



HAL
open science

TEAMViz: systèmes multi-dispositifs, multi-vues et multi-surfaces pour la visualisation interactive de scènes 3D

Rami Ajaj

► **To cite this version:**

Rami Ajaj. TEAMViz: systèmes multi-dispositifs, multi-vues et multi-surfaces pour la visualisation interactive de scènes 3D. Informatique [cs]. Université Paris Sud - Paris XI, 2009. Français. NNT : . tel-00617855

HAL Id: tel-00617855

<https://theses.hal.science/tel-00617855>

Submitted on 30 Aug 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



TEAMViz : systèmes multi-dispositifs, multi-vues et multi-surfaces pour la visualisation interactive de scènes 3D

THESE

Soutenue publiquement le 16 novembre 2009

pour l'obtention du

Doctorat de l'Université Paris-Sud 11

(spécialité informatique)

par

Rami Ajaj

LIMSI-CNRS

Université Paris-Sud 11 - Orsay

Composition du jury :

<i>Présidente du jury</i>	Laurence Nigay, Professeur, Université Joseph Fourier, Grenoble
<i>Rapporteurs</i>	Christophe Chaillou, Professeur, Université des sciences et technologies de Lille Gilles Coppin, Professeur, Télécom Bretagne, Brest
<i>Examineur</i>	Morten Fjeld, Maître de conférence, Université technologique de Chalmers, Suède
<i>Directeur de thèse</i>	Christian Jacquemin, Professeur, Université Paris-Sud 11, Orsay
<i>Co-directeur de thèse</i>	Frédéric Vernier, Maître de conférence, Université Paris-Sud 11, Orsay

Remerciements

La vie est une succession d'expériences. Certaines d'entre elles sont positives tandis que d'autres négatives mais toutes laissent des traces dans notre vie. La thèse est une expérience exceptionnelle pour ses hauts et ses bas, ses voyages en conférences et ses périodes de travail intenses pour une deadline ... Je n'aurais jamais réussi à arriver au bout de ma thèse sans l'aide et le soutien direct et indirect de plusieurs personnes.

Je tiens à remercier premièrement Christian Jacquemin et Frédéric Vernier pour m'avoir donné l'opportunité de réaliser cette thèse, pour leurs conseils, pour leur aide et surtout pour leur patience avec moi. Leurs qualités personnelles et professionnelles m'ont permis de faire le lien entre leurs deux thématiques de recherche tout en ayant une certaine autonomie dans mon travail. Les fréquentes discussions scientifiques (et non scientifiques) que nous avons m'ont donné goût à la recherche et m'ont fait apprécier mon travail. Je remercie Christian pour l'encre qui coulait des versions du manuscrit et des articles qu'il corrigeait et pour ses réponses à mes mails à 2h du matin. Je remercie Fred pour les discussions sur des idées de techniques d'interaction très originales.

Un grand merci aux membres du jury d'avoir accepté de rapporter et d'examiner ma thèse. Je les remercie également pour les questions soulevées durant la soutenance et pour leurs nombreux conseils.

Je remercie également tous mes collègues et amis pour les fructueuses discussions, leur soutien, la bonne ambiance et les bons moments passés ensemble. Certains de mes collègues sont devenus des amis proches. Je remercie spécialement : les AMIs, Anne (la petite), Céline (Cend), François (Fend), François, les Gaetans, Guillaume, John, Juju, Kishman, Marc, Mathieu, Matt (ShaderMan), Sonia, les Sylvains, Tifanie, Wai et plein d'autres.

Sans ma famille je n'aurais pas eu l'occasion de réaliser mon parcours universitaire. Je

la remercie ainsi que la famille d'Anja pour m'avoir soutenu dans mes choix, de m'avoir soutenu malgré les distances qui nous séparent et pour la belle ambiance familiale à chaque fois que nous nous retrouvions. Choukran aala kelshi. Dziękuję za wszystko. Thank you for everything. Merci pour tout. Dankeschön für alles.

Mon dernier remerciement est dédié à ma chérie Anja qui m'a soutenu durant les trois années de thèse. Je te remercie d'avoir été à mes cotés dans les moments de bonheurs et les moments difficiles, d'avoir supporté mes retours tardifs du laboratoire et les weekends studieux. Sans toi je n'aurais jamais réussi. Merci d'exister.

Résumé

Les environnements virtuels sont des espaces immersifs et intuitifs mais la conception des interfaces d'édition et de navigation y est complexe en raison de la diversité des dispositifs d'interaction et des tâches à réaliser et en raison des difficultés d'orientation dans un espace 3D. Les tables tactiles offrent des interactions intuitives mais proposent seulement un nombre limité de degrés de liberté. La réalisation d'une interface multimodale combinant une table tactile, des dispositifs d'entrée complémentaires augmentant le nombre de degrés de liberté et plusieurs vues offrant des points de vue supplémentaires sur la scène permet de mieux couvrir les besoins d'interaction dans des environnements 3D virtuels. Les TEAMViz (Tabletop Environment Augmented with Mural Visualization soit en français environnement tabletop augmenté par une visualisation murale) constituent un sous-ensemble de ce type d'interface. Ils combinent des interactions réalisées au moyen d'une vue 2D d'un environnement virtuel présentée sur une table tactile et des interactions au moyen d'une vue 3D en perspective de ce même environnement projetée sur un écran vertical.

La contribution principale de notre recherche est la réalisation d'une méthode de conception pour la combinaison d'interactions effectuées avec plusieurs dispositifs d'entrée sur des vues multiples présentées sur des surfaces d'affichage distinctes pour des tâches d'édition et de navigation dans un environnement virtuel. Cette méthode de conception prend en compte les relations entre les degrés de liberté des dispositifs d'entrée dans le monde réel, les tâches à réaliser et les relations entre les vues. Elle permet de combiner des interactions réalisées avec des dispositifs dont le couplage fort entre l'entrée et la sortie est requis comme dans le cas des surfaces tactiles.

La première étape de la méthode, appelée conception par mapping, est une conception à haut niveau des combinaisons des interactions basée sur un modèle mettant en relation les degrés de liberté d'un dispositif d'entrée dans le monde physique et les degrés de liberté des tâches souhaitées dans le monde numérique. Cette étape de conception génère une combinatoire élevée de relations dont nous réduisons la taille au moyen de règles d'élimination des combinaisons afin d'éviter la surcharge cognitive des utilisateurs

et de prendre en compte les préférences exprimées lors des expériences. Une évaluation sur une tâche de positionnement et de redimensionnement d'un objet virtuel permet de comparer et de valider les solutions obtenues lors de cette étape de conception.

La seconde étape, appelée conception par composantes logiques, définit des relations entre des dispositifs en entrée et le contrôleur de dialogue qui est une composante de haut niveau articulant les tâches de l'utilisateur, l'interaction et l'accès aux données. Elle repose sur les composantes logiques qui définissent les correspondances directes entre degrés de liberté en entrée et dimensions en sortie et qui doivent permettre de préserver un couplage fort entre les entrées et les sorties.

La combinaison d'interactions sur plusieurs dispositifs en entrée et sur des vues multiples implique une concurrence entre les interactions. La troisième étape de la méthode de conception est la gestion de la concurrence au moyen d'une architecture logicielle centralisée reposant sur un système de partage de ressources communes. La gestion des permissions est accomplie par la modélisation de machines à états. Plusieurs niveaux de granularité sont étudiés allant de la gestion de la concurrence entre les applications connectées aux dispositifs d'entrée et/ou de sortie jusqu'aux degrés de liberté. Une expérience de positionnement d'un objet virtuel et de navigation permet de comparer les deux niveaux de granularité les plus fins des machines à états (niveau objet virtuel ou navigation, et niveau degrés de liberté).

Le chapitre suivant illustre l'évolutivité d'un système de type TEAMViz conçu avec notre espace de conception au moyen de deux études : la première modifiant le repère de la vue 2D tandis que la seconde étend le système existant à un système de visualisation interactive de réflexions sonores dans un environnement virtuel. Ces études mettent en relief la spécificité de chacune des étapes de conception et leur paramétrage sur un cas de conception d'une interface complète.

Le dernier chapitre de cette thèse présente une étude centrée sur la navigation dans un système de type TEAMViz. Il se focalise sur la relation dans le monde physique entre les deux vues et sur la limitation des efforts cognitifs nécessaires pour les mettre en correspondance. Deux des huit techniques de navigation présentées dans ce chapitre sont développées dans le but de faciliter la compréhension de cette relation entre les vues. L'expérience utilisateur comparant l'une de ces deux techniques à la manipulation directe d'une représentation du point de vue 3D montre toutefois des résultats contradictoires entre les préférences et les performances des utilisateurs qui illustre la difficulté de concevoir un système satisfaisant l'ensemble des contraintes que nous avons identifiées dans notre méthode de conception.

Table des matières

Introduction	1
1 Etude des systèmes pour la conception des TEAMViz	5
1.1 Introduction	5
1.2 3DGUI	5
1.2.1 Dispositifs d'entrée	6
1.2.2 Techniques d'interaction	10
1.3 Tabletop	14
1.3.1 Les types de dispositifs tabletop	15
1.3.2 Avantages et inconvénients de l'utilisation des dispositifs tabletops	18
1.3.3 Tabletop et degrés de liberté	19
1.4 Combinaison de dispositifs d'entrée, de surfaces et de vues multiples	23
1.4.1 Combinaison de dispositifs d'entrées	24
1.4.2 Combinaison de vues multiples	25
1.4.3 Combinaison de surfaces multiples	26
1.4.4 Interfaces 2D/3D	27
1.5 Modèles pour l'interaction	28
1.5.1 Modèles d'architecture logicielle en IHM	29
1.5.2 Modèles pour la multimodalité	33
1.5.3 Modèles pour la réalité augmentée et la réalité mixte	35
1.6 Synthèse	37
1.7 Conclusion	39

2	Du monde physique au monde numérique	41
2.1	Introduction	41
2.2	Modèle physique/numérique	42
2.3	Mapping	43
2.4	Conception par <i>mapping</i>	45
2.4.1	Contraintes	47
2.4.2	Méthodologie de conception	48
2.4.3	Spécification d'un prototype d'interfaces selon la méthodologie de conception	52
2.4.4	Choix retenus et comparaison	56
2.5	Expérimentation	57
2.5.1	Conception de l'expérimentation	57
2.5.2	Conditions de l'expérimentation	65
2.5.3	Résultats et analyses	67
2.5.4	Conclusion de l'expérimentation	77
2.6	Conclusion	77
3	Architecture logicielle de la combinatoire des composantes logiques	79
3.1	Introduction	79
3.2	Les composantes logiques	80
3.2.1	Définition	80
3.2.2	Composantes logiques et modalités	82
3.2.3	Composante logique spatiale	83
3.2.4	Types de composantes logiques	84
3.3	Conception par composantes logiques	85
3.3.1	Définition	85
3.3.2	Du Mapping aux composantes logiques	85
3.3.3	Règles de conception de composantes logiques spatiales	88
3.4	Conception d'un simulateur de char d'assaut par composantes logiques	95
3.5	Conclusion	101

4	Contribution à la gestion de la concurrence par un modèle de machines à états	105
4.1	Introduction	105
4.2	Enjeux	105
4.3	Architecture logicielle	108
4.4	Gestion de la concurrence par machines à états	110
4.4.1	Gestion de la concurrence par un système de permission	111
4.4.2	Machines à états des composantes logiques	113
4.4.3	Machines à états du contrôleur de dialogue	116
4.4.4	Granularité de la gestion de la concurrence	118
4.5	Expérimentation	123
4.5.1	Conception de l'expérimentation	123
4.5.2	Conditions de l'expérimentation	127
4.5.3	Résultats et analyses de l'expérimentation	129
4.5.4	Conclusion sur l'expérimentation	135
4.6	Conclusion	135
5	Evolutivité des systèmes et conséquences sur les trois phases de conception	137
5.1	Introduction	137
5.2	Modification du repère du point de vue 2D dans MC	138
5.3	Extension à un environnement de simulation acoustique	142
5.3.1	Système de rendu visuel et audio interactif de l'acoustique des salles	142
5.3.2	Extension selon les étapes de conception	144
5.4	Conseils d'extension et de modification d'un système	148
5.5	Conclusion	152
6	Navigation multi-surfaces dans une scène 3D	155
6.1	Introduction	155
6.2	Disposition des vues	156

6.2.1	Disposition des points de vue dans le monde virtuel	156
6.2.2	Disposition des surfaces de projection des vues dans le monde physique	158
6.3	Techniques d'interaction pour la navigation	160
6.3.1	Technique d'interaction par manipulation de caméra	161
6.3.2	Technique d'interaction de suivi du doigt	162
6.3.3	Technique d'interaction d'orientation du doigt	163
6.3.4	Technique d'interaction de manipulation à deux doigts	165
6.3.5	Technique d'interaction de couplage des vues	166
6.3.6	Technique d'interaction de suivi du doigt augmentée	169
6.4	Conception comparative des techniques d'interaction pour la navigation .	169
6.4.1	Conception par <i>mapping</i>	170
6.4.2	Conception par composantes logiques	173
6.5	Expérimentation	176
6.5.1	Objectif	177
6.5.2	Protocole	177
6.5.3	Résultats et analyse	180
6.5.4	Discussion	182
6.6	Conclusion	189
	Conclusion	191
	Bibliographie	197
	Glossaire	207

Table des figures

1	Système de type TEAMViz	2
1.1	Slimblade Trackball : un dispositif de type trackball	7
1.2	SpaceNavigator : un dispositif de type SpaceMouse	7
1.3	INTUOS4 : un dispositif de type tablette graphique	8
1.4	Degrés de liberté d'un dispositif de type SpaceMouse	8
1.5	CAT : dispositif à six degrés de liberté augmenté d'une tablette graphique	9
1.6	Table tactile avec la technologie FTIR	15
1.7	Table tactile avec la technologie DSI	17
1.8	Table tactile avec la technologie DI	17
1.9	Orientation de la représentation des données par manipulation d'un ma- nipulateur	20
1.10	Orientation continue de la représentation des données selon sa position sur l'affichage	20
1.11	Orientation discrète de la représentation des données selon sa position sur l'affichage	21
1.12	Orientation de la représentation des données selon sa trajectoire de ma- nipulation	21
1.13	Manipulation à deux doigts pour la translation et la rotation d'une re- présentation des données	22
1.14	Modèle Seeheim	30
1.15	Modèle ARCH	31
1.16	Modèle MVC	32
1.17	Architecture générale des systèmes multimodaux	33

1.18	Continuum entre le réel et le virtuel	36
1.19	Modèle pour l'interaction dans un environnement mixte	36
2.1	Modèle Physique/Numérique	43
2.2	Modèle Mapping	43
2.3	Mapping d'une application de station de travail permettant la translation des photos sur l'écran en utilisant une souris	44
2.4	Etapes de conception par mapping	48
2.5	Etapes initiales de la conception par mapping	49
2.6	Etapes finales de la conception par mapping	50
2.7	Exemple de conception par mapping	53
2.8	Combinaisons possibles de l'exemple de conception par mapping	55
2.9	Composante tâche de l'expérimentation	58
2.10	Dispositif SpaceTraveler	58
2.11	Tables des techniques d'interaction	60
2.12	Mapping initial de l'EDE T	60
2.13	Solution du mapping de l'EDE T	61
2.14	Mapping initial de l'EDE S	61
2.15	Solution du mapping de l'EDE S	62
2.16	Solution du mapping de l'EDE St	62
2.17	Solution du mapping de l'EDE TS	63
2.18	Première solution du mapping de l'EDE TSt	64
2.19	Seconde solution du mapping de l'EDE TSt	64
2.20	Utilisateur dans les conditions de l'expérimentation	65
2.21	Phase d'apprentissage	65
2.22	Phase d'évaluation	66
2.23	Possibilité de sélection de l'expérience	67
2.24	Temps moyen de sélection	68
2.25	Préférence des utilisateurs du SEDE de sélection	69
2.26	Temps moyen de manipulation	70

