
Analyse de techniques de coopération en environnements virtuels 3D

Laurent Aguerreche* — **Thierry Duval**** — **Bruno Arnaldi***

* INSA de Rennes – IRISA – projet Bunraku

** Université de Rennes 1 – IRISA – projet Bunraku

IRISA

UMR 6074

Campus de Beaulieu

35042 Rennes Cedex

{laurent.aguerreche, thierry.duval, bruno.arnaldi}@irisa.fr

RÉSUMÉ. La réalité virtuelle a permis le développement de nombreuses méthodes d'interaction entre des utilisateurs humains et des objets dans des espaces en 3D. Aujourd'hui, de nouveaux développements apparaissent pour permettre des interactions coopératives, c'est-à-dire permettant à plusieurs utilisateurs d'agir simultanément sur des objets partagés. Cette coopération nécessite de nouvelles métaphores qui doivent d'une part permettre à plusieurs outils d'accéder en même temps à un objet, et d'autre part offrir de nouveaux retours d'informations vers les utilisateurs expliquant ces actions multiples. Enfin, au niveau des plates-formes logicielles, des problèmes techniques apparaissent tels que la synchronisation et le maintien de cohérence entre sites.

ABSTRACT. Virtual reality has lead to development of many methods for interacting between real humans and virtual objects in 3D spaces. Nowadays, some new developments are made for cooperative interactions which means that several persons can act simultaneously on shared objects. A cooperation requires new metaphors to make several persons acting on same objects possible and also to have some new feedbacks to users to explain shared actions. Finally, some new technical issues in platforms appear like synchronization and consistency between sites.

MOTS-CLÉS : État de l'art, interaction, coopération, collaboration, réalité virtuelle, environnements virtuels collaboratifs.

KEYWORDS: State of the art, interaction, cooperation, collaboration, virtual reality, collaborative virtual environments.

1. Introduction

La réalité virtuelle trouve aujourd'hui des applications dans de nombreux domaines. Pour le grand public, elle peut par exemple être un moyen de s'immerger dans la reconstitution d'une cité antique. Du côté des applications professionnelles, des industriels peuvent concevoir des maquettes à distance, faire de la revue de projet, éviter des déplacements de leurs employés : la réalité virtuelle sert de moyen de communication avancé autorisant des interactions distantes avec des objets d'une façon qui se veut naturelle. Certaines de ces tâches peuvent être coopératives. Ainsi, lors d'une revue de projet, un présentateur peut être en train de modifier des points de vue autour d'un objet à l'étude devant un groupe de spectateurs, tandis qu'une autre personne agit sur différents éléments de l'objet au gré du discours tenu par le présentateur. Un troisième opérateur peut, pendant la même présentation, enregistrer des séquences vidéos ou prendre des photos pour fournir par la suite une sorte de résumé de la présentation qui a eu lieu. L'interaction avec des objets du monde réel passe souvent par le besoin de donner une impression de réalisme, voire d'immersion, à l'utilisateur. Malheureusement, cette impression est difficile à fournir. En premier lieu, l'humain doit se projeter dans l'environnement virtuel, mais la proprioception est souvent dégradée par le nombre souvent limité de types de retours sensoriels vers l'utilisateur : visuels, sonores et haptiques, ainsi que par leur qualité qui dépend du matériel à disposition. Ensuite, il est difficile de superposer correctement un objet réel à l'environnement virtuel (EV) pour donner une impression de plongement du réel dans le virtuel : c'est le problème de la colocalisation. Enfin, les utilisateurs souhaitent interagir comme dans la réalité en ayant la sensation de saisir et de manipuler des objets. De nombreux efforts ont été faits dans ce sens, tant à travers la qualité du matériel que des représentations d'actions effectuées par des utilisateurs. Malheureusement, ces efforts ont souvent été restreints au cas d'un seul utilisateur interagissant avec un objet. Les travaux concernant la coopération sont encore peu nombreux et montrent que de nouveaux développements sont nécessaires, tant sémantiques que techniques. Il y a donc besoin de fournir aux utilisateurs des représentations (pas seulement visuelles) de leurs actions simultanées et des résultats de ces actions, et d'améliorer les services techniques offerts par les plates-formes logicielles.

Par conséquent, cet état de l'art présente tout d'abord des métaphores d'interactions pour un seul utilisateur, ainsi que la perception d'un utilisateur. La plupart de ces métaphores ne peuvent être employées directement dans un environnement coopératif parce que la base logicielle n'est pas prévue pour gérer des accès concurrents à un objet, parce qu'un utilisateur n'a aucun moyen de savoir qu'il est train d'interagir simultanément sur un objet avec quelqu'un d'autre, parce qu'il est rarement prévu de laisser des traces de « qui a fait quoi » lors d'une interaction. Certaines métaphores d'interaction peuvent cependant être rapidement rendues collaboratives parce qu'elles ne requièrent que l'ajout de retours d'informations aux utilisateurs. Cet état de l'art traite donc de la coopération selon un plan similaire à celui des interactions mono-utilisateur, et il aborde enfin les difficultés liées à la coopération d'un point de vue technique au travers des plates-formes logicielles.

2. Réalité virtuelle et interactions

Les définitions concernant la réalité virtuelle sont nombreuses et peuvent être très détaillées comme celle présente dans le traité de la réalité virtuelle (Fuchs *et al.*, 2006). Nous choisissons ici une définition compacte qui est utilisée dans le cadre du projet ANR/RNTL Part@ge¹ : « la *réalité virtuelle* peut être définie comme un ensemble de techniques matérielles et logicielles destinées à permettre à un ou plusieurs utilisateurs d'interagir de la façon la plus naturelle possible avec des données numériques ressenties par le biais de canaux sensoriels ».

Les interactions entre des utilisateurs humains et des objets d'un monde virtuel se font au travers de techniques ou de métaphores d'interaction. Selon l'état de l'art du projet RNTL Perf-RV : « une technique, ou paradigme, d'interaction désigne une méthode, une stratégie, un scénario d'utilisation, permettant à l'utilisateur d'accomplir une tâche précise dans l'univers virtuel : naviguer, sélectionner des objets, manipuler des objets, contrôler l'application » ; « une métaphore d'interaction est utilisée par les fonctions de transfert. Une métaphore est la transposition d'un objet ou d'un concept réel utilisé pour faciliter l'interaction ».

Dans un EV, un interacteur effectue plusieurs types de tâches : navigation, sélection/manipulation d'objets, contrôle d'application (Hand, 1997). Nous choisissons de nous intéresser à la sélection/manipulation d'objets. L'interacteur interagit avec les éléments d'un EV par le biais de techniques d'interaction. Par ailleurs, il doit également recevoir des informations sur le résultat de ses actions et de celles des autres interacteurs dans le cas d'un environnement où plusieurs utilisateurs travaillent de concert. La première partie de la section suivante concerne les techniques d'interaction et est suivie d'une partie traitant de la perception du potentiel interactif des objets.

2.1. Techniques de sélection/manipulation

Nous allons présenter un panorama des principales métaphores d'interaction rencontrées dans les EV. Selon (Poupyrev *et al.*, 1998), elles peuvent être classées en deux grands groupes : d'un côté, les interactions *exocentriques*, ou encore du point de vue de Dieu², et de l'autre les interactions *égocentriques*.

2.1.1. Interactions exocentriques

Les interactions exocentriques placent un interacteur, c'est-à-dire une personne qui interagit, dans une position extérieure à l'environnement virtuel de sorte que l'environnement puisse être perçu dans son ensemble. Dans cette situation, l'interacteur n'est plus forcément contraint par la gravité ou encore sa taille vis-à-vis des éléments de l'EV.

1. Site du projet Part@ge : <http://www.rntl-partage.fr/>.

2. « God's eye viewpoint ».

Dans la technique dite du monde en miniature (WIM³, (Stoakley *et al.*, 1995)), l'environnement est représenté en miniature et l'utilisateur peut ainsi bénéficier d'une vue d'ensemble pour voir, par exemple, où se situe un objet ou la représentation d'un utilisateur, mais peut également interagir avec un objet dans le monde miniature et voir les résultats dans le monde de grande taille et *vice versa* (cf. figure 2).

(Mine *et al.*, 1997) proposent de diminuer la taille d'un monde ou d'augmenter la taille d'un utilisateur, en fonction de la position de cet utilisateur et de ses mains. Lorsque l'utilisateur relâche l'objet, le monde reprend sa taille initiale (cf. figure 1).

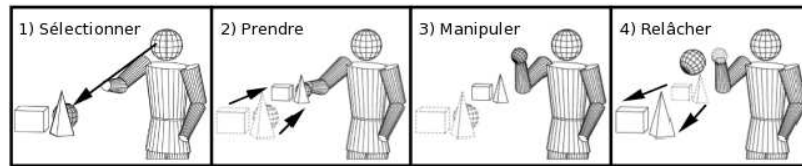


Figure 1. Principe de fonctionnement de la métaphore de redimensionnement automatique du monde appelée « Scaled-World Grab » (Mine *et al.*, 1997)

Les « poupées vaudou⁴ » (Pierce *et al.*, 1999) fournissent un moyen de manipuler des objets par des représentations miniatures. La main non-dominante⁵ détermine le contexte de la manipulation tandis que l'autre main permet à l'utilisateur de manipuler un objet par rapport à ce contexte. Par exemple, un utilisateur peut déplacer avec sa main dominante le téléphone qui est sur le bureau tenu par sa main non-dominante. Lorsqu'il en a fini avec le bureau, il peut le relâcher ce qui le fait retourner à sa position (avec les modifications éventuelles apportées, ici le téléphone déplacé).

Le positionnement global de l'interacteur est intéressant par le point de vue qu'il offre à un interacteur mais se révèle peu compatible avec un environnement partagé par plusieurs utilisateurs. En effet, un acteur sur le monde en miniature est généralement placé à l'extérieur du monde ce qui le rend invisible pour des utilisateurs présents à l'intérieur. Dans le cas où l'interacteur devient très grand, il peut également devenir invisible pour un interacteur qui se situerait « entre les jambes » du géant. Par ailleurs, les actions d'un géant pourraient avoir une force décuplée (mouvements très amples) par rapport à celle d'un interacteur de taille normale. Si l'on préférerait faire varier la taille du monde, les ordres de variations pourraient être contradictoires entre plusieurs interacteurs. Par ailleurs, il n'est pas forcément évident de se rendre compte de la variation de la taille du monde. En ce qui concerne les « Voodoo dolls », seul le résultat d'une manipulation est visible, l'outil n'est même pas intégré à l'environnement. En effet, un utilisateur manipule une poupée dans ses mains et fait venir devant lui un objet, en superposition au monde. Il serait néanmoins possible de colorer d'une façon

3. « World In Miniature ».

4. « Voodoo dolls ».

