



MASTER RESEARCH INTERNSHIP



RAPPORT BIBLIOGRAPHIQUE

Sailor vs Poseidon : Simulation pour l'Apprentissage en Environnement Virtuel Collaboratif 3D

Domaine : Technologie pour l'Apprentissage Humain, Interaction Homme-Machine

Auteur :
Gwendal LE MOULEC

Superviseurs :
Valérie GOURANTON
Thi Thuong HUYEN NGUYEN
Fernando ARGELAGUET
Anatole LÉCUYER
Hybrid

Résumé

Ce rapport bibliographique présente l'étude de méthodes employées en Environnement Virtuel Collaboratif et en simulations dédiées à l'apprentissage afin de faciliter les interactions entre les utilisateurs et l'efficacité du processus d'apprentissage et d'évaluation par un entraîneur. Les premières méthodes présentées sont celles qui permettent de favoriser la compréhension des faits et des événements liés à l'environnement et aux interactions des utilisateurs, ce que l'on appelle l'*awareness*. Elles comprennent plus particulièrement les techniques de guidage pour l'exécution de tâches précises, et celles qui facilitent la manipulation d'objets virtuels. Ces techniques se basent sur le concept de métaphore, qui consiste en un ensemble de moyens pour représenter dans l'environnement virtuel des faits ou événements en faisant appel à l'expérience que l'utilisateur a du monde réel. Les moyens d'organiser les tâches et de synchroniser les interactions collaboratives sont également étudiés. Quelques normes sociales qu'il faut prendre en compte pour la mise en œuvre de ces métaphores sont présentées. L'étude se terminera par la présentation des "zones d'ombre" qu'il nous faudra étudier par la suite. Nous produirons pour cela une application de Réalité Virtuelle pour l'apprentissage de la navigation, permettant à l'entraîneur d'agir dans l'environnement comme un utilisateur pourvu de capacités supérieures à celles d'un apprenant.

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | <i>Awareness</i> et interactions | 3 |
| 2.1 | Compréhension de l'environnement et des événements | 3 |
| 2.2 | Guidage | 5 |
| 2.3 | Manipulation d'objets | 6 |
| 2.3.1 | Exploitation des 6DDL | 6 |
| 2.3.2 | Prise en compte de l'influence d'un coopérateur | 8 |
| 3 | Aspects sociaux | 8 |
| 4 | Gestion automatique de l'apprentissage et des interactions collaboratives | 9 |
| 4.1 | Scénario d'apprentissage | 9 |
| 4.2 | Synchronisation des interactions collaboratives | 11 |
| 5 | Conclusion | 13 |

1 Introduction

La Réalité Virtuelle (RV) *"est un domaine scientifique et technique exploitant l'informatique et des interfaces comportementales en vue de simuler dans un monde virtuel le comportement d'entités 3D, qui sont en interaction en temps réel entre elles et avec un ou des utilisateurs en immersion pseudo-naturelle par l'intermédiaire de canaux sensori-moteurs"* [AFT03]. Dans certaines applications, plusieurs utilisateurs peuvent collaborer dans le même environnement virtuel. De tels environnements sont appelés Environnements Virtuels Collaboratifs (EVC). Les EVC doivent permettre la collaboration d'utilisateurs physiquement présents au même endroit (e.g grâce à un dispositif immersif de type CAVE¹ [CNSD⁺92]), ou bien répartis sur des sites distants, et même avec des humains virtuels. On distingue alors trois niveaux de collaboration [MAP99] :

1. Les utilisateurs sont conscients les uns des autres grâce à une représentation sous forme d'avatars et peuvent communiquer entre eux. C'est un niveau de base qui peut être suffisant pour des applications de téléconférence comme MASSIVE [GB95] ou des réseaux sociaux par exemple.
2. Les utilisateurs peuvent en plus interagir avec des objets de l'environnement de manière individuelle. Il est par exemple possible à un utilisateur d'attraper et de déplacer un objet ou de changer sa couleur. la collaboration se fait à ce niveau encore par la communication entre les utilisateurs, mais elle n'a pas une vocation uniquement sociale. Les utilisateurs peuvent en effet communiquer pour se mettre d'accord sur des transformations à opérer sur l'environnement, en vue d'atteindre un but donné. C'est le cas de Calvin [LJ96], une application de conception architecturale permettant à plusieurs utilisateurs de s'immerger dans un même environnement afin de modifier et de visualiser en direct l'apparence d'une scène d'intérieur.
3. Les interactions avec les objets de l'environnement peuvent être collaboratives. Il est par exemple possible de manipuler un objet à plusieurs. On distingue en général deux sous-niveaux : (3.1) seules des interventions indépendantes sont permises en simultané sur un même objet (e.g déplacement et changement de la couleur) ou (3.2) le système gère les interventions interdépendantes (e.g en calculant la couleur moyenne lors de deux changements de couleur différents effectués simultanément).

L'apprentissage humain est un des principaux champs d'application de la RV. L'utilisation d'outils virtuels permet en effet de s'entraîner dans des conditions favorables et paramétrables avant de se confronter à des situations réelles, qui peuvent être trop risquées ou trop complexes pour un apprenti [FG06]. Les opérations de maintenance de certaines pièces mécaniques peuvent par exemple s'avérer dangereuses pour une personne inexpérimentée. Dans certaines situations d'apprentissage, le matériel n'est pas toujours disponible ou est très coûteux, il est donc pratique d'avoir à disposition du matériel virtuel réservé à l'entraînement, comme des simulateurs de conduite de voitures ou d'avions [FG06]. Le but de l'apprenti est de réaliser un certain nombre de tâches définies par l'entraîneur. Ce dernier évalue les performances de l'apprenti, par exemple en mesurant de temps passé pour la réalisation d'une action ([FMCB06], chapitre 2 p.42).

Les EVC pour l'apprentissage impliquent donc une collaboration entre les apprentis. De plus, un apprenti doit comprendre les directives et les conseils donnés par l'entraîneur, qui doit lui-même

1. Dispositif sous forme de salle à 3, 4, 5 ou 6 murs donnant l'impression d'être plongé dans l'environnement virtuel 3D.

avoir un retour de l'apprenti pour pouvoir évaluer ses performances et comprendre ses difficultés. Les apprentis collaborant doivent quant à eux prendre conscience mutuellement de la présence des uns et des autres, et des opérations qu'ils exécutent. Cette prise de conscience est appelée l'*awareness*. Plus formellement, l'*awareness* peut être définie comme le fait d'avoir une compréhension complète de l'environnement, des personnes qui y travaillent et des interactions que ces personnes ont entre elles et avec l'environnement, afin d'utiliser cette information comme contexte pour ses propres activités et le travail collaboratif [DB92, End95]. Elle est rendue possible grâce à l'utilisation de métaphores, qui sont des représentations symboliques virtuelles d'un concept issu du monde réel [FG06]. Cela permet de faire appel à l'imagination ou à l'expérience d'un utilisateur pour donner une compréhension instantanée d'un événement ou d'un fait (e.g représentation des autres utilisateurs sous forme d'avatars).