2.27	Différence de temps moyen de sélection	71
2.28	Précision moyenne de position pour chaque SEDE	71
2.29	Précision moyenne de dimension pour chaque SEDE	71
2.30	Précision moyenne de position pour chaque SEDE de manipulation selon chaque axe	72
2.31	Précision moyenne de dimension pour chaque SEDE de manipulation selon chaque axe	72
2.32	Préférence des utilisateurs pour la SEDE de manipulation	73
2.33	Temps d'interaction moyen de la sélection et de manipulation	74
2.34	Notation des quatre métriques subjectives selon chaque SEDE	75
2.35	Somme des notations	75
2.36	Préférence des utilisateurs de la SEDE pour l'ensemble de l'expérience . .	76
3.1	Composante logique dans le modèle physique/numérique	80
3.2	Exemple d'utilisation de composantes logiques avec un écran tactile . . .	81
3.3	Composante logique spatiale	83
3.4	Liste des types des composantes logiques	84
3.5	Etapes de la conception par composantes logiques	86
3.6	Exemple d'un élément du contrôleur de dialogue	87
3.7	Sketch des vues du simulateur de tank	95
3.8	Dispositifs physiques utilisés pour le simulateur de tank	96
3.9	Eléments du contrôleur de dialogue utilisés pour le simulateur de tank . .	96
3.10	Types possibles pour la composante logique de la vue de la tourelle . . .	97
3.11	Définition des composantes logiques liées aux vues pour le simulateur de tank	98
3.12	Ensemble des composantes logiques pour le simulateur de tank	99
3.13	Composantes logiques et leurs connexions pour la solution Sol1	100
3.14	Composantes logiques et leurs connexions pour la solution Sol2	101
3.15	Composantes logiques et leurs connexions pour la solution Sol3	102
3.16	Composantes logiques et leurs connexions pour la solution Sol4	103

4.1	Approches de gestion des permissions	108
4.2	Architecture logicielle centralisée avec l'intégration des éléments du contrôleur de dialogue dans l'application centrale et des composantes logiques dans les applications IHM	109
4.3	Architecture de MC	110
4.4	Gestion des permissions par demande de permission et mise à jour	112
4.5	Gestion des permissions par demande de permission	112
4.6	Architecture avec deux composantes logiques <i>CL1</i> et <i>CL2</i> connectées à un même élément du contrôleur de dialogue <i>CD</i>	114
4.7	Etats d'une machine à états d'une composante logique d'entrée/sortie <i>CL1</i> dans le cas de deux composantes logiques <i>CL1</i> et <i>CL2</i> reliées au même élément du contrôleur de dialogue	114
4.8	Machine à états de la composante logique d'entrée/sortie <i>CL1</i> de l'architecture de la figure 4.6	115
4.9	Etats de la machine à états de l'élément du contrôleur de dialogue <i>CD</i> de la figure 4.6	116
4.10	Machine à états de l'élément du contrôleur de dialogue <i>CD</i> de l'architecture de la figure 4.6	118
4.11	Exemple de degrés de liberté concurrents	119
4.12	Exemple des états de machine à états d'un degré de liberté d'un élément du contrôleur de dialogue	120
4.13	Exemple d'une machine à états au niveau d'un degré de liberté d'un élément du contrôleur de dialogue	122
4.14	Exemple de deux machines à états au niveau d'un degré de liberté d'un élément du contrôleur de dialogue après simplification	122
4.15	Vues initiales de la tâche de positionnement d'un objet virtuel	123
4.16	Plan de l'expérience avec un utilisateur interagissant avec la table tactile et un autre avec la souris	124
4.17	Différences entre le repère du monde virtuel et le repère du plan de projection pour la tâche de positionnement d'un cube virtuel	125
4.18	Composantes logiques d'entrée/sortie et l'élément du contrôleur de dialogue nécessaires pour la tâche de positionnement d'un objet virtuel	125

4.19	Vues initiales de la tâche de navigation	126
4.20	Différences entre le repère du monde virtuel et le repère du plan de projection pour la tâche de navigation	126
4.21	Composantes logiques d'entrée/sortie et l'élément du contrôleur de dialogue nécessaires pour la tâche de navigation	126
4.22	Machines à états développées au niveau des composantes logiques et de l'élément du contrôleur de dialogue pour le mode <i>exclusif</i>	128
4.23	Machines à états développées au niveau des degrés de liberté de l'élément du contrôleur de dialogue représentant le cube virtuel pour la tâche de positionnement d'un objet virtuel	129
4.24	Machines à états développées au niveau des degrés de liberté de l'élément du contrôleur de dialogue représentant le point de vue 3D pour la tâche de navigation	130
4.25	Deux utilisatrices durant l'expérience	130
4.26	Différence de temps entre le mode <i>exclusif</i> et le mode <i>coopératif</i> pour effectuer la tâche de positionnement d'un objet virtuel	131
4.27	Différence de temps entre le mode <i>exclusif</i> et le mode <i>coopératif</i> pour la réalisation de la tâche de navigation	131
4.28	Nombre d'interactions réalisées simultanément par les utilisateurs pour la réalisation de la tâche de positionnement d'un objet virtuel	132
4.29	Nombre d'interactions réalisées simultanément par les utilisateurs pour l'accomplissement de la tâche de navigation	132
4.30	Facilité des interactions selon les utilisateurs pour chacune des tâches et pour chaque niveau de granularité	134
4.31	Préférence des utilisateurs pour le niveau de granularité de la gestion de la concurrence	134
4.32	Importance de la possibilité de réalisation d'interactions simultanément sur un même élément au moyen de deux vues	135
5.1	Architecture et tâches réalisables dans MC	138
5.2	Conception par <i>mapping</i> des interactions pour la manipulation d'un objet virtuel dans le système MC initial avec projection des axes X et Y du repère du monde virtuel sur la table tactile	139

5.3	Conception par <i>mapping</i> des interactions dans MC changement de repère de la vue 2D	140
5.4	Modification de la composante logique de MC correspondant à la représentation 2D d'un objet virtuel et de ses connexions à l'élément du contrôleur de dialogue	141
5.5	Vue 3D du système de rendu visuel et audio de l'acoustique des salles . .	142
5.6	Architecture du système de rendu visuel et audio de l'acoustique des salles	143
5.7	Exemple récapitulatif des trois phases de conception	151
6.1	Les trois angles de rotation d'un point de vue	155
6.2	Présentation de la disposition des points de vue dans le monde virtuel dans le système développé par Forlines <i>et al.</i>	157
6.3	Navigation dans BUILD-IT en utilisant des "briques"	158
6.4	Disposition des surfaces de projection dans le monde physique dont l'une disposée horizontalement et l'autre verticalement	159
6.5	Présentation de la disposition des surfaces de projection des vues dans le monde physique dans le système développé par Wigdor <i>et al.</i>	160
6.6	Translation par manipulation directe d'une icône en forme de caméra représentant le point de vue 3D affichée sur la vue 2D	162
6.7	Icône en forme de caméra représentant le point de vue 3D affichée sur la vue 2D et le manipulateur associé pour la réalisation des rotations dans <i>TIMC</i>	162
6.8	Technique d'interaction du suivi du doigt sur la table tactile pour la navigation dans un environnement 3D avec orientation de la caméra dans la direction de la tangente à la courbe de déplacement	163
6.9	Déplacement nécessaire pour se positionner parallèlement à la vue 3D courante en un autre point situé perpendiculairement à l'axe de vue en utilisant <i>TISD</i>	164
6.10	<i>TIOD</i> sur la table tactile pour la navigation dans un environnement 3D par orientation de la caméra en fonction de l'orientation du doigt	164
6.11	Technique d'interaction de la manipulation à deux doigts sur la table tactile pour la navigation dans un environnement 3D	166
6.12	Technique d'interaction de couplage des vues pour la navigation dans un environnement 3D	167

6.13	Solutions de la conception par <i>mapping</i> de <i>TIMC</i> , <i>TISD</i> et <i>TIOD</i> . . .	171
6.14	Solutions de la conception par <i>mapping</i> de <i>TIMDD</i>	172
6.15	Solutions de la conception par <i>mapping</i> de <i>TICV</i>	173
6.16	Solution de la conception par composantes logiques de <i>TIMC</i> , <i>TISD</i> , <i>TIOD</i> et <i>TIMDD</i>	174
6.17	Solution de la conception par composantes logiques de <i>TICV</i>	175
6.18	Vues 2D et 3D de l'environnement virtuel pour la tâche <i>arbre</i> en utilisant <i>TIMCJ</i>	178
6.19	Vues 2D et 3D de l'environnement virtuel pour la tâche <i>boule rouge</i> en utilisant <i>TIMCJ</i>	179
6.20	Utilisateur participant à l'expérimentation de comparaison de <i>TIMCJ</i> et <i>TICVJ</i>	180
6.21	Temps moyen nécessaire pour exécuter la tâche <i>arbre</i>	180
6.22	Temps moyen nécessaire pour exécuter la tâche <i>boule rouge</i>	181
6.23	Somme de toutes les notations des utilisateurs pour la facilité, la satis- faction, l'intuitivité et l'efficacité des interactions	181
6.24	Préférences des utilisateurs entre <i>TIMCJ</i> et <i>TICVJ</i>	182
6.25	Caméras multiples dans la vue 2D dont une active présentée sur l'écran vertical	183
6.26	Présentation d'une vue inactive réduite incrustée dans la vue active 3D afin de fournir un retour visuel 3D à l'utilisateur s'il manipule une caméra inactive	184
6.27	Position de l'icône de caméra sur la vue 2D	186
6.28	Disposition de quatre utilisateurs autour d'une table tactile	187
6.29	Deux possibilités de disposition des caméras sur la vue 2D en utilisant <i>TICVJ</i> dans la situation d'écrans verticaux multiples	188

Liste des tableaux

1.1	Relations temporelles non dirigées d'Allen	26
1.2	Classification des travaux de recherche sur la combinaison de surfaces multiples et de vues multiples	29
3.1	Notations des éléments nécessaires pour les règles de conception par composantes logiques	90
5.1	Phases de conception touchées par la modification des caractéristiques d'un système	150

Introduction

L'être humain apprend dès son plus jeune âge à manipuler des objets jusqu'au moment où cela devient naturel. Certaines avancées scientifiques et technologiques en réalité virtuelle ont pour but d'offrir la possibilité d'interagir avec des objets virtuels de la même façon que l'être humain le fait dans la vie réelle. Les techniques d'interaction développées pour ces systèmes demandent quelquefois un temps d'apprentissage important et l'effet de fatigue est parfois significatif. De plus, certaines techniques d'interaction sont adaptées à certaines tâches mais pas à d'autres.

Contrairement à la souris pour les environnements en deux dimensions de type stations de travail, il n'existe pas actuellement un dispositif d'entrée universel adapté pour interagir avec un environnement en trois dimensions quelles que soient les tâches. Les dispositifs actuels intègrent plusieurs (3, 6 ou même 12) degrés de liberté augmentant leur complexité d'utilisation. Le manque de dispositifs d'entrée adaptés à des interactions tridimensionnelles ainsi que des techniques d'interaction faciles et intuitives incite à l'utilisation de dispositifs d'entrées bidimensionnels.

Depuis le prototypage en 1963 et la démonstration en 1968 de la souris par Douglas C. Engelbart, elle est devenue le dispositif d'entrée indispensable pour l'utilisation des ordinateurs. La facilité et l'intuitivité d'utilisation de la souris avec ses deux degrés de liberté ont encouragé le développement de plusieurs techniques d'interaction dont le célèbre "Drag and Drop". Depuis, les avancées technologiques et scientifiques ont permis de développer des systèmes d'interaction 3D avec une souris. Ces systèmes permettent la manipulation d'objets virtuels selon plusieurs degrés de liberté en utilisant la notion de modes ou en ajoutant des interacteurs graphiques. Cependant, l'utilisation de modes augmente la complexité du système et peut induire des erreurs de la part des utilisateurs [Norman, 1981]. L'utilisation de la souris est donc imparfaite pour ce type de tâches. Dans le domaine des Interactions Hommes-Machines (IHM), les efforts pour remplacer la souris se focalisent sur les surfaces tactiles mais les mêmes problèmes subsistent pour les interactions à plusieurs degrés de liberté avec des objets virtuels.

La solution que nous proposons pour remédier à ce manque de techniques d'interac-

tion faciles et intuitives dédiés aux environnements tridimensionnels est la combinaison d'interactions au moyen d'une vue 2D présentée sur une surface tactile et des interactions sur une vue 3D (voir figure 1). Nous appelons le type de système réalisant cette combinaison *TEAMViz** de *Tabletop Environment Augmented with Mural Visualization* (soit en français environnement tabletop augmenté par une visualisation murale). L'étude de cette combinaison est très complexe, d'une part, à cause de l'étude séquentielle de la tâche dans le domaine de la réalité virtuelle et, d'autre part, à cause de la spécificité de plusieurs études de techniques d'interaction pour les surfaces tactiles (e.g. gestion de photos). Néanmoins la complémentarité entre l'intuitivité et la facilité d'interaction bidimensionnelle sur une surface tactile et la possibilité de réaliser des tâches selon plusieurs degrés de liberté à travers une vue 3D justifient l'intérêt des TEAMViz.

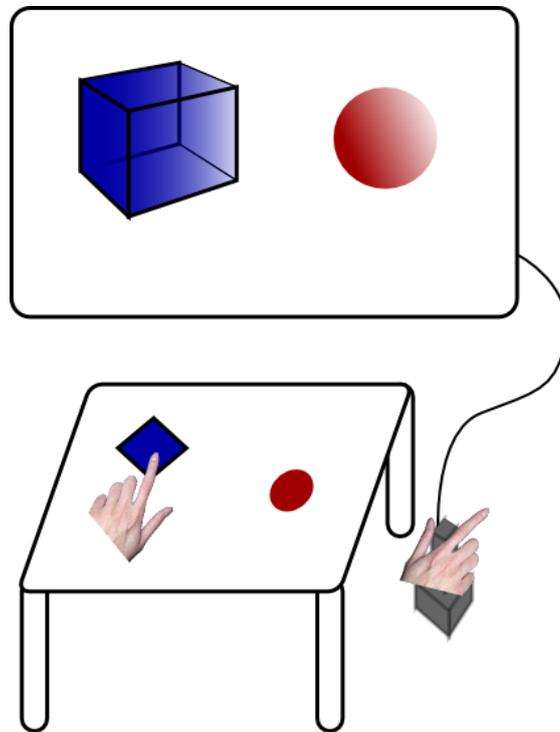


FIGURE 1 Système de type TEAMViz.

Plusieurs systèmes similaires aux TEAMViz existent dans la littérature mais aucun espace de conception n'est réellement adapté aux TEAMViz. Les modèles existant pour la conception des interfaces et/ou des interactions sont nombreux en IHM. Ces modèles couvrent plusieurs types d'interactions allant des interactions spécifiques aux stations de travail jusqu'aux interactions multimodales et se situent à différents niveaux des phases de conception. Plusieurs de ces modèles semblent intéressants pour notre étude mais la majorité d'entre eux séparent la modélisation des interactions effectuées en entrée de la présentation de l'information en sortie. La modélisation des interactions en entrée ainsi que la présentation de l'information en sortie est essentielle dans le type

d'interactions que nous souhaitons concevoir. Les seules exceptions à la séparation des entrées des sorties lors de la conception des systèmes sont les modèles spécifiques à la réalité augmentée et la réalité mixte. Contrairement aux types d'interactions que nous souhaitons concevoir, plusieurs de ces modèles ne prennent pas en considération la multimodalité et/ou ne sont pas appliqués aux interactions dans un environnement virtuel tridimensionnel. Cela justifie l'intérêt de développer un espace de conception adapté aux TEAMViz.