5. Pour un droitier, la main non-dominante est la main gauche.

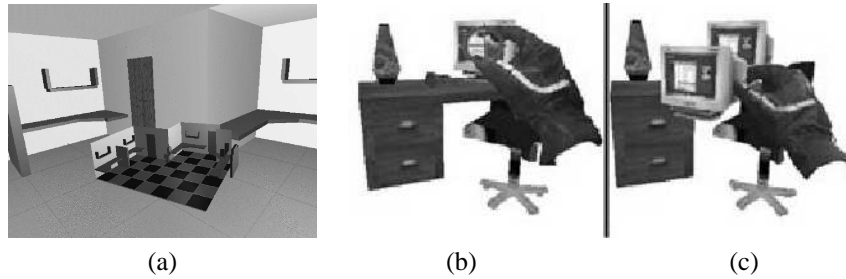


Figure 2. (a) Illustration d'un monde en miniature (au centre de l'image) (Stoakley et al., 1995). (b) Illustration du fonctionnement des « Voodoo dolls » : l'utilisateur désigne un objet grâce à une cible qui apparaît au bout de ses doigts lorsqu'il les écarte et la sélectionne en les pinçant. (c) L'utilisateur tient un objet au bout de ses doigts (Pierce et al., 1999)

particulière un objet lorsqu'il est manipulé par une « Voodoo doll », mais comment montrer que plusieurs poupées manipulent un objet et surtout quelles propriétés ? Le positionnement détaché du monde présente des difficultés pour passer des techniques exocentriques d'un environnement mono-utilisateur à celles d'un environnement partagé.

2.1.2. Interactions égocentriques

Les interactions égocentriques placent l'interacteur à l'intérieur du monde virtuel. Selon (Poupyrev *et al.*, 1998), elles peuvent être classées en deux catégories : celles utilisant une main ou l'extension du bras de l'utilisateur, et celles utilisant un pointeur.

2.1.2.1. Main virtuelle

La méthode la plus naturelle consiste à afficher une main ressemblant à une main humaine. Sa position ou encore le mouvement de ses doigts dépendent alors de l'interface matérielle d'interaction employée par l'utilisateur comme, par exemple, un gant de données. Avec cette technique, l'utilisateur a une meilleure sensation d'immersion si la main est finement représentée parce qu'il aura l'impression que sa propre main est affichée. Un utilisateur désigne un objet en plaçant sa main au-dessus et le sélectionne, par exemple, en la refermant. Les mouvements de la main de l'utilisateur peuvent ensuite être directement imprimés sur ceux de la main virtuelle, ou être accélérés/amplifiés en fonction de l'amplitude des mouvements (Mine, 1995) pour atteindre des zones plus rapidement. Malheureusement, cette métaphore n'a qu'un champ d'action très limité. En effet, un utilisateur ne peut sélectionner un objet que s'il est à portée de sa main, sinon, il doit se déplacer, ce qui peut l'amener à abandonner ses actions en cours et le ralentir dans ses travaux. Pour résoudre ce problème, les articles (Poupyrev *et al.*, 1996a), (Poupyrev *et al.*, 1996b) proposent la métaphore *Go-Go*. Le principe repose sur une extension linéaire du bras virtuel lorsque la main de l'utilisateur est

proche de son corps, puis en une extension exponentielle à partir d'un certain éloignement. Par conséquent, l'utilisateur peut manipuler précisément un objet situé près de lui tandis qu'un déplacement lointain se fait par des grands mouvements. Néanmoins, cette technique limite les déplacements en profondeur de la main virtuelle à la longueur du bras de l'utilisateur. Pour cette raison, (Bowman *et al.*, 1997) ont proposé deux nouvelles versions du Go-Go autorisant une extension infinie : *i*) le « *stretch Go-Go* » où l'espace entourant l'utilisateur est divisé en trois couches concentriques, la couche la plus en avant faisant allonger le bras et celle la plus en arrière le faisant raccourcir ; *ii*) l'« *indirect Go-Go* » où l'allongement du bras est commandé par les boutons d'une souris 3D. Cependant, un bras de l'utilisateur ne peut s'allonger, ainsi la colocalisation⁶ du bras de l'utilisateur n'est plus possible.

2.1.2.2. Pointeur virtuel

En bureautique, un utilisateur utilise généralement une souris 2D pour déplacer un curseur 2D à l'écran. Dans un environnement virtuel en 3D, le curseur 2D précédent se déplace à la « surface » de l'écran et un objet n'est alors sélectionnable que s'il est projeté sur cette surface (Mine, 1995) (cf. figure 3). Pour pouvoir désigner des objets occultés ou lointains, il faut un curseur 3D pouvant se déplacer dans tout l'environnement 3D. Un curseur est donc un moyen abstrait de sélection. Par ailleurs, il permet de lever naturellement des contraintes : par exemple le déplacement d'un nombre élevé d'objets à la fois contrairement à une main qui sera limitée à un ou deux objets si elle respecte les contraintes du monde réel. Malheureusement, l'utilisation d'un pointeur n'est pas forcément très naturelle si elle est pratiquée comme décrite ci-avant. À la place, il serait souhaitable de pouvoir désigner un objet en le pointant avec une main (Mine, 1995) (cf. figure 3). Dans ce cas, l'utilisateur pourrait, par exemple, sélectionner effectivement un objet pointé en appuyant sur un bouton d'un dispositif matériel.

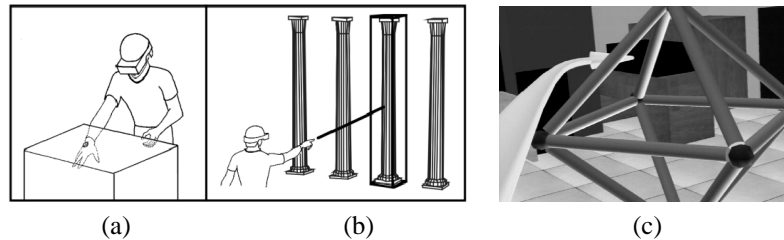


Figure 3. (a) Sélection d'un objet en approchant suffisamment un curseur 2D d'un objet projeté à la surface de l'écran. (b) Sélection d'un objet par pointage avec une main (Mine, 1995). (c) Illustration d'un rayon courbé pointant vers un objet partiellement occulté par la structure au premier plan (Owal et al., 2003)

6. La colocalisation du bras de l'utilisateur consiste à le superposer visuellement à sa représentation dans l'EV. L'utilisateur ressent que son bras réel est plongé dans l'EV.

Un rayon, quant à lui, permet de sélectionner assez facilement un objet, même distant, mais il présente l'inconvénient que, une fois sélectionné, un objet reste à l'extrémité du rayon. Il devient alors très difficile de lui appliquer des rotations (à l'exception de celles autour de l'axe induit par le rayon lui-même). De plus, comme ce rayon a une longueur fixe, le positionnement en profondeur de l'objet dépend des possibilités d'allongement du bras de l'utilisateur. Néanmoins, cette contrainte trouve une solution par la méthode du « ray-casting with reeling » (Bowman *et al.*, 1997), c'est-à-dire par la possibilité de faire glisser l'objet sélectionné le long du rayon *via* des boutons en avant/en arrière d'un dispositif matériel. Notons que le rayon virtuel tel que présenté jusqu'à présent ne permet pas de sélectionner un objet entièrement ou partiellement occulté par un autre. Constatant que de nombreuses personnes ont tendance à plier leurs bras ou leurs poignets pour désigner un objet placé derrière un autre, (Owal *et al.*, 2003) ont suggéré l'utilisation de rayons coudés (cf. figure 3) où l'orientation de la main de l'utilisateur désigne la courbure du rayon et l'éloignement de celle-ci par rapport au corps, la longueur du rayon.

Dans les illustrations données précédemment, la facilité de sélection de ces techniques a été mise en avant ainsi que les problèmes de manipulation qu'elles posent. Par conséquent, ces techniques savent surtout se montrer pratiques pour sélectionner des objets, mêmes s'ils sont (partiellement) occultés.

2.1.2.3. Combinaison de métaphores

Parmi les métaphores d'interactions égocentriques, l'une d'elles utilise à la fois un pointeur virtuel et une main virtuelle : il s'agit de la méthode HOMER⁷ (Bowman *et al.*, 1997). Cette technique utilise un rayon pour sélectionner un objet, mais ensuite une main virtuelle va venir s'attacher à l'objet pour permettre de modifier sa position selon trois degrés de liberté. Cette technique regroupe les qualités des techniques de sélection par rayon virtuel et des techniques de manipulation par une main virtuelle.

2.1.3. Bilan intermédiaire

Un panorama plus détaillé des techniques d'interaction à un utilisateur est disponible dans (Bowman *et al.*, 2005). Les métaphores exocentriques peuvent être délicates à employer en coopération, contrairement aux métaphores égocentriques. Dans une situation de coopération, ces dernières peuvent permettre à un utilisateur de mieux comprendre ce qu'un autre utilisateur est en train de faire au cours d'une interaction. Par ailleurs, les métaphores égocentriques fournissent à l'utilisateur une meilleure sensation d'immersion et facilitent l'interaction dans un EV.

2.2. Perception du potentiel interactif des objets

Pour pouvoir interagir avec un objet, tout interacteur a besoin de connaître les possibilités interactives qu'offre cet objet : *i*) les propriétés modifiables de l'objet (posi-

7. *Hand-centered object manipulation extending ray-casting.*

tion, couleur, etc.) ; *ii*) les zones interactives : où le saisir, comment, à combien de personnes, etc. ; *iii*) les limites et contraintes d'interaction : impossibilité de faire traverser un mur à un objet, mouvements contraints par rapport à un autre objet, etc. Ces informations peuvent utiliser des retours haptiques, visuels ou sonores qui se rajoutent à l'EV. À l'image de la réalité augmentée qui synthétise puis ajoute des éléments à un monde réel, il est alors possible de parler de *virtualité augmentée*.

2.2.1. Propriétés modifiables

L'une des premières interactions qu'un objet peut offrir à un interacteur est celle de son propre déplacement ou des rotations qu'il peut subir. L'un des premiers systèmes à afficher le fait qu'un objet puisse être translaté ou tourné a été Inventor de SGI⁸. Ce kit de développement permettait de construire facilement des mondes en trois dimensions grâce à l'utilisation et l'extension d'objets fournis. Inventor affichait un curseur dont la forme indiquait les translations, ou rotations, possibles. Cette idée a été souvent exploitée, par exemple par (Duval *et al.*, 2004). Cependant, ce type d'information est rarement affiché dans les environnements virtuels, notamment parce qu'il surcharge le monde visuel. De plus, l'utilisateur obtient souvent implicitement les possibilités de translation d'un objet en essayant directement de le déplacer.

Enfin, outre l'affichage d'un symbole représentant les degrés de liberté d'un objet, il est également possible de donner ces informations implicitement par l'affichage de l'outil strictement réservé à une translation ou à une rotation (Cutler *et al.*, 1997) dès que l'une ou l'autre de ces opérations devient accessible.

2.2.2. Objets et parties interactives

Pour pouvoir interagir avec un objet, un interacteur doit savoir si l'objet est interactif et en connaître les parties interactives. Par exemple, une porte peut être poussée pour l'ouvrir et sa poignée peut être tournée. Néanmoins, une porte ne pourra être poussée que si la poignée a été actionnée. Il faut donc fournir à l'interacteur un moyen de savoir que la poignée peut être tournée et la porte poussée. Cependant, il faut aussi montrer à l'utilisateur que la porte ne peut être poussée que si la poignée a été tournée.

