but des visiteurs.
– Les suribirds, oiseaux stylisés au comportement de suricats, font partie de l'exposition. Outre les comportements "normaux" de guet et de picorage, ils prennent la fuite quand un visiteur (que l'utilisateur contrôle) s'approche trop près de leur groupe.

Un système de trigger (détecteur de présence) à été implémenté et testé. Il permettra aux

FIG. 13 – *visite du musée.*

scénarios de posséder des informations sur la localisation et l'identité des acteurs.

Un scénario détaillé est en cours d'écriture avec la cité des sciences afin de décrire le comportement de quelques personnages particulier tels qu'un voleur, un photographe, un gardien, une mère et son enfant. Cette application est un bon champ d'expérience tant pour la plate-forme GASP, que pour nos travaux sur les langages de scénario.

7.2 CSTB : simulation d'éclairage

debug : Module de nom « contrats-cstb » pour le projet Siames, section contrats

Participants : Kadi Bouatouch, Éric Maisel.

Mots clés : simulation d'éclairage, éclairage naturel, éclairage artificiel, approche spectrale.

Résumé :

Réalisation d'un logiciel de simulation d'éclairage appelé GIN dans le cadre de bourses de thèse financées par le CSTB de Nantes.

debug : corps du module.

En collaboration avec le CSTB de Nantes et dans le cadre de deux bourses financées par cet organisme, nous avons développé un logiciel de simulation d'éclairage appelé *GIN* basé sur une approche physique et spectrale, simulant de vrais luminaires, prenant en compte l'éclairage naturel, permettant de modifier les caractéristiques physiques des matériaux et de recalculer de manière instantanée l'éclairage en effectuant un maillage hiérarchique. Contrairement aux autres logiciels (à part Lise développé par le LCPC) l'approche adoptée est spectrale et le nombre et les valeurs des longueurs d'ondes des échantillons sont déterminés automatiquement par le logiciel en fonction de la forme des spectres des sources et des réflectances des matériaux. Le logiciel intègre un certain nombre de critères de confort visuel.

7.3 CSTB Grenoble : simulation de propagation d'ondes radioélectriques

debug : Module de nom « contrats-cstb » pour le projet Siames, section contrats

Participants : Kadi Bouatouch.

Mots clés :

.

Résumé : *Le CSTB de Grenoble s'intéresse à la communication entre stations de travail au moyen d'un réseau local Hertzien. L'avantage de ce choix réside dans sa souplesse de reconfiguration. Par contre il nécessite l'utilisation d'outils de simulation de la propagation du rayonnement radioélectrique. L'objectif de notre collaboration avec le CSTB est d'apporter les connaissances que nous avons acquises lors de l'étude des algorithmes d'éclairement globaux dans le domaine du rayonnement électromagnétique visible et de les transposer au domaine radioélectrique.*

debug : corps du module.

Le projet SIAMES participe avec le CSTB de Grenoble à la conception d'outils de simulation de la propagation d'ondes radioélectriques. Le problème posé consiste à pouvoir déterminer comment placer dans plusieurs pièces des stations de travail dotées d'émetteurs-récepteurs Hertziens ainsi que des relais (dans la bande de fréquence de 60GHz) de façon à échanger des informations en évitant des artefacts tels que les affaiblissements de signaux ou les interférences provoquées par les cloisons ou l'ameublement des pièces. L'intérêt du média Hertzien réside dans la possibilité de reconfigurer le réseau sans avoir à recabler les pièces.

Mais les outils de conception sont en nombre assez réduit et ne conviennent que peu à ce problème : les campagnes de mesure sont coûteuses tandis que l'utilisation de logiciels reposant sur l'emploi de statistiques conduit à des résultats trop imprécis. Contrairement à ce qui se passe pour d'autres applications, les ondes radioélectriques peuvent ici se réfléchir aussi bien de façon spéculaire que diffuse. La simulation du rayonnement radioélectrique doit donc s'effectuer au moyen d'un algorithme de radiosité suivi d'une deuxième passe spéculaire. Les transferts d'énergie ne se faisant pas de façon instantannée, ils sont représentés par le biais de facteurs de forme temporels calculés à l'aide d'une variante de l'algorithme de l'hémicube. La méthode de résolution est une adaptation de la radiosité progressive. Une validation de la méthode a été effectuée par comparaison avec des mesures effectuées sur un réseau déployé sur deux pièces.