La contribution majeure de cette thèse est le développement d'un espace de conception permettant la création facile de systèmes combinant plusieurs vues et plusieurs dispositifs d'entrée (dont les TEAMViz) pour des tâches de navigation et de manipulation 3D. Cet espace de conception se divise en trois phases allant de la combinaison des interactions à haut niveau jusqu'à la gestion de la concurrence. L'originalité de cet espace de conception est la prise en considération des entrées et des sorties du système final ainsi que le filtrage de la combinatoire des couplages possibles. La focalisation de cet espace sur les caractéristiques spatiales (e.g. dimensions, degrés de liberté, etc.) permet de mieux relier les entrées et les sorties. Par rapport aux travaux existants dans le domaine, cet espace de conception permet de mettre en relation le monde virtuel présenté sur plusieurs vues avec le monde physique ainsi que l'étude des interactions en entrée et en sortie de façon couplée.

Ce manuscrit de thèse se compose de six chapitres. Nous présentons dans le chapitre 1 l'état de l'art lié à la conception des TEAMViz. Nous présentons respectivement dans les chapitres 2, 3 et 4 les trois phases de notre espace de conception. Nous étudions ensuite l'évolutivité d'un système conçu avec notre approche dans le chapitre 5. Enfin nous présentons un exemple d'application de notre espace de conception dans le chapitre 6.

Dans le chapitre 1 nous commençons par étudier les différents dispositifs d'entrée, techniques d'interaction et systèmes existants dans le domaine des interfaces graphiques tridimensionnelles surtout pour la réalisation de tâches spatiales. Ensuite nous exposons les travaux de recherche effectués dans le domaine des tabletops et plus spécifiquement les tables tactiles. Nous poursuivons par la présentation de différents systèmes combinant séparément plusieurs dispositifs en entrée, des vues multiples et des surfaces multiples. Nous étudions ensuite les modèles de conception des interfaces et des interactions existants surtout dans les domaines de la multimodalité et de la réalité mixte et augmentée. Nous concluons ce chapitre par un récapitulatif des différentes études présentées en précisant à quels niveaux elles interviennent dans ce manuscrit.

L'ensemble de l'espace de conception que nous élaborons et ses trois phases sont présentés au début du chapitre 2. Nous décrivons dans ce chapitre la première phase de

notre espace de conception qui permet la définition des dispositifs d'entrée et de sorties à utiliser, la définition des couplages entrée/sortie possibles et la mise en relation des dispositifs d'entrée avec les tâches spatiales. Cette phase de conception est dédiée à la combinaison à haut niveau des interactions réalisées au moyen de plusieurs dispositifs en entrée et plusieurs vues. Nous poursuivons par la présentation d'une expérimentation et de ses résultats en se basant sur cette première phase de l'espace de conception. Une tâche spatiale précise est définie pour cette expérimentation qui combine plusieurs dispositifs en entrée et deux vues en sorties d'une même scène virtuelle.

Après la combinaison des interactions à haut niveau, nous présentons la seconde phase de notre espace de conception (appelée conception par composantes logiques) qui est proche de la conception logicielle dans le chapitre 3. Nous définissons dans ce chapitre la notion de composante logique avant de présenter leur conception permettant la mise en relation des entrées avec les sorties du système. Nous concluons ce chapitre par un exemple illustrant cette phase de conception.

La combinaison d'interactions effectuées à travers des vues multiples d'une même scène virtuelle et en utilisant plusieurs dispositifs d'entrée entraîne une concurrence des interactions. Dans le chapitre 4, nous présentons la troisième phase de l'espace de conception liée à la gestion de la concurrence (étude de la concurrence des interactions puis architecture logicielle). Nous présentons ensuite notre contribution à la gestion de la concurrence avant de conclure avec une expérimentation illustrant cette troisième phase de l'espace de conception.

Après la présentation de l'espace de conception pour la combinaison des interactions en utilisant plusieurs dispositifs d'entrées au moyen de vues multiples pour une tâche spatiale, nous étudions la généralité de cette approche dans le chapitre 5. Dans ce chapitre nous exposons deux études de cas de modification et d'extension d'un système de type TEAMViz déjà conçu avec notre espace de conception. Nous concluons ce chapitre avec des règles de généralité de notre approche dans le but de faciliter l'extension ou la modification des systèmes par les concepteurs.

Dans les chapitres 2, 3, 4 et 5 nous nous focalisons principalement sur des tâches de manipulation d'objets virtuels. Nous étudions dans le chapitre 6 la navigation dans un système de type TEAMViz combinant une vue 2D d'un environnement virtuel présentée sur une table tactile et une vue 3D de ce même environnement présentée sur un écran vertical. Nous décrivons plusieurs techniques de navigation et revisitons notre espace de conception pour la modélisation de certaines d'entre elles. Nous présentons ensuite une expérimentation qui se focalise sur la relation entre les vues multiples d'une même scène virtuelle et le monde physique.

Chapitre 1

Etude des systèmes pour la conception des TEAMViz

1.1 Introduction

Dans ce chapitre nous décrivons la problématique de nos travaux pour le développement des TEAMViz en étudiant dans un premier temps les interfaces graphiques tridimensionnelles (nommées dans la littérature les 3DGUI ou 3D Graphical User Interfaces) et les travaux de recherche sur les tabletops interactifs. Nous montrons que ces deux thématiques de recherche ne permettent pas de trouver une solution parfaite pour la réalisation de tâches spatiales. Cependant elles sont complémentaires et justifient le développement d'interactions combinant plusieurs dispositifs en entrée, de vues multiples et de surfaces multiples.

Dans le cadre de la combinaison des interactions effectuées au moyen de deux vues distinctes et combinant des dispositifs d'entrée différents affectant une même scène virtuelle dans les TEAMViz, nous devons étudier les modèles d'interaction existants. Nous examinons ensuite dans ce chapitre les différents modèles de conception des interfaces et les modèles d'interaction pour définir les améliorations à apporter dans le cadre de nos travaux.

1.2 3DGUI

De façon générale, les interfaces utilisateurs 3D, appelées dans la littérature 3DUI (acronyme de 3D User Interfaces), sont des interfaces utilisateurs qui impliquent des interactions 3D [Bowman *et al.*, 2004, p. 7]. Le domaine des 3DUIs regroupe des interfaces

visuelles, auditives, haptiques et tactiles. Nous restreignons notre étude aux interfaces visuelles appelées 3DGUI uniquement qui font référence aux interactions visuelles.

1.2.1 Dispositifs d'entrée

Les avancées technologiques et scientifiques ont entraîné le développement d'un nombre très important de dispositifs d'entrée pour les 3DGUI en constante augmentation. Chacun de ces dispositifs présente des caractéristiques différentes : degrés de liberté, discrets ou non, isotoniques (capture du déplacement) ou isométriques (capture de la force), etc. Pour classifier les dispositifs selon leurs différentes caractéristiques, plusieurs taxonomies ont été conçues dont celle de William Buxton [1983]. Buxton classe les dispositifs d'entrée dans cette taxonomie selon trois propriétés : le nombre de dimensions (1, 2 ou 3), les propriétés des mécanismes de capture des interactions (position, mouvement ou pression) et selon l'existence d'un élément séparant l'utilisateur des capteurs (e.g. utilisation d'un stylet ou manipulation directe avec le touché). Les trois propriétés considérées dans cette taxonomie présentent quelques limites comme la considération des dimensions au lieu des degrés de liberté ainsi que la classification des dispositifs continus exclusivement (et non discret). Une autre taxonomie a été proposée par Mackinlay *et al.* [1990] qui est inspirée de Buxton et prend en considération les degrés de liberté ainsi que le fait que le dispositif soit discret ou continu. Cette taxonomie classe aussi les dispositifs d'entrée selon trois propriétés : les degrés de liberté (trois axes de translation et trois axes de rotation), les propriétés des mécanismes de capture ainsi que le nombre de valeurs mesurées par le dispositif. Cette taxonomie permet une classification plus fine des dispositifs d'entrées que celle proposée par Buxton. Les points qui nous semblent intéressants dans cette taxonomie sont la classification des dispositifs d'entrée selon les degrés de liberté dans le monde réel et la relation (combinée ou séparée) existant entre ces degrés de liberté. La séparation et la combinaison des degrés de liberté d'un dispositif en entrée influence la façon de réaliser une tâche. Considérons le déplacement d'un objet virtuel en 2D d'une position A vers une position B . Cette tâche est réalisable de façon "diagonale" si les deux degrés de liberté de translation sont combinés et "en escalier" sinon.

Le clavier et la souris sont les deux dispositifs d'entrée les plus utilisés pour les stations de travail. Le clavier est un dispositif de type discret et la souris permet l'obtention de valeurs continues ce qui explique la complémentarité entre ces deux dispositifs. La combinaison de ces deux dispositifs peut sembler idéale à cause de leur popularité mais elle présente quelques limites : deux degrés de liberté uniquement avec la souris, une nécessité de poser la souris sur un support, difficulté de dessin avec la souris etc. D'autres



FIGURE 1.1 Slimblade Trackball : un dispositif de type trackball. Source : [Kensington].



FIGURE 1.2 SpaceNavigator : un dispositif de type SpaceMouse. Source : [3Dconnexion].

dispositifs ont été aussi créés pour les stations de travail comme le trackball qui est un dispositif ressemblant à une souris inversée (voir figure 1.1). Avec l'invention des jeux vidéos par Baer [2005, p. 1] en 1966, plusieurs dispositifs de type manette et joystick ont été développés dont le premier joystick en 1967. Ces dispositifs sont souvent utilisés actuellement dans les jeux vidéos et les stations de travail mais ne sont pas réellement adaptés pour la manipulation d'objets virtuels en trois dimensions. Un domaine d'application proche de la manipulation d'objets virtuels en trois dimensions est la modélisation tridimensionnelle. Nous retrouvons pour ce domaine d'application deux types de dispositifs : le premier est de type SpaceMouse (voir figure 1.2) et le second est de type tablette graphique (voir figure 1.3). Les dispositifs de type SpaceMouse sont des dispositifs d'entrée à six degrés de liberté (voir figure 1.4), disponibles pour le grand public, et dont les applications principales sont les modeleurs 3D en complément de la souris. L'avantage avancé de ce type de dispositif est l'intégration des six degrés de liberté permettant une utilisation simple pour les experts mais il reste très difficile pour

les novices. Contrairement à la SpaceMouse, les tablettes graphiques intègrent deux degrés de liberté uniquement facilitant le tracé et le dessin en deux dimensions. Plusieurs recherches ont été menées pour l'utilisation des tablettes graphiques dans le but de modéliser des objets virtuels tridimensionnels. Ces travaux soulignent le désagrément des utilisateurs par rapport au changement de modes [Kara *et al.*, 2007] et la complexité de la reconstruction d'objets tridimensionnel non droits (e.g. des objets ne contenant pas des variations angulaires importantes) [Masry *et al.*, 2005].



FIGURE 1.3 INTUOS4 : un dispositif de type tablette graphique. Source : [Wacom].



FIGURE 1.4 Degrés de liberté d'un dispositif de type SpaceMouse. Source : [3Dconnexion].

En parallèle des avancés sur les dispositifs d'entrée pour les stations de travail, un autre type de dispositif d'entrée basé sur le "tracking" (suivi ou capture) est en constant développement. Les systèmes de tracking permettent souvent de récupérer six degrés de liberté. Il existe six types de tracking d'après Welch et Foxlin [2002] : mécanique, inertiel, acoustique, magnétique, optique et le tracking basé sur les ondes radio ou microondes. L'un des éléments intéressants de ces systèmes est la variété de leurs applications. L'une de ces applications est le suivi de la tête d'un utilisateur dans le but d'adapter le rendu visuel au point de vue de utilisateur augmentant ainsi sa perception de la profondeur [Bowman *et al.*, 2004, p. 217-218]. La manipulation d'objets virtuels selon les six degrés de liberté est aussi une application possible. De plus des six degrés de liberté, certains dispositifs augmentent les systèmes de tracking avec des boutons (e.g. le dispositif Bat [Ware & Jessome, 1988]) ou d'autres dispositifs pour accroître le nombre de degrés de liberté du dispositif (e.g. le Cubic Mouse [Frohlich *et al.*, 2000]). L'utilisation de systèmes de tracking semble très prometteuse mais Welch et Foxlin révèlent l'existence de problèmes importants selon les systèmes dont : la lourdeur de la mise en place de

l'infrastructure des systèmes, le coût, la robustesse, la latence et le manque de confort. Ils soulignent aussi dans leur travail de recherche que les utilisateurs souhaitent des systèmes de tracking avec *une infrastructure réduite, une robustesse améliorée et une réduction de la latence*. Pour répondre à certains de ces problèmes, Johnny Chung Lee [2008] transforme une wiimote (manette de jeu contenant des accéléromètres, une caméra infrarouge et des boutons) en un système de tracking. Ce travail permet d'acquérir un système de tracking bon marché mais présente aussi des problèmes d'occlusion et de précision des mesures.

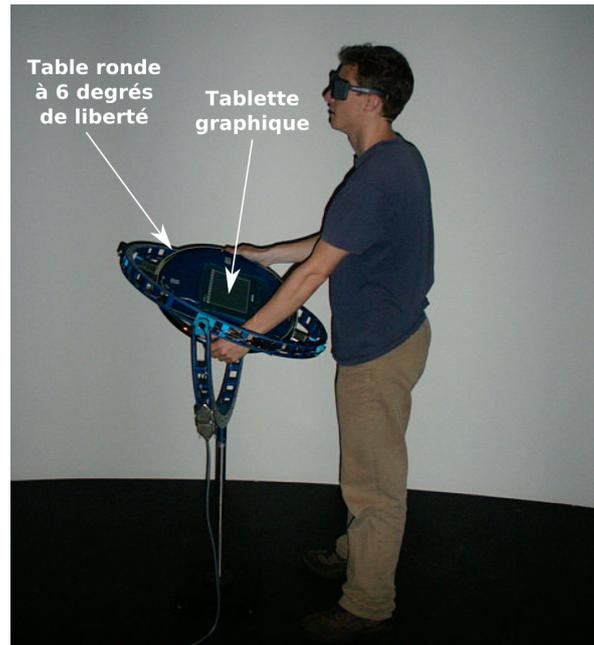


FIGURE 1.5 CAT : dispositif à six degrés de liberté augmenté d'une tablette graphique. Source : [Hachet *et al.*, 2003].

En plus des dispositifs d'entrée pour les stations de travail et les systèmes de tracking, il existe d'autres dispositifs dédiés aux interfaces utilisateurs tridimensionnels. CAT (Control Action Table soit en français table de contrôle d'action) [Hachet *et al.*, 2003] est un dispositif sous forme d'une table ronde rotative selon trois degrés de liberté (voir figure 1.5). De plus de ces trois rotations, des capteurs de pression existent dans ce dispositif pour capturer les pressions exercées selon les trois dimensions rendant ainsi le CAT un dispositif à six degrés de liberté. Le point que nous trouvons intéressant dans CAT est l'intégration d'une tablette graphique permettant des interactions selon deux dimensions. Les développeurs de CAT soulignent l'utilité de cette tablette pour certains types de tâches comme l'annotation qui ne nécessitent que deux degrés de liberté. Un autre dispositif qui nous semble intéressant est le CubTile [De La Rivière *et al.*, 2008] qui est un dispositif tactile à cinq facettes formant un cube. Les développeurs de ce dispositif soulignent les difficultés existant pour la manipulation de dispositifs

qui intègrent plusieurs degrés de liberté. Pour contrer ces difficultés, CubTile sépare plusieurs degrés de liberté selon les facettes permettant ainsi des interactions simples pour des tâches tridimensionnelles. CubTile et CAT illustrent le fait qu'un dispositif à plusieurs degrés de liberté ne convient pas à toutes les tâches et que des manipulations bidimensionnelles sont plus simples pour certaines tâches telles que l'annotation ou le positionnement d'un objet virtuel dans un plan.

La liste des dispositifs d'entrée dédiés aux interfaces tridimensionnelles présentée auparavant n'est pas exhaustive. D'autres types de dispositifs existent comme les dispositifs haptiques, les bras à retour d'effort, les gants numériques et d'autres encore. Nous pensons que le but de ces dispositifs est de mimer au mieux les interactions de la vie courante et donc d'améliorer la sensation de réalisme. Pour plus de détail voir [Bowman *et al.*, 2004, chap. 4] qui présente une liste exhaustive des dispositifs en entrée et en sortie pour les interfaces tridimensionnelles jusqu'à sa date de parution en 2004.