La représentation des parties interactives d'un objet sert à fournir à un utilisateur humain les informations nécessaires à une interaction avec un objet, mais elle peut aussi permettre d'aboutir à une perception « logicielle » rendant des agents autonomes capables d'interagir avec ces objets. Le concept de *smart object* (Kallmann *et al.*, 1998), (Kallmann, 2001) permet de décrire les possibilités interactives qu'offrent des objets. En outre, ces objets possèdent des comportements qui décrivent la position qu'un agent doit avoir avant de démarrer une interaction, puis lui dictent le comportement qu'il doit suivre ensuite. (Badawi *et al.*, 2004) font remarquer que le fait de placer toutes les informations d'interactions, ainsi que des actions, dans un objet transforme l'agent autonome en une sorte de coquille vide. Pour ne pas figer autant le comporte-

8. <http://oss.sgi.com/projects/inventor/>, aujourd'hui développé par la société Mercury Computer (<http://www.mc.com/>).

ment des agents autonomes, ils proposent le concept d'*objets synoptiques* qui offrent à un agent autonome un sommaire des interactions qu'ils fournissent, ainsi que la connaissance de leur état et des actions disponibles à partir de celui-ci. Néanmoins, ces objets prévoient rarement le cas d'interactions partagées par plusieurs agents, voire interacteurs humains.

2.2.3. Limites et contraintes des interactions

De la même façon que dans le monde réel, un objet peut être muni de contraintes telles que le fait de ne pas pouvoir passer au travers d'un mur ou d'un autre objet. Les limites, ou contraintes, sur des mouvements peuvent donc empêcher certaines actions dans un EV afin de reproduire des contraintes du monde réel. Cependant, la limitation des mouvements due aux contraintes peut aussi rendre des actions plus précises ou plus faciles à effectuer en servant de guide géométrique : par exemple, des tremblements de l'utilisateur seront moins répercutés si la pièce qu'il manipule est contrainte à des translations le long d'un plan.

L'assemblage ou le positionnement relatif de pièces peut être délicat pour au moins trois raisons : *i*) soit il n'y a pas de système de détection de collisions et donc les pièces se traversent les unes les autres, *ii*) soit pour une raison ou une autre l'utilisateur ne peut pas placer son point de vue à sa guise et voit mal les placements des pièces qu'il manipule, *iii*) soit les interfaces matérielles qu'il utilise n'offrent pas un retour haptique et il devient difficile de correctement estimer l'instant et les conséquences du contact. Une solution aidant l'utilisateur à emboîter des pièces peut consister à limiter les degrés de liberté de la pièce qu'il manipule : lorsque la pièce devient coplanaire à une autre, il est alors impossible de la tourner autrement qu'autour de l'azimut ou de la translater autrement qu'en la conservant coplanaire à l'autre comme c'est la cas dans VLEGO (Kiyokawa *et al.*, 1998) (cf. figure 4).

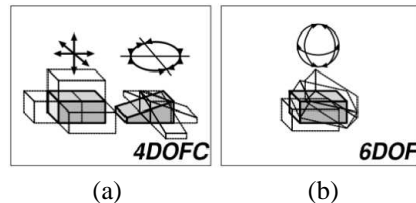


Figure 4. Illustration des degrés de liberté disponibles dans VLEGO (Kiyokawa *et al.*, 1998) pour une pièce. (a) Une pièce non limitée. (b) Les rotations sont limitées à un seul axe

Les contraintes peuvent donc apparaître comme une aide : il faut alors permettre à l'utilisateur de faire la part des choses entre les contraintes qui peuvent l'aider et celles qui modélisent de vraies limitations. Dans les deux cas cependant, les retours exprimant les limitations sont de trois natures : visuelle, sonore et haptique.

2.2.3.1. Retour haptique

La restitution haptique peut transmettre une contrainte au moment d'un contact ou d'une collision. Un bras à retour d'efforts va pouvoir freiner le mouvement d'un utilisateur lorsque ce dernier entrechoquera la pièce qu'il manipule avec une autre pièce présente dans l'EV. D'autres systèmes permettent à l'interacteur de « ressentir », restituer, un contact de son propre corps⁹ avec un objet virtuel (utilisation d'un système vibrotactile (Bloomfield *et al.*, 2007)). L'haptique se divise donc en deux types de retour : le retour de force et le retour tactile. La décision d'utiliser un système à retour d'efforts ou tactile peut dépendre de contraintes pratiques : un bras à retour d'efforts précis est cher, souvent encombrant et d'une amplitude de mouvements faible. Les systèmes tactiles sont généralement plus légers même s'il faut noter l'existence de versions plus compactes de systèmes à retour d'efforts, par exemple pour le Spidar¹⁰ (Kim *et al.*, 2000). Le choix d'un système à retour d'efforts dépend également des objectifs d'une expérience : un système tactile s'avère être meilleur qu'un système à retour d'efforts pour sentir un contact, mais un système à retour de force permet de mieux évaluer la grandeur d'une force alors qu'un système tactile fait surtout ressentir son existence (Massimino *et al.*, Fall 1993). Le coût, la faible précision de certains systèmes, leur encombrement, ou la difficulté de leur mise en œuvre (coûts de calculs de la simulation élevés, nécessité de disposer de données physiques pour les objets simulés, etc.), font que les systèmes haptiques sont parfois évités.

2.2.3.2. Retour visuel

Pour les raisons évoquées précédemment, beaucoup d'utilisateurs d'EV profitent uniquement d'indications visuelles pour leur faire comprendre les contraintes en présence. Une contrainte peut être montrée par l'affichage : *i*) d'une trajectoire à respecter, *ii*) d'une zone dans laquelle les déplacements d'une pièce sont autorisés, *iii*) de butées (cf. figure 5). Par ailleurs, il est possible de combiner une interface haptique passive à un retour visuel adapté afin de simuler des propriétés haptiques ; ce type de retour est qualifié de *pseudo-haptique* (Lécuyer *et al.*, 2000). Ainsi, sur un périphérique isotonique¹¹, l'utilisateur doit effectuer de grands mouvements pour faire un peu bouger un objet qui est freiné par une friction dans l'EV : la difficulté pour le déplacement amène l'utilisateur à imaginer une friction.

2.2.3.3. Retour sonore

Le retour sonore peut être utilisé pour les mêmes raisons que le retour visuel. De plus, il peut permettre de fournir une aide à des utilisateurs ayant des difficultés visuelles. Enfin, un retour sonore peut améliorer les performances dans la réalisation d'une tâche d'assemblage de pièces comme dans (Massimino *et al.*, Fall 1993) où

9. Bien entendu, il faut ici imaginer une immersion du corps de l'utilisateur dans l'EV.

10. *Space interface device for artificial reality*.

11. La position d'un objet manipulé par un périphérique isotonique est fonction des déplacements physiques de ce périphérique. Une souris 2D est un périphérique isotonique.

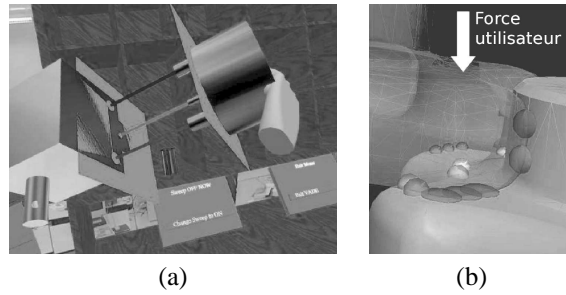


Figure 5. (a) La contrainte sur les mouvements d'un objet est matérialisée par des lignes droites entre les deux pièces qui représentent la trajectoire (Jayaram et al., 1999). (b) Les forces appliquées sont représentées par des sphères colorées qui s'aplatissent (Sreng et al., 2006)

l'expérimentation qui consiste à enfiler une cheville dans un trou¹² est réalisée plus rapidement avec un retour sonore. Ce retour intervient pour signaler un choc (son ponctuel) ou la grandeur d'une force (variation des basses du son).

2.2.3.4. Retour multimodal

Enfin, une tendance très forte est actuellement de combiner de différentes façons des retours visuels, sonores et haptiques comme dans (Zhang *et al.*, 2005) où les utilisateurs effectuent l'expérience « Peg-in-Hole » plus rapidement lorsqu'ils ont un retour visuel et sonore combiné que lorsqu'ils n'ont qu'une seule modalité de retour. L'article (Hecht *et al.*, 2006) conclut que la multiplication des modalités simultanées améliore les performances d'un utilisateur d'un EV (tant que le nombre de ces modalités reste peu élevé pour ne pas surcharger le système cognitif d'un individu). Cette expérience emploie des retours haptiques, visuels et sonores. Dans (Sreng *et al.*, 2007), autant de retours (cf. figure 6 pour les retours visuels) sont utilisés lorsque, par exemple, une roue crantée impacte une barre : un jet de particules ayant le point de collision comme origine apparaît, un son bref représentant le choc est émis, le bras à retours d'efforts bloque le mouvement de l'utilisateur. Les retours visuels et sonores permettent ici de renforcer la sensation de contact donnée à l'utilisateur.

2.2.4. Bilan intermédiaire

Un utilisateur doit être informé du potentiel interactif des objets pour lui permettre de comprendre comment interagir avec un objet et même améliorer ses performances en réalisant des opérations plus rapidement avec cet objet, que cet objet soit un outil ou non. Il faut donc permettre à l'utilisateur de déterminer si le déplacement ou le changement d'orientation d'un objet est possible. Ensuite, l'utilisateur devra être

12. En anglais « Peg-in-Hole ».

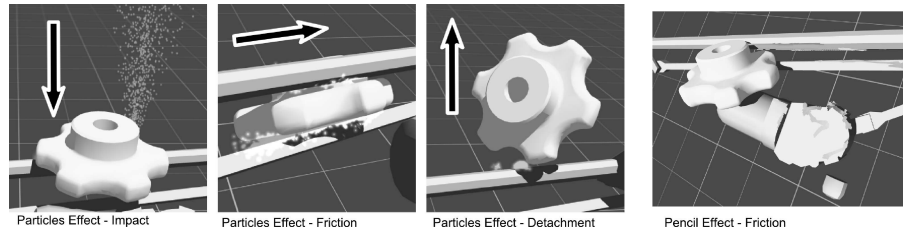


Figure 6. Illustrations de collisions (Sreng et al., 2007)

doté de moyens (d'outils) permettant d'exprimer des nouvelles positions ou orientations afin de modifier ces propriétés. Il en est de même pour toutes les propriétés de l'objet (comme la couleur par exemple) : l'utilisateur doit être averti du potentiel interactif d'un objet et de la façon de l'utiliser, c'est-à-dire avec quel outil interagir, ou bien quelles sont les opérations applicables. Cependant, afin d'éviter à l'utilisateur des mouvements ou des opérations inutiles, afin même de ne pas lui permettre certaines actions pour mieux mimer des contraintes physiques du monde réel, certaines opérations doivent rester impossibles ou limitées. Ainsi apparaissent des retours haptiques ou sonores donnant des indications à un utilisateur dans le guidage d'une pièce le long d'un conduit par exemple : l'utilisateur devient à la fois contraint dans ses actions, mais également assisté pour le rendre plus précis et plus rapide. Certains retours pourraient, *a priori*, être envoyés aux interacteurs partageant une interaction, notamment les retours visuels ou sonores s'ils ne sont pas trop nombreux et s'ils ne font pas confondre à l'utilisateur le résultat de ses actions avec celui d'autres personnes.