7.4 Délinéateurs intelligents

debug : Module de nom « contrats-PREDIT » pour le projet Siames, section contrats

Participants : Annick Leroy, Stéphane Donikian, Bruno Arnaldi,.

Mots clés : prototypage virtuel, simulation comportementale, simulateur.

Résumé : *Notre environnement de simulation comportementale est utilisé comme plate-forme de prototypage virtuel d'un délinéateur intelligent.*

debug : corps du module.

Ce projet, financé par le PREDIT (programme de recherche national sur les transports), est mené en collaboration avec les sociétés Lacroix Technologie et Edixia. Il s'agit de proposer l'architecture d'un délinéateur intelligent capable d'assurer une surveillance temps réel et continue de sections routières. Cette surveillance concerne la détection de véhicules à l'arrêt, en marche arrière, roulant en sens inverse ou encore d'obstacles inattendus.

Outre un état de l'art du domaine, le projet DELICE a pour une bonne part consisté à examiner un ensemble de solutions techniques potentielles concernant l'acquisition de données (technologie capteur) et les technologies de transmission et d'alimentation. L'objectif de cette étude était de déterminer la meilleur association respectant les objectifs de performance du délinéateur intelligent.

La solution retenue consiste à équiper le bas côté de la chaussée de caméras visant des marqueurs, passifs ou diodes infra-rouge, placés sur le rail central. Tout objet passant sur la chaussée aura pour effet d'occulter un ou plusieurs marqueurs dans le champ d'au moins deux caméras. La mise en correspondance des marqueurs occultés par un logiciel embarqué permet de situer immédiatement la position et la dimension approximatives de l'objet sur la chaussée.

La surveillance globale d'une portion de chaussée est réalisée grâce à l'emission à intervalles réguliers par les délinéateurs de paquets de données à destination de l'unité centrale de traitement temps réel.

La modèlisation d'une portion de route équipée de délinéateurs et de capteurs est réalisée à l'aide du modeleur VUEMS développé dans l'équipe. Notre outil de simulation permet de prendre en compte l'ensemble des paramètres définis et de procèder à la simulation avec visualisation graphique de scénario. Ces derniers décrivent la circulation de véhicules roulant à des vitesses différentes sur une portion de route à 4 voies. Chaque délinéateur transmet au superviseur un codage de l'image indiquant le degré de visibilité de chaque marqueur. L'observation des données fournies par un délinéateur constitue un faisceau qui permet d'évaluer la vitesse et la densité du trafic en ce point. Ci-dessous le faisceau observé pendant quelques secondes par un capteur au passage de 5 voitures :

image 6.gif : passage des véhicules devant un délinéateur, ici matérialisé par un réverbère. Au dessus de la rambarde centrale apparaissent des réflecteurs verts espacés d'un mètre. Le recoupement des informations fournies par deux délinéateurs consécutifs permet de détecter les obstructions de la chaussée. Les simulations ont été effectuées avec des angles de caméra différents, des coefficients de brouillard variables. La lumière réfléchie par les capteurs a été dégradée aléatoirement pour simuler la saleté ou des pannes.

7.5 CCETT : application interactive 3D multimédia

debug : Module de nom « contrats-CNET » pour le projet Siames, section contrats

Participants : Bruno Arnaldi, Rémi Cozot, Bernard Valton.

Mots clés : multimédia, VRML, Java.

Résumé : *Les applications interactives font de plus en plus appel à des modèles sophistiqués d'objets 3D et à des modèles d'animation évolués. De plus, les applications distribuées mettent en évidence la complexité de gérer la cohérence spatiale et temporelle des objets d'une scène en interaction avec plusieurs utilisateurs distants.*

Ce travail a pour objectif d'étudier, d'une part, les apports potentiels des récentes avancées dans le domaine de l'informatique graphique appliquée au domaine du multimédia interactif et, d'autre part, l'apport des nouveaux standards tels que VRML 2.0 et Java.

debug : corps du module.

Cette collaboration, matérialisée par l'encadrement d'un doctorant (Bernard Valton) financé par le CCETT, concerne les applications interactives 3D multimédia et plus particulièrement la notion de forums interactifs (lieux de rencontre virtuels). On trouve dans ce genre d'application un certain nombre de composants liés au domaine de l'informatique graphique : la modélisation 3D d'un environnement virtuel, la visualisation 3D temps réel, l'animation réaliste de personnages (avatars), la simulation distribuée, etc ...