En résumé, les dispositifs d'entrée existants pour les interfaces graphiques tridimensionnelles présentent chacun des avantages et des inconvénients. L'utilisation de dispositifs 2D de type souris ou tablette graphique permet d'effectuer des interactions simples. Mais ces dispositifs ne couvrent pas tous les degrés de liberté nécessaires pour des tâches tridimensionnelles. Avec ces périphériques, les tâches 3D nécessitent un recours aux modes (e.g. mode de translation XY et mode de translation YZ), donc un apprentissage par les utilisateurs et une certaine discontinuité des interactions. Contrairement à la souris et la tablette graphique, les dispositifs d'entrée à plusieurs degrés de liberté couvrent jusqu'à la totalité des degrés de liberté nécessaires mais présentent d'autres inconvénients. Bérard *et al.* [2009] montrent que les dispositifs 3D sont moins performants que la souris pour une tâche de placement d'un objet et que le niveau de stress des utilisateurs est plus important avec un dispositif basé sur un système de tracking. Il n'existe pas un dispositif d'entrée idéal pour les interfaces graphiques tridimensionnelles permettant d'effectuer des interactions intuitives, simples et couvrant tous les degrés de liberté nécessaires pour des tâches spatiales.

1.2.2 Techniques d'interaction

Nous séparons deux types de systèmes 3DGUI. Le premier type de systèmes regroupe les systèmes dédiés aux stations de travail (dont les modeleurs 3D et les jeux vidéos). Le second type est plus spécifique à la réalité virtuelle immersive et regroupe des systèmes permettant leur utilisation dans une CAVE (système audio-visuel combinant plusieurs écrans de projection formant un cube presque complet [Cruz-Neira *et al.*, 1992]). Ces

deux types de systèmes affectent les mêmes tâches mais se distinguent au niveau des dispositifs (d'entrée et/ou de sortie) utilisés et donc des techniques d'interaction.

Dans le domaine des interfaces graphiques tridimensionnelles, il existe une séparation des tâches en trois types principaux : les tâches spatiales de manipulation d'objets virtuels, les tâches de navigation et les tâches de contrôle. Nous nous intéressons essentiellement aux tâches de manipulation d'objets virtuels mais décrivons brièvement les techniques d'interaction existantes pour les tâches de navigation et de contrôle dans les environnements tridimensionnels immersifs.

Techniques d'interaction de 3DGUI pour les stations de travail

Les interfaces graphiques tridimensionnelles sont utilisées dans plusieurs types de logiciels dédiés aux stations de travail et surtout dans les modeleurs 3D (e.g. Blender [Blender], 3ds Max [Autodesk 3ds Max], AutoCad [AutoCad], ArchiCad [ArchiCad], etc.). Les dispositifs d'entrée utilisés le plus souvent pour ces logiciels sont d'abord le clavier et la souris impliquant des interactions planes. Une technique d'interaction nommée sphère virtuelle, a été développée pour la rotation d'objets virtuels. Hinckley *et al.* [1997] ont montré que l'utilisation de cette technique était moins rapide que l'utilisation d'un dispositif 3D mais que la différence de précision n'est pas importante. Le manque de degrés de liberté suffisants au niveau des dispositifs d'entrée oblige l'augmentation du système par les modes de manipulation (e.g. translation, rotation, etc.). L'utilisation d'un dispositif à six degrés de liberté (e.g. SpaceNavigator de 3Dconnexion figure 1.2) complémentaire permet de réaliser des interactions bimanuelles et apporte une solution à ce manque de degrés de liberté. L'utilisation d'un tel dispositif est difficile pour des utilisateurs novices comme nous le montrons dans la section 2.5.

Pour la navigation dans les environnements tridimensionnels sur des stations de travail, nous différencions deux techniques. La première technique consiste à déplacer le point de vue. Pour cette technique d'interaction, trois possibilités existent. La première possibilité affecte certains degrés de liberté de navigation à un dispositif et d'autres à un autre dispositif d'entrée. L'exemple le plus connu d'une telle technique d'interaction est l'utilisation de la souris pour la rotation et le clavier pour le déplacement dans les jeux vidéos. Dans la seconde possibilité, l'utilisateur manipule une représentation de son point de vue sur une carte de l'environnement virtuel (e.g. icône en forme de caméra ou triangle). La troisième possibilité consiste à transposer les actions effectuées dans le monde réelle sur un dispositif d'entrée aux degrés de liberté de navigation numérique. Cette possibilité est utilisée pour la navigation en utilisant le joystick dans les simulateurs de vol et en utilisant un dispositif de type SpaceMouse pour les éditeurs

de modèles 3D. Dans la seconde technique d'interaction, une carte de l'environnement virtuel est présentée à l'utilisateur. Pour effectuer la navigation, l'utilisateur définit le point final où il sera téléporté. Dans Navidget [Hachet *et al.*, 2008], un utilisateur ayant uniquement une vue sur l'environnement virtuel peut choisir un objet d'intérêt et placer la nouvelle position du point de vue relativement au point d'intérêt.

Techniques d'interaction de 3DGUI pour la réalité virtuelle

Bowman *et al.* [1999] différencient trois étapes pour la manipulation d'un objet dans un environnement virtuel : la sélection, la manipulation et la désélection. Ils comparent dans leurs travaux neuf techniques d'interaction pour la manipulation d'un objet virtuel dans un environnement immersif. La première technique est la Go-Go technique d'interaction [Poupyrev *et al.*, 1996] qui consiste à étendre la main virtuelle pour sélectionner et manipuler un objet virtuel. Les autres techniques combinent deux techniques de sélection, deux techniques pour attraper l'objet virtuel et deux techniques pour le déplacement de l'objet virtuel (position absolue de la main et l'utilisation de boutons). Les deux techniques de sélection sont le lancé de rayon et l'occlusion qui sont deux techniques bidimensionnelles. Les utilisateurs ont porté des dispositifs de type Head Mounted Displays (HMD, soit en français un dispositif d'affichage porté sur la tête). Ils ont été suivis par un système de tracking magnétique. Les résultats obtenus montrent que les temps de sélection avec les techniques de sélection bidimensionnelles sont inférieurs à ceux obtenus avec une technique de sélection tridimensionnelle. De plus, les résultats montrent que la précision est meilleure pour les tâches à deux degrés de liberté que pour les tâches à six degrés de liberté. Ce travail illustre la complexité d'effectuer la sélection et la manipulation d'objets virtuels avec des techniques d'interaction tridimensionnelles.

Pinho *et al.* [2002] séparent les degrés de liberté de manipulation d'un objet virtuel entre deux utilisateurs muni d'un dispositif de type HMD et un système de tracking. Pinho *et al.* justifient cette nécessité de collaboration entre plusieurs utilisateurs par la difficulté ou même l'impossibilité d'effectuer certains types de tâches. Cette séparation des degrés de liberté semble intéressante sachant que certaines tâches élémentaires peuvent être effectuées d'une façon plus efficace d'un certain point de vue sur la scène 3D que d'un autre (e.g. déplacements latéraux dans une vue d'un objet virtuel au lieu d'un déplacement en profondeur dans une seconde vue). L'unique inconvénient est que les auteurs définissent au préalable la séparation des degrés de liberté entre les utilisateurs. La séparation des degrés de liberté selon un ou des utilisateur(s) ayant des points de vue différents pour faciliter la réalisation de la tâche nous semble très prometteuse.

Pour la navigation dans les environnements virtuels immersifs, nous différencions la navigation physique de la navigation numérique. La navigation physique nécessite le déplacement et/ou la rotation physique du corps de l'utilisateur contrairement à la navigation virtuelle [Bowman *et al.*, 2004, p. 189]. Il existe plusieurs dispositifs pour la navigation physique dans un environnement virtuel comme le CirculaFloor [Iwata *et al.*, 2005], les Powered Shoes [Iwata *et al.*, 2006] ou aussi le Cyberwalk [Schwaiger *et al.*, 2007]. Le but de ces dispositifs est de permettre aux utilisateurs de se déplacer dans le monde virtuel en marchant dans le monde réel. Les déplacements physiques réels sont compensés mécaniquement pour limiter l'espace réel utilisé. Une autre technique pour permettre la navigation physique tout en restant dans un espace restreint est la simulation du déplacement droit dans le monde virtuel qui correspond à un déplacement selon une courbe dans le monde réel [Steinicke *et al.*, 2008]. La navigation physique est intéressante mais dans le cadre de nos travaux, l'utilisation d'une table tactile avec ce type de navigation est impossible. Pour la navigation numérique 3D, certaines techniques d'interaction étendent celles dédiées pour les stations de travail comme le Navidget à la navigation dans des environnements immersifs [Knödel *et al.*, 2008]. Le déplacement d'une représentation de l'utilisateur sur une carte de l'environnement virtuel ou sur une représentation en miniature de l'environnement [Stoakley *et al.*, 1995] pour la navigation est une technique à base de changement d'échelle et de projection de l'espace 2D. Cette technique nous semble très intéressante à cause d'une vue d'ensemble de l'environnement, du positionnement du point de vue souhaité par l'utilisateur sur cette vue et de la possibilité de son adaptation aux tables tactiles. Par ailleurs, Ware et Osborne [1990] présentent trois techniques d'interaction pour la navigation. La première, appelée *eyeball in hand* (soit en français œil dans la main), consiste à mettre un tracker dans la main de l'utilisateur qui simulera le point de vue 3D. Ainsi la navigation s'effectue par le déplacement de la main de l'utilisateur. La seconde technique, appelée *scene in hand* (soit en français scène dans la main), consiste à coupler le monde virtuel avec le tracker utilisé par l'utilisateur. La manipulation du monde permet la navigation dans le monde virtuel. La troisième technique d'interaction, appelée *flying vehicle control* (soit en français contrôle d'un véhicule volant), consiste à contrôler la vitesse de déplacement et la vitesse angulaire en utilisant le tracker. Les deux premières techniques d'interaction nous semblent très intéressantes et leurs extensions à notre travail sont étudiées dans le chapitre 6.

Nous nous sommes moins intéressés aux autres types de tâches comme les tâches de contrôle du système. Ces tâches permettent un changement de mode d'interaction ou encore une demande au système de réaliser une tâche précise. L'utilisation de menus bidimensionnels ou la reconnaissance des gestes répondent à ces besoins. Une autre tech-

nique est l'utilisation d'une surface plane tenue dans une main et d'un stylet dans l'autre. Cette technique a été développée par Poupyrev *et al.* [1998] dans le Virtual Notepad pour l'annotation dans les environnements virtuels. Schalmsteig *et al.* [1999] développent un système semblable mais dont les fonctionnalités sont différentes comme la sélection par le dessin d'un lasso autour des objets souhaités. Ces deux travaux montrent que, pour certaines tâches, l'utilisation de dispositifs permettant des interactions planes est plus appropriée que l'utilisation de dispositifs à plusieurs degrés de liberté.

En résumé, les tâches de manipulation nécessitent plusieurs degrés de liberté. Les techniques d'interaction pour les stations de travail tentent de remédier aux manques des degrés de liberté de la souris et du clavier en implémentant une interaction modale. L'utilisation de modes d'interaction ne semble pas être la meilleure solution en raison des erreurs utilisateurs [Norman, 1981]. Au niveau des interfaces graphiques immersives, les techniques d'interaction qui nécessitent l'utilisation de systèmes de tracking sont fatigantes pour l'utilisateur [Zhai, 1998]. De plus, la sélection bidimensionnelle est plus rapide que la sélection tridimensionnelle et le taux d'erreurs est inférieur pour des tâches bidimensionnelles comparé à celui des tâches nécessitant six degrés de liberté. La réduction et la séparation des degrés de liberté manipulés simultanément semble une solution intéressante et justifiée par plusieurs études convergentes. Selon Bowman et ses collègues : *Il n'existe pas une seule meilleure technique de manipulation* [Bowman *et al.*, 2004, p. 181].

1.3 Tabletop

Le terme tabletop désigne la surface plate d'une table. Ce terme a été étendue dans la recherche en informatique aux tables interactives qui permettent l'interaction avec des données numériques présentées sur une table. Les dispositifs tabletops se caractérisent selon leurs types tels que les tables tactiles (interactions par le touché sur la table) et les tables tangibles (interactions en manipulant des objets physiques disposés sur la table).

L'utilisation des surfaces tactiles devient de plus en plus répandue. Certaines surfaces sont verticales comme le CityWall [Peltonen *et al.*, 2008] d'autres horizontales formant des tables tactiles comme la DiamondTouch [Dietz & Leigh, 2001]. Ces surfaces tactiles permettent d'effectuer des interactions planes uniquement. La combinaison de deux surfaces (tactiles ou non) complémentaires (i.e. une horizontale et l'autre verticale) permet d'obtenir les trois dimensions au lieu de deux. Nous présentons dans cette section les différents types de tables tactiles existants puis nous étudions les avantages et inconvénients de l'utilisation des tables tactiles. Nous concluons cette section par l'étude des

degrés de liberté pour les tables tactiles.

1.3.1 Les types de dispositifs tabletop

La limite séparant les dispositifs tabletops des autres types de dispositifs reste floue jusqu'à présent. Selon Grossman et Wigdor [2007], les travaux de recherche des dernières années se sont concentrés sur les dispositifs tabletops pour des applications et des interactions bidimensionnelles. Néanmoins, il existe des travaux de recherche se focalisant sur des dispositifs tabletops 3D qui sont parfois considérés comme des dispositifs non-tabletop. Grossman et Wigdor considèrent qu'un dispositif tabletop 3D est un dispositif présentant un environnement tridimensionnel sur une surface horizontale. ARTHUR de Penn *et al.* [2005] et le système développé par Kato *et al.* [2000] sont deux exemples intégrant de tels dispositifs où les utilisateurs, munis d'un dispositif de type HMD, interagissent avec des objets 3D présentés sur une table. Nous reprenons la définition des tabletops 3D proposée par Grossman et Wigdor. Dans leurs travaux de recherche, ils suggèrent aux concepteurs de considérer des interactions planes même si l'utilisation de dispositif 3D en entrée est possible. En accord avec ce principe nous concentrons notre étude des tabletops uniquement aux types de tabletops permettant d'effectuer des interactions planes.

Les tabletops tangibles sont un exemple de dispositifs tabletops bidimensionnels. Dans ces dispositifs, des objets physiques dont la position et parfois l'orientation sur une table sont trackées (e.g. par un système de vision par ordinateur). L'utilisateur peut ainsi interagir avec le système en manipulant ces objets physiques. Un exemple de tels dispositifs est la ReacTable [Jordà *et al.*, 2007].

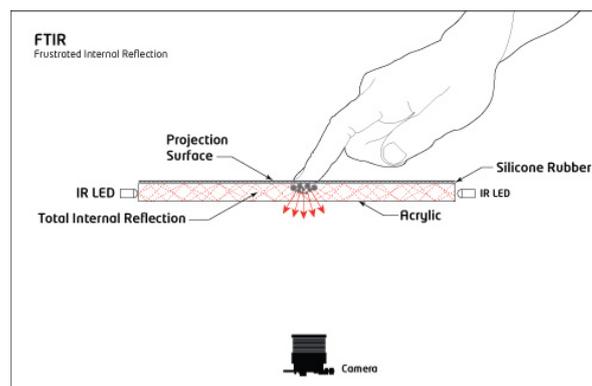


FIGURE 1.6 Table tactile avec la technologie FTIR. Source : [NUI Group].

Un autre type de tabletops bidimensionnels est les tables tactiles que nous différencions en deux types de dispositifs : la tables tactiles utilisant la vision par ordinateur

et les tables tactiles utilisant des capteurs intégrés. Un ensemble assez complet à notre connaissance des technologies pour les surfaces tactiles a été étudié par Grit Schuster dans sa thèse de master [Schuster, 2008]. L'idée principale des technologies utilisant la vision par ordinateur pour les tables tactiles est la reconnaissance des points d'appui sur la surface de la table en utilisant une caméra qui filme cette surface. A notre connaissance, il existe cinq types de tables tactiles utilisant la vision par ordinateur de la lumière infrarouge et qui se différencient par la façon de projeter la lumière infrarouge sur la surface :

- le Frustrated Total Internal Reflection (FTIR) [Han, 2005]. La projection de la lumière s'effectue des côtés d'une surface en plexiglas. La lumière infrarouge est redirigée vers le bas au niveau des points d'appui sur la surface et ainsi la caméra infrarouge capte ces points (voir figure 1.6) ;
- la technologie Diffused Surface Illumination (DSI) est semblable au FTIR mais elle diffuse la lumière d'une façon homogène sur toute la surface sauf au point d'appui (voir figure 1.7) ;
- la technologie Diffuse Illumination (DI) [Matsushita & Rekimoto, 1997] se décline en deux selon si elle utilisée de façon frontale comme dans le système playAnyWhere [Wilson, 2005] ou de derrière. Cette technologie repose sur le positionnement d'un projecteur infrarouge qui illumine la surface (voir figure 1.8). Cette technologie est utilisée par Microsoft pour le système surface [Microsoft Surface] ;
- l'utilisation de laser infrarouge projetant la lumière tangentiellement à la surface d'interaction (en dessus). Lorsqu'un doigt ou un objet est dans la surface de la lumière infrarouge, la lumière est redirigée à cette position vers le bas de la table où se situe la caméra. En comparaison avec les techniques FTIR, DSI et DI, cette technique évite le besoin d'appuyer fort sur la surface tactile ;
- l'utilisation de LED infrarouges directement sur les doigts. Cette technique est intrusive à cause de la nécessité d'équiper les doigts des utilisateurs avec des LED infrarouges.

ThinSight [Hodges *et al.*, 2007] est une autre technologie basée sur la vision par ordinateur pour les surfaces tactiles. Elle consiste à poser des petits émetteurs/récepteurs infrarouges derrière une dalle LCD. Cette technique semble prometteuse surtout qu'aucune calibration n'est nécessaire entre la surface d'interaction et les émetteurs et capteurs infrarouge contrairement aux autres technologies. A notre connaissance cette technologie n'a pas encore été appliquée sur les tables tactiles.

Ces technologies permettent de reconnaître plusieurs doigts et ainsi offrent des interactions dites multi-points (multi-touch). Les désavantages de ces technologies sont un possible temps de traitement important [Wilson, 2005] qui entraîne de la latence

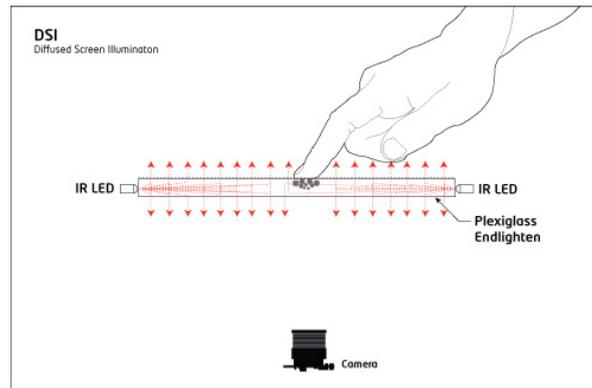


FIGURE 1.7 Table tactile avec la technologie DSI. Source : [NUI Group].

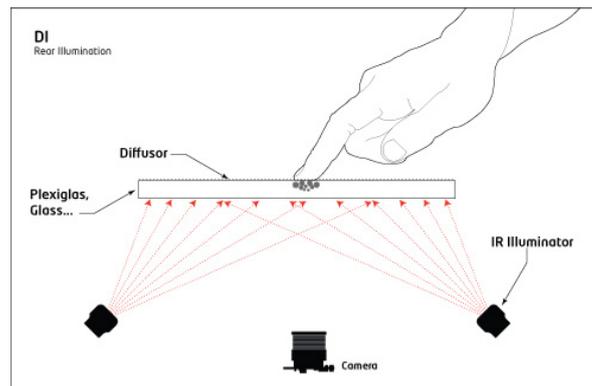


FIGURE 1.8 Table tactile avec la technologie DI. Source : [NUI Group].

gênante dans l'utilisation, des problèmes avec la luminosité de l'environnement ambiant [Hodges *et al.*, 2007] et l'impossibilité de différencier les utilisateurs. Pour permettre la différenciation des utilisateurs avec les surfaces tactiles utilisant la vision par ordinateur infrarouge, deux solutions ont été proposées dans la littérature : la reconnaissance des empreintes et le tracking. La première solution nécessite une résolution importante de la caméra infrarouge et un traitement de données conséquent. La seconde solution a été proposée par une équipe finlandaise sur le dispositif MultiTouch Cell [Multitouch] qui consiste à tracker la main et non uniquement les doigts lorsqu'ils touchent la surface.

Les tables tactiles capacitives comme la Diamondtouch [Dietz & Leigh, 2001] de Mitsubishi Electric Research Laboratories (MERL) constituent un autre type de tables tactiles. La DiamondTouch nécessite que l'utilisateur touche un tapis ou un stylet relié à la table créant, lors du touché avec la table, un circuit capacitif. Ainsi, avec des antennes implantées dans le dispositif, la reconnaissance de la position du touché est effectuée. Elle permet de différencier plusieurs utilisateurs chacun touchant un tapis ou utilisant un stylet instrumenté différent. Les désavantages de cette technologie sont la nécessité de toucher un tapis ou d'utiliser un stylet par l'utilisateur et la projection frontale sur la surface tactile impliquant des occlusions. La SmartSkin [Rekimoto, 2002] est une autre

table tactile capacitive. Elle est semblable à la DiamondTouch avec la différence que les émetteurs et les capteurs sont directement implantés dans la surface tactile n'obligeant pas l'utilisateur à toucher l'émetteur. Néanmoins, la SmartSkin présente toujours le problème d'occlusion obtenu par la projection frontale sur la surface et l'impossibilité de reconnaissance des utilisateurs.

1.3.2 Avantages et inconvénients de l'utilisation des dispositifs tabletops

Les dispositifs tabletops créent des espaces de travail adaptés pour la collaboration entre plusieurs utilisateurs. Maher et Kim [2006] étudient les différences des aspects collaboratifs entre deux systèmes de conception tridimensionnelle : le premier utilise une interface graphique sur une station de travail tandis que le second utilise un dispositif tangible de type tabletop. Cette étude illustre que, dans le cas du dispositif tabletop tangible, les utilisateurs ont recours aux gestes pour la communication des idées en pointant par exemple sur les objets tangibles. De plus les utilisateurs ont explicité des relations existant entre les différents objets tridimensionnels avec le dispositif tabletop, alors qu'avec la station de travail ils ont pensé à chaque objet séparément. Dans un autre travail de recherche, Rogers et Lindley [2004] étudient les effets de l'orientation d'un dispositif d'affichage interactif (en utilisant un stylet) sur la collaboration entre plusieurs utilisateurs pour une tâche de planification et de résolution de problèmes. Ils analysent les comportements des utilisateurs sous trois conditions : surface horizontale interactive, surface verticale interactive et un ordinateur personnel. Les résultats de cette étude montrent que le nombre de suggestions proposées par les utilisateurs est supérieur lorsqu'ils utilisent la surface horizontale. De plus le changement de rôle (personne utilisant le stylet et les commentateurs) durant l'expérience a été supérieur en utilisant cette disposition. Les auteurs soulignent aussi qu'un espace de travail horizontal a permis aux utilisateurs de faire attention aux actions des autres. Les deux travaux précédents illustrent l'apport des tabletops sur les aspects collaboratifs par rapport aux ordinateurs personnels et aux surfaces interactives verticales.

Les tables tactiles présentent une caractéristique supplémentaire aux aspects collaboratifs des tabletops qui est la manipulation par "Direct Touch" (touché direct). L'interaction directe peut sembler "naturelle" mais, comme signalé par Forlines *et al.* [2007], l'aspect naturel est difficile à mesurer. Néanmoins, Shen *et al.* [2006] soulignent la possibilité qu'ont les tables tactiles *de prendre en entrée des gestes manuels naturels et des manipulations intuitives*. Certaines tables tactiles actuelles permettent des interactions à plusieurs doigts et des interactions avec une ou plusieurs main(s). Wu et Balakrishnan [2003] développent plusieurs techniques d'interaction utilisant un doigt, deux doigts,

une main ou deux mains pour le prototype RoomPlanner. Les aspects bimanuels des interactions directes sur les tables tactiles ont été étudiés par Forlines *et al.* [2007] en les comparant aux interactions bimanuelles avec deux souris. Cette étude montre que les interactions bimanuelles directes sont préférées par les utilisateurs et présentent des temps de sélection, de déplacement et de redimensionnement d'objets bidimensionnels inférieurs comparées aux interactions bimanuelles avec deux souris.

Chaque dispositif possède des avantages et des inconvénients. L'inconvénient principale des tables tactiles est l'imprécision [Shen *et al.*, 2006]. Cette imprécision est montrée dans les travaux de Forlines *et al.* [2007] en comparant la sélection par touché direct à la sélection avec une souris. Forlines et ses collègues proposent plusieurs hypothèses pour justifier cette imprécision. La première est la distortion possible de l'objet à sélectionner selon sa distance de l'utilisateur. La seconde hypothèse est la disposition de la main et des doigts selon la distance de l'objet à sélectionner de l'utilisateur. Un objet proche est sélectionné par la pointe de l'index et les autres doigts de la main sont loin de la surface tactile. Contrairement à la sélection des objets proches, la sélection des objets à une distance importante de l'utilisateur s'effectue avec la pulpe de l'index augmentant la surface du contact et rapprochant les autres doigts de la surface tactile provoquant des sélections non souhaitées. De plus de l'imprécision, la co-localisation du dispositif d'entrée et de la surface de présentation de l'information présente aussi des inconvénients dont l'occlusion et le partage de l'espace de travail. Le problème d'occlusion, soulevé dans plusieurs travaux de recherche [Shen *et al.*, 2006, Forlines *et al.*, 2007, Wu & Balakrishnan, 2003], est dû à l'utilisation des mains sur la surface d'affichage pour effectuer les interactions. Deux types d'occlusions existent : les occlusions de la vue (e.g. un objet virtuel obstrue un autre objet virtuel) et les occlusions de la projection (e.g. la main occulte un objet virtuel).

1.3.3 Tabletop et degrés de liberté

Les degrés de liberté des interactions pour les tabletops varient selon leurs types. Les tables tactiles permettent d'effectuer des interactions en entrée selon deux dimensions uniquement. Le déplacement de données numériques (e.g. photos, objets géométriques, etc.) s'effectue facilement selon deux degrés de liberté en les sélectionnant et les déplaçant avec un doigt. La volonté de manipulation de données numériques avec plus de degrés de liberté en utilisant les tables tactiles a incité les chercheurs à développer de nouvelles techniques d'interaction comme l'utilisation de deux doigts pour le redimensionnement de cartes [Rekimoto, 2002].

Pour la rotation horizontale sur les tables tactiles, Hancock *et al.* [2006] présentent

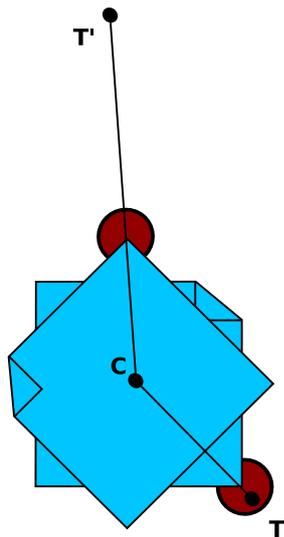


FIGURE 1.9 Orientation de la représentation des données par manipulation d'un manipulateur [Hancock *et al.*, 2006]. C est le centre de rotation. T et T' sont respectivement le point du premier et du dernier contact du doigt avec la surface tactile.

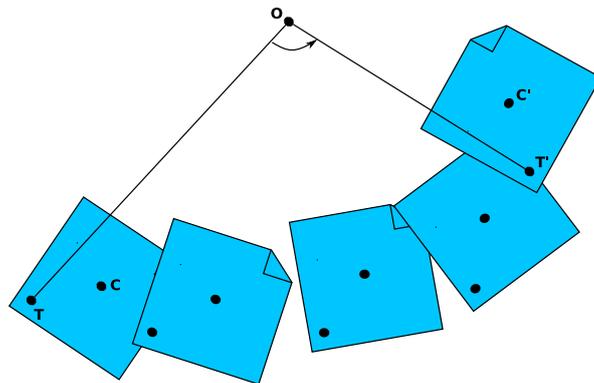


FIGURE 1.10 Orientation continue de la représentation des données selon sa position sur l'affichage [Hancock *et al.*, 2006, Shen *et al.*, 2004]. O est le centre de la vue. C et C' sont les centres de la représentation des données respectivement au début et à la fin de l'interaction. T et T' sont respectivement le point du premier et du dernier contact du doigt avec la surface tactile.

cinq techniques différentes :

1. Définition d'un espace dans la représentation des données dont la manipulation modifie l'orientation (voir figure 1.9). Cette approche ressemble à l'ajout d'un manipulateur et permet d'effectuer la rotation indépendamment de la translation.
2. Orientation de la représentation des données par le système selon sa position sur l'affichage (voir figure 1.10). Cette technique permet d'orienter un document vers le bord le plus proche de la surface d'affichage. L'orientation est toujours identique à un endroit donné de l'affichage. Cette technique est aussi développée par Shen *et al.* [2004].

3. Orientation avec des valeurs d'angles discrètes de la représentation des données par le système selon sa position sur l'affichage (voir figure 1.11). De même que dans la technique 2, l'orientation est toujours identique à un endroit donné de l'affichage.
4. Orientation de la représentation des données selon la trajectoire de sa manipulation (voir figure 1.12). Elle est semblable à la technique "Rotate'N Translate" (RNT) [Kruger *et al.*, 2005, Beaudouin-Lafon, 2001].
5. Manipulation à deux doigts de la position et de l'orientation des représentations des données (voir figure 1.13). Cette manipulation implique un changement d'échelle pour conserver l'invariance des points de contacts par rapport à l'élément manipulé.

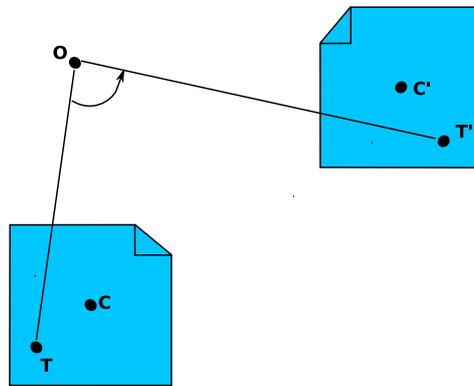


FIGURE 1.11 Orientation discrète de la représentation des données selon sa position sur l'affichage [Hancock *et al.*, 2006]. O est le centre de la vue. C et C' sont les centres de la représentation des données respectivement au début et à la fin de l'interaction. T et T' sont respectivement le point du premier et du dernier contact du doigt avec la surface tactile.

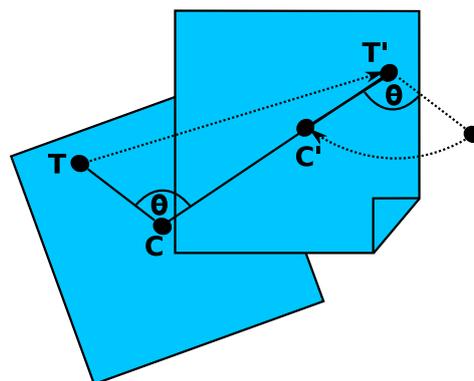


FIGURE 1.12 Orientation de la représentation des données selon sa trajectoire de manipulation [Hancock *et al.*, 2006]. C et C' sont les centres de la représentation des données respectivement au début et à la fin de l'interaction. T et T' sont respectivement le point du premier et du dernier contact du doigt avec la surface tactile.

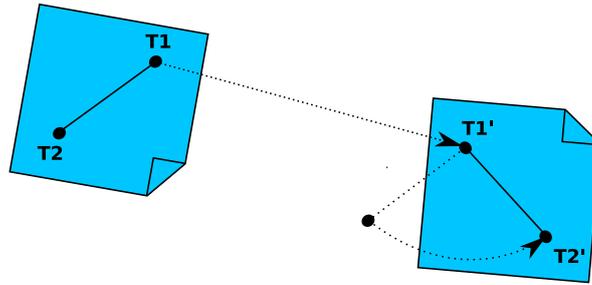


FIGURE 1.13 Manipulation à deux doigts pour la translation et la rotation d'une représentation des données [Hancock *et al.*, 2006]. T1 et T2 sont les points du premier contact des deux doigts avec la surface tactile. T1' et T2' sont les points du dernier contact des deux doigts avec la surface tactile.

En plus de ces cinq techniques dédiées à la rotation, une autre technique nommée TNT a été développée [Liu *et al.*, 2006]. Cette technique se décline en trois types : TNT-hand, TNT-block et TNT-hybrid. Dans la technique TNT-hand un capteur est positionné sur l'un des doigts de l'utilisateur permettant au système de connaître son orientation. Ainsi l'orientation de la représentation des données numériques est modifiée lorsque la main de l'utilisateur change d'orientation tout en touchant la surface tactile à la position de la représentation. Dans la TNT-block, l'utilisateur manipule un objet physique contenant un capteur pour effectuer les rotations. Dans la dernière technique appelée TNT-hybrid, la surface de la représentation des données numériques est séparée en deux : la sélection du centre de la représentation permet d'effectuer des interactions selon la technique TNT-hand tandis que la sélection des parties extérieures de la représentation permet de réaliser les interactions selon la technique RNT.

En plus de la translation horizontale et de la rotation horizontale, certains travaux de recherche se sont focalisés sur le développement de systèmes intégrant d'autres degrés de liberté en rajoutant des manipulateurs 3D ou en utilisant plusieurs points de contacts. Peter Ørbæk [2004] suggère l'utilisation de manipulateurs pour permettre la transformation de fenêtres d'application selon plusieurs degrés de liberté. Cette solution peut sembler adéquate mais le nombre conséquent de manipulateurs ajoute une complexité visuelle importante à l'interface. Trois autres solutions ont été proposées par Hancock *et al.* [2007] pour la manipulation d'un objet virtuel selon six degrés de liberté. Elles se différencient selon le nombre de doigts que l'utilisateur utilise pour effectuer les interactions. La première solution consiste à déplacer et à orienter la représentation de l'objet virtuel selon la technique RNT mais appliquée en 3D en utilisant un seul doigt. La seconde solution consiste à utiliser la technique RNT avec un doigt pour réaliser deux degrés de liberté de translation et un degré de liberté de rotation exclusivement. Les deux degrés de liberté de rotation restants sont pris en charge par la manipulation plane d'un second doigt. Les deux techniques précédentes sont couplées avec la possibi-

lité d'effectuer des translations et des rotations séparément selon la zone de sélection. La troisième solution proposée est l'utilisation du premier doigt pour deux degrés de liberté de translation, le second doigt pour un degré de liberté de rotation et le troisième pour les deux degrés de liberté de rotation restants. Les trois solutions proposées permettent de réaliser des manipulations d'un objet virtuel selon cinq degrés de liberté. Les auteurs soulignent néanmoins quelques problèmes comme la non correspondance de la position sélectionnée par l'utilisateur de l'objet virtuel tout au long de l'interaction dans les deux dernières solutions.

Certains tabletops combinés avec des dispositifs de types HMD permettent l'affichage d'objets virtuels en 3D et leurs manipulations selon plusieurs degrés de liberté. Dans ARTHUR [Penn *et al.*, 2005], les utilisateurs déplacent des objets tangibles sur la surface de la table pour translater et orienter horizontalement les objets virtuels. En plus des objets tangibles à trois degrés de liberté, les utilisateurs de ARTHUR peuvent utiliser un pointeur à cinq degrés de liberté pour les menus et la manipulation des objets virtuels. L'utilisation d'un dispositif annexe est intéressante pour prendre en charge les degrés de liberté impossibles avec les objets tangibles, mais Grossman et Wigdor [2007] suggèrent l'utilisation d'interactions planes s'il est possible de les réaliser avec les dispositifs d'entrée 3D complémentaires.

En résumé, les tables tactiles permettent d'effectuer des interactions par manipulation directe intuitives et faciles. Néanmoins, leur utilisation pour des manipulations d'objets tridimensionnels reste difficile à cause du manque de degrés de liberté et de leur imprécision. Nous montrons dans notre étude que la combinaison d'une table tactile avec d'autres dispositifs en entrée et d'autres vues apporte une solution à ces limites des surfaces tactiles.

1.4 Combinaison de dispositifs d'entrée, de surfaces et de vues multiples

Les combinaisons des dispositifs en entrée, des vues et des surfaces ont été étudié dans la littérature de façon disjointe. La seule exception est le domaine de l'informatique ubiquitaire qui *prend en considération l'environnement naturel des utilisateurs et qui permet aux ordinateurs de disparaître dans le milieu* [Weiser, 1999]. Cette section illustre d'abord divers travaux réalisés dans la combinaison des dispositifs en entrée, des vues multiples et des surfaces multiples. Nous décrivons ensuite des travaux combinant un dispositif de type tabletop avec une surface verticale présentant des informations tridimensionnelles. Nous concluons cette partie par un tableau classifiant les différents

travaux qui combinent des vues et/ou des surfaces multiples et positionnant les TEAMViz par rapport à ces travaux.

1.4.1 Combinaison de dispositifs d'entrées

Dans le domaine de l'informatique ubiquitaire, la combinaison de plusieurs dispositifs en entrée et en sortie est un élément essentiel. GAIA [Roman *et al.*, 2002] est une plate-forme logicielle centralisée permettant la combinaison de plusieurs dispositifs et dédiée aux systèmes ubiquitaires. Elle se base sur un modèle distribué du modèle Model-View-Controller (MVC) appelé Model, Presentation, Controller, Coordinator (MPCC). Tandler [2004] développe aussi une plate-forme logicielle (appelée BEACH) basée sur une architecture avec quatre niveaux d'abstraction (tâche, générique, modèle et cœur) et cinq modèles (interaction, environnement, interface utilisateur, application et données). L'élément intéressant dans BEACH et GAIA est l'utilisation du réseau comme un moyen de communication entre les différentes parties des systèmes. Cependant, l'objectif des recherches dans le domaine de l'informatique ubiquitaire est de faire disparaître les ordinateurs de la vue des utilisateurs (pervasivité) et la prise en considération de l'environnement humain par les ordinateurs (nomadisme) [Weiser, 1999]. Ces deux points ne sont pas essentiels dans notre travail de recherche même si l'environnement, dans le cadre de la disposition de plusieurs surfaces d'affichage, est important au stade de la conception des interactions.

A un niveau de conception inférieur, l'ICON [Dragicevic & Fekete, 2004] est une boîte à outil permettant une configuration rapide et facile des dispositifs en entrée avec les applications d'un système de type bureau de travail. Un exemple d'application est la configuration d'une souris et d'une tablette graphique pour une application graphique. Le point intéressant de cette boîte à outil est la prise en considération des degrés de liberté d'un dispositif en entrée et leurs liaisons avec l'application. Néanmoins, son utilisation intervient à un niveau très bas de la conception des interfaces. iStuff [Ballagas *et al.*, 2003] est une boîte à outil permettant la conception d'interactions en utilisant des dispositifs hétérogènes pour les systèmes ubiquitaires. La classification des dispositifs selon les dimensions en entrée et en sortie est intéressante dans iStuff mais elle n'illustre pas les relations existant entre ces dispositifs. Dans un autre travail de recherche, Kobayashi *et al.* [2005] proposent un espace de travail pour les systèmes multimodaux et multi-dispositifs en entrée. Cet espace de travail permet la mise en relation de dispositifs d'entrée et les services proposés par un système ubiquitaire. Cette approche ne se focalisant pas uniquement sur les applications graphiques est alors plus générique que l'ICON.

1.4.2 Combinaison de vues multiples

La combinaison de vues multiples offre la possibilité de visualiser des informations complémentaires de différentes façons. Elle constitue la base des modeleurs 3D (e.g. Blender [Blender]). L'utilisation de vues multiples a été étudiée dans la réalité virtuelle ainsi que pour des applications bidimensionnelles. L'un de ces travaux est le World In Miniature (WIM, soit en français monde en miniature) qui est une technique d'interaction dans un environnement virtuel présentant l'information à l'échelle normale de plus d'une version en miniature [Stoakley *et al.*, 1995]. Ainsi l'utilisateur, équipé d'un dispositif de type HMD, peut sélectionner un objet virtuel au moyen de la vue en miniature en utilisant un dispositif 3D en entrée, puis effectuer la manipulation de cet objet à travers la vue à l'échelle normale. Ces vues multiples permettent ainsi de sélectionner des objets virtuels qui ne sont pas dans le champ de vue de l'utilisateur ou qui ne sont pas dans le champ de sélection de l'utilisateur (i.e. en utilisant un dispositif de tracking et une technique d'interaction faisant une correspondance 1 : 1 entre le monde physique et le monde numérique). L'efficacité de la version bidimensionnelle de WIM s'appelant la "Radar view" (soit en français vue de radar) a été montrée par Nacenta *et al.* [2005]. WIM et la Radar view illustrent l'utilité de vues multiples pour les interactions en entrée même si les vues sont redondantes.

Roberts [1998] soutient l'utilisation de vues multiples dans un système de visualisation du fait qu'un *utilisateur peut mieux comprendre l'information de différentes perspectives, surmonter d'éventuelles interprétations erronées et effectuer une visualisation interactive d'investigation en réalisant une corrélation entre les vues*. Il souligne des éléments essentiels qui doivent être pris en considération lors de la combinaison de vues multiples tels que la cohérence et le couplage des vues. Ces éléments sont aussi importants lors de la combinaison d'interactions effectuées au moyen de vues multiples affichées sur des surfaces distinctes.

Baldonado *et al.* [2000] ont étudié la combinaison de vues multiples d'un point de vue général et exposent huit règles à respecter pour la conception de systèmes à vues multiples. Les huit règles sont :

- **Diversité** : utiliser des vues multiples lorsque les attributs, les modèles, les profils utilisateurs, les niveaux d'abstraction et les genres sont variés ;
- **Complémentarité** : utiliser des vues multiples lorsque celles-ci soulignent les corrélations et les différences ;
- **Décomposition** : décomposer les données complexes sur plusieurs vues ;
- **Minimalisation** : utiliser le minimum de vues possibles ;
- **Optimisation des ressources temporelles et spatiales** : équilibrer entre le

- coût et les apports des vues multiples au niveau spatial et temporel ;
- **Evidence** : rendre évident à l'utilisateur les relations entre les vues en utilisant des techniques de perception ;
- **Cohérence** : les interfaces et les états des vues doivent être cohérents ;
- **Gestion de l'attention** : gérer l'attention de l'utilisateur sur une vue à un instant donné en utilisant des techniques de perception.

Ces règles nous semblent essentielles pour la combinaison de vues multiples. Elles interviennent au niveau de la combinaison de plusieurs vues et non de plusieurs surfaces de présentation de l'information. Cependant, ces règles sont toujours applicables lorsque plusieurs surfaces avec des vues multiples sont combinées comme dans notre travail.

Equal	Overlaps	Meets	Before	During
				

TABLE 1.1 Relations temporelles non dirigées d'Allen [1983].

Dans un autre travail de recherche, Vernier [2001] étend les relations temporelles non dirigées d'Allen [1983] (voir la table 1.1) aux relations spatiales, articulatoires, syntaxiques et sémantiques. Cette extension est dédiée aux modalités en sortie dont les vues. Les relations spatiales (disjointe, adjacente, intersectée, imbriquée et recouvrance) entre les vues sont importantes dans le cadre de combinaison de vues multiples d'un même environnement virtuel et de surfaces multiples. Nous nous inspirons des relations proposées par Vernier pour réaliser une classification des systèmes et des études illustrés dans cette section.

1.4.3 Combinaison de surfaces multiples

La combinaison de surfaces multiples a été étudiée dans le domaine des interfaces ubiquitaires. Lachenal [2004] présente les systèmes couplant des surfaces et des dispositifs multiples. L'un de ces systèmes est le PointRight [Johanson *et al.*, 2002] qui permet de combiner plusieurs souris et claviers avec plusieurs surfaces mais avec un seul dispositif de pointage par surface. L'élément intéressant de ces travaux est l'étude des relations existant entre les différentes surfaces et le couplage entre l'espace numérique et l'espace physique. En comparaison avec ces travaux, nous n'étudions ni les aspects de couplage dynamique ni les aspects de migration des dispositifs et des surfaces.

Dans un autre type d'approche, l'augmentation d'une table tactile par des écrans verticaux permet d'accroître l'espace d'affichage de l'information. La combinaison de plusieurs surfaces verticales avec une table tactile est réalisée par Wigdor *et al.* [2006].

La table tactile représente l'espace affiché sur les surfaces verticales en miniature, créant un couplage entre l'espace numérique et l'espace physique. Les interactions en entrée s'effectuent uniquement au moyen de la table tactile. Pour interagir indirectement avec les vues présentées verticalement, un pointeur est affiché sur ces vues qui représente le point du touché de l'utilisateur sur la vue en miniature (présentée sur la table tactile).

Dans un travail semblable, Forlines *et al.* [2006] combinent plusieurs surfaces verticales avec une table tactile pour une application géo-spatiale. L'élément intéressant de ce travail est la complémentarité des vues (2D sur la table tactile, 3D sur une surface verticale et 2D augmentée d'informations supplémentaires sur un deuxième écran vertical). L'affichage d'une représentation de caméra sur la vue 2D permet la navigation dans la vue 3D présentée verticalement. Les interactions s'effectuent par manipulation directe sur la table tactile ainsi que sur une tablettePC pour les menus nécessitant un espace d'affichage important. Les manipulations tridimensionnelles sont impossibles dans ce système.

En résumé, les surfaces tactiles permettent non seulement d'effectuer des interactions directes et des manipulation intuitives, mais elles peuvent également se combiner avec des surfaces d'affichage supplémentaires pour permettre un affichage d'une quantité d'information plus importantes. Cette combinaison permet également l'affichage d'information complémentaire à celle de la surface tactile. Toutefois, l'information sur la relation existant entre les différentes surfaces doit être facilement perceptible par l'utilisateur afin de ne pas le désorienté.

1.4.4 Interfaces 2D/3D

La combinaison d'un dispositif tabletop bidimensionnel, sur lequel est projetée une vue 2D, avec une surface verticale affichant une vue 3D des mêmes données numériques a été développée dans plusieurs travaux de recherche. BUILD-IT [Fjeld *et al.*, 1998] est un système combinant une table tangible avec une surface d'affichage verticale. L'utilisateur interagit dans BUILD-IT en utilisant des objets tangibles (des bricks). Une des applications possibles de ce système est l'aide à la conception 3D en affichant une vue bidimensionnelle sur la table tangible et une vue 3D sur la surface verticale. Kuchar *et al.* [2007] utilisent une table tactile sur laquelle est affichée la carte d'un château virtuel dont une vue 3D est également présentée sur un écran vertical. Les interactions s'effectuent sur la table tactile pour la navigation dans la vue 3D.

En plus des environnements architecturaux, la combinaison d'une vue 2D présentée sur un dispositif tabletop avec une vue 3D affichée sur un écran vertical a été développée

pour des applications en biologie. NAVRNA [Bailly *et al.*, 2006] est un système combinant une table tangible avec une projection verticale pour la navigation, l'exploration et l'édition d'une séquence d'acide ribonucléique (ARN). Une représentation bidimensionnelle dite structure secondaire est présentée sur une table avec laquelle un utilisateur interagit au moyen de deux objets tangibles. Une représentation tridimensionnelle plus complexe est projetée verticalement sur un écran. Aliakseyeu *et al.* [2002] développent un système combinant une table tactile avec un écran vertical pour des applications de planification chirurgicale. La différence entre ce travail de recherche et les trois autres cités dans cette partie est la présentation de plusieurs vues sur une même surface d'affichage. De plus, une portion de la vue horizontale (appelée "3to2D window") est réservée pour simuler l'existence de l'objet présenté en 3D réellement sur cette partie. Ainsi, en déplaçant un objet physique tracké en forme de carré sur la "3to2D window", l'utilisateur peut visualiser différentes coupes de l'objet 3D. Des applications architecturales et géo-spatiales sont aussi proposées dans ce travail.

Les travaux présentés dans cette section illustrent différentes applications de la combinaison d'un dispositif tabletop avec une surface d'affichage verticale. Par rapport à notre objectif de combinaison de plusieurs surfaces avec plusieurs dispositifs en entrée, aucune interaction n'est effectuée ici de façon directe (e.g. par le touché) ou partiellement directe (e.g. par pointage) au moyen de la surface verticale.

En s'inspirant des relations entre les différentes modalités en sortie proposé par Vernier [2001], nous classifions les travaux exposés dans cette section selon les relations sémantiques existant entre les différentes vues dans la table 1.2. Ces relations sémantiques sont dans notre classification des relations de l'espace virtuel présenté sur chacune des vues. Nous considérons que chaque surface contient au moins une vue sur l'espace numérique. Ainsi l'étude de la combinaison de plusieurs surfaces est effectuée dans la table 1.2 selon la relation spatiale existant entre les espaces numériques affichés sur ces vues. Nous différencions dans cette table trois types de combinaisons selon le nombre de vues et le nombre de vues actives à un instant donné. La concurrence en entrée entre les surfaces peut exister uniquement dans les cases contenant le symbole *****. Dans le cadre de notre travail de recherche sur les TEAMViz, nous nous situons dans la case contenant le symbole **+**.

1.5 Modèles pour l'interaction

Après avoir exposé les interfaces graphiques tridimensionnelles et les tablespots pour la réalisation de tâches spatiales et après avoir présenté les études se focalisant sur la

	Une surface active	Plusieurs surfaces dont une active	Plusieurs surfaces actives
	[Stoakley <i>et al.</i> , 1995] [Nacenta <i>et al.</i> , 2005] [Baldonado <i>et al.</i> , 2000]		*
	[Baldonado <i>et al.</i> , 2000]	[Fjeld <i>et al.</i> , 1998] [Aliakseyeu <i>et al.</i> , 2002] [Bailly <i>et al.</i> , 2006] [Kuchar <i>et al.</i> , 2007]	*⋄
	[Baldonado <i>et al.</i> , 2000]	[Wigdor <i>et al.</i> , 2006]	[Lachenal, 2004] [Johanson <i>et al.</i> , 2002]
	[Baldonado <i>et al.</i> , 2000]		[Johanson <i>et al.</i> , 2002]
	[Baldonado <i>et al.</i> , 2000]		*

TABLE 1.2 Classification des travaux de recherche sur la combinaison des surfaces et des vues multiples selon les relations spatiales entre les vues et les surfaces actives en entrée.

combinaison de dispositifs en entrée, de vues multiples et de surfaces multiples, l'étude des architectures et des modèles pour la réalisation des TEAMViz est essentielle. Dans le domaine de l'IHM, une distinction existe entre les modèles d'architecture, qui se focalisent sur les aspects d'architectures logicielles des systèmes, et les modèles de conception qui sont dédiés à la conception des interactions et des interfaces. Nous étudions dans un premier temps les architectures développées dans le domaine de l'IHM. La multimodalité étant un aspect important de notre travail, nous présentons ensuite les modèles de conception d'interactions et d'interfaces développés pour ce domaine de recherche. Nous concluons cette section avec les modèles de conception d'interactions et d'interfaces pour les réalités augmentée et mixte qui sont des domaines intéressants dans le cadre de la combinaison de surfaces d'affichages physiques multiples et de vues numériques multiples d'un même espace virtuel.

1.5.1 Modèles d'architecture logicielle en IHM

Dans le domaine de l'IHM, plusieurs modèles ont été développés pour les architectures des systèmes interactifs. La majorité de ces modèles ont été illustrés dans la thèse de Dragicevic [2001]. Seeheim [Pfaff, 1985] est un modèle inspiré de la linguistique qui se base sur trois composantes : l'*interface avec le noyau fonctionnel*, le *contrôleur de dialogue* et la composante *présentation* (voir figure 1.14). Le rôle de l'*interface avec le noyau fonctionnel* est défini comme la partie sémantique d'un système d'interaction.

Le *contrôleur de dialogue* est quant à lui défini comme la partie syntaxique d'un tel système tandis que la composante *présentation* est la partie lexicale. Les points négatifs de cette architecture sont l'absence de précision sur le rôle de chaque composante [Nigay & Coutaz, 1991] ainsi que la séquentialité de ce système.

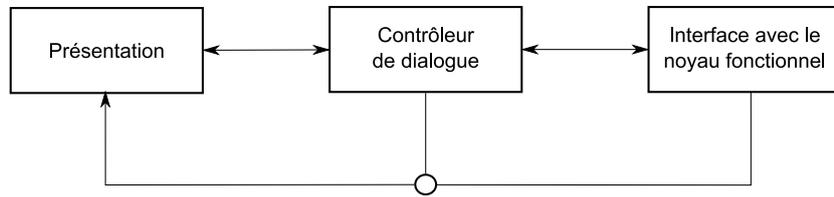


FIGURE 1.14 Modèle Seeheim [Pfaff, 1985].

ARCH [UIMS, 1992] est un modèle d'architecture qui étend le modèle Seeheim. Il se base sur cinq composantes (voir figure 1.15) :

- le *noyau fonctionnel* qui réalise la gestion et la représentation des données. Il contient aussi des fonctionnalités spécifiques aux données ;
- l'*adaptateur du noyau fonctionnel* est un médiateur entre le *noyau fonctionnel* et le *contrôleur de dialogue*. Les tâches qui sont nécessaires mais qui ne sont pas fournies dans le *noyau fonctionnel* sont développées à ce niveau ;
- le *contrôleur de dialogue* gère le séquençement des tâches pour l'interface utilisateur ainsi que pour la partie du *noyau fonctionnel* qui pourrait être altérée par l'utilisateur. De plus, il gère la communication entre l'interface utilisateur et le *noyau fonctionnel*. Dans le cas de vues multiples, cette composante gère aussi la cohérence des vues ;
- la *présentation* est le médiateur entre le *contrôleur de dialogue* et la *boite à outil d'interaction*. Elle fournit des objets indépendants de la *boite à outil d'interaction* pour le *contrôleur de dialogue* et gère la représentation des objets interactifs ;
- la *boite à outil d'interaction* est connectée aux dispositifs d'entrée et de sortie.

Le modèle ARCH définit aussi les mécanismes de communication entre ses composantes :

- les *objets du domaine* sont utilisés par le *noyau fonctionnel* ainsi que l'*adaptateur du noyau fonctionnel* mais leurs rôles diffèrent dans les deux cas. Dans le *noyau fonctionnel*, les objets du domaine sont utilisés pour fournir des fonctionnalités qui ne sont pas en relation avec l'interface utilisateur (comme la récupération d'une base de données d'une liste de produits et de prix). Dans l'*adaptateur du noyau fonctionnel*, les objets du domaine fournissent des fonctionnalités qui sont en relation avec l'interface utilisateur (comme le tri des produits selon leurs prix) ;
- les *objets de présentation* sont des objets qui contiennent les événements pouvant être générés par les utilisateurs ainsi que les données à présenter ;

- les *objets d'interaction* sont fournis par la composante *boite à outil d'interaction* à plusieurs niveaux de complexité.

La notion d'objet ne fait pas référence à un objet spécifique ou à la programmation objet mais elle est utilisée pour représenter un élément quelconque. Les composantes *noyau fonctionnel* et *boite à outil d'interaction* sont souvent prédéfinies avant le développement des interfaces. C'est pour cette raison que ces deux composantes se présentent comme les bases du modèle. Un élément intéressant de ce modèle est le fait que la définition des limites des composantes n'est pas fixe et se base sur un méta-modèle appelé Slinky (comme le nom du jouet qui a inspiré ce méta-modèle).

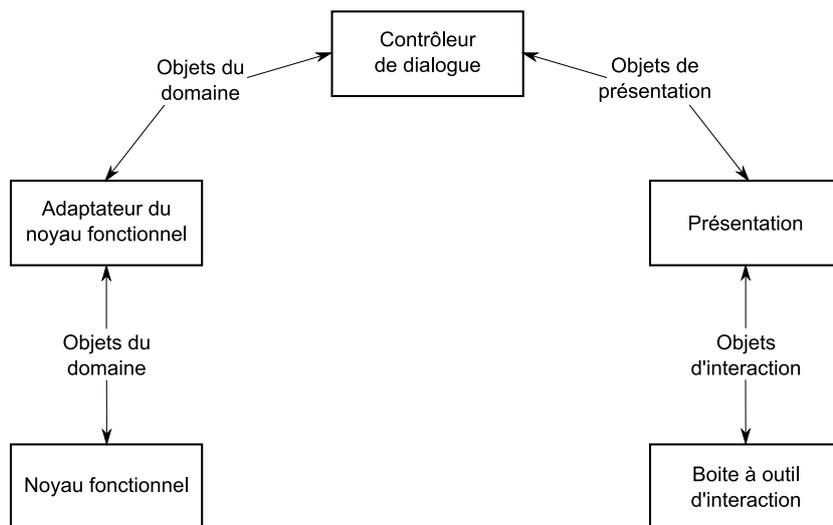


FIGURE 1.15 Modèle ARCH [UIMS, 1992].

L'architecture MVC (Model-View-Controller ou en français Modèle-Vue-Contrôleur) [Krasner & Pope, 1988] est apparue avec l'environnement Smalltalk dans les années quatre-vingt. Cette architecture est une architecture multi-agents composée de trois composantes :

- le *modèle* représente des données ou des objets complexes ;
- la *vue* est la partie visible par l'utilisateur ;
- le *contrôleur* reçoit les interactions en entrée de l'utilisateur.

Lorsqu'un *contrôleur* reçoit des interactions en entrée d'un utilisateur, il transmet immédiatement ces informations à la composante *modèle* et éventuellement à la composante *vue* pour un retour d'information rapide à l'utilisateur. Lorsque des éléments ou l'état de la composante *modèle* sont modifiés, il en informe la composante *vue* pour que celle-ci présente cette information à l'utilisateur. L'un des inconvénients de ce modèle est sa limitation aux interfaces graphiques. Néanmoins, il a été utilisé dans plusieurs travaux de recherche et a été étendue à d'autres modèles. L'un de ces modèles est le MPCC (Model, Presentation, Controller, Coordinator ou Modèle, Présentation, Contrôleur et

Coordinateur), qui remplace la *vue* par *présentation* et intègre une nouvelle composante qui est le coordinateur [Roman *et al.*, 2002]. Ce modèle a été conçu pour les systèmes ubiquitaires qui gèrent plusieurs dispositifs en entrée et en sortie.

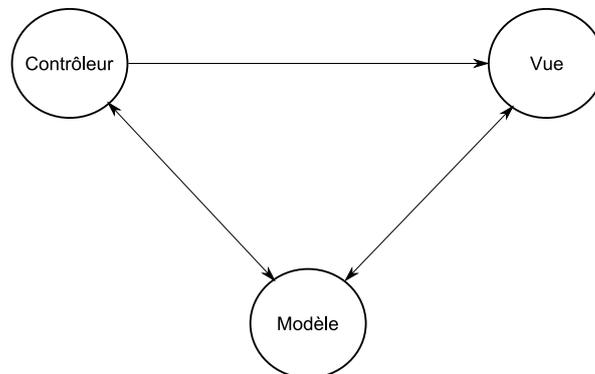


FIGURE 1.16 Modèle MVC [Krasner & Pope, 1988].

Le modèle PAC (Présentation, Abstraction et Contrôle) développé par Coutaz [1987] est un modèle hiérarchique à base d'agents. Chaque agent possède trois composantes ou facettes :

- la *présentation* définit les entrées et les sorties perçues par l'utilisateur ;
- l'*abstraction* représente la sémantique du système dans laquelle sont définies les fonctionnalités. Ces fonctionnalités se déduisent d'une analyse des tâches ;
- le *contrôle* fait la liaison entre la composante abstraction et la composante présentation et préserve leur cohérence. Il effectue la liaison avec les autres agents PAC (au-dessus et au-dessous dans la hiérarchie).

Ce modèle permet de mettre en relation la partie noyau fonctionnel (ou la partie sémantique) avec la partie interface graphique d'un système interactif au moyen de la composante *contrôle*. Cette composante n'existe pas directement dans le modèle MVC.

Le modèle PAC-Amodeus [Nigay & Coutaz, 1991] combine le modèle PAC avec le modèle ARCH. Nigay et Coutaz proposent d'ajouter des agents PAC au niveau du *contrôleur de dialogue* du modèle ARCH. Ainsi, en intégrant des agents PAC au niveau du *contrôleur de dialogue*, il offre la possibilité de mieux le définir et le développer. Dans ce modèle, la composante *adaptateur du noyau fonctionnel* est connecté à l'élément *abstraction* de chaque agent PAC du *contrôleur de dialogue* tandis la composante *présentation* est connectée à l'élément *présentation* de chacun de ces agents.

Les modèles présentés dans cette partie sont de haut niveau. Leur intérêt est qu'ils sont génériques et sont applicables à un grand nombre de systèmes interactifs. Néanmoins, ils restent très généraux et ne permettent pas par exemple de bien définir les techniques d'interaction. Ces modèles ne sont pas assez spécifiques pour un espace de

conception d'interactions tel que le notre combinant plusieurs vues, plusieurs surfaces et plusieurs dispositifs en entrée.

1.5.2 Modèles pour la multimodalité

Un système multimodal est un système intégrant plusieurs modalités en entrée et/ou en sortie. Nigay et Coutaz [1995] définissent une modalité comme étant un couple dispositif/langage. Un exemple de modalité est le couple microphone/langage naturel. Cette définition de modalité s'applique aux modalités en entrée ainsi qu'aux modalités en sortie. L'architecture des systèmes multimodaux est souvent divisée en trois parties (voir figure 1.17) :

- fusion de l'information : les informations provenant des modalités en entrée sont fusionnées et envoyées au contrôleur de dialogue ;
- contrôleur de dialogue : il relie les modalités en entrée aux modalités en sortie. Il gère aussi la cohérence des données ;
- fission de l'information : reçoit les informations du contrôleur de dialogue et renvoie les informations selon les modalités en sortie.

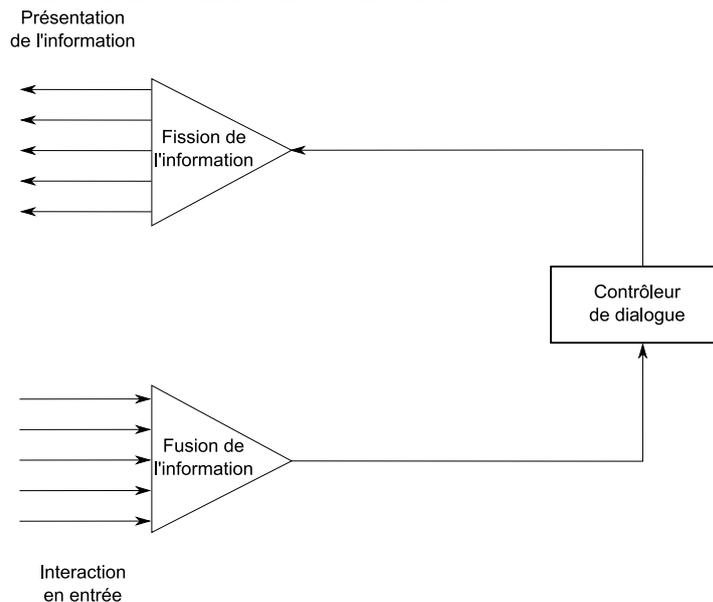


FIGURE 1.17 Architecture générale des systèmes multimodaux.

Plusieurs travaux de recherche se sont focalisés sur la définition des relations entre plusieurs modalités. Martin [1998] développe l'approche TYCOON (de TYpes and goals of COOperationN soit en français types et buts de la coopération) pour caractériser les modes de coopération des modalités dans un système multimodal. Cette caractérisation se base sur cinq types :

- *Transfert* : il existe lorsqu'une information produite par une modalité est utilisée par une autre modalité ;

- *Equivalence* : elle intervient lorsqu’une information peut être traitée par plusieurs modalités mais une d’entre elles suffit pour effectuer le traitement ;
- *Spécification* : elle existe lorsqu’un type d’information ne peut être traité que par une unique modalité ;
- *Redondance* : elle intervient lorsque plusieurs modalités traitent simultanément la même information de façon redondante ;
- *Complémentarité* : plusieurs informations sont traités de façon complémentaires par plusieurs modalités mais doivent être fusionnées.

Les propriétés CARE [Coutaz *et al.*, 1995] décrivent et caractérisent les relations intermodales selon quatre propriétés organisées selon deux axes (portée des choix et combinaison des choix) :

- *Complémentarité* : des modalités doivent être utilisées de façon complémentaires pour la réalisation d’une tâche ;
- *Assignment* (ou affectation) : une tâche ne peut être réalisée qu’au moyen d’une unique modalité ;
- *Redondance* : plusieurs modalités sont utilisées de façon redondante pour accomplir une tâche ;
- *Equivalence* : plusieurs modalités existent pour la réalisation d’une tâche donnée mais l’utilisation d’une seule modalité suffit pour sa réalisation.

Ces propriétés s’appliquent aux modalités en entrée mais peuvent être étendues aux modalités en sortie.

Inspirées par les propriétés CARE, ICARE [Bouchet & Nigay, 2004, Bouchet *et al.*, 2004b] est une plate-forme basée sur l’architecture ARCH pour la combinaison de modalités en entrée. ICARE intervient sur les composantes *boite à outil d’interaction et présentation* du modèle ARCH. La combinaison des modalités en entrée est définie selon la relation entre elles. En effet, plusieurs dispositifs liés à leurs langage sont reliés à des boites de complémentarité, d’équivalence ou de redondance/équivalence. Celles-ci sont ensuite reliées à des tâches spécifiques. Pour une affectation, un dispositif et son langage sont directement liés à la tâche. ICARE a été étendu au domaine ubiquitaire en prenant en compte le contexte [Bouchet *et al.*, 2004a]. Cette plate-forme permet la combinaison de modalités en entrée mais ne tient pas compte des modalités en sortie.

Dans un autre travail de recherche, Palanque et Schyn [2003] proposent l’utilisation d’un formalisme appelé ICO [Bastide *et al.*, 1998] pour la fusion de l’information en entrée dans un système multimodal. Le formalisme ICO utilise les réseaux de Pétri [Petri, 1962] pour décrire les aspects dynamiques d’un système interactif. Ainsi, une technique d’interaction est décrite formellement en modélisant des réseaux de Pétri. L’utilisation du formalisme ICO pour la multimodalité a été étendue aux systèmes

multimodaux de réalité virtuelle [Schyn *et al.*, 2003, Navarre *et al.*, 2005].

Au niveau de la multimodalité en sortie, Rousseau *et al.* [2005] proposent un modèle conceptuel appelé WWHT (What, Which, How, Then). Ce modèle décrit le cycle de vie de la présentation d'une information multimodale basée sur les quatre notions :

- *What : Quelle information présenter ?* Dans cette phase, une décomposition de l'information sémantique en plusieurs informations élémentaires est réalisée ;
- *Which : Quelles modalités choisir pour la présenter ?* Une allocation des modalités aux informations élémentaires est réalisée dans cette phase selon des critères de sélection prenant en compte le contexte d'interaction ;
- *How : Comment présenter cette information à travers ces modalités ?* Dans cette partie, une instanciation de la présentation est réalisée au niveau des modalités en définissant la syntaxe et les attributs des modalités ;
- *Then : Comment faire évoluer cette présentation ?* Cette partie gère l'évolution de la présentation de l'information soit par des actions des utilisateurs soit par le contexte du système.

Ce modèle prend en compte le contexte dans lequel se situe le système multimodal à deux niveaux. Le premier niveau se situe dans les parties *What*, *Which* et *How* où une vérification de la cohérence de la présentation est effectuée. Le deuxième niveau se situe dans la partie *Then* où la présentation évolue selon le contexte et les actions des utilisateurs. Le modèle WWHT est spécifique à la présentation multimodale de l'information en sortie.

En résumé, dans le domaine de la multimodalité, des propriétés ont été conçues pour classifier les relations entre les différentes modalités comme les propriétés CARE ou TYCOON. Des modèles et des plate-formes ont été développés pour la combinaison des modalités en entrée et des modalités en sortie. Néanmoins, ces travaux de recherche étudient la multimodalité en entrée et la multimodalité en sortie de façon disjointe.

1.5.3 Modèles pour la réalité augmentée et la réalité mixte

Pour bien définir la diversité des environnements mixtes allant du réel au virtuel, Milgram et Kishino [1994] développent le concept de continuum virtuel (voir figure 1.18). La réalité mixte est à mi-chemin entre le monde réel et le monde virtuel. Elle essaie de fusionner le monde réel avec un monde virtuel. La réalité augmentée constitue une partie de la réalité mixte qui tend à augmenter le monde réel avec des données numériques. Un exemple de dispositifs de réalité augmentée est la table tactile.

ASUR [Dubois *et al.*, 2001, 2002] est une notation pour décrire des systèmes interactifs de réalité mixte. Il se base sur quatre éléments :



FIGURE 1.18 Continuum entre le réel et le virtuel [Milgram & Kishino, 1994].

- *Adaptateur* : un adaptateur effectue la liaison du réel vers le système (adaptateur d'entrée) et ou dans le sens inverse (adaptateur de sortie) ;
- *Système* : système informatique ;
- *Utilisateur* : utilisateur qui manipule des outils ;
- *Objet réel* : objet réel nécessaire pour la tâche. Cet objet peut être un outil que l'utilisateur manipule ou l'objet de la tâche en elle même.

ASUR permet de décrire les liaisons entre un système et le monde réel dans un système de réalité mixte.

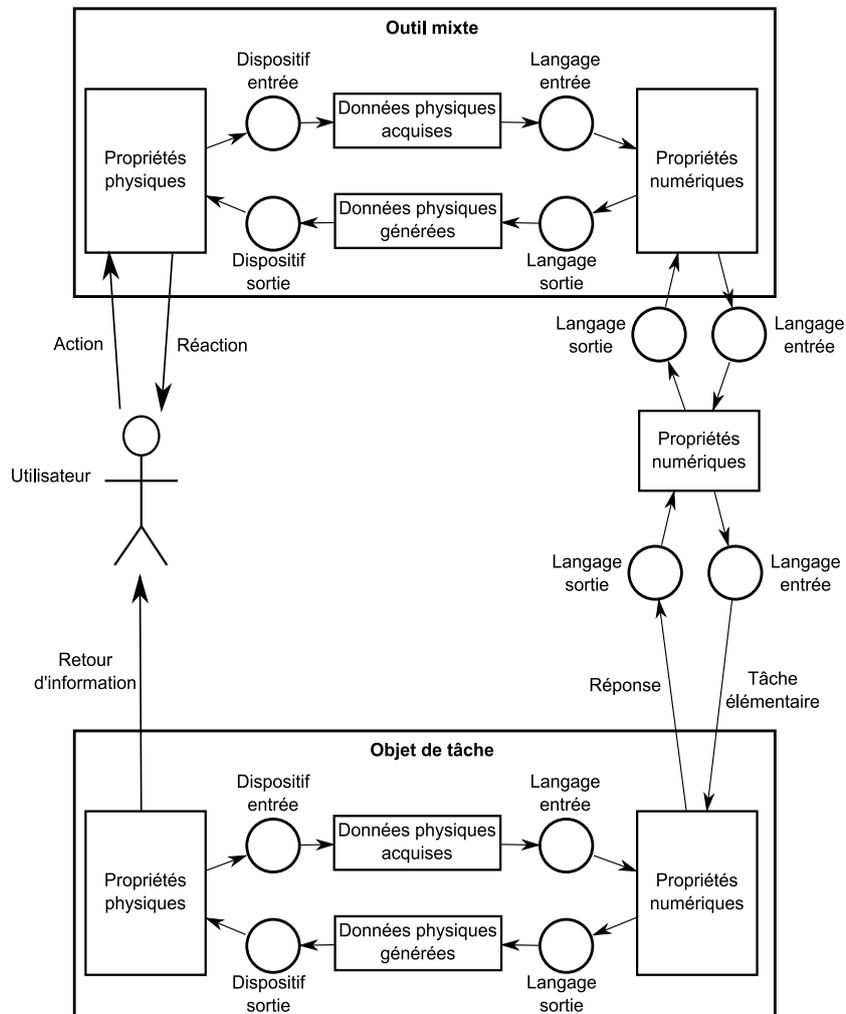


FIGURE 1.19 Modèle pour l'interaction dans un environnement mixte développé par [Coutrix & Nigay, 2006].

Coutrix et Nigay [2006] développent un modèle pour l'interaction dans des environ-

nements mixtes. Ce modèle relie le monde réel et le monde numérique au niveau d'un outil mixte (voir figure 1.19). Les propriétés physiques sont acquises par un dispositif physique et un langage interprétant les données qui les transmettent aux propriétés numériques. Dans l'autre direction, les propriétés numériques génèrent des informations qui seront traduites par un langage et envoyées ensuite à un dispositif qui rendra l'information perceptible physiquement au niveau des propriétés physiques. Un outil mixte peut être relié à un outil/objet logique qui lui sera relié à un autre objet. Ce modèle a été ensuite appliqué pour l'étude de la modélisation d'un objet mixte [Coutrix & Nigay, 2008].

Un modèle d'architecture pour les systèmes de réalité mixte collaboratifs a été développé par Chalon et David [2004]. Ce modèle s'appelle IRVO (Interacting with Real and Virtual Objects ou en français interaction avec des objets réels et virtuels). Il est basé sur trois entités principales :

- *Utilisateur* : il représente l'utilisateur ainsi que trois de ses canaux sensoriels (l'ouïe, la vue et le canal kinesthésique/haptique) ;
- *Objet/Outil* : un objet ou un outil sont réels ou virtuels. Un objet réel est une feuille de papier par exemple alors qu'un stylet est un outil réel. Un objet virtuel est par exemple un document numérique et le pointeur de la souris est un outil virtuel ;
- *Modèle* : il représente le système.

Le passage entre le monde réel et le monde virtuel s'effectue par les dispositifs d'entrée et de sortie du système. Le point intéressant de ce modèle est la modélisation du système en mettant en relation le monde numérique avec le monde physique jusqu'aux canaux sensoriels de chaque utilisateur.

En résumé, les modèles et les architectures pour les systèmes de réalité mixte et augmentée tentent de mettre en relation les données numériques avec le monde réel. L'utilisateur est considéré comme un élément important dans ce type de système. Dans les TEAMViz, l'étude de la relation entre le monde virtuel et le monde physique est importante en raison des vues multiples sur la scène virtuelle présentée sur des surfaces d'affichage distinctes. De plus, la table tactile est un objet physique (i.e. un mobilier dans une pièce) qui garde ses propriétés de table et qui est augmenté par l'affichage de données numériques.

1.6 Synthèse

Dans ce chapitre, nous avons étudié les travaux de recherche pouvant aider à la conception des TEAMViz. Tout d'abord, nous avons présenté les travaux dédiés aux

3DGUI qui ont commencés dans les années soixante. Ce domaine de recherche est en constant développement tant au niveau de la conception des interactions et des systèmes qu'au niveau des dispositifs. Les types de tâches identifiées pour les interfaces tridimensionnels sont : la manipulation d'un objet virtuel (que nous étudions aux chapitres 2 et 4), la navigation dans un environnement 3D (que nous étudions partiellement au chapitre 4 et plus en détail au chapitre 6) et les tâches de contrôle. Les dispositifs d'entrée dédiés à ce type d'interfaces sont classifiés selon les degrés de liberté qu'elles intègrent. Cette classification nous a incité à les considérer comme des éléments centraux de la conception des interactions (voir les chapitres 2, 3 et 4).

Le domaine des tablesps et plus précisément des tables tactiles est un domaine plus jeune mais est aussi en constante expansion. Cette expansion est due à l'aspect intuitif des interactions (dans le cas des tables tactiles) ainsi qu'aux aspects collaboratifs. Plusieurs techniques d'interaction existant pour les tables tactiles ont été considéré dans la première phase de conception que nous présentons au chapitre 2. De plus, ces techniques nous ont inspirées dans la conception des techniques de navigation dans un système de type TEAMViz (voir chapitre 6).

Les systèmes présentées dans ce chapitre pour la combinaison de dispositifs multiples en entrée relient les dispositifs aux tâches élémentaires à réaliser. Cette démarche de connecter les dispositifs en entrée aux tâches élémentaires nous a inspirée pour définir la première étape de notre espace de conception (chapitre 2). Les systèmes de type TEAMViz combinent, de plus des dispositifs multiples en entrée, des vues multiples affichées sur des surfaces différentes. D'une part, les travaux de recherche dédiés à la combinaison de vues multiples exposent des règles à respecter pour faciliter la détermination des vues à intégrer dans un système. D'autre part, les travaux de recherche sur la combinaison de surfaces d'affichage multiples étudient la connectivité des surfaces. Ces deux domaines d'étude nous ont inspiré dans l'élaboration de certaines règles de conception du chapitre 2.

Les modèles pour l'architecture des systèmes interactifs dans le domaine de l'IHM sont nombreux. Nous définissons dans le chapitre 2 un modèle basé sur le modèle ARCH pour positionner les différentes étapes de notre espace de conception. En IHM, le domaine qui se rapproche au mieux de nos travaux est la multimodalité. Certains travaux dans ce domaine (dont les propriétés CARE) définissent des propriétés pour caractériser les relations entre les différentes modalités. Nous caractérisons les relations entre les différentes techniques d'interaction selon les tâches à réaliser avec certaines des propriétés CARE. Finalement, la réalité augmentée permet d'augmenter le monde réel avec des données numériques. Les modèles conçues pour la réalité augmentée tentent de

construire une liaison entre les données du monde réel et les données numériques. Cette liaison entre les interactions dans le monde réel et l'environnement virtuelle est étudiée au chapitre 2. Nous étudions aussi les différents types de relation entre les vues dans les TEAMViz selon si elles intègrent ou non les relations physiques entre les surfaces d'affichage dans le chapitre 6.

1.7 Conclusion

En résumé de ce chapitre, il n'existe ni un dispositif en entrée ni une technique d'interaction idéale pour les tâches nécessitant plusieurs degrés de liberté. Le nombre de dispositifs d'entrée conçus pour les 3DGUI est conséquent mais chacun présente des inconvénients ne lui permettant pas d'être le dispositif universel pour ce type d'interfaces. Les inconvénients majeurs de ces dispositifs sont la fatigue et la manipulation de plusieurs degrés de liberté en même temps les rendant difficiles à utiliser. Les dispositifs tabletops et surtout les tables tactiles présentent plusieurs avantages dont le couplage entre l'espace d'interaction en entrée et la présentation de l'information en sortie permettant des interactions simples et intuitives. Néanmoins, les tables tactiles ne sont pas idéales pour la réalisation de tâches tridimensionnelles en raison du manque de degrés de liberté en entrée. L'intérêt des TEAMViz se situe donc dans la combinaison d'une table tactile avec une surface d'affichage complémentaire et un autre dispositif en entrée. Les TEAMViz peuvent contrer le manque de degrés de liberté en entrée des tables tactiles et faciliter les interactions dans les interfaces graphiques tridimensionnelles.

Les systèmes immersifs de type CAVE permettent de visualiser et d'interagir avec une scène virtuelle à une grