

PARTAGE D'APPLICATIONS ET IMMERSION DANS UN MONDE 3D VIRTUEL HABITE

José Marques Soares, Patrick Horain, André Bideau

GET / INT / EPH - Intermedia

9 rue Charles Fourier, 91011 EVRY CEDEX France

{Jose_Marques.Soares,Patrick.Horain,Andre.Bideau}@int-evry.fr

Résumé :

Dans un travail collaboratif à distance, chaque utilisateur a une perception limitée des actions des autres participants. L'utilisation d'humanoïdes virtuels dans un environnement virtuel collaboratif permet d'améliorer la représentation des utilisateurs. Nous proposons une interface hybride 2D+3D consistant à immerger l'interface 2D d'une application partagée dans un monde virtuel habité et à représenter chacune des actions d'un utilisateur sur l'application 2D par une action de son avatar dans l'espace 3D.

MOTS-CLES : travail collaboratif, environnements virtuels

1. Introduction

Le travail collaboratif, lorsqu'il est réalisé à distance, limite les sensations d'un travail de groupe en "présentiel": la perception des interactions est amoindrie. Quelques environnements de travail collaboratif, tels que NetMeeting [MIC], utilisent la vidéoconférence pour limiter ce manque. Cependant, comme l'affirme Leung [LEU 01], dans ce type de système le sens de l'immersion est perdu parce que chaque participant voit les autres dans différentes fenêtres et ne sait pas directement qui agit sur l'application partagée.

Des environnements de réalité virtuelle permettent de représenter chaque utilisateur par un objet virtuel appelé avatar afin d'améliorer la perception du groupe. Dans ces environnements, les effets gênants occasionnés par l'absence du contact physique entre utilisateurs distants peuvent être minimisés au moyen de

canaux classiques de communication par le texte, la voix ou la vidéo. Toutefois, la communication naturelle entre des interlocuteurs concerne aussi l'activité gestuelle réalisée par des mouvements du corps et du visage. Cette communication non verbale est d'une part un instrument d'expression complémentaire de celui qui agit et d'autre part un élément qui augmente la capacité de compréhension de celui qui observe. Dans un environnement virtuel collaboratif, ce genre de communication peut être établi par des avatars humanoïdes articulés [VUI 99] et l'absence d'immersion peut alors être réduite par l'utilisation d'un espace 3D habité.

La plate-forme VReng [DAX] utilise VNC [ATT] pour proposer, dans un monde 3D, un tableau virtuel comme interface pour afficher et interagir avec des applications 2D. Malheureusement, dans VReng les actions de l'utilisateur sur l'application immergée ne sont pas représentées par des actions des avatars.

NetICE [ADV] permet le partage d'un tableau blanc dans son interface 3D et offre aux utilisateurs un ensemble d'expressions faciales et corporelles pour leurs avatars, mais ces animations sont déclenchées à partir d'un menu et dissociées des actions réalisées sur le tableau blanc.

Un autre problème est la mauvaise lisibilité des interfaces 2D lorsqu'elles sont immergées dans des environnements virtuels. Pour contourner ces limitations, NetICE permet d'interagir avec un tableau blanc en quittant temporairement le monde 3D pour présenter l'interface applicative 2D, au prix d'une perte de l'immersion et de la visualisation des avatars des autres participants.

Nous proposons de garder une vue permanente de l'interface de l'application 2D et de l'augmenter avec un monde virtuel 3D habité où les actions des utilisateurs sur l'application 2D sont reproduites par leurs avatars.

2. Une interface hybride 2D/3D

Cette interface est composée de deux espaces distincts : *l'espace applicatif* et *l'espace immersif collaboratif*. *L'espace applicatif* présente, sans dégradation de l'affichage, l'application 2D partagée sur laquelle les utilisateurs interagissent directement. *L'espace immersif collaboratif* est un monde 3D virtuel multi-utilisateur qui contient un *tableau virtuel* sur lequel est projeté *l'espace applicatif*. Des avatars humanoïdes, définis suivant la norme H-ANIM [HUM 01], représentent les utilisateurs qui participent au travail. Chaque interaction d'un utilisateur sur *l'espace applicatif* est représentée sur *l'espace immersif collaboratif* par un déplacement de son avatar vers le *tableau* sur lequel il reproduit les mouvements associés aux actions de l'utilisateur.

Les événements dans *l'espace applicatif* peuvent déclencher des animations prédéfinies ou calculées en temps réel.

Les animations prédéfinies sont :

- **Entrée dans l'espace collaboratif immersif.** Après la connexion d'un utilisateur au système, son avatar est positionné alternativement à gauche ou à droite du tableau. De même, l'avatar quitte l'environnement lors de la déconnexion de l'utilisateur.

- **Positionnement devant le tableau virtuel.** Un seul participant peut être actif dans le partage d'une application mono-utilisateur. Après la réquisition d'un utilisateur, son avatar est positionné devant le tableau lorsqu'il est libre. La libération du tableau par le participant actif est indiquée par le retour de son avatar à la position de repos.
- **Lever de bras de l'avatar.** Cette posture d'un avatar indique l'attente de l'utilisateur qui demande à interagir avec une application mono-utilisateur partagée.
- **Autres communications par geste.** Quelques animations prédéfinies peuvent être déclenchées par l'utilisateur à partir d'une liste.

Pour permettre aux participants observateurs d'avoir une perception visuelle sur le comportement du participant actif, l'avatar, qui se trouve devant le tableau, suit de la main la position des événements associés aux actions réalisées par l'utilisateur sur l'*espace applicatif*. Pour cela, les angles de rotation des articulations du bras sont calculés en temps réel par cinématique inverse.

3. Prototypes et Architectures

Pour valider l'interface décrite précédemment, deux prototypes ont été développés. Le premier utilise une application multi-utilisateur de type tableau blanc (Figure 1). Le deuxième permet de partager une application mono-utilisateur quelconque en le plongeant dans un espace virtuel multi-utilisateur (Figure 3).

Ces deux prototypes ont été construits sur une architecture client serveur écrite en Java. Un serveur Java contrôle l'état général du système, gère les connexions établies avec les clients et diffuse des informations qui permettent aux clients de réaliser dans le monde virtuel des actions telles que l'actualisation du tableau VRML et l'animation des avatars humanoïdes.

Le calcul des angles des articulations du bras de l'avatar est effectué par cinématique inverse en utilisant la bibliothèque IKAN [CEN][TOL 00]. Pour effectuer cela, le client récupère les paramètres de position de l'avatar dans le monde 3D et le point d'interaction sur la fenêtre de l'application 2D. A partir de ce dernier on pourra déterminer le point de contact que l'avatar devra toucher sur le tableau virtuel. Comme IKAN est écrit en C++, on a créé une bibliothèque dynamique (DLL) appelée en utilisant *Java Native Interface*.

Les caractéristiques particulières de chaque prototype sont présentées par la suite.

3.1. Le Tableau blanc

Dans le prototype du tableau blanc l'avatar touche les points extrêmes de l'objet dessiné afin de reproduire les actions du participant actif (Figure 1).

Pour effectuer cette animation, les données de l'objet dessiné par l'utilisateur sont extraites de la fenêtre de l'application 2D et envoyées avec les coordonnées de l'avatar actif vers le *serveur d'animation et de gestion du tableau blanc*. Celui-ci effectue les calculs de cinématique inverse et rediffuse vers les clients les

informations relatives à l'objet créé ainsi que les paramètres pour animer des avatars.

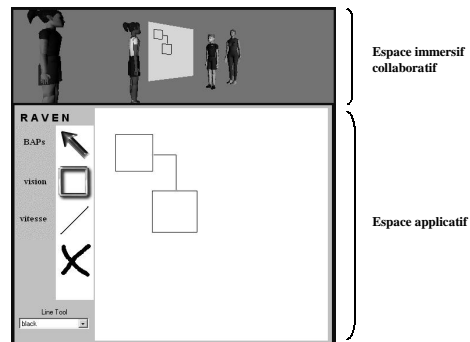


Figure 1 – Immersion du tableau blanc

Suivant une infrastructure technologique utilisée dans les mondes virtuels distribués tels que DeepMatrix [REI 98] ou VNet [WHI], on a construit ce premier prototype sur une architecture client serveur *Web*. Une application de tableau blanc développée par Huges [HUG 97] a été intégrée dans notre interface 3D. Un fichier VRML et un applet Java sont stockés dans un serveur HTTP et chargés par le client à travers un navigateur Web. Le monde virtuel est affiché dans le navigateur Web au moyen d'un navigateur VRML comme Cortona [PAR] ou Blaxxun Contact [BLA]. L'interface EAI (*External Authoring Interface*) [VRM 02] permet d'interagir avec le monde VRML et, ainsi, d'animer des objets humanoïdes. La Figure 2 montre l'architecture utilisée.

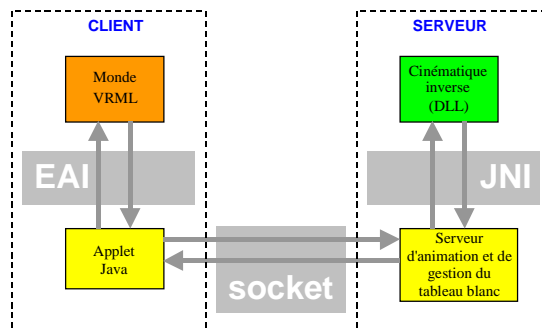


Figure 2 – Architecture 1 : tableau blanc

Le serveur d'animation garde une liste des objets créés sur le tableau. Pour chaque objet dessiné dans l'espace applicatif, il est créé un objet VRML au moyen de l'interface EAI dans l'espace immersif collaboratif avec position et taille proportionnelles. Dans ce prototype il n'y a donc pas d'images introduites dans le monde 3D.

Pour permettre d'immerger n'importe quelle application 2D dans un monde 3D selon une approche plus flexible que celle de manipuler un objet VRML à chaque action, on a évolué vers un deuxième prototype présente ci-après.

3.2. Le client VNC

Dans ce deuxième prototype, les interactions avec le monde virtuel sont réalisées au moyen de Java3D et Xj3D [Xj3D]. Ce dernier permet d'obtenir les coordonnées du corps de l'avatar et l'animer au travers des rotations et translations de ses articulations.

L'application 2D immergée dans le monde 3D est un client VNC (*Virtual Network Computing*) [ATT] qui permet de voir l'interface graphique d'un ordinateur distante. Pour simuler les actions du participant actif l'avatar suit de la main la position de la souris sur l'image affichée dans *l'espace immersif collaboratif* (Figure 3).

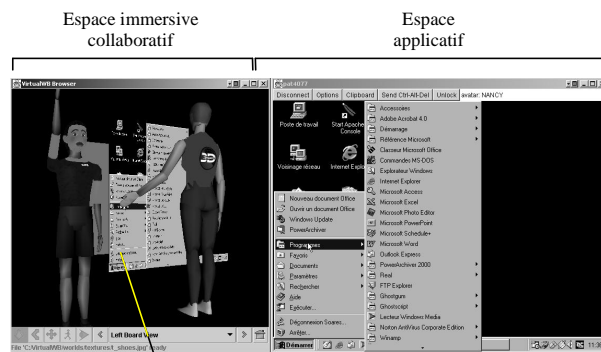


Tableau blanc virtuel

Figure 3 – Immersion d'un client VNC

La communication entre le client et le serveur in VNC est basée sur le protocole RFB [RIC 98] au travers d'une connexion TCP. A chaque interaction d'un client avec l'application partagée, le serveur VNC diffuse des parties modifiées de l'interface graphique distante. Malheureusement, l'identité du client qui a agi et les coordonnées de l'événement ne sont pas envoyées avec les fragments de l'image, ce qui nous est nécessaire pour représenter ces actions avec les avatars dans le monde 3D. Pour régler ce problème, un client VNC *open source* écrite en Java est étendu pour envoyer les coordonnées de l'action de l'utilisateur vers un serveur d'animation écrit en Java qui, à son tour, les diffuse vers les autres clients. Cette communication est établie par datagrammes UDP afin de minimiser le retard entre les animations et l'actualisation du tableau dans le monde 3D.

Notre serveur réalise aussi le contrôle d'accès à l'application pour prévenir des interactions simultanées des participants. Lors du changement d'utilisateur, le système diffuse cette information à chaque client qui positionne l'avatar

correspondant. Notre client est une application Java (et non une applet) et l'architecture 2 intègre aussi un serveur VNC, comme le montre la Figure 4.

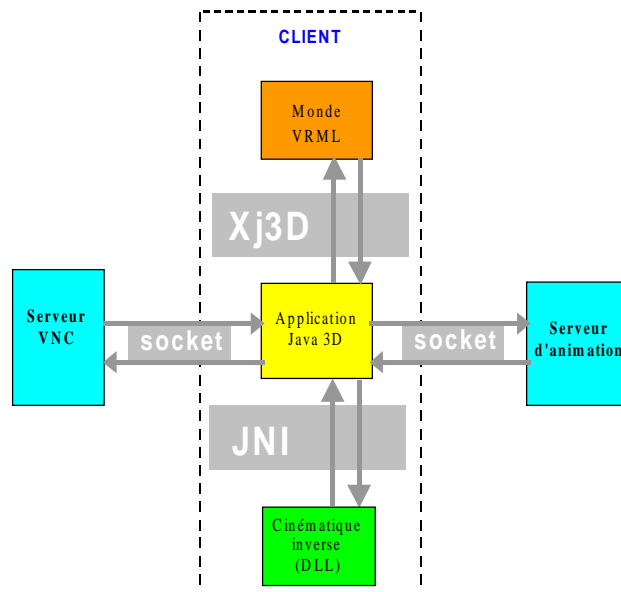


Figure 4 – Architecture 2: client VNC

3.3. Comparaison entre les architectures 1 et 2

L'architecture 1 offre les mêmes avantages que les applications Web typiques: les modifications du code client peuvent être rapidement et uniformément diffusées et il est facile de développer des clients indépendants du matériel avec l'utilisation de navigateurs Web et applets Java.

Cependant, cette approche présente deux limitations. La première est liée aux contraintes de sécurité des applets non signées. Ces contraintes empêchent d'établir des connexions par sockets avec une machine autre que leur serveur HTTP et ne permettent pas l'accès aux ressources locales du client. Dans l'architecture 2 cela devient critique puisque l'immersion de l'application 2D utilise aussi un serveur VNC. Cette limitation a été réglée en mettant en œuvre une application Java cliente plutôt qu'une applet, ce qui permet l'accès à un serveur VNC quelle que soit son adresse dans l'Internet.

L'application java cliente peut aussi accéder directement à la bibliothèque dynamique pour effectuer les calculs de cinématique inverse, ce qui était réalisé par le serveur dans l'architecture 1. Ce déplacement réduit les éventuels retards du serveur pour la diffusion des données qui animent les avatars.

Une autre limitation de l'architecture 1 apparaît lors du changement de l'image affichée sur le tableau virtuel dans l'espace 3D, ce qui est réalisé fréquemment par le client VNC. L'affichage des images stockées dans la mémoire du client est peu

performant en utilisant l'interface EAI des navigateurs VRML. Pour régler ce problème dans l'architecture 2, on utilise la représentation Java3D du tableau VRML, ce qui permet une actualisation plus efficace des images affichées.

4. Conclusion

Afin d'améliorer la perception des interactions des utilisateurs qui partagent une application 2D, nous avons réalisé une plate-forme dans laquelle les actions des utilisateurs sont reproduites dans un monde virtuel 3D.

Nous proposons d'augmenter l'espace de partage usuel avec une *espace immersif collaboratif*. Avec cette nouvelle interface, il est possible :

- de voir qui est présent pendant le travail partagé;
- de percevoir la sortie ou l'entrée de nouveaux utilisateurs;
- d'être conscient de qui agit sur l'application;
- de manifester l'envie d'interagir et de voir qui souhaite interagir.

D'autres moyens que *l'espace 3D collaboratif* proposé pourraient aussi être employés afin de fournir ce même ensemble d'informations, comme le fait NetMeeting par vidéo et plusieurs outils type IRC par texte. Toutefois un monde 3D habité permet une interface visuelle plus intuitive que, par exemple, un message de texte ou un signal sonore.

L'idée d'utiliser, dans une même interface, deux espaces visuels séparés peut représenter un nouveau paradigme. Dans une solution hybride (2D+3D) on peut ajouter le sens d'immersion manquant au travail collaboratif à distance sans perdre les aspects ergonomiques bien élaborés des interfaces 2D. Un utilisateur peut bénéficier des avantages de la 3D, traduites par la possibilité d'observer l'action des autres participants aux travaux partagés, sans en subir les inconvénients liés à la navigation 3D et au changement de mode d'interaction pour accéder à l'interface 2D de l'appliquatif.

Notre objectif actuel est d'intégrer une interface d'acquisition de gestes par vision artificielle [HOR 02]. La restitution distante des gestes acquis peut améliorer la communication non verbale et ainsi augmenter le sens d'immersion entre les participants d'un travail collaboratif. En particulier, les actions d'un participant qui travaille sur un tableau réel augmenté avec vidéoprojecteur et capteur MIMIO [VIR] peuvent actuellement être restituées par son avatar dans le monde virtuel partagé. Nous étudions la possibilité d'acquérir par caméra en temps réel l'ensemble des gestes de l'utilisateur (pas seulement ses actions sur le tableau) et de les restituer dans le monde 3D.

Remerciement. Ce travail a été réalisé avec le soutien financier partiel du gouvernement brésilien au travers du projet CAPES/COFECUB n° 266/99-I.

5. Bibliographie

- [MIC] MICROSOFT CORPORATION, "NetMeeting Home",
<http://www.microsoft.com/windows/netmeeting>
- [LEU 01] W. H. LEUNG & T. CHEN, "Creating a Multiuser 3-D Virtual Environment",
IEEE Signal Processing Magazine, 18, pp. 9-16, May 2001
- [VUI 99] A. VUILLEME-GUYE; T. K. CAPIN; I. PANDZIC; N. THALMAN; D.
THALMAN, "Nonverbal Communication Interface for Collaborative Virtual Environments",
Virtual Reality J., vol.4, pp. 49-59, 1999
- [DAX] P. DAX, "VREng Virtual Reality Engine", <http://www.infres.enst.fr/net/vreng>
- [ATT] AT&T LABORATORIES, "VNC - Virtual Network Computing",
<http://www.uk.research.att.com/vnc>
- [ADV] ADVANCES MULTIMEDIA PROCESSING LAB, CARNEGIE MELLON, "Projet
NetICE", <http://amp.ece.cmu.edu/projects/NetICE/>
- [HUM 01] HUMANOID ANIMATION WORKING GROUP, "H-ANIM 2001 specification",
<http://www.h-anim.org/Specifications/H-Anim2001/>
- [CEN] CENTER FOR HUMAN MODELLING AND SIMULATION, UNIVERSITY OF
PENNSYLVANIA, "IKAN: Inverse Kinematics using ANalytical Methods",
<http://hms.upenn.edu/software/ik/ik.html>
- [TOL 00] D. TOLANI; A. GOSWAMI; N. BADLER, "Real-time inverse kinematics
techniques for anthropomorphic limbs", *Graphical Models* 62 (5), Sept. 2000, pp. 353-388
- [REI 98] G. REITMAYR; S. CARROLL; A. REITEMEYER, "DeepMatrix – An open
technology based virtual environment system", Octobre 1998,
<http://www.geometrek.com/developers/whitepapers.html>
- [WHI] S. WHITE, "VNet: Multi-User VRML System",
<http://www.csclub.uwaterloo.ca/u/sfwhite/vnet>
- [HUG 97] M. HUGES, "Networking our whiteboard with Java 1.1", Décembre 1997
<http://www.javaworld.com/javaworld/jw-12-1997/jw-12-step.html>
- [PAR] PARALLEL GRAPHICS, "Cortona vrml client",
<http://www.parallelgraphics.com/products/cortona/>
- [BLA] BLAXXUN INTERACTIVE, "Blaxxun contact - Multimedia communications client",
<http://developer.blaxxun.com>
- [VRM 02] "The Virtual Reality Modeling Language (External Authoring Interface) ISO/IEC
FDIS 14772-2:2002", <http://www.web3d.org/technicalinfo/specifications/eai/index.html>
- [Xj3D] SOURCE CODE MANAGEMENT TASK GROUP, Web3D Consortium, "Xj3D
Open Source VRML/X3D Toolkit", <http://www.web3d.org/TaskGroups/source/xj3d.html>
- [RIC 98] T. RICHARDSON; K. R. WOOD, "The RFB Protocol, Version 3.3", Janvier 1998,
<http://www.uk.research.att.com/vnc/protocol.html>
- [HOR 02] P. HORAIN, M. BOMB, "Acquisition du geste humain 3D par vision
monoscopique", *Actes des 8^{èmes} journées d'études et d'échanges "Compression et
Représentation des Signaux Audiovisuels" (CORESA'03)*, Lyon, 16-17 janvier 2003, pp. 269-
272.
- [VIR] VIRTUAL INK CORPORATION, "MIMIO products",
<http://www.mimio.com/index.shtml>