A travers l'utilisation de nouveaux standards de développement tels que VRML2.0 (pour l'environnement 3D et les personnages) et Java (pour le développement de l'applicatif), nous étudions l'apport des récentes avancées du domaine de l'informatique graphique dans le contexte du forum interactif : les modèles haut niveau de génération de mouvement (modèle générateur, modèle comportemental), les modèles de contrôle du mouvement (locomotion) et les niveaux de détail géométrique pour les grandes bases de données.

Après avoir effectué une bibliographie étendue sur le sujet (modèles d'animation et outils), une première expérimentation sur la locomotion d'un humanoïde a été effectuée sur la base d'une modélisation géométrique en VRML2.0 et du développement du moteur de marche en Java.

7.6 Cryo-Interactive : humain virtuel temps réel

debug : Module de nom « contrats-Cryo » pour le projet Siames, section contrats

Participants : Bruno Arnaldi, Rémi Cozot, Franck Multon.

Mots clés : humanoïde, temps réel, locomotion, habillage, jeu vidéo.

Résumé :

Le projet "Humain Virtuel Temps Réel" a pour but la création d'un module d'animation temps réel d'un humain virtuel. Ce module est intégré dans la plateforme de développement CryoGen de la société Cryo Interactive. Les principales fonctions du module d'animation sont la locomotion (marche, course), la préhension d'objet et l'habillage de l'humain. Le module est entièrement paramétrable. Les

mouvements produits sont calculés à partir de mouvements de référence spécifiés par les concepteurs de jeux.

debug : corps du module.

Le projet “Humain Virtuel Temps Réel” a pour but la création d'un module d'animation temps réel d'un humain virtuel. Ce module est intégré dans la plate-forme de développement CryoGen de la société Cryo en collaboration avec l'université de Rennes 2. Les principales fonctions du module d'animation sont :

- la locomotion (marche, course)
- la préhension d'objet
- l'habillage de l'humain.

Le module est entièrement paramétrable : les mouvements produits sont issus de mouvements de référence spécifiés par les concepteurs de jeux.

Le module de locomotion calcule automatiquement les mouvements de marche et de course qui s'adaptent à la configuration du sol. La locomotion humaine est générée à partir de trajectoires de référence définies par un animateur ou obtenues par acquisition de mouvements. Ces trajectoires sont décomposées automatiquement en suites de positions clés qui sont modifiées grâce à des règles issues de la biomécanique. Ces règles décrivent l'évolution des cycles de locomotion en fonction de la vitesse et de paramètres externes (configuration du sol). Le système peut alors calculer très rapidement (garantissant largement une animation temps réel) les positions intermédiaires. Le mouvement ainsi produit préserve les caractéristiques du mouvement de référence et vérifie les règles issues de la biomécanique.

Le module de préhension détermine automatiquement les mouvements de prise d'objets simples (ne faisant pas intervenir des configurations complexes des doigts) par un humain virtuel. L'un des objectifs est d'obtenir des mouvements respectant les caractéristiques de chaque personnage. A cet effet, le module utilise et déforme des trajectoires de référence fournies par le concepteur de jeux. Le résultat assure que la main approche l'objet dans une configuration permettant de le saisir dans de bonnes conditions.

Le module d'habillage en temps réel de l'humanoïde substitue à la modélisation par corps rigides une modélisation à partir d'un squelette et d'une peau (et des vêtements) déformables. Traditionnellement, le corps humain est abstrait par un ensemble de volumes indéformables articulés. Cette modélisation introduit des effets de cassure au niveau des articulations. Le module remplace cette modélisation en prenant en compte un modèle hiérarchique composé d'un squelette rigide articulé et d'une peau qui entoure le squelette. Le rôle du module est de calculer la déformation de la peau et des vêtements en fonction de la configuration du squelette. Cette dernière est calculée par les deux premiers modules : locomotion et préhension.

8 Actions régionales, nationales et internationales

8.1 Relations bilatérales internationales

debug : Module de nom « international-bilat » pour le projet Siames, section international

8.1.1 Amérique (du nord et du sud)

Participants : Stéphane Donikian, Gwenola Thomas, Frédéric Devillers.

debug : corps du module.