Nous nous proposons d'étudier un système d'apprentissage en EVC dans lequel l'entraîneur a la possibilité de modifier les paramètres de l'environnement et d'intervenir en tant que "super utilisateur", c'est-à-dire de collaborer avec un apprenti comme le ferait un autre apprenti "normal" dans l'EVC, mais avec la possibilité d'utiliser des "pouvoirs" supplémentaires. Certaines idées seraient de lui laisser la possibilité d'éditer les caractéristiques du scénario en créant et en supprimant des objets, de se téléporter n'importe où dans l'environnement ou encore de prendre momentanément le contrôle de l'avatar de l'apprenti pour faire des démonstrations. C'est une collaboration asymétrique, c'est à dire une collaboration pour laquelle la perception qu'un utilisateur a de lui-même (niveau 1 de la collaboration) et son champ d'action (niveaux 2 et 3) sont respectivement différents de la perception qu'en ont les autres utilisateurs et de leurs actions, comme le définit l'article [DR14]. Les auteurs y étudient un jeu de chasse au trésor en équipe dans lequel un joueur peut percevoir son avatar "rouge" alors qu'il est vu "gris" par ses coéquipiers, ce qui peut avoir une influence sur la dynamique de l'équipe. Le schéma de principe de la figure 1 représente l'ensemble des paramètres qu'il faut prendre en compte dans un tel contexte. Un apprenti doit comprendre les instructions et les conseils que lui donne son entraîneur. Comme certaines tâches se font en collaboration avec d'autres utilisateurs (dont l'entraîneur peut faire partie), il est nécessaire de faciliter les interactions multiples sur un objet, tant sur le plan de l'*awareness* (pour que chaque utilisateur impliqué soit conscient des interactions des autres) que sur le plan de la synchronisation des actions au niveau du système sous-jacent. Le fait que l'entraîneur puisse être un collaborateur a un impact sur les choix qui seront faits afin de faciliter la collaboration, l'entraîneur ayant besoin d'un retour de l'apprenti et l'apprenti ayant besoin d'être guidé.

Dans une première partie, nous étudierons les différents moyens qu'il existe afin de faciliter l'*awareness* dans trois cas de figure : la compréhension de l'environnement global, le guidage lors de la réalisation d'une tâche et la manipulation d'objets. Ensuite, nous nous pencherons sur les moyens existants pour garantir des interactions entre utilisateurs respectueuses des normes sociales communément admises, notamment le respect d'une certaine zone de discrétion autour de chacun. Enfin, nous étudierons les mécanismes qui permettent de planifier et de synchroniser automatiquement les tâches d'apprentissage en EVC grâce à un scénario de supervision ².

2. A la différence de l'entraîneur qui contrôle des paramètres de l'environnement tels que la luminosité, l'utilisation de certaines métaphores ou la présence ou l'absence de certains objets, le scénario définit l'ordonnancement des tâches. Il peut pour cela activer ou désactiver certaines commandes, dont celles de l'entraîneur. L'entraîneur garde un contrôle sur le scénario en tant qu'éditeur de celui-ci.

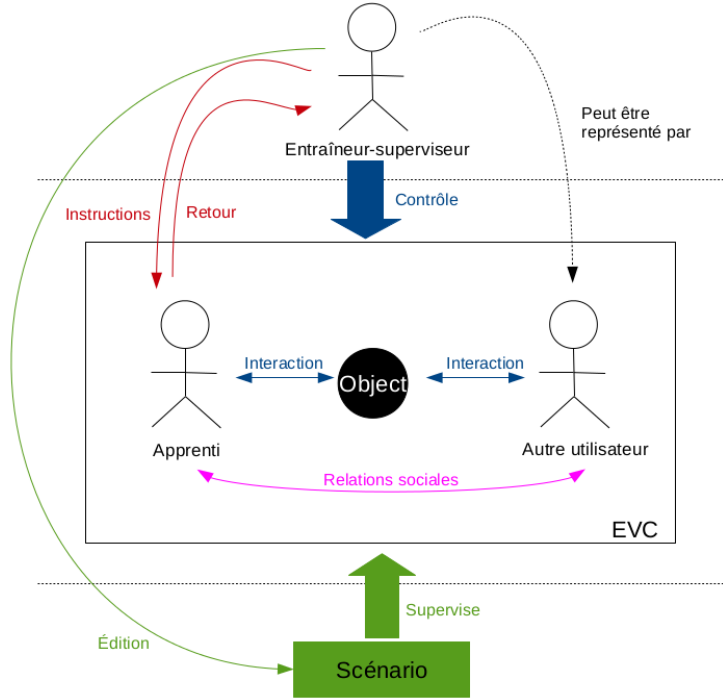


FIGURE 1 – Schéma de principe du système à étudier. **Rouge** : *awareness*, **Bleu** : interaction avec l’environnement, **Vert** : scénario, **Rose** : relations sociales.

2 *Awareness* et interactions

Cette section traite de la partie rouge du schéma de principe en figure 1.

2.1 Compréhension de l’environnement et des événements

Un des aspects de l’*awareness* est la compréhension des événements se déroulant dans un environnement virtuel (EV), pour que les utilisateurs sachent ce qu’il se passe autour d’eux. Quelques fois, ces événements doivent être pris en compte pour la réalisation de tâches. Par exemple, l’article [LBLE⁺14] présente un cas d’usage où l’utilisateur doit demander de l’aide à des experts spécifiques à certaines tâches. Il doit pour cela bien comprendre quel expert contrôle quel avatar. Ce qui complique la tâche, c’est le fait que les experts peuvent échanger leurs avatars. Il faut donc bien suivre le déroulement de ces échanges !

Les faits et événements peuvent être regroupés en différentes catégories (voir [ND⁺14]) :

- **les utilisateurs** : il faut pouvoir retranscrire leurs activités. Pour cela, les supports des métaphores sont leurs avatars. A partir de cette représentation d’un utilisateur, il est possible de donner des compléments d’information utiles pour comprendre ses actions. L’article [FBHH99] propose par exemple de représenter le champ de vision sous forme d’un fuseau de demi-droites.
- **l’environnement virtuel** : des objets ou des zones de la scène peuvent se caractériser par leur état : certains objets sont manipulables, d’autres non, d’autres sont en cours de

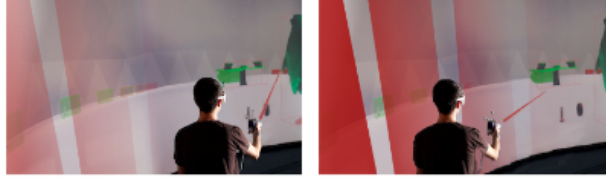


FIGURE 2 – Limites de l’espace physique représentées par un verre semi-transparent devenant de plus en plus visible au fur et à mesure que l’utilisateur s’approche [FCD⁺10].

manipulation. Il peut alors être intéressant de coder cette information par un code couleur, comme cela est proposé dans [PBF02].