3. La coopération

Cette section décrit les fonctions de la coopération, les méthodes d'interactions coopératives, et la prise de conscience de soi et des autres. Il y a un écart sensible entre les besoins exprimés par les utilisateurs et les solutions existantes qui seront présentées. En effet, des industriels pourraient souhaiter que leurs employés puissent reproduire dans des EV des actions du monde réel comme des tâches coopératives d'assemblages complexes, hélas la littérature se limite souvent à proposer des solutions pour seulement déplacer et orienter des objets alors que la coopération permettrait naturellement des interactions plus avancées. L'article (Bayon *et al.*, 2006) montre comment un ingénieur, un designer et un responsable marketing pourraient, au même moment et de façon colocalisée, profiter des compétences de chacun : le designer peut bouger le point de vue à l'intérieur d'une voiture pour parler de l'allure de l'habitacle pendant que l'ingénieur peut, lui aussi, manipuler des parties de l'habitacle et mieux comprendre leur réalisation, enfin la personne chargée du marketing peut enregistrer des séquences vidéos exploitables pour des supports commerciaux. Cet usage de la coopération, dans une phase de prototypage, permet de gagner du temps et réduire les coûts

de conception. L'intérêt d'industriels pour les interactions coopératives se retrouve ainsi naturellement dans le projet ANR/RNTL Part@ge.

3.1. Fonctions

Dans le projet ANR/RNTL Part@ge, la coopération¹³ est considérée comme « la possibilité de contribuer à une tâche globale grâce à des actions complémentaires de plusieurs utilisateurs. Ces actions peuvent se faire de différentes façons, en alternance, ou en simultané, sur un même objet ou sur des objets différents ».

La coopération amène plusieurs utilisateurs à pouvoir travailler ensemble pour réaliser une tâche grâce à des mécanismes permettant d'interagir alternativement ou simultanément. D'une façon générale, des interactions à plusieurs posent des difficultés pour montrer à un utilisateur ce qu'un partenaire est en train de faire ou a fait. Un utilisateur doit avoir conscience¹⁴ de l'autre : sa présence, ses actions passées, courantes, voire futures s'il est possible de les inférer. Lorsqu'une interaction est le résultat d'une combinaison d'autres interactions, c'est le cas de la simultanéité, il apparaît un nouveau problème : celui de faire comprendre à un utilisateur que le résultat d'une de ses actions est fonction de sa combinaison avec celles d'autres utilisateurs.

3.2. Interactions coopératives

Les interactions coopératives entre interacteurs humains sont de deux types : soit les personnes sont dans un même endroit, disons une même salle, soit dans des lieux géographiquement distants. La coopération locale (*colocative*) entraîne la difficulté de présenter des retours visuels et sonores à un ensemble d'individus, alors qu'ils ne sont pas forcément en train d'effectuer les mêmes actions, sans les perturber. La coopération distante entraîne des latences dans les communications réseau qui peuvent devenir gênantes lorsque des utilisateurs doivent se coordonner précisément pour interagir. Les plates-formes logicielles autorisant des interacteurs répartis géographiquement doivent disposer de mécanismes de synchronisation forte.

3.2.1. Une approche « simple » : la séparation des degrés de liberté

Pour réaliser des interactions coopératives, il faut combiner des commandes provenant de plusieurs interacteurs. Les problèmes suivants doivent alors être résolus : *i*) Comment le système recevant des commandes d'interaction venant d'utilisateurs peut-il les reconnaître (par exemple, ne pas confondre des translations selon un axe x et un

13. Certains auteurs distinguent « coopération » et « collaboration » en considérant que l'un désigne des actions simultanées et l'autre des actions simplement coordonnées à la résolution d'une tâche. Malheureusement, dans ces mêmes publications, les définitions d'un mot et de l'autre s'échangent au gré des auteurs. Par conséquent, nous considérerons les deux mots comme synonymes en explicitant les situations.

14. En anglais « awareness ».

axe y) ?, *ii*) Comment combiner des commandes dans le bon ordre (les transformations géométriques ne sont généralement pas commutatives) ? Une première approche consiste à séparer les degrés de liberté selon les utilisateurs en proposant deux types de techniques coopératives : homogènes ou hétérogènes (Pinho *et al.*, 2002).

3.2.1.1. Métaphores coopératives homogènes

Les utilisateurs utilisent une même métaphore mono-utilisateur, c'est-à-dire à l'usage d'un seul utilisateur à la fois, par exemple une main virtuelle. Par ailleurs, la séparation des degrés de liberté implique que si l'un des utilisateurs peut manipuler l'orientation d'un objet, alors un autre ne peut s'occuper que de sa position. Un autre exemple est l'utilisation d'un rayon par un premier utilisateur sur lequel un deuxième utilisateur va pouvoir faire glisser l'objet qui est attaché grâce à son propre rayon. Par sa position, le premier utilisateur n'a pas forcément une bonne perception de la profondeur, ou bien ne peut voir correctement la pièce qu'il manipule du fait de l'éloignement. L'autre utilisateur apporte une aide grâce à son point de vue complémentaire.

3.2.1.2. Métaphores coopératives hétérogènes

Les utilisateurs utilisent des métaphores mono-utilisateur différentes. Par exemple, un premier utilisateur applique des translations à un objet au moyen d'un rayon virtuel pendant qu'un deuxième utilisateur peut lui aussi faire glisser l'objet le long du rayon mais peut également lui appliquer des rotations (cf. figure 7).

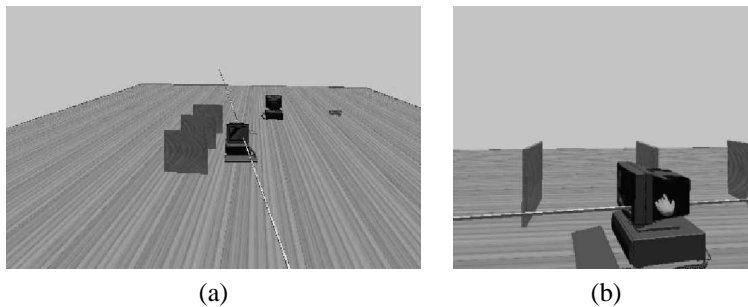


Figure 7. (a) Un utilisateur applique des translations à un objet, y compris le long du rayon. (b) La main virtuelle permet de faire glisser l'objet le long du rayon et de lui appliquer des rotations (Pinho *et al.*, 2002)

3.2.1.3. Résultats

Ce type de technique permet l'utilisation d'outils classiques (un rayon et une main virtuelle) pour faire de la coopération. Chaque utilisateur dispose d'un point de vue sur la scène qui peut être différent. Ces techniques s'avèrent très adaptées pour faire des ajustements et des manipulations difficiles (par exemple des manœuvres dans des espaces étroits) où elles conduisent à la réalisation plus rapide de tâches.

3.2.2. Accès concurrent à un même degré de liberté

Nous allons présenter ici le cas où tous les degrés de liberté disponibles sont donnés à chaque interacteur suivant le classement déjà employé avec les techniques d'interactions égocentriques.

3.2.2.1. Main virtuelle

Il serait possible de positionner une seule main virtuelle, résultante de la moyenne des mouvements de plusieurs interacteurs, mais cette solution n'offrirait aucun retour sur le nombre d'interacteurs et rendrait impossible l'interaction sur plusieurs points de contrôle d'un même objet. Il est donc nécessaire de conserver l'usage de plusieurs mains virtuelles.

L'article (Noma *et al.*, 1997) décrit une méthode où plusieurs utilisateurs peuvent interagir sur un même objet au travers de bras à retour d'efforts. Chaque utilisateur actionne son propre bras à retour d'efforts et bouge ainsi la main virtuelle qui correspond à un bras. Le mouvement de l'objet est alors le résultat de l'équilibre des forces appliquées par les utilisateurs. Pour éviter des à-coups brusques suite à des délais par exemple, un système de ressort¹⁵ est utilisé entre la main de l'utilisateur et le point d'application sur l'objet. La figure 8 montre l'application de ces mains sur un objet dont l'axe de rotation, au point *C* est perpendiculaire au support et évolue avec les déplacements de l'objet. Cette technique entraîne des incohérences entre la position des mains virtuelles et celle des mains réelles mais, d'après ses auteurs, si l'écart des positions est faible, il ne devrait pas gêner les utilisateurs parce que le retour visuel influence plus l'utilisateur que le retour haptique.

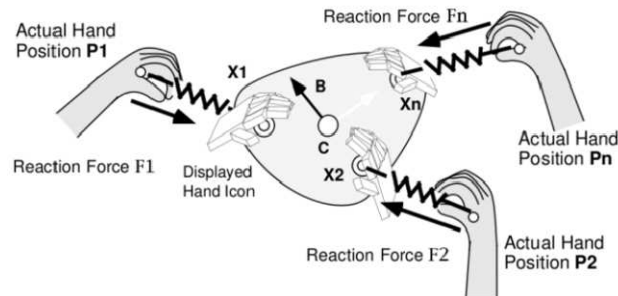


Figure 8. Application d'une force sur un objet virtuel au travers de ressorts virtuels (Noma *et al.*, 1997)

Il est cependant important de noter que l'utilisation d'une main virtuelle pour le coopératif demande l'utilisation d'un retour haptique ou le recours à une métaphore. En effet, si un deuxième interacteur agit sur un objet, la main virtuelle du premier

15. Ce système est un exemple de *couplage virtuel*.

utilisateur qui été posée dessus devra se déplacer pour suivre l'objet. Or, autant le déplacement de la main peut être ressenti par un retour d'effort, autant un simple retour visuel pourrait être difficile à utiliser pour ressentir correctement les actions de l'autre interacteur et agir rapidement en conséquence. De plus, comment un utilisateur qui se serait absenté pourrait-il comprendre que sa main virtuelle a bougé pendant qu'il n'était pas là ? Est-ce que la main doit être déformée, le bras allongé, etc. ?

3.2.2.2. Curseur 3D

La main virtuelle ne permet pas de représenter correctement les interactions multiples simultanées, ni l'historique des interactions. Ces problèmes peuvent être en partie résolus par l'utilisation de pointeurs virtuels au niveau d'abstraction plus élevé, comme les curseurs 3D que nous allons voir à présent.

D'une façon générale, un curseur 3D est un objet 3D déplaçable dans l'espace et qui permet de sélectionner des objets (par proximité, par contact, etc.). L'article (Duval *et al.*, 2006b) présente le *SkeweR* dont le déplacement s'effectue par des curseurs 3D. Un utilisateur approche son curseur près d'un point de contrôle pour manipuler l'objet. S'il n'y a qu'un utilisateur, seuls des mouvements de rotations autour du centre géométrique de l'objet peuvent lui être imprimés. À partir de deux utilisateurs, l'objet se manipule tel une brochette sur laquelle les utilisateurs appliquent des translations ou des rotations. Cela a l'avantage de ne pas limiter la préhension d'un objet uniquement en son centre géométrique, mais elle rend inaccessible l'emploi d'un degré de liberté induit par le « *SkeweR* » lui-même : la rotation autour de l'axe formé par les deux extrémités. Cette limitation disparaît avec trois utilisateurs ayant des points de contrôle formant un plan. Au-delà des trois interactions, les mouvements d'un « *SkeweR* » deviennent sur-contraints et nécessitent un arbitrage.