Dans le cadre d'une collaboration soutenue par le programme NSF[2]/ INRIA, des échanges ont été mis en place entre notre équipe et un laboratoire américain de l'Université de l'Iowa[3] sur le thème des scénarios pour la simulation de conduite. L'équipe américaine *Hank* travaille dans le domaine de la simulation de conduite (Iowa Driving Simulator) et utilise un modèle de représentation du comportement (HCSM) proche du nôtre (HPTS). L'activité de cette année a principalement porté sur le développement des langages de scénario, à savoir l'analyseur du langage et les différents éléments nécessaires pour pouvoir exécuter des scénarios.

Joe Kearney (professeur à l'Université d'Iowa) et Terry Hostetler (doctorant à l'Université d'Iowa) ont effectué un séjour de quinze jours au printemps dans notre projet. Nicolas Courty (doctorant dans le projet SIAMES) a effectué un séjour de quinze jours à l'Université d'Iowa en janvier.

8.2 Actions nationales

debug : Module de nom « international-nat » pour le projet Siames, section international
debug : corps du module.

Autres actions

- GREVE - Groupe de Réflexion sur les Environnements VirtuEls
  Depuis fin 1997, quatre équipes de recherche (le projet SIAMES de l'IRISA, équipe Informatique graphique du LaBRI, le LISSE de l'ENSM.SE et I3D) se sont regroupées au sein d'un groupe de réflexion dans le domaine de la réalité virtuelle baptisé GREVE. L'objectif premier du groupe de réflexion GREVE est le regroupement des forces de plusieurs laboratoires de recherche sur le thème de la réalité virtuelle. A travers une structure souple, les objectifs de GREVE sont poursuivis é travers des actions communes sur les trois points suivants :

  - Recherche : étude des complémentarités entre les membres du groupe et mise en place de collaborations de recherche bi ou multilatérale.
  - Valorisation : Développement commun de démonstrateurs, et mise à disposition de matériels spécifiques.
  - Veille technologique : mise en commun d'informations scientifiques et techniques relatives à la réalité virtuelle.

  Dans le cadre de GREVE, une Action de Recherche Coopérative (ARC) de l'INRIA, d'une durée de deux ans est en cours. Cette ARC a pour objectif de développer une

2. National Science Foundation
3. http://www.cs.uiowa.edu/~hank/index.html

plate-forme française ouverte et modulaire dédiée à la réalité virtuelle, qui soit multi-sites et multi-configurations et qui permette les développements (communs ou non) de chacune des équipes ou d'équipes extérieures.

- Pôle mouvement : Le laboratoire “Physiologie et Biomécanique de l'Exercice Musculaire” de l'Université de Rennes 2, l'École de kinésithérapie, la rééducation fonctionnelle de Pontchaillou ainsi que le projet SIAMES de l'Irisa se sont associés pour acquérir un système d'acquisition de mouvements VICON370 sur le site de Rennes. Cette action a été soutenue par le ministère de la Jeunesse et des Sports qui a décerné à ce groupe rennais, associé pour l'occasion à l'Institut National du Sport et de l'Éducation Physique, le titre de “Pôle de compétences mouvement” pour trois ans à partir de juillet 1998. Ce titre implique une participation à des projets d'analyse et de synthèse du geste posés par diverses fédérations sportives françaises. Le site rennais permet de proposer des solutions allant de la mesure à la restitution et la simulation du mouvement sportif aussi bien selon des critères visuels, cinématiques, mécaniques, physiologiques ou neuro-moteurs. Ces solutions passent par la définition d'outils croisés d'analyse et de synthèse du mouvement sportif et de produits finis directement utilisables par les entraîneurs.
- Pôle MESI :

  Le projet MESI repose sur un groupement de laboratoires qui mettent 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'Institut des Sciences pour l'Ingénieur (Ispi de l'École 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'Ispi 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.

  Un endoscope est défini par une structure mécanique basée sur un assemblage de segments articulés, des actionneurs, des lois de commande associées à ces actionneurs. Nous avons développé une boucle de simulation et d'optimisation qui permettra de définir automatiquement l'endoscope le mieux adapté à une intervention donnée.

  Les simulations sont menées sur la plate-forme de simulation modulaire GASP dont la conception est parfaitement adaptée à la cohabitation de modules de différents mondes de la physique (mécanique, asservissement et commande, optimisation).

  Le premier modèle mécanique développé utilisait le langage de description DREAM qui, à partir d'une interface de description, engendre un module directement utilisable sur

la plate-forme. Ce langage utilise le principe des travaux virtuels pour engendrer sous forme symbolique (calcul formel) le simulateur de l'objet considéré. Pour être intégré dans la plate-forme, ce module doit être compilé. L'optimisation nécessite un grand nombre de simulations et une possibilité de modifier les modèles, notamment au niveaux du nombre de segments constituant l'endoscope. Ceci impose, si nous utilisons DREAM) de précompiler les modèles envisageables (sans en oublier). Cette approche ne peut donc raisonnablement aboutir dans un contexte d'optimisation.
Un module de contrôle et de commande spécifique a également été développé. Il permet le contrôle en position du modèle d'endoscope. Il est basé sur un modèle d'actionneur qui devrait être réalisé à l'aide d'alliages à mémoire de forme.
Nous avons développé un prototype de nouveau module de la plate-forme basé sur un algorithme de simulation de type Newton-Euler récursif. Il permet de mieux répondre aux contraintes de modification de la structure de l'endoscope par le module d'optimisation puisqu'il permet d'échanger les modèles mécaniques. Nous avons défini une structure générique de description d'endoscopes qui permet de jouer sur le nombre de segments, leur longueur, leurs propriétés inertielles ainsi que sur les types des liaisons qui les relient.
Le développement d'un module de génération de canalisations basé sur des primitives géométriques simples a été effectué par l'intermédiaire d'un stage de maîtrise. Il a été complété par un module de traitement des interactions entre l'endoscope et cette canalisation. Le but poursuivi est de pouvoir faire interagir un endoscope virtuel avec un corps humain virtuel. Intéressons nous au rapport de rigidité entre une chaîne polyarticulée (modélisant l'endoscope) et un modèle déformable de vaisseaux humains (entourés de tissus). Le premier est le plus rigide, il est donc pertinent de considérer un modèle d'interaction basé sur une méthode de pénalité (ou pénalisation). La constante de pénalité sera identifiée plus tard par une approche mécanique générique (de type éléments finis) des organes ou vaisseaux considérés. Ceci nous ouvre une piste de collaboration potentielle avec le LTSI/INSERM.
Les principaux résultats concernent l'intégration, au sein de la plate-forme, des différents modèles définis plus haut :

- Modèle mécanique paramétrable d'endoscopes basé sur une succession de segments articulés ;
- Modèle d'un actionneur piloté par des AMF ;
- Modèle de description des réseaux de tuyauterie.

Le processus d'optimisation de structure d'endoscope a été connecté aux modules de simulation existants. Il est défini comme un superviseur qui peut arrêter une simulation, pour en définir une suivante et enfin l'exécuter. Pour l'instant, l'évaluation de la qualité d'une simulation n'est pas mesurée. Des travaux restent à poursuivre dans ce domaine. Le LRP travaille sur le sujet.
L'essentiel des travaux qui restent à réaliser sont les suivants :

- Améliorer et enrichir les outils de description des mécanismes d'endoscopes. C'est le but poursuivi par l'arrivée d'un stagiaire Post Doctoral au début du mois de novembre de cette année ;

  - Enrichir le modèle de description de réseaux de tuyaux, en définissant un modèle de réseau de canalisations souples : nous disposons de bases de données médicales acquises par scanner (acquisition médicale routinière) avec lesquelles nous nous proposons de faire interagir les modèles mécaniques d'endoscopes ;
  - Enrichir le modèle d'interaction entre l'endoscope et ce même réseau : ceci passera par la réalisation de modèles mécaniques génériques du comportement des tissus humains identifiés par des essais ;
  - Établir les connexions entre la simulation et un moyen de commande extérieur à retour d'effort qui permettra de "sentir" la pénétration dans les tuyaux pour faire tester les prototypes virtuels à des médecins. Ce point est l'objet d'une thèse qui début à l'ENS de Cachan, Antenne de Bretagne et fera l'objet d'une collaboration interne à l'équipe sur le retour d'efforts.

- distribution de notre plate-forme de simumlation GASP
  GASP est distribué, depuis le mois de novembre 1999, à d'autres laboratoires français qui travaillent dans le domaine de la réalité virtuelle. Certains de ces laboratoires participent à l'Action de Recherche Coopérative GREVE de l'INRIA :

  - I3D de l'INRIA-Rocquencourt,
  - équipe informatique graphique du LaBRI,
  - LISSE de l'Ecole des Mines de Saint-Etienne,
  - projet Imagis de l'INRIA Rhone-Alpes,
  - LIMSI de la faculté d'Orsay ;

  d'autres font partie du pôle MESI :
  - ENS Cachan,
  - Laboratoire de Robotique de Paris.

  Cette diffusion a été accompagnée d'une formation offerte aux futurs utilisateurs de GASP. De plus, un post-doc INRIA ira assurer le support utilisateur sur trois des sites partenaires, tout en participant à leurs developpements réalisés autour de GASP. Il commencera par apprendre les principes de fonctionnement et de développement de GASP en développant des modules à l'IRISA.
- 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 RNRT 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.

– Projets RNTL
  Nous avons soummis quatre projets RNTL pour l'appel d'offre 2000, ces quatres projets ont été labellisés en juin 2000. La contractualisation de ces projets est actuellement en cours.
  – Dramæra travaille depuis 2 ans sur un modèle d'environnement narratif, à travers une recherche poussée, et la réalisation d'une fiction interactive sur cédérom, "The Insider". Elle souhaite aujourd'hui étendre cette recherche et adapter sa méthode à de nouveaux environnements, en particulier Internet et la télévision interactive. Le projet de recherche proposé consiste à développer un nouveau modèle d'environnement narratif, adapté aux différents supports visés et à créer un outil d'assistance à l'écriture, qui permettra aux auteurs de créer des scénarios directement exploitables par le modèle d'environnement narratif que nous aurons créé.
  – La société IWI a développé une technologie innovante et unique qui lui permet une quasi-automatisation du processus de modélisation de villes numériques d'un très grand niveau de réalisme, à partir d'un ensemble de données géoréférencées. L'objectif du projet est de mettre en commun nos outils de modélisation de comportement humain et le savoir faire en modélisation d'IWI afin de pouvoir proposer dans dix-huit mois des maquettes virtuelles de villes réelles, dans lesquelles l'utilisateur pourra se déplacer et interagir avec des acteurs autonomes. Les problématiques scientifiques abordées seront 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. Un soin tout particulier sera porté au réalisme tant pour les environnements que pour les comportements.
  – Avec la société de jeux Infogram, le LETI et l'université de Rennes 2, nous allons travailler sur l'adaptation de nouvelles technologies de capteurs de mouvement dans un contexte d'acquisition de gestes sportifs en conditions réelles. Le domaine applicatif cible de cette étude est le jeu vidéo.
  – Un projet de plate-forme RNTL (PERF-RV : réalité virtuelle et bureau d'étude du futur) concerne la mise en commun de ressources matérielles et logicielle entre de nombreux partenaires académiques et industriels.

8.3 Actions financées par la Commission Européenne

debug : Module de nom « international-CEE » pour le projet Siames, section international
debug : corps du module.

TMR : PAVR

Projet PAVR (Training and Mobility Research) : *Platform for Animation and Virtual Reality*. L'objectif de ce projet est de créer un environnement européen expérimental d'animation et de réalité virtuelle. Nos partenaires sont l'Université de Las Baleares, l'University of Bath, Technical University of Vienna, l'EPFL, University of Geneva, le projet iMAGIS de l'Inria Rhône-Alpes, l'École des Mines de Nantes, University of Glasgow, University of Limburg. Jordi Regincos effectue, dans le cadre de ce projet TMR, une post-doc dans le projet.

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 plateforme ouverte et échelonnable pour fournir des services interactives de qualité à travers un réseau. Les différents objectifs visés sont :

- plateforme 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.
- plateforme de contenu : le projet étudie, définit et intègre une plateforme de haute performance modulaire et ouverte pour le codage et la transmissioon (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.
- plateforme de réseau : pour supporter efficacement les différentes applications prévues, le projet spécifie et intègre une plateforme 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 plateforme de laboratoire pour réaliser des expérimentations visant à évaluer les performances de la plateforme OPENISE compte tenu des objectifs du projet.

8.4 Relations bilatérales internationales

debug : Module de nom « international-bilat » pour le projet Siames, section international

8.4.1 Amérique (du nord et du sud)

Participants : Stéphane Donikian, Gwenola Thomas, Frédéric Devillers.

debug : corps du module.