- **les interactions** : lorsqu’un utilisateur interagit avec un objet, il peut être utile de représenter cette information par un lien entre l’utilisateur et l’objet. L’article [FBHH99] propose de représenter des bras s’étendant jusqu’à un objet lors d’une saisie (qui peut en réalité se faire à distance, hors de portée de l’utilisateur réel, e.g à l’aide d’un pointeur).
- **l’environnement physique** : les utilisateurs évoluent dans un environnement réel constitué d’obstacles, par exemple les murs dans un système de type CAVE. Il faut au moins diminuer les risques de collision. Une possibilité est d’afficher une barrière virtuelle destinée à dissuader l’utilisateur, comme présenté sur la figure 2.
- **Les erreurs internes** : certaines erreurs du système peuvent introduire des incohérences dans l’EV. C’est notamment le cas lors de problèmes de réseau pour un EVC : si le système chargé de gérer un objet ou un avatar particulier ne communique plus de données, les interactions qu’auraient alors les utilisateurs avec cet avatar ou cet objet seraient incohérentes. Il peut y avoir aussi un déphasage entre ce que voient deux utilisateurs différents. Dans ces deux cas, il est important de se rendre compte de l’anomalie. Par exemple, si un utilisateur distant se déconnecte anormalement alors qu’il était en train d’interagir avec un objet, il peut être intéressant de représenter sur tous les autres sites une vue ”gelée” de l’objet en question, comme proposé par [DZ⁺06].

Par nature, l’évaluation d’une métaphore se fait de manière qualitative. La qualité d’une métaphore se mesure par sa capacité à transmettre une information correcte, compréhensible et complète, ainsi que par son influence sur le sentiment d’immersion éprouvé par les utilisateurs, comme le montre le chapitre 2 de [FMCB06], dont le thème est la conception et l’évaluation d’un environnement virtuel. Une expérience consistant à simuler le comportement de consommateurs dans un magasin y est menée. Pour tirer des conclusions applicables au monde réel, il faut s’assurer que les comportements ne sont pas biaisés par le fait que le magasin est seulement virtuel (e.g, les utilisateurs n’hésitent pas à toucher les produits fragiles car ils savent qu’ils ne peuvent pas être dégradés).

Le ”coût” d’une métaphore doit bien souvent être pris en compte. En général, il se caractérise par la distraction inutile que peut introduire une métaphore, ou l’encombrement de la scène. Par exemple, dans le cas d’un objet vu sous différents états selon les utilisateurs à cause de problèmes liés au réseau, une solution consiste à faire apparaître à côté de l’objet un fantôme par état différent [DZ⁺06]. Cependant, faire apparaître trop d’états peut encombrer l’espace, ce qui n’est pas souhaitable.

Pour illustrer, considérons un jeu de rôle de RV dans lequel les joueurs peuvent se connecter et se déconnecter en cours de partie. Pour prévenir les participants qu'un joueur s'est connecté ou déconnecté, nous pouvons imaginer deux métaphores différentes :

1. l'avatar qui vient d'apparaître ou qui est sur le point de disparaître clignote avec une certaine couleur pendant quelques secondes ;
2. une simple popup apparaît et indique que le joueur s'est connecté ou déconnecté.

Suite à des tests réalisés sur plusieurs personnes, les critères de qualité et de coût peuvent être évalués en soumettant aux participants un ensemble d'affirmations (e.g "La métaphore est facile à comprendre" ou "La métaphore distrait de la tâche en cours") qui doivent être notées par un degré de satisfaction allant de 1 à 5. Les résultats obtenus peuvent par exemple révéler que la première métaphore procure une meilleure impression d'immersion mais est trop distrayante alors que la deuxième facilite la compréhension de ce qu'il se passe tout en étant moins plaisante.

Dans le cas d'une collaboration asymétrique, il est intéressant de faire varier l'apparence des avatars pour indiquer une différence de rôles ou de capacités entre les utilisateurs. Bien plus qu'une indication, ces différences d'apparence peuvent avoir une influence sur le comportement des utilisateurs, comme le révèle une expérience reportée dans l'article [PHM09] : des volontaires incarnent chacun un chevalier Jedi dans un jeu basé sur l'univers de Star Wars. Des groupes sont formés : un groupe de Jedis revêtant des bures blanches et un autre groupe de Jedis revêtant des bures noires. À part cette différence, tous les avatars ont la même apparence neutre. Chaque groupe doit collaborer séparément afin de décider comment doit être puni le fait d'attaquer un joueur désarmé. L'expérience montre que les Jedis noirs privilégient des peines lourdes alors que les Jedis blancs sont plus cléments.

2.2 Guidage

Lorsqu'un utilisateur doit effectuer une tâche précise, par exemple atteindre une zone cible, il peut être guidé au moyen de métaphores. Ces indications ont pour but d'améliorer la performance d'un utilisateur ou de le diriger rapidement vers les points d'intérêt.

Certains guidages ont un but purement didactique, c'est à dire qu'ils s'utilisent durant une phase d'entraînement pour apprendre à l'utilisateur comment effectuer une tâche. Un bon exemple est celui qui est cité dans l'article [ASKK⁺10], où un entraîneur montre à un apprenti des pièces de moteur de voiture en les mettant en évidence³. L'apprenti subit ensuite une phase d'évaluation au cours de laquelle il tente de retrouver les pièces successivement montrées sans avoir recours à la mise en évidence. D'autres guidages sont inhérents à la tâche à effectuer et le fait de les enlever perdrait l'utilisateur, même très expérimenté. Un exemple est celui qui est reporté dans l'article [ADL⁺09], qui montre une technique de manipulation d'objet avec 6 Degrés De Liberté (DDL) à base de 3 points de contact. Ces points sont représentés par des boules rouges. Leur absence empêcherait d'avoir un repère spatial pour maîtriser correctement la manipulation.

Le critère permettant d'évaluer une métaphore de guidage est son efficacité, c'est à dire sa capacité à rendre la tâche plus facile. Les paramètres que l'on recherche à minimiser sont le temps

3. La technique est alors un peu particulière car elle consiste simplement à rendre invisibles ou transparents les obstacles qui occultent la pièce du champ de vision de l'apprenti.

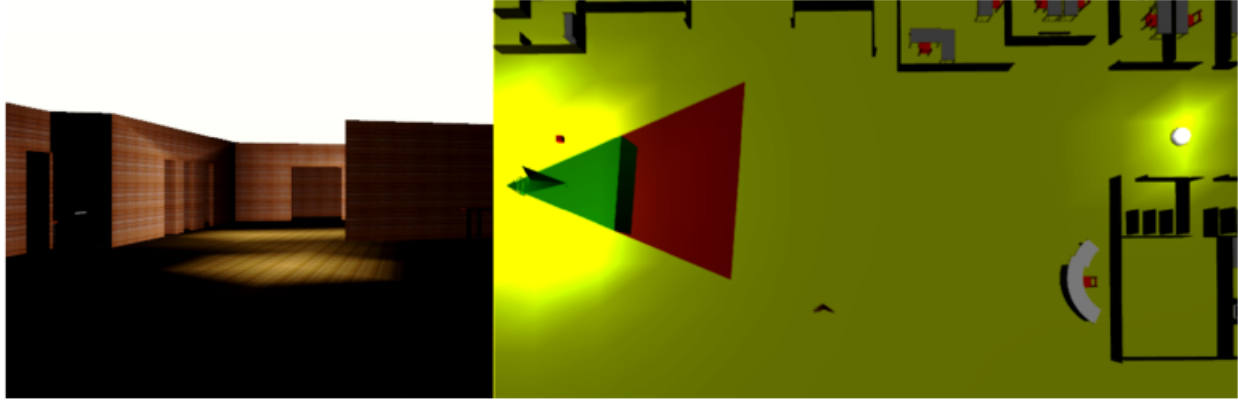


FIGURE 3 – Éclairage d’un chemin et assombrissement des alentours [DNF⁺13]. Vue de l’apprenti (à gauche) et de l’entraîneur (à droite).

d’exécution de la tâche ([FMCB06], chapitre 2 p.42) et le niveau de stress ou de fatigue que la métaphore induit. Par exemple, [ASKK⁺10] teste l’influence de l’absence de métaphore pour mettre en évidence les objets, ce qui oblige l’apprenti à se déplacer jusqu’à l’entraîneur pour regarder par-dessus son épaule. Ceci introduit une gêne due à la proximité (voir la section 3).

Dans le cas de métaphores de guidage introduites par un entraîneur, la perception que celui-ci a de son apprenti et du contexte dans lequel celui-ci se trouve est également à prendre en compte. En effet, l’entraîneur peut vouloir adapter dynamiquement la métaphore pour rendre la tâche plus aisée ou au contraire plus difficile, selon les performances de l’apprenti. Il est également possible d’envoyer des métaphores visuelles permettant de donner à l’apprenti des indications sur ses performances. Il faut alors représenter les informations dont à besoin l’entraîneur. Un cas d’usage est donné dans [DNF⁺13] : un apprenti est immergé dans un EVC où il doit trouver des cibles. L’entraîneur est quant à lui en face d’une carte interactive de l’EVC sur laquelle il peut observer la cible, l’utilisateur, son champ de vision et les métaphores de guidage testées une à une : un ensemble de flèches traçant le chemin, une boussole accompagnant l’apprenti et une source de lumière provenant de la cible. L’entraîneur peut zoomer sur la carte et même emprunter le point de vue de l’apprenti. Il peut lui envoyer des signaux de couleur pour lui indiquer s’il se dirige dans la bonne ou la mauvaise direction. Il peut aussi adapter les paramètres de la métaphore de guidage, par exemple en ajoutant des flèches de guidage. La figure 3 montre un exemple de vue de l’apprenti (à gauche) et de l’entraîneur (à droite). Notons que l’influence des métaphores sur l’efficacité de la tâche de l’entraîneur (l’évaluation) n’a pas été étudiée dans cet article.

2.3 Manipulation d’objets

Cette section concerne la partie bleue du schéma de principe en figure 1.

2.3.1 Exploitation des 6DDL

L’un des problèmes les plus étudiés en manipulation d’objet est l’exploitation des 6DDL (translations et rotations selon les trois axes du repère 3D de l’objet). En effet, la saisie d’objets se fait

en général avec des appareils bien moins complexes que nos deux mains, avec de simples capteurs de position par exemple. L'interaction n'est pas aussi naturelle que dans le monde réel. Il est par exemple parfois difficile d'opérer une rotation autour de l'axe défini par les deux mains qui maintiennent un objet virtuel, comme c'est le cas pour la méthode citée dans [CFH97]. Il faut donc permettre d'exploiter les 6DDL aisément. Une solution est proposée par [Ngu14] (chapitre 2). Elle consiste à définir sept points de contact entre l'opérateur et l'objet. Ces points sont en réalité décorrélés de l'objet et placés sur un outil, le *7-Handle*, constitué d'un triangle. Les sept points de contact sont les sommets du triangle (points principaux), les milieux des segments (points secondaires) et son centre de gravité (point tertiaire). L'outil est associé avec l'objet à manipuler en mettant en correspondance leur centre de gravité. La manipulation se fait alors en deux phases :

1. **Configuration** : il est possible d'adapter la taille du *7-Handle* et la position des points en fonction de l'objet à manipuler.
2. **Manipulation** : l'utilisateur peut manipuler les points du *7-Handle*, chacun d'entre eux asservissant un point correspondant de l'objet. Il est possible de verrouiller un ou plusieurs points principaux, qui cessent alors de suivre le mouvement en adoptant une position fixe. Cela permet de décomposer un mouvement complexe en une succession de mouvements plus simples (rotations autour d'un point, d'un axe, ou de simples translations). Par addition de ces mouvements contraints, les 6DDL peuvent être utilisées facilement. Les points secondaires et le point tertiaire peuvent être utilisés à condition qu'aucun point adjacent ne soit déverrouillé. Dans le cas contraire, ces points sont désactivés (on ne peut plus les saisir mais ils suivent le mouvement). la figure 4 permet de mieux comprendre le fonctionnement du *7-Handle*.

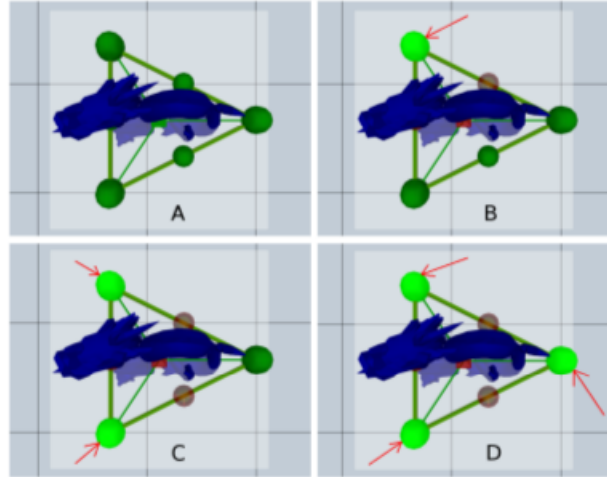


FIGURE 4 – *7-Handle* [Ngu14]. (A) Tous les points de contact sont verrouillés (vert foncé), l'objet ne peut pas bouger. (B) Un point de contact est déverrouillé (vert clair), la rotation autour de l'axe opposé est possible. 2 points secondaires et le point tertiaire sont désactivés (rouge). (C) les rotations autour du point verrouillé sont possibles. (D) Tous les mouvements sont possibles.

Bien sûr, il est possible de faire jouer le rôle du verrou par un humain, qui se contenterait de maintenir un ou deux des trois points de contact principaux pendant qu'un autre opérerait les

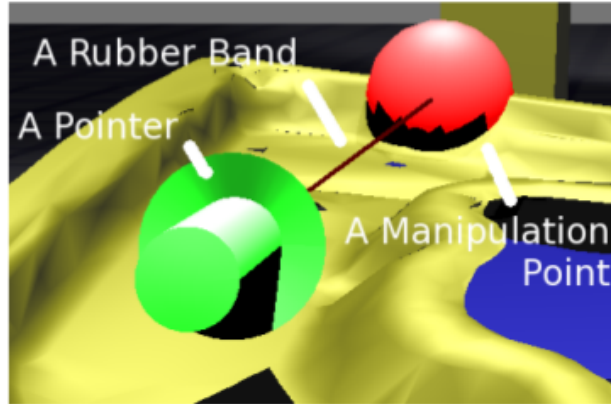


FIGURE 5 – Points de contact et pointeurs d'utilisateur [ADL⁺09].

manipulations qu'il souhaite par le biais des points de contact restants. Cette solution est proposée dans [ADL⁺09]. Cependant, elle n'est viable qu'en environnement collaboratif.

2.3.2 Prise en compte de l'influence d'un coopérateur

Lorsque dans la vraie vie deux personnes déplacent un meuble lourd, chacune ressent les forces exercées par l'autre. C'est ce que l'on appelle le retour haptique (ou retour de force). Cela permet d'adapter son effort à celui de l'autre pour équilibrer le meuble. En EVC, le retour haptique n'est possible qu'avec certains appareils. Le bras à retour d'effort est l'un de ceux qui fournissent les retours les plus réalistes, mais il est très coûteux. Il existe des appareils plus accessibles et moins réalistes, comme la Wiimote de Nintendo.

Un problème connexe est de connaître le positionnement d'un opérateur par rapport à l'objet, autrement dit les points de contact. En effet, dans certains cas, l'utilisateur contrôle les points de contact au moyen de pointeurs, les mains réelles ne sont alors pas en contact direct avec l'objet 3D. On peut avoir recours à une représentation virtuelle d'un pointeur, reliée au point de contact et indiquant la direction de l'opérateur ainsi que son identité (par un code couleur), comme c'est le cas dans [ADL⁺09] (voir figure 5). La proximité entre l'appareil et le point de contact est ici également représentée par un code couleur (vert si l'appareil est éloignée, rouge s'il est proche). En effet, une main éloignée du point de contact crée beaucoup d'instabilités.

3 Aspects sociaux

Comme vu en section 2.2, la gêne induite par une métaphore est à minimiser (partie rose de la figure 1). En EVC immersif, il existe une gêne particulière liée à la disposition des acteurs dans l'espace : chaque individu définit une zone "intime" autour de lui, dont le rayon varie entre 15cm et 40cm selon les personnes. En règle générale, si une personne pénètre cette zone, cela provoque une réaction de l'individu, qui va d'un simple recul à un énervement. Cette notion a été introduite par Edward T. Hall dans l'ouvrage [Hal69]. Cette conception de l'espace s'appelle la proxémique. Elle est mise en jeu si deux acteurs physiques se retrouvent dans le même espace immersif.

En EVC, et particulièrement dans les applications d'apprentissage, il existe plusieurs cas qui impliquent une forte proximité physique entre deux acteurs : un entraîneur peut avoir besoin de prendre le point de vue de son apprenti si celui-ci demande de l'aide, un apprenti lui-même peut avoir besoin d'observer son entraîneur lors d'une démonstration, deux collaborateurs peuvent avoir besoin de manipuler un même petit objet en même temps... en guise d'exemple, l'article [ASKK⁺10] présente un cas d'usage mettant en jeu d'une part un apprenti devant apprendre les positions de pièces dans un moteur de voiture, et d'autre part son entraîneur qui lui montre lesdites pièces. Dans certains cas, l'apprenti doit observer par dessus l'épaule de l'entraîneur, ce qui provoque effectivement une gêne pour ce dernier.

Pour le problème de la manipulation, la solution est d'utiliser des points de contact distants de l'objet (voir la section 2.3). Pour le problème de l'observation, la solution au problème consiste à utiliser un pointeur projetant un rayon de lumière virtuel pour mettre en évidence l'objet. Comme la nécessité de regarder par dessus l'épaule est en général due à un obstacle qui bloque le champ de vision, le rayon doit effacer partiellement l'obstacle. L'article [DWE03] donne quelques exemples de méthodes, notamment le *cutaway*, qui consiste à faire une coupe dans l'obstacle pour observer ce qui se trouve derrière. Ces techniques n'améliorent pas l'efficacité de l'apprentissage, mais limitent au moins la gêne.

4 Gestion automatique de l'apprentissage et des interactions collaboratives

4.1 Scénario d'apprentissage

En EVC, les utilisateurs ont en général un but à atteindre qui nécessite l'exécution de plusieurs tâches successives. Cependant, tout ne doit pas nécessairement s'opérer de manière linéaire. L'ensemble des utilisateurs peut se scinder en plusieurs groupes qui réaliseront des tâches en parallèle et des choix peuvent être offerts entre plusieurs tâches possibles... ce qui génère des embranchements complexes. On appelle cela un scénario (partie verte de la figure 1), qu'il faut formaliser afin de gérer efficacement l'enchaînement des tâches.

Il existe principalement deux définitions du concept de scénario : ce peut être vu comme une séquence de buts à atteindre [CGBA14] (atteindre un but permet de débloquer la tâche suivante) ou comme un ensemble de règles contraignant les interactions. Dans ce dernier cas, l'utilisateur est laissé libre de ses mouvements mais le scénario n'avance que s'il exécute des actions dans un certain ordre. S'il se trompe d'action, une indication peut lui être envoyée pour le prévenir qu'il s'est trompé [PHM⁺03]. Cette approche a l'inconvénient d'être moins précise, car c'est à un superviseur réel ou virtuel de décider si une action est conforme ou non au scénario. L'apprenti est ainsi installé dans un "bac à sable" et on veut qu'il trouve de lui-même la bonne séquence d'actions à réaliser.

Dans l'approche "séquence de buts", le cycle de vie d'une tâche est constitué des étapes successives "définition, exécution, conséquence". Pour illustrer, considérons un scénario simple constitué de deux tâches : un utilisateur doit placer une pièce sur un support, puis lui appliquer des transformations. Le moteur de scénario prépare d'abord les commandes pour la première tâche (définition) :

des *handlers*⁴ de manipulation (haut, bas, gauche, droite, en avant, en arrière) sont mis en place pour déplacer la pièce à l'aide d'une machine par exemple. Puis l'opérateur exécute les commandes une à une afin de placer la pièce sur le support (exécution). Une fois le but atteint, le système désactive les *handlers* (conséquence) et prépare les commandes pour la tâche suivante (définition) et ainsi de suite.