3.2.2.3. Rayon virtuel

Les rayons classiques explicités précédemment ne permettent pas d'interactions simultanées sur des objets même s'il est facile d'imaginer plusieurs rayons reliés à un même objet. En effet, si un interacteur bouge son rayon, l'objet bouge également et, soit les autres rayons vont le suivre mais ils ne vont alors pas rester cohérents avec la position imprimée par leur utilisateur, sauf s'il est possible de bouger ces rayons avec un retour d'effort, soit les autres rayons ne vont pas pouvoir suivre l'objet car ils vont rester dans la position imprimée par leur utilisateur. En l'absence de retour d'efforts, on place généralement l'objet à une position qui est le barycentre des positions proposées par les rayons et il va donc falloir montrer sur chaque rayon les contraintes résultantes de l'interaction coopérative. Une façon naturelle de traiter ce problème est de courber les rayons en fonction des forces appliquées sur ceux-ci. L'article (Duval *et al.*, 2002) propose trois types de rayons (cf. figure 9) : *i*) l'élastique qui, si un objet ne suit pas une trajectoire imposée par l'interacteur à cause d'une contrainte, force un élastique à s'étirer entre l'interacteur et l'objet ; *ii*) le rayon coudé qui se plie vers l'objet manipulé ; *iii*) le rayon déformable qui se dédouble avec une partie qui reste raide (celle qui a permis la sélection) et une autre qui se plie pour conserver un contact avec

l'objet dont les mouvements sont contraints par la coopération ; la courbure montre l'écart entre la position que l'objet devrait avoir et celle qu'il a réellement. Nous noterons que, parallèlement, (Riege *et al.*, 2006) ont eux aussi proposé une métaphore basée sur des rayons coulés appelée *Bent Pick Ray*.

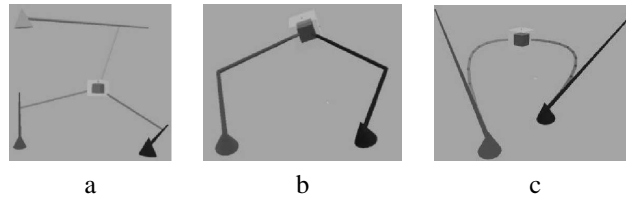


Figure 9. De (a) à (c), des rayons élastiques, des rayons coulés et des rayons déformables (Duval *et al.*, 2002)

3.2.3. Bilan intermédiaire

Les techniques d'interactions coopératives peuvent diminuer les temps d'exécution de tâches et rendre des interactions plus naturelles parce que plusieurs utilisateurs peuvent interagir simultanément sur des parties communes d'objets. Mais ces techniques entraînent le problème du passage d'un utilisateur à plusieurs et *vice versa* qui doit se faire de la façon la plus transparente possible. Par ailleurs, il faut montrer en permanence à l'utilisateur que son action est le fruit de la sienne combinée à celles des autres utilisateurs. La différence entre l'action qu'il souhaite et celle réellement obtenue doit donc être montrée *via* des métaphores. Enfin, il est important de rappeler qu'une plate-forme logicielle doit limiter le plus possible les latences réseau pour permettre à des interacteurs de se coordonner.

3.3. La conscience de soi et des autres

Dans un environnement peuplé par plusieurs utilisateurs, il faut informer chacun des actions que les autres effectuent pour *i)* qu'il puisse comprendre grâce à quel moyen des éléments de l'EV ont évolué, *ii)* qu'il puisse prendre la suite d'une interaction. Ces informations sont encore plus importantes lors d'actions coopératives puisque toute action d'un utilisateur est issue de l'observation de celles des autres.

3.3.1. Représentation d'un utilisateur

Permettre à plusieurs utilisateurs d'interagir simultanément sur des objets nécessite de montrer à un utilisateur la présence des autres. Mais, avant de voir les autres, un utilisateur doit lui-même comprendre qu'il est « dans » l'EV. Une approche simple permettant de plonger un utilisateur dans un environnement virtuel est l'utilisation d'un *avatar* qui est une représentation, souvent en trois dimensions, d'un utilisateur.

Cette représentation va de la représentation la plus simple et abstraite, une sphère par exemple, à l'hyperréalisme en passant par tous les intermédiaires. Le cas le plus simple (Carlsson *et al.*, 1993) permet surtout de montrer l'emplacement d'un utilisateur aux autres, ou encore de montrer la direction de son regard (pour exprimer ses intentions, ou afin de se voir soi-même pour envisager la façon d'entreprendre des actions) cette méthode est directement utilisée dans bon nombre de jeux vidéo (III, 1999) et est qualifiée de « vue à la troisième personne », ou encore de « vue subjective ». Il est possible d'affiner un peu cette représentation pour obtenir des visages véhiculant des expressions, néanmoins seul un avatar réaliste (Dumas *et al.*, 1999), (Salem *et al.*, 2000), pourra couvrir l'ensemble des expressions possibles du visage et permettra, en outre, une communication non verbale véhiculant plus d'informations (Benford *et al.*, 1997), (Mania *et al.*, 1998). Les gestes sont un moyen direct de communication non verbale, la posture exprime l'état d'un utilisateur (sa fatigue par exemple), les contacts du corps avec un autre montrent des signes d'affection ou des liens sociaux, les expressions faciales transmettent des émotions ou donnent des sens différents à des mots prononcés, la représentation réaliste du corps permet de reconnaître une personne réelle et de mieux envisager certaines actions, etc. Par ailleurs, l'incarnation de nombreuses personnes dans un EV augmente la sensation de présence, la sensation d'un monde partagé par plusieurs personnes. (Gerhard *et al.*, 2004) montrent notamment comment utiliser des agents autonomes pour conserver la présence d'avatars dans un EV même lorsque leurs manipulateurs humains se sont absents.

3.3.2. Perception des autres

Comment percevoir les autres lorsque le champ de vision est limité par un écran dont la forme gêne une vision périphérique correcte ? De la même façon, comment faire avec des lunettes stéréoscopiques utilisées dans des grands systèmes immersifs, tel qu'un CAVE™ (Cruz-Neira *et al.*, 1992), qui peuvent partiellement occulter la vue latérale ? Même les systèmes de casques ou de lunettes embarquant des écrans à cristaux liquides ne fournissent généralement pas une vision latérale à grand champ. Il est pourtant indispensable de percevoir les actions des autres et, souvent, cela passe par la vision de ce qu'ils font. Comment également voir ce que les autres regardent lorsque les avatars utilisés ne sont pas très réalistes ou que le système d'affichage n'a pas une finesse de détail suffisante pour bien distinguer la direction du regard ?

Ces problèmes influent directement sur le flot des actions menées au cours d'une interaction (Hindmarsh *et al.*, 1998). Dans le monde réel, on peut être amené à aider quelqu'un d'autre parce qu'on a été capable de le percevoir en train d'effectuer une action. Cette situation peut se reproduire dans un EV. Cependant, un EV inhibe certains sens comme le toucher, en altère d'autres comme la vue ou l'ouïe, et induit le recours quasi systématique à une communication verbale : les gens ont besoin de se dire ce qu'ils font en plus de simplement le faire. Il faut donc des solutions pour compenser les problèmes suscités par les environnements virtuels.

3.3.2.1. Perception des autres par la vue

L'amélioration de la perception des autres par la vue peut passer par l'extension du champ de vision, la focalisation de la vue, ou encore la matérialisation du champ de vision. La première méthode est très adaptée au cas des systèmes fournissant un champ de vision restreint, les autres sont plus générales. Enfin, la vue permet aussi de se rendre compte qu'un objet a été sélectionné ou est en cours de manipulation.

(Fraser *et al.*, 1999) proposent d'étendre le champ de vision par le biais d'une métaphore exploitant l'idée de lentilles périphériques pour la vue. Ils partent du constat que le champ de vision est souvent limité au tiers de celui d'un humain (environ 55°) avec un environnement virtuel. Cette situation ne vient pas directement de l'environnement virtuel mais du fait que l'écran créerait des distorsions si le champ de vision était plus largement couvert et affiché dessus. La figure 10 (a), illustre l'utilisation des lentilles où la vue centrale est favorisée tandis que les vues latérales sont présentes mais compressées latéralement. Ces lentilles permettent surtout de pouvoir jeter des coups d'œil latéraux. Lorsque l'utilisateur souhaite voir correctement en périphérie, il peut basculer une vue secondaire en principale, la vue centrale devient alors distordue tandis que la vue de gauche devient la vue principale (cf. figure 10 (a)).

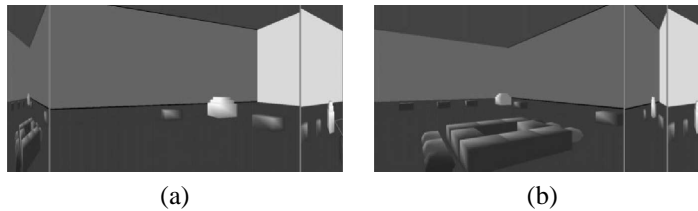


Figure 10. Illustration de l'utilisation des lentilles périphériques. (a) La vue centrale est placée comme vue principale. (b) C'est au tour de la vue de gauche (Fraser *et al.*, 1999)

À l'inverse d'une extension du champ de vision, il pourrait être envisagé de diriger le regard d'un utilisateur vers des points où d'autres utilisateurs interagissent. Des altérations de l'image (renforcements de contrastes par exemple) peuvent ainsi diriger le regard (Baudisch *et al.*, 2003).

Alors que les deux méthodes précédentes travaillent sur le champ de vision à travers lequel un utilisateur voit le monde, on peut chercher à montrer (Hindmarsh *et al.*, 1998), (Fraser *et al.*, 1999) aux autres utilisateurs le champ de vision d'un utilisateur en matérialisant sa pyramide de vue par des fils de fer (cf. figure 11). En pratique, les utilisateurs ne se servent malheureusement des fils de fer de la pyramide que pour repérer la position d'un interacteur.

La pyramide de vue, explicitée précédemment, ne permet pas de montrer qu'un utilisateur est en train de regarder un objet si cet utilisateur est en dehors du champ de vision du premier utilisateur. Cette situation est tout à fait commune dans le monde

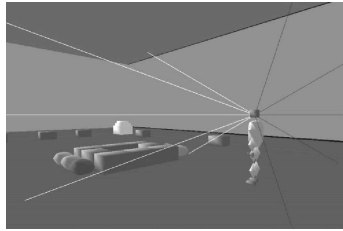


Figure 11. Illustration de la délimitation du champ de vision d'un interacteur par des fils de fer : les fils clairs délimitent la zone non distordue de la vue et les fils foncés celles des lentilles périphériques (Fraser et al., 1999)

réel mais sa fréquence d'apparition est augmentée dans un environnement virtuel si le champ de vision est réduit. Certes les lentilles périphériques permettent de voir partiellement sur les côtés mais la distorsion qu'elles entraînent peut tout à fait ne pas permettre de se rendre compte qu'un utilisateur fixe un objet en particulier. Il faut alors un moyen indiquant clairement la sélection d'un objet et par quel interacteur il est sélectionné. Une solution est de modifier la géométrie ou l'apparence d'un objet lorsqu'il est sélectionné. (Fraser *et al.*, 1999) basculent ainsi la géométrie d'un objet en fils de fer (cf. figure 12) et étirent le bras de l'avatar de l'utilisateur jusqu'à l'objet en cours de manipulation. L'article (Duval *et al.*, 2004) donne plusieurs façons de montrer la sélection des portières d'une voiture, par exemple par le biais de volumes englobants transparents de couleurs différentes (cf. figure 12).



Figure 12. Illustration d'un objet sélectionné : il apparaît en fils de fer et se trouve au bout des bras de l'avatar de l'interacteur (Fraser et al., 1999). Sur (a), la sélection est montrée via des boîtes englobantes (Duval et al., 2004)

3.3.2.2. Perception des autres par les sons

Prenons deux origines de sons : la parole et des bruits, ceux dûs à des entrechoquements de pièces par exemple. La parole est un moyen simple de communication et l'un des plus naturels. De plus, grâce à sa mise en œuvre relativement aisée, plusieurs plates-formes de réalité virtuelle permettent l'utilisation de la voix : MAS-

SIVE (Greenhalgh *et al.*, 1995), Spin3D (Dumas *et al.*, 1999), etc. Mais la voix entraîne des discontinuités lors de coopérations (Bowers *et al.*, 1996), notamment lorsqu'un utilisateur lance un appel aux autres pour démarrer une action mais finalement en arrive à agir lui-même par défaut. C'est ce moment, où l'on attend une réponse de la part de quelqu'un, qu'un délai est introduit. De même quand un utilisateur veut s'assurer que personne n'a l'intention de faire ce qu'il souhaite effectuer, il lance un appel à tout le monde et attend des réponses avant de se décider à agir. Enfin, la voix devient difficile à distinguer lorsque plusieurs utilisateurs parlent en même temps ou lorsque les micros sont dans un environnement bruyant. Par contre, lorsque deux personnes parlent ensemble pour se dire ce qu'elles font ou souhaitent faire, le dialogue est fluide.

En dehors de la parole, le bruit produit par l'entrechoquement de pièces ou d'outils lors d'une interaction est un indicateur des événements importants pour l'utilisateur et ceux qui l'assistent lors de la réalisation d'une tâche (cf. section 2.2.3.3).

3.3.2.3. Perception des autres par l'haptique

Lorsque plusieurs interacteurs maintiennent ou « touchent » un objet en même temps, son déplacement par l'un des interacteurs va provoquer un retour de force chez les autres interacteurs, ou encore une vibration. L'inconvénient avec un système à retour de force est qu'il est difficile de faire la différence, dans certaines situations, entre le retour de force dû à des collisions et celui dû à l'action opposée d'un autre interacteur. Néanmoins, cette modalité introduit une information kinesthésique fondamentale de l'activité des autres participants.

3.3.3. *Bilan intermédiaire*

La conscience de soi au sein d'un EV et de celle des autres passe, au minimum, par une représentation physique et une information de position. L'avatar élaboré permet des communications plus riches car non verbales. La perception des autres peut se faire *via* différents canaux sensoriels : la vue, l'ouïe et le toucher. Deux problèmes récurrents sont : *i*) ne pas « noyer » les utilisateurs par un flot trop important d'informations, *ii*) permettre à l'utilisateur de faire la différence entre les résultats de ses seules actions et les résultats de combinaisons d'interactions de plusieurs utilisateurs.

4. Aspect technique

Nous allons parler à présent des solutions techniques permettant de mettre en œuvre la coopération en EV : comment faire en sorte que leurs utilisateurs situés potentiellement à des endroits géographiques différents puissent partager un même monde simultanément et y agir ? Nous verrons que la résolution de ce problème passe par le choix d'une architecture logicielle et d'algorithmes pour la distribution afin de maintenir la cohérence entre sites.

4.1. Architectures logicielles

Les architectures logicielles employées dans les EV peuvent se regrouper sous trois grandes catégories : centralisée, répliquée et hybride. Nous proposons dans cette section d'analyser les qualités et les défauts de chacune de ces catégories.

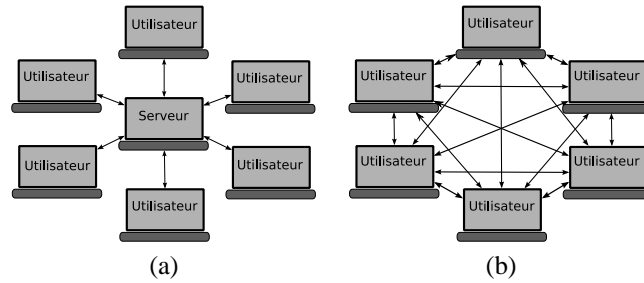


Figure 13. (a) Illustration d'un réseau centralisé où, de fait, une base de données décrivant l'environnement virtuel est partagée avec l'ensemble des utilisateurs. (b) Illustration d'un réseau distribué. Ces deux illustrations sont basées sur (Gossweiler et al., 1994)

4.1.1. Architecture centralisée

L'ensemble des données décrivant l'environnement virtuel est stocké sur un seul serveur. Ces données décrivent les objets du monde virtuel par leurs positions dans l'espace, leur état ou encore leurs parties interactives. De leur côté, les sites impliqués dans la simulation interactive ne stockent pas de données descriptives mais disposent généralement, en local, des géométries des objets de l'EV pour diminuer le trafic réseau qu'engendrerait l'échange de ce type de données. Dans le cas « extrême » de centralisation, le serveur calcule l'image résultat d'une action à envoyer sur l'environnement de visualisation d'un interacteur. Le serveur calcule tout, et le client affiche le résultat qu'il a demandé au travers d'une action à effectuer. La puissance de calcul graphique du serveur doit alors être très importante.

Lorsqu'un interacteur agit sur un objet, le site où se trouve physiquement l'interacteur envoie au serveur central les données d'une demande de mise à jour : une nouvelle position, une nouvelle couleur pour une partie d'un objet, une modification de la géométrie d'un objet, etc. Le serveur a alors à sa charge, d'une part, d'accepter ou de refuser les modifications, et d'autre part de les propager aux autres sites, y compris l'émetteur des commandes, lorsqu'elles ont été acceptées. L'architecture centralisée a ainsi pour avantage la simplicité de son principe de fonctionnement et donc de sa mise en œuvre : chaque site ne peut recevoir des modifications que d'un seul point, le serveur. De plus, les clients déployés sur chaque site ont un schéma de fonctionnement très simple : toute tentative de modification locale est envoyée au serveur et seul le résultat provenant du serveur est affiché. Le serveur doit donc gérer tout

conflit entre différentes actions concurrentes. Sauf si les clients sont à des distances très variables du serveur ou utilisent des liaisons aux performances très différentes, ils reçoivent les résultats d'actions au même moment. Cette synchronisation conduit à faciliter les interactions coopératives nécessitant une forte coordination. Enfin, cette architecture assure la cohérence entre sites sans avoir recours à des mécanismes complexes à mettre en place. Néanmoins, le fait que toutes les communications passent par le serveur peut le transformer en un goulot d'étranglement s'il reçoit beaucoup de commandes d'interactions de clients ou même s'il doit mettre à jour un grand nombre de clients. Enfin, cette architecture suppose que le serveur doit toujours rester opérationnel pour pouvoir fonctionner, ou que les ruptures de communications doivent rester extrêmement brèves.

4.1.2. Réplication de données sur chaque site

La *réplication active* est un modèle de distribution d'objets où l'ensemble des données décrivant l'EV est stocké sur chaque site. L'état de chaque site est initialisé à son démarrage (certains sites peuvent rejoindre l'EV plus tard au cours de la simulation) et leur évolution est effectuée indépendamment sur chacun. Une plate-forme utilisant cette architecture est Spin-3D (Gaeremynck *et al.*, 2006) de Orange Labs.

Une interaction sur un site doit communiquer aux autres sites les changements qu'elle a entraînés. Chaque site est donc responsable des interactions qu'il autorise et des calculs des résultats, ces derniers impliquent de fortes charges sur chaque machine. Ainsi, le retour d'informations à un interacteur est immédiat. Malheureusement, des synchronisations sont parfois nécessaires et peuvent s'avérer délicates : comment arbitrer l'accès à un objet par exemple ? Quel interacteur a modifié l'objet en premier ? Il est également possible d'émettre en permanence les changements de tout objet (dûs à une interaction ou au simple déroulement de l'évolution de l'EV) et à les transmettre aux autres sites.

De plus, l'interacteur qui agit est le premier à être informé des changements qu'il a produits tandis qu'un autre interacteur qui travaille en coopération avec cette personne ne sera informé des changements que plus tard. Le délai introduit est problématique dans le cas où des interacteurs ont besoin de fortement se coordonner. Néanmoins, ce type d'architecture *pair-à-pair*¹⁶ présente aussi des avantages, notamment celui d'être potentiellement robuste aux pannes parce que la disparition d'un site n'interrompt pas le fonctionnement de l'ensemble de l'EV. Toutefois, il devient important de se poser la question de la gestion du cas où un interacteur manipule un objet lorsque son site tombe en panne. Est-ce que l'objet en cours d'interaction doit revenir à l'état qu'il avait avant l'interaction ? Est-ce que l'objet doit rester dans sa position actuelle (par exemple, rester artificiellement en l'air) ? En ce qui concerne la communication, l'envoi de messages par *multicast*¹⁷ (voire par *broadcast*¹⁸) est nécessaire pour pouvoir envoyer à chaque site concerné de nouveaux états. Mais ce système a le désavantage

16. « Peer to Peer » (P2P) en anglais.

17. Le « multicast » permet d'envoyer des messages à un groupe de machines sur un réseau.

18. Le « broadcast » permet l'envoi de messages à toutes les machines connectées sur un réseau.

de produire une très forte activité réseau et d'être peu exploitable sur des réseaux à grande échelle (ex : l'internet) d'où le recours à des solutions logicielles qui donnent l'illusion d'un « multicast » à un EV.

4.1.3. Architecture hybride

L'architecture répliquée représente un mieux par rapport à l'architecture centralisée du point de vue de la robustesse aux pannes, de la vitesse du retour d'informations à l'interacteur et des possibilités de passage à l'échelle. Cependant, l'activité réseau est très forte et des problèmes importants d'arbitrage dans les interactions simultanées subsistent. Pour ces raisons, des architectures hybrides ont été développées.

4.1.3.1. Réplication à la demande

Pour diminuer les échanges réseaux entre sites, tels que ceux d'un système de réplication active, il est préférable d'effectuer l'envoi de messages par *multicast*. Un site ne possède donc que les données qui lui importent ; pour cette raison, ce modèle de distribution d'objets se nomme *réplication à la demande*.

4.1.3.2. Clients/serveurs et partitionnement de l'EV

La base de données décrivant l'EV peut être distribuée sur plusieurs sites. Toutes les communications passent par un serveur qui sait où se situent les objets qu'un interacteur peut souhaiter utiliser. Ce système est destiné à afficher plusieurs vues à plusieurs groupes d'utilisateurs tout en ayant un grand nombre d'objets dans l'EV. En pratique, cette architecture entraîne une très forte activité réseau concentrée sur le serveur (exactement comme dans le cas de l'architecture centralisée qui n'est autre qu'une variation de l'architecture de celle-ci). Pour contrer ce problème, il est possible de disposer de plusieurs serveurs avec, de fait, des difficultés pour les synchroniser entre eux. Ce cas est exploité par la plate-forme RING (Funkhouser, 1995).