Dans le cadre d'une collaboration soutenue par le programme NSF [4] / INRIA, des échanges ont été mis en place entre notre équipe et un laboratoire américain de l'Université de l'Iowa [5] sur le thème des scénarios pour la simulation de conduite. L'équipe américaine *Hank* travaille

4. National Science Foundation
5. http://www.cs.uiowa.edu/~hank/index.html

dans le domaine de la simulation de conduite (Iowa Driving Simulator) et utilise un modèle de représentation du comportement (HCSM) proche du nôtre (HPTS). L'activité de cette année a principalement porté sur la spécification de plusieurs niveaux de langages de scénarisation allant d'une version interprétée très légère à une version compilée très complète. Ces travaux ont donné lieu à une première publication commune [?].

Pete Willemsen (doctorant au sein de l'équipe Hank de l'université de l'Iowa) a effectué un séjour d'un mois au début de l'année 1999 dans notre projet, tandis que Stéphane Donikian a effectué un séjour de dix jours à l'Université d'Iowa en juin dernier.

9 Diffusion de résultats

9.1 Animation de la communauté scientifique

debug : Module de nom « international-commun » pour le projet Siames, section diffusion
debug : corps du module.

- Coordination du groupe de travail français sur l'animation et la simulation (G. Hégron). Ce groupe de travail a pour objectif de réunir la communauté scientifique française étudiant l'animation par ordinateur, d'organiser périodiquement des séminaires thématiques.
- Animation du groupe de travail français sur les techniques de rendu (co-responsable K. Bouatouch). Ce groupe de travail a les mêmes objectifs que celui sur l'animation : réunir et organiser des séminaires.
- Participation aux travaux de l'association française d'informatique graphique AFIG – secrétaire de l'association : S. Donikian.
- Participation au groupe de travail national sur la réalité virtuelle (GT-RV), dans le cadre du GDR ISIS (Communication Homme-Machine) : G. André assure la représentation de l'Irisa.

9.2 Enseignement universitaire

debug : Module de nom « diffusion-univ » pour le projet Siames, section diffusion
debug : corps du module.

- Responsabilité de la formation 3ème cycle en informatique de l'université de Rennes 1 (B. Arnaldi), responsabilité de la filière *Image* du DEA d'informatique.
- DEA d'informatique Ifsic : synthèse d'image et animation (K. Bouatouch, G. Hégron).
- DEA STIR et DEA d'informatique Ifsic : synthèse d'image (B. Arnaldi).
- DIIC LSI & INC, DESS ISA Ifsic : cours sur les interfaces homme-machine et la conception d'applications interactives (S. Donikian).
- DEA d'informatique : participation au cours sur le raisonnement spatial et temporel (S. Donikian).
- DIIC-INC : Ifsic, Rennes 1, (K. Bouatouch, Ð. Maisel, R. Cozot) option image.
- DESS ISA Ifsic : partie synthèse d'image du cours d'imagerie numérique (É Maisel).
- DESS CCI Ifsic : option informatique graphique((R. Cozot, É. Maisel).

- ENSTB Brest : cours sur la synthèse d'image avancée(K. Bouatouch).
- École des Mines de Sophia-Antipolis (ISIA), cours de synthèse d'image et d'animation (K. Bouatouch).
- MAÎTRISE D'INFORMATIQUE Ifsic : partie synthèse d'image de l'option images numériques (É Maisel).
- Magistère de mathématiques, université de Rennes 1 : cours sur la modélisation géométrique et la synthèse d'images (R. Cozot)
- DESS de mathématiques, université de Rennes 1 : cours sur la modélisation géométrique et la synthèse d'images (R. Cozot)
- Mastère Réalité Virtuelle de l'ENI de Brest : cours sur la coopération en univers 3D et sur la multimodalité (T. Duval)
- Module Interface de l'ENI de Brest : cours sur la conception, les modèles d'architecture et l'ergonomie des IHM (T. Duval)
- DESS ISA option Génie Logiciel, Ifsic : cours sur les interfaces homme-machine (lien avec les Design Patterns) (T. Duval).

10 Références bibliographiques

Articles et chapitres de livre

[1] E. Guillou, D. Meneveaux, E. Maisel, K. Bouatouch, « Using vanishing points for camera calibration and coarse 3D reconstruction from a single image », *The Visual Computer 16*, 7, 2000.

[2] D. Maurel, S. Donikian, *Intelligent Vehicle Technologies*, Arnold, 2000, ch. ACC Systems - Overview and examples.

[3] F. Multon, S. Menardais, B. Arnaldi, « Human motion coordination: a juggler as an example », *The Visual Computer 16*, 2000, To appear.

Communications à des congrès, colloques, etc.

[4] S. Carré, J. Deniel, E. Guillou, K. Bouatouch, « handling dynamic changes in hierarchical radiosity through interaction meshes », *in : Pacific Graphics'2000*, p. 40–51, 2000.

[5] F. Chapelle, G. Dumont, O. Chocron, « Prototypage virtuel de micro-endoscopes par algorithmes évolutionnaires », *in : Journées des Jeunes Chercheurs en Robotique*, jjcr13, juin 2000.

[6] S. Donikian, « Towards more realistic Autonomous Agents », *in : Workshop 7: "Achieving Human-Like Behavior in Interactive Animated Agents" of the International Conference on Autonomous Agents*, J. Rickel (éditeur), ACM, p. 23–29, Barcelone, Espagne, juin 2000.

[7] S. Donikian, « The use of Spatial Cognition for Behavioural Simulation of Autonomous Pedestrians and Car Drivers evolving in Virtual Cities », *in : International Conference on Spatial Cognition*, Pabst Science Publishers, Rome, Italie, décembre 2000.

[8] G. Dumont, F. Chapelle, O. Chocron, P. Bidaud, « Prototypage virtuel d'un micro-endoscope », *in : Journée thématique PRIMECA, Prototypage Virtuel : Mythe ou réalité*, Primeca, mars 2000.

[9] G. Dumont, F. Chapelle, « Simulation multi-physique pour la conception en micro-robotique », *in : Troisièmes journées du pôle micro-robotique*, juin 2000.

[10] R. DUMONT, K. BOUATOUCH, « Combining Hierarchical Radiosity and LODs », *in : IASTED, Computer Graphics and Imaging'2000*, 2000.

[11] J. Szewczyk, D. Sars, P. Bidaud, G. Dumont, « An active intelligent steering device for endoscopes », *in : proceedings of ISER2000 (International Symposium on Experimental Robotics)*, dec 2000.

[12] T. Duval, A. Chauffaut, J. Regincós, B. Arnaldi, « Intégration de modes d'interaction à l'aide de GASP », *in : Colloque "Multimodalité, 10 ans !"*, Grenoble, France, mai 2000.

[13] T. Duval, D. Margery, « Building Objects and Interactors for Collaborative Interactions with GASP », *in : Proceedings of the Third International Conference on Collaborative Virtual Environments (CVE'2000)*, ACM, San Francisco, USA, septembre 2000.

[14] T. Duval, D. Margery, « Using GASP for Collaborative Interactions within 3D Virtual Worlds », *in : Proceedings of the Second International Conference on Virtual Worlds (VW'2000)*, Springer LNCS/AI, volume 1834, p. 65–76, Paris, France, juillet 2000.

[15] T. Duval, F. Pennaneac'h, « Using the PAC-Amodeus Model and Design Patterns to Make Interactive an Existing Object-Oriented Kernel », *in : Proceedings of the TOOLS EUROPE 2000 Conference*, juin 2000.

[16] T. Duval, J. Regincós, A. Chauffaut, D. Margery, B. Arnaldi, « Interactions collectives locales en immersion dans des univers virtuels 3D avec GASP », *in : Actes de la conférence ERGO-IHM 2000*, Biarritz, France, octobre 2000.

[17] T. Duval, J. Regincós, D. Margery, B. Arnaldi, A. Chauffaut, « Una experiencia de interaccion colectiva en mundos virtuales 3D utilizando GASP », *in : Actes de CEIG'2000 (X Congreso Español de Informática Gráfica)*, Castelló, Espagne, juin 2000.

[18] G. Thomas, S. Donikian, « Modelling virtual cities dedicated to behavioural animation », *in : EUROGRAPHICS'2000*, M. Gross, F. Hopgood (éditeurs), *19:3*, Blackwell Publishers, Interlaken, Switzerland, août 2000.

[19] G. Thomas, S. Donikian, « Virtual Humans Animation in Informed Urban Environments », *in : Computer Animation*, IEEE Computer Society Press, p. 129–136, Philadelphia, PA, USA, May 2000.

Rapports de recherche et publications internes

[20] C. Gach, B. Arnaldi, A. Leroy, M. Ollivier, S. Donikian, « Delice : article de synthèse », *rapport de recherche*, Rapport de convention Predit, mai 2000.

[21] C. Gach, B. Arnaldi, A. Leroy, M. Ollivier, S. Donikian, « Delice : rapport final », *rapport de recherche*, Rapport de convention Predit, mars 2000.