L'approche "bac à sable" peut être intéressante pédagogiquement. Cependant, elle ne décrit pas les tâches avec précision, ce qui rend le scénario moins "divisible". Il est alors plus difficile de mesurer la performance de l'utilisateur sur chaque tâche comme nous le souhaiterions. C'est pour cela que nous privilégions l'approche "séquence de buts". Néanmoins, il est intéressant d'avoir recours à des tâches "bac à sable" pour ne pas trop contraindre l'utilisateur lors de l'exécution de ces tâches. Le système séquentiel StoryNets [SHG⁺06] permet de définir ce type de tâches.

Selon cette approche, les scénarios sont souvent modélisés sous forme de machine à états [CKP95]. Les états représentent chacun une tâche et les transitions sont gardées par les conditions de fin de tâche. Cependant, cette représentation a l'inconvénient de ne pas définir directement la notion d'états courants⁵. Pour cela il faut avoir recours à des visiteurs du graphe.

La manière de représenter les événements de transition d'un état au suivant varie d'un système à l'autre. Elle est en général fortement corrélée à la définition des commandes, un événement correspondant à l'exécution d'une commande. Cette approche est contraignante car elle rend le moteur de scénario dépendant du système de commandes. StoryNets est par exemple basé sur des commandes vocales. Le type d'événement est donc très spécifique. Or, il peut être intéressant de se libérer de ce genre de contrainte pour définir des événements sur mesure, directement liés au scénario (e.g définir une zone qu'un utilisateur doit atteindre afin de passer à la tâche suivante).

D'autres modèles se basent sur les réseaux de Petri, comme #SEVEN⁶ [CGBA14], car ils ont l'avantage de fournir la notion d'états courants grâce aux tokens. Un exemple simple permettant de comprendre la signification exacte des tokens consiste à considérer que chaque token représente l'un des terminaux physiques impliqués dans l'application collaborative. Cela permet de paralléliser l'exécution du scénario en fournissant à chaque terminal informatique des actions différentes. Par exemple, lors d'une opération d'assemblage de deux pièces, l'un des terminaux permettra de déplacer une seule des deux pièces et réciproquement. Quant aux places du réseau de Petri, elles représentent les tâches du scénario et les transitions représentent les événements permettant le passage à une tâche suivante. #SEVEN introduit en plus la notion de capteurs et d'effecteurs, qui sont attachés aux transitions : le capteur d'une transition écoute l'environnement en attendant la validité de la condition de fin de tâche (e.g la pièce a atteint son support). Une fois celle-ci validée, il active la transition qui fait passer les tokens aux états suivants, puis l'effecteur s'active en produisant les conséquences de l'action sur l'environnement (e.g désactiver les commandes devenues inutiles et activer celles qui correspondent aux états suivants).

4. Entité logicielle qui permet de détecter une requête d'interaction et qui en gère les paramètres. Un *handler* peut être matérialisé dans l'EV par une boule de couleur par exemple (voir la figure 5).

5. pluriel : dans un scénario il peut y avoir plusieurs états courants car des tâches peuvent être effectuées en parallèle.

6. La vidéo <https://vimeo.com/104937822> illustre parfaitement #SEVEN en action et permet de bien comprendre son fonctionnement.

#SEVEN utilise en réalité une hiérarchie de réseaux de Petri : les places du réseau principal sont en fait des réseaux de Petri fils. En effet, une tâche peut elle même être décomposée en sous-tâches. Par exemple, le fait de déplacer une pièce au moyen des commandes *haut*, *bas*, *gauche*, *droite*, *en avant*, *en arrière* peut être vu comme une succession en boucle de tâches correspondant chacune à l'une de ces commandes. Cela permet notamment de définir des tâches "bac à sable". Chaque tâche principale est ainsi décomposée en un réseau de Petri constitué de places initiales et de places finales. Lorsqu'un token atteint une place principale, celle-ci n'est pas immédiatement considérée atteinte. D'abord, le token est dupliqué sur les places initiales du sous-réseau. Lorsque les places finales sont toutes atteintes par un token, la place principale reçoit effectivement le token original.

Ce modèle propose également une formalisation de la répartition des tâches entre les acteurs. Chacun d'entre eux est associé à un token. Chaque token est assigné à un ensemble d'actions possibles (les commandes activées), qui définissent ce que l'on appelle un rôle. Ainsi, chaque acteur hérite du rôle du token auquel il appartient. Les effecteurs peuvent modifier l'ensemble des actions qui caractérisent un token afin de faire évoluer les rôles.

Le réseau de Petri proposé en figure 6 représente un scénario simple géré par #SEVEN.

Deux collaborateurs (U1 et U2) doivent assembler deux composants (pièces A et B) puis réaliser des finitions. La table de montage doit être laissée propre. Les deux utilisateurs commencent par transporter chacun une pièce sur la table de montage puis collaborent pour effectuer le montage. A la fin, l'un d'eux s'occupe des finitions tandis que l'autre nettoie la table de montage.

4.2 Synchronisation des interactions collaboratives

La gestion de la collaboration ne peut pas se passer de mécanismes de synchronisation. En effet, la transmission des événements sur un réseau peut conduire à des ordres différents de réception et donc à des incohérences entre les états de chaque processus. Selon le niveau de collaboration⁷ de l'application, les mécanismes de synchronisation ne sont pas les mêmes. Au premier niveau, qui ne permet que la communication entre les utilisateurs, il n'y a pas besoin de mécanismes particuliers. Au deuxième niveau, permettant des interactions individuelles avec les objets, il faut pourvoir ces derniers de verrous qui empêchent l'intervention de plus d'une personne. Le troisième niveau, permettant des interactions simultanées sur un même objet, est traité dans l'article [MAP99]. Il faut garantir deux propriétés :

1. Tous les processus doivent s'accorder sur l'ordre d'exécution des opérations sur un même objet.
2. Tous les processus doivent s'accorder sur la simultanéité des opérations sur un même objet, c'est à dire que si un processus considère que deux opérations ont eu lieu simultanément, alors tous les processus doivent considérer la même chose. Cette propriété est utile pour le niveau 3.2, qui permet les interactions simultanées interdépendantes.

Avant tout, il faut définir ce qu'est une opération sur un objet. L'article [MAP99] la définit comme étant le changement de valeur d'un *handler*. Ainsi les opérations peuvent être associées à des événements ordonnés dans le temps.

7. Rappel : les différents niveaux de collaboration sont définis en introduction.

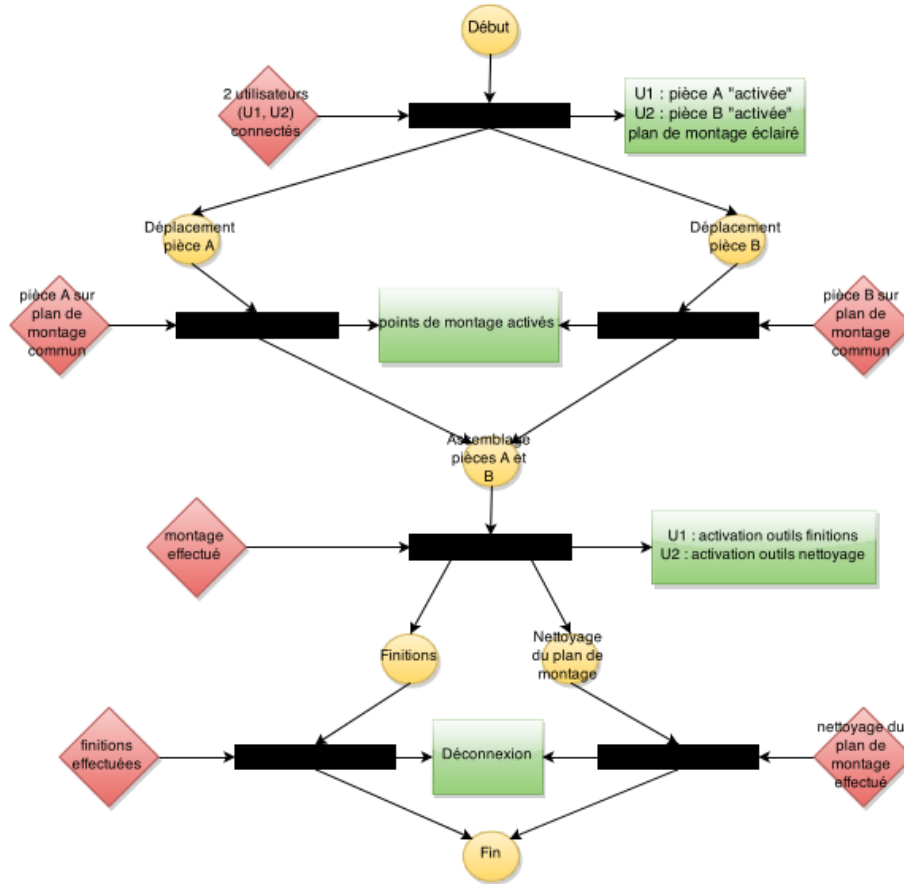


FIGURE 6 – Exemple de scénario #SEVEN (réseau de Petri). Losanges : capteurs. Rectangles : effecteurs. Cercles : places.

La première propriété est garantie à l'aide d'une contrainte qui doit être imposée sur les messages échangés sur le réseau⁸ et dont se charge l'objet concerné : il faut que les messages soient diffusés en ordre complet, c'est à dire que tous les processus les reçoivent dans le même ordre.

La deuxième propriété s'obtient grâce à un mécanisme d'inscription / désinscription au moment où un utilisateur commence à interagir avec un objet (resp. termine son interaction). A chaque exécution d'opération, l'objet considère que tous les utilisateurs inscrits sont en activité simultanée et opère d'abord la diffusion d'un message contenant le nombre d'inscrits. Ensuite, à chaque opération effectuée, la valeur du *handler* correspondant est à son tour diffusée. Chaque processus attend donc le nombre d'opérations correspondant au nombre reçu au préalable, avant de calculer la transformation résultante. Si un utilisateur reste trop longtemps "détenteur" d'un *handler* sans pour autant réaliser d'interaction, la valeur du *handler* est considérée constante et un signal spécial est diffusé afin de ne pas faire attendre inutilement les processus.

8. Un message représente alors une opération.

5 Conclusion

Nous nous sommes proposé d'étudier un système d'apprentissage en EVC. Nous avons montré que cela nécessitait de prendre en compte les problèmes liés à l'*awareness* : l'apprenti doit comprendre les instructions de son entraîneur, qui doit comprendre le retour de son apprenti. Il faut aussi comprendre les événements ayant lieu dans l'environnement, notamment les événements dus à l'intervention de l'entraîneur. L'*awareness* est rendue possible grâce à l'utilisation de métaphores qu'il faut évaluer selon leur influence sur la qualité et la rapidité d'exécution des tâches, l'impresion d'immersion qu'elles procurent, ainsi que sur leur simplicité. En particulier, des métaphores notifiant des événements divers ne doivent pas distraire l'utilisateur de sa tâche principale et des métaphores visant la facilitation de tâches ne doivent pas induire de stress. Il faut aussi respecter les normes sociales qui s'imposent naturellement entre les utilisateurs afin de ne pas provoquer de gêne.

De plus, nous avons montré que la manipulation collaborative nécessitait l'utilisation de métaphores spécifiques afin que chaque utilisateur soit conscient de l'influence des autres sur un objet manipulé, notamment par la représentation des points de contact. Pour renforcer le réalisme des manipulations, il faut pouvoir exploiter les 6DDL par le biais d'un outil spécifique, le *7-Handle*, ou plus simplement par l'utilisation de 3 points de contact. Toujours dans un souci de réalisme et d'utilisabilité, il est nécessaire d'assurer un retour synchrone de l'état de l'EVC entre tous les utilisateurs. cela peut se faire par un mécanisme d'inscription / désinscription des utilisateurs pour l'interaction avec un objet. Enfin, il a été montré comment superviser la succession des tâches d'apprentissage grâce à l'utilisation d'un scénario gérant la notion de rôles, qui peuvent être changés dynamiquement au cours de l'apprentissage.

Cependant, la perception que l'apprenti a de son entraîneur "super-utilisateur" est à notre connaissance peu étudiée. Beaucoup d'articles étudient les perceptions qu'ont des utilisateurs "égaux" entre eux, mais pas celle qu'implique la présence d'un super-utilisateur aux pouvoirs surnaturels, comme la possibilité d'apparaître et de disparaître à volonté. L'article [ASKK⁺10] par exemple, n'étudie que le cas d'un entraîneur physiquement présent. Dans le cas d'un entraîneur représenté par un avatar, le problème de violation de l'espace intime reste à étudier. D'autres métaphores peuvent avoir un effet intrusif a priori. Par exemple, on peut imaginer que l'entraîneur prenne le contrôle de l'avatar d'un apprenti pour lui faire une démonstration, ou que des mains virtuelles apparaissent aux côtés de l'apprenti. De telles métaphores sont susceptibles d'introduire une gêne qu'il faut évaluer. Il faut veiller à ce que les pouvoirs de l'entraîneur "n'effraient pas" l'apprenti, en particulier sa faculté d'apparaître et de disparaître à chaque instant. Cela nécessite d'introduire des métaphores permettant à l'apprenti de comprendre un phénomène "surnaturel" dans l'environnement. Cela peut également permettre d'éviter des erreurs d'interprétation. Par exemple, le fait de voir disparaître un objet subitement peut être interprété comme une erreur interne, alors qu'il s'agit d'une action de l'entraîneur. Un autre problème sera de faire en sorte que l'avatar de l'entraîneur permette bien d'identifier celui-ci et de ne pas le confondre avec celui d'un utilisateur "classique". La perception que l'entraîneur a de l'apprenti est aussi à étudier. En effet, les métaphores permettant à l'entraîneur de se rendre compte des conditions dans lesquelles se trouve l'apprenti ne sont en général pas évaluées, comme la métaphore présentée en figure 3, en section 2.2. Il pourrait être intéressant de comparer plusieurs possibilités, comme une vue de dessus ou l'adoption du point de vue de l'utilisateur par exemple. Nous tenterons de répondre à ces questions grâce à l'application *Sailor vs Poseidon*, qui intégrera un apprenti et un entraîneur super-utilisateur.