4.1.3.3. Référentiels et miroirs

Chaque site peut se voir doté de certains objets de l'EV, qu'ils soient visuels ou non. Lorsque un site possède un objet, cet objet est un référentiel. Le référentiel est chargé de faire évoluer l'objet après toute demande de modification d'un état de cet objet (position, forme, couleur, etc.). Lorsqu'un interacteur présent sur un site distant essaie d'interagir avec un objet, il passe par un *miroir*, appelé aussi proxy ou fantôme. Un miroir reçoit régulièrement les mises à jour d'un référentiel et les passe systématiquement aux objets (s'il y en a) qui lui sont connectés. L'utilisation d'un miroir diminue les activités réseau parce que la seule communication qui a lieu est celle alimentant le miroir. En effet, les autres objets présents sur le même processus qu'un miroir ne seront en liaison qu'avec ce miroir. D'une certaine façon, les référentiels jouent le rôle de serveur tandis que les miroirs jouent celui de client. Ce modèle est exploité par OpenMASK (Margery *et al.*, 2002).

4.1.3.4. Migration

Plutôt que de placer statiquement et de façon arbitraire un objet sur un site (par exemple le site où il y a l'interacteur susceptible d'interagir le plus, ou le site le moins chargé au moment de la création de l'objet), il peut être préférable de le placer dynamiquement en fonction d'autres critères, notamment la charge d'une machine au cours de l'évolution de l'EV. Par exemple, il devient ainsi possible d'éviter d'avantager un outil par rapport à un autre au niveau de la vitesse de la communication. Lorsqu'au cours de l'utilisation de l'EV, un objet peut être déplacé d'un site vers un autre : on parle de *migration* d'objets. Des travaux ont été réalisés dans ce sens sur OpenMASK (Duval *et al.*, 2006a). Notons que la géométrie des objets, si elle est déjà présente sur chaque site, n'est généralement pas migrée pour éviter de trop fortes charges réseau.

4.1.4. Bilan intermédiaire

Aucune architecture ne présente une solution universelle au problème de la cohérence entre sites. En effet, à l'exception de délais relativement maîtrisés autorisant donc des interactions coopératives fortement coordonnées, l'architecture centralisée présente trop d'inconvénients majeurs pour être intéressante (faible robustesse, puissance de calcul requise du serveur considérable, difficultés de passage à l'échelle, etc.). En utilisant des architectures hybrides, le modèle décentralisé est un meilleur choix même s'il est plus complexe : il offre des interactions avec un retour rapide d'informations (ce qui est très profitable aux interactions coopératives) ainsi que des arbitrages entre interactions simultanées, il utilise raisonnablement le réseau et permet des passages à l'échelle plus aisés.

4.2. Algorithmes

Parmi les nombreuses difficultés introduites par les architectures distribuées, notamment hybrides, nous choisissons d'en voir deux qui nous semblent parmi les plus essentielles : la communication entre sites et le maintien de la cohérence entre sites.

4.2.1. Communication entre sites

En un même lieu, la communication entre deux entités est immédiate. En distribué, le réseau influe sur la qualité de cette communication en augmentant les délais, voire en perdant des données¹⁹. Une activité réseau trop importante augmente ces deux aléas. Deux formes de communication entre sites sont utilisables : l'événement et le flot de données. L'événement peut véhiculer une valeur qui est, bien souvent, une valeur ponctuelle : par exemple, un booléen qui passe de faux à vrai et *vice versa*. L'envoi peut souffrir des problèmes décrits précédemment : pertes de données et retards de livraison trop importants. Le flot de données consiste à envoyer des informa-

19. La perte de données dépend surtout du protocole de communication sous-jacent employé. Ainsi TCP est un protocole assurant la livraison des paquets réseau, tandis que UDP est plus rapide mais non fiable.

tions en permanence sous la forme de flots de données. De cette façon, un récepteur trouve toujours une donnée à exploiter, même si cette dernière n'est pas forcément la plus à jour. Cette méthode est peu adaptée à des changements de valeurs ponctuels. Par ailleurs, la charge réseau induite par ce système est importante. Ainsi, un mélange entre flots, pour les positions d'objets en mouvements par exemple, et d'événements, pour des communications ponctuelles, peut être envisagé comme dans OpenMASK. Enfin, des communications par appel de fonctions distantes peuvent être employées, selon le modèle CORBA²⁰ par exemple.

4.2.2. Cohérence entre sites

Dans le cas d'une architecture centralisée, le calcul de l'évolution de chaque entité s'effectue sur une même machine ; les seules difficultés techniques proviennent des entrées concurrentes qu'il faut arbitrer/combiner. En distribué répliqué, chaque entité évolue indépendamment, mais l'état de chaque entité doit rester le même sur chaque site au fil du temps ; cela est assuré par le calcul de l'évolution de chaque entité dont l'avancement est basé sur les tics d'une horloge globale. En distribué hybride, deux cas sont possibles : soit un miroir fait suivre les valeurs des sorties d'un référentiel (voir OpenMASK), soit un miroir est un objet qui calcule par lui-même les valeurs de sortie tant que ses propres calculs ne s'écartent pas trop des valeurs calculées par le référentiel (voir NPSNET (Macedonia *et al.*, 1994)). Cette technique s'appelle du *dead-reckoning* (Gossweiler *et al.*, 1994) et s'emploie notamment pour du calcul de trajectoires. Le but de ce modèle est de fortement réduire l'activité réseau et aussi de maintenir un système en marche pendant des petites coupures réseau.

5. Conclusion et perspectives

Si l'on se réfère à la classification fonctionnelle qui identifie les fonctions des collecticiels selon les espaces de production/coopération, de coordination et de communication (Ellis *et al.*, 1991), on peut voir que les systèmes d'EVC permettent effectivement aux utilisateurs d'agir dans un espace de production commun. Ensuite, on constate que ces systèmes assurent généralement une coordination entre utilisateurs seulement selon des modes d'interaction exclusifs. Enfin, ces systèmes cherchent à faciliter la communication entre utilisateurs grâce à des métaphores de rendu, visuelles, sonores ou haptiques, mettant en évidence les actions des autres afin d'en faciliter la perception.

Cependant, bien que s'appuyant sur des techniques d'interactions 3D éprouvées, les techniques coopératives sont encore peu nombreuses et souvent trop spécialisées ; en particulier elles n'abordent souvent que les problèmes de position et d'orientation d'objets 3D. Côté coordination, des problèmes d'ordre technique persistent, liés à la difficulté de maintenir la cohérence entre sites, au besoin de généraliser les combinaisons de commandes provenant de plusieurs utilisateurs, ou encore à la nécessité de

20. Site de CORBA : <http://www.omg.org/>

diminuer les délais ressentis par l'utilisateur au cours d'interactions distantes. Enfin, il manque un formalisme de description des interactions coopératives : non seulement aucun format de fichier ne permet de stocker ce type d'information mais il n'existe même pas de taxonomie de toutes les coopérations possibles ni des retours pouvant être attendus (visuels, sonores ou haptiques). Les formats de données tels que VRML ou X3D permettent uniquement le stockage d'informations simples pour l'interaction : il est par exemple impossible de préciser des comportements évolués comme les limites (contraintes), les droits d'accès, ou encore les retours offerts à l'utilisateur. De même, aucun format n'existe pour décrire la coopération : le fait qu'un objet soit saisissable par plusieurs personnes, l'arbitrage entre les interactions simultanées, ou encore les retours adéquats concernant des actions coopératives.

Il nous semble donc qu'un effort doit être porté : d'une part sur la description de la dimension coopérative des objets de cet espace de production ; d'autre part sur les moyens de faire percevoir le potentiel interactif et coopératif des objets ainsi que les actions (coopératives ou non) en cours dans l'environnement virtuel partagé. Pour cela, il faut proposer un formalisme de description des interactions coopératives et des retours qu'elles offrent aux utilisateurs. Il est donc nécessaire de créer des formats de données plus complets, cela doit passer par une formalisation des interactions coopératives : la description des interactions offertes, des types de retours fournis aux utilisateurs, etc.

Pour aller dans ce sens, dans le cadre du projet Part@ge, nous avons contribué à proposer des extensions au format Collada afin de prendre en compte les caractéristiques non seulement interactives mais aussi coopératives des objets virtuels qui peuplent les EVC, ainsi qu'un langage de communication entre objets interactifs et outils d'interaction exploitant ces caractéristiques. Nous travaillerons ensuite à la proposition d'outils d'édition permettant de décrire plus facilement ces aspects coopératifs et permettant de créer plus facilement de nouveaux outils adaptés aux interactions coopératives, c'est-à-dire de nouvelles métaphores de coopération.

Une fois définis de nouvelles techniques d'interaction coopérative et les moyens de les percevoir, une fois résolus les problèmes techniques liés aux architectures distribuées, une fois définis des formats d'interactions coopératives basées sur un formalisme précis, une fois décrits des protocoles permettant d'échanger des données au cours de sessions dans des EV, alors il sera possible d'atteindre un niveau d'abstraction suffisant pour obtenir de l'interopérabilité entre différentes plates-formes de développement d'EV, à partir du chargement des données jusqu'à la communication entre les plates-formes pendant l'exécution.

Remerciements

Ces travaux ont été réalisés dans le cadre du projet Part@ge 06 TLOG 031 : plate-forme RNTL financée par l'ANR.

6. Bibliographie

- Badawi M., Donikian S., « Autonomous Agents Interacting With Their Virtual Environment Through Synoptic Objects », *CASA 2004*, 2004.
- Baudisch P., DeCarlo D., Duchowski A. T., Geisler W. S., « Focusing on the Essential : Considering Attention in Display Design », *Commun. ACM*, vol. 46, n° 3, p. 60-66, 2003.
- Bayon V., Griffiths G., Wilson J. R., « Multiple decoupled interaction : An interaction design approach for groupware interaction in co-located virtual environments », *Int. J. Hum.-Comput. Stud.*, vol. 64, n° 3, p. 192-206, 2006.
- Benford S., Bowers J., Fahlén L. E., Greenhalgh C., Snowdon D., « Embodiments, avatars, clones and agents for multi-user, multi-sensory virtual worlds », *Multimedia Systems*, vol. 5, n° 2, p. 93-104, 1997.
- Bloomfield A., Badler N. I., « Collision Awareness Using Vibrotactile Arrays », *Virtual Reality Conference, 2007. VR'07. IEEE*, p. 163-170, March, 2007.
- Bowers J., Pycock J., O'Brien J., « Talk and Embodiment in Collaborative Virtual Environments », *CHI*, p. 58-65, 1996.
- Bowman D. A., Hodges L. F., « An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments », *SI3D '97 : Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, p. 35-38., 1997.
- Bowman D. A., Kruijff E., Joseph J. LaViola J., Poupyrev I., *3D User Interfaces : Theory and Practice*, Addison-Wesley/Pearson Education, 2005.
- Carlsson C., Hagsand O., « DIVE – a Multi-User Virtual Reality System », *IEEE VRAIS '93*, p. 394-400, Sept, 1993.
- Cruz-Neira C., Sandin D. J., DeFanti T. A., Kenyon R. V., Hart J. C., « The CAVE : audio visual experience automatic virtual environment », *Commun. ACM*, vol. 35, n° 6, p. 64-72, 1992.
- Cutler L. D., Fröhlich B., Hanrahan P., « Two-Handed Direct Manipulation on the Responsive Workbench », *SI3D '97 : Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, p. 107-114, 1997.
- Dumas C., Degrande S., Saugis G., Chaillou C., Plénacoste P., Viaud M.-L., « Spin : a 3-D Interface for Cooperative Work », *Virtual Reality Society Journal*, May, 1999.
- Duval T., el Zammar C., « A Migration Mechanism to Manage Network Troubles while Interacting within Collaborative Virtual Environments », *VRCIA '06 : Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, ACM Press, New York, NY, USA, p. 417-420, 2006a.
- Duval T., Fenals A., « Faciliter la perception de l'interaction lors de manipulations coopératives simultanées en environnements virtuels 3D », *annex of the Proceedings of IHM 2002*, Poitiers, France, p. 29-32, November, 2002.
- Duval T., Lécuyer A., Thomas S., « SkeweR : a 3D Interaction Technique for 2-User Collaborative Manipulation of Objects in Virtual Environments », *3DUI '06 : Proceedings of the 3D User Interfaces (3DUI'06)*, IEEE Computer Society, p. 69-72, 2006b.
- Duval T., Tenier C. L., « Interactions 3D coopératives en environnements virtuels avec Open-MASK pour l'exploitation d'objets techniques », *Mécanique & Industries*, vol. 5, p. 129-137, 2004.

- Ellis C. A., Gibbs S. J., Rein G., « Groupware : some issues and experiences », *Commun. ACM*, vol. 34, n° 1, p. 39-58, 1991.
- Fraser M., Benford S., Hindmarsh J., Heath C., « Supporting Awareness and Interaction through Collaborative Virtual Interfaces », *UIST '99 : Proceedings of the 12th annual ACM symposium on User interface software and technology*, ACM Press, p. 27-36, 1999.
- Fuchs P., Berthoz A., Vercher J.-L., *Traité de la réalité virtuelle*, vol. 1, 3e edn, Presses de l'École des Mines de Paris, chapter Introduction à la réalité virtuelle, p. 3-21, 2006.
- Funkhouser T. A., « RING : a client-server system for multi-user virtual environments », *SI3D '95 : Proceedings of the 1995 symposium on Interactive 3D graphics*, ACM, p. 85-92., 1995.
- Gaeremynck S., Degrande S., Grisoni L., Chaillou C., « Utilisation des composants pour la représentation des objets virtuels », *Journées AFIG – Bordeaux 2006*, 2006.
- Gerhard M., Moore D., Hobbs D., « Embodiment and copresence in collaborative interfaces », *Int. J. Hum.-Comput. Stud.*, vol. 61, n° 4, p. 453-480, 2004.
- Gossweiler R., laferriere R. J., Keller M. L., Pausch R., « An Introductory Tutorial for Developing Multi-User Virtual Environments », *Presence : Teleoperators and Virtual Environments*, vol. 3, n° 4, p. 255-264, 1994.
- Greenhalgh C., Benford S., « MASSIVE : A Collaborative Virtual Environment for Teleconferencing », *ACM Trans. Comput.-Hum. Interact.*, vol. 2, n° 3, p. 239-261, 1995.
- Hand C., « A Survey of 3D Interaction Techniques », *Computer Graphics Forum*, vol. 16, n° 5, p. 269-281, 1997.
- Hecht D., Reiner M., Halevy G., « Multimodal Virtual Environments : Response Times, Attention, and Presence », *Presence*, vol. 15, n° 5, p. 515-523, 2006.
- Hindmarsh J., Fraser M., Heath C., Benford S., Greenhalgh C., « Fragmented Interaction : Establishing mutual orientation in virtual environments », *ACM conference on Computer-Supported Cooperative Work (CSCW'98)*, ACM Press, p. 217-226, November, 1998.
- III R. R., « What's your perspective ? », *SIGGRAPH CG*, vol. 33, n° 3, p. 9-12, 1999.
- Jayaram S., Jayaram U., Wang Y., Tirumali H., Lyons K., Hart P., « VADE : A Virtual Assembly Design Environment », *IEEE CGA*, vol. 19, n° 6, p. 44-50, 1999.
- Kallmann M., Object Interaction in Real-Time Virtual Environments, PhD thesis, École Polytechnique Fédérale de Lausanne, 2001.
- Kallmann M., Thalmann D., « Modeling Objects for Interaction Tasks », *Computer Animation and Simulation '98*, p. 73-86, 1998.
- Kim S., Ishii M., Koike Y., Sato M., « Development of Tension Based Haptic Interface and Possibility of its Application to Virtual Reality », *VRST '00 : Proceedings of the ACM symposium on Virtual reality software and technology*, ACM, p. 199-205, 2000.
- Kiyokawa K., Takemura H., Katayama Y., Iwasa H., Yokoya N., « VLEGO : A Simple Two-Handed 3D Modeler in a Virtual Environment », *Electronics and Communications in Japan, Part 3 : Fundamental Electronic Science*, vol. 81, n° 11, p. 18-28, November, 1998.
- Lécuyer A., Coquillart S., Kheddar A., Richard P., Coiffet P., « Pseudo-Haptic Feedback : Can Isometric Input Devices Simulate Force Feedback ? », *VR*, vol. 00, p. 83, 2000.
- Macedonia M. R., Zyda M. J., Pratt D. R., Barham P. T., Zeswitz S., « NPSNET : A Network Software Architecture for Large-Scale Virtual Environment », *Presence*, vol. 3, n° 4, p. 265-287, 1994.

- Mania K., Chalmers A., A Classification for User Embodiment in Collaborative Virtual Environments, Technical report, University of Bristol, Bristol, UK, UK, 1998.
- Margery D., Arnaldi B., Chauffaut A., Donikian S., Duval T., « OpenMASK : Multi-threaded | Modular Animation and Simulation Kernel | Kit : un bref survol », *VRIC 2002, Laval, France*, S. Richir, P. Richard, B. Tavel, juin, 2002.
- Massimino M. J., Sheridan T. B., « Sensory Substitution for Force Feedback in Teleoperation », *Presence*, vol. 2, n° 4, p. 344-352, Fall 1993.
- Mine M. R., Virtual Environment Interaction Techniques, Technical Report n° TR95-018, University of North Carolina, Department of Computer Science, 1995.
- Mine M. R., Brooks, Jr. F. P., Sequin C. H., « Moving Objects in Space : Exploiting Proprioception In Virtual-Environment Interaction », *Computer Graphics*, vol. 31, n° Annual Conference Series, p. 19-26, 1997.
- Noma H., Miyasato T., « Cooperative Object Manipulation in Virtual Space using Virtual Physics », *ASME-DSC*, vol. 61, p. 101-106, 1997.
- Owal A., Feiner S., « The Flexible Pointer : An Interaction Technique for Selection in Augmented and Virtual Reality », *Conference Supplement of UIST '03 (ACM Symposium on User Interface Software and Technology)*, p. 81-82, November, 2003.
- Pierce J. S., Stearns B. C., Pausch R., « Voodoo dolls : seamless interaction at multiple scales in virtual environments », *IBD '99 : Proceedings of the 1999 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, p. 141-145, 1999.
- Pinho M. S., Bowman D. A., Freitas C. M., « Cooperative Object Manipulation in Immersive Virtual Environments : Framework and Techniques », *VRST '02 : Proceedings of the ACM symposium on Virtual reality software and technology*, ACM Press, p. 171-178, 2002.
- Poupyrev I., Billinghamurst M., Weghorst S., Ichikawa T., « The Go-GO Interaction Technique for Direct Manipulation in VR », 1996a, Unpublished SIGGRAPH Technical Sketch.
- Poupyrev I., Billinghamurst M., Weghorst S., Ichikawa T., « The Go-GO Interaction Technique : Non-Linear Mapping for Direct Manipulation in VR », *Proceedings of the 9th annual ACM symposium UIST*, ACM Press, p. 79-80, 1996b.
- Poupyrev I., Weghorst S., Billinghamurst M., Ichikawa T., « Egocentric Object Manipulation in Virtual Environments : Empirical Evaluation of Interaction Techniques », *Computer Graphics Forum*, 1998.
- Riege K., Holtkamper T., Wesche G., Frohlich B., « The Bent Pick Ray : An Extended Pointing Technique for Multi-User Interaction », *3DUI '06 : Proceedings of the 3D User Interfaces (3DUI'06)*, IEEE Computer Society, Washington, DC, USA, p. 62-65, 2006.
- Salem B., Earle N., « Designing a non-verbal language for expressive avatars », *CVE '00 : Proceedings of the third international conference on Collaborative virtual environments*, ACM Press, New York, NY, USA, p. 93-101, 2000.
- Sreng J., Bergez F., Garrec J. L., Lécuyer A., Andriot C., « Using an Event-Based Approach to Improve the Multimodal Rendering of 6DOF Virtual Contact », *Proceedings of ACM International Symposium on Virtual Reality Software and Technology*, 2007.
- Sreng J., Lécuyer A., Mégard C., Andriot C., « Using Visual Cues of Contact to Improve Interactive Manipulation of Virtual Objects in Industrial Assembly/Maintenance Simulations », *IEEE TVCG*, vol. 12, n° 5, p. 1013-1020, 2006.

- Stoakley R., Conway M. J., Pausch R., « Virtual Reality on a WIM : Interactive Worlds in Miniature », *CHI '95 : Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co., p. 265-272, 1995.
- Zhang Y., Sotudeh R., Fernando T., « The Use of Visual and Auditory Feedback for Assembly Task Performance in a Virtual Environment », *SCCG '05 : Proceedings of the 21st spring conference on Computer graphics*, ACM Press, New York, NY, USA, p. 59-66, 2005.

Article reçu le 19 novembre 2007

Accepté après révisions le 24 juillet 2008

Laurent Aguerreche a effectué ses études à l'université de Bordeaux I où il a obtenu un Master recherche en 2006 spécialisé dans la réalité virtuelle et la synthèse d'images. Il est actuellement doctorant à l'INSA de Rennes et effectue ses recherches à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) à propos des interactions collaboratives dans des environnements virtuels immersifs dans le cadre du projet Part@ge.

Thierry Duval est Maître de Conférences en informatique à l'IFSIC, à l'Université de Rennes I, et effectue ses recherches à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) dans l'équipe projet Bunraku. Il a obtenu un doctorat en informatique de l'École Centrale de Nantes et de l'Université de Nantes en 1993. Ses thèmes de recherche sont l'interaction 3D, la coopération en environnements virtuels, et les architectures logicielles pour l'interaction. Il est membre de l'AFRV et de l'AFIHM, et il a participé à plusieurs projets de plates-formes RNRT, RNTL et ANR : VTHD, VTHD++, PERF-RV, Part@ge et Collaviz.

Bruno Arnaldi est Professeur à l'INSA de Rennes (Institut National des Sciences Appliquées) et effectue ses recherches à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) dans l'équipe projet Bunraku. Il a obtenu un doctorat en informatique de l'Université de Rennes I en 1988 et une habilitation à diriger les recherches à l'université de Rennes I en 1994. Il a été le coordinateur de la plate-forme française de réalité virtuelle PERF-RV et il coordonne actuellement la plate-forme ANR Part@ge. Il est président de l'Association Française de Réalité Virtuelle (AFRV).