Références

- [ADL⁺09] Laurent Aguerreche, Thierry Duval, Anatole Lécuyer, et al. Short paper : 3-hand manipulation of virtual objects. *JVRC 2009*, 2009.
- [AFT03] Bruno Arnaldi, Philippe Fuchs, and Jacques Tisseau. Chapitre 1 du volume 1 du traité de la réalité virtuelle. *Les presses de l'école de Mines de Paris*, 2003.
- [ASKK⁺10] Fernando Argelaguet Sanz, André Kunert, Alexander Kulik, Bernd Froehlich, et al. Improving co-located collaboration with show-through techniques. 2010.
- [CFH97] Lawrence D Cutler, Bernd Fröhlich, and Pat Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 107–114. ACM, 1997.
- [CGBA14] Guillaume Claude, Valérie Gouranton, Rozenn Bouville Berthelot, and Bruno Arnaldi. Short paper :# seven, a sensor effector based scenarios model for driving collaborative virtual environment. In *ICAT-EGVE, International Conference on Artificial Reality and Telexistence, Eurographics Symposium on Virtual Environments*, pages 1–4, 2014.
- [CKP95] James Cremer, Joseph Kearney, and Yiannis Papelis. Hcsm : a framework for behavior and scenario control in virtual environments. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 5(3) :242–267, 1995.
- [CNSD⁺92] Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. The cave : audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6) :64–72, 1992.
- [DB92] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114. ACM, 1992.
- [DNF⁺13] Thierry Duval, Thi Thuong Huyen Nguyen, Cédric Fleury, Alain Chauffaut, Georges Dumont, and Valérie Gouranton. Improving awareness for 3d virtual collaboration by embedding the features of users' physical environments and by augmenting interaction tools with cognitive feedback cues. *Journal on Multimodal User Interfaces*, pages 1–11, 2013.
- [DR14] Ignacio X Dominguez and David L Roberts. Asymmetric virtual environments : Exploring the effects of avatar colors on performance. 2014.
- [DWE03] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Interactive cutaway illustrations. In *Computer Graphics Forum*, volume 22, pages 523–532. Wiley Online Library, 2003.
- [DZ⁺06] Thierry Duval, Chadi Zammar, et al. Managing network troubles while interacting within collaborative virtual environments. In *CSAC*, pages 85–94, 2006.
- [End95] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors : The Journal of the Human Factors and Ergonomics Society*, 37(1) :32–64, 1995.
- [FBHH99] Mike Fraser, Steve Benford, Jon Hindmarsh, and Christian Heath. Supporting awareness and interaction through collaborative virtual interfaces. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 27–36. ACM, 1999.

- [FCD⁺10] Cédric Fleury, Alain Chauffaut, Thierry Duval, Valérie Gouranton, Bruno Arnaldi, et al. A generic model for embedding users' physical workspaces into multi-scale collaborative virtual environments. In *ICAT 2010 (20th International Conference on Artificial Reality and Telexistence)*, 2010.
- [FG06] Philippe Fuchs and Pascal Guitton. Introduction à la réalité virtuelle. *Le traité de la réalité virtuelle-3ème édition*, pages Volume–4, 2006.
- [FMCB06] Philippe Fuchs, Guillaume Moreau, Sabine Coquillart, and Jean-Marie Burkhardt. *Le traité de la réalité virtuelle*, volume 2. Presses des MINES, 2006.
- [GB95] Chris Greenhalgh and Steven Benford. Massive : a collaborative virtual environment for teleconferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3) :239–261, 1995.
- [Hal69] Edward T Hall. *The hidden dimension*, volume 1990. Anchor Books New York, 1969.
- [LBLE⁺14] Thomas Lopez, Rozenn Bouville, Emilie Loup-Escande, Florian Nouviale, Valérie Gouranton, and Bruno Arnaldi. Exchange of avatars : Toward a better perception and understanding. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4) :644–653, 2014.
- [LJ96] Jason Leigh and Andrew E Johnson. Calvin : an immersimedia design environment utilizing heterogeneous perspectives. In *Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on*, pages 20–23. IEEE, 1996.
- [MAP99] David Margery, Bruno Arnaldi, and Noel Plouzeau. *A general framework for cooperative manipulation in virtual environments*. Springer, 1999.
- [ND⁺14] Thi Thuong Huyen Nguyen, Thierry Duval, et al. A survey of communication and awareness in collaborative virtual environments. In *2014 International Workshop on Collaborative Virtual Environments (3DCVE)*, 2014.
- [Ngu14] Thi Thuong Huyen Nguyen. *Proposition of new metaphors and techniques for 3D interaction and navigation preserving immersion and facilitating collaboration between distant users*. PhD thesis, INSA Rennes, 2014.
- [PBF02] Márcio S Pinho, Doug A Bowman, and Carla MDS Freitas. Cooperative object manipulation in immersive virtual environments : framework and techniques. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 171–178. ACM, 2002.
- [PHM⁺03] Michal Ponder, Bruno Herbelin, Tom Molet, Sebastien Schertenlieb, Branislav Ulicny, George Papagiannakis, Nadia Magnenat-Thalmann, and Daniel Thalmann. Immersive vr decision training : telling interactive stories featuring advanced virtual human simulation technologies. In *Proceedings of the workshop on Virtual environments 2003*, pages 97–106. ACM, 2003.
- [PHM09] Jorge Peña, Jeffrey T Hancock, and Nicholas A Merola. The priming effects of avatars in virtual settings. *Communication Research*, 36(6) :838–856, 2009.
- [SHG⁺06] William Swartout, Randall Hill, Jonathan Gratch, W Lewis Johnson, Chris Kyriakakis, Catherine LaBore, Richard Lindheim, Stacy Marsella, David Miraglia, and Ben Moore. Toward the holodeck : Integrating graphics, sound, character and story. Technical report, DTIC Document, 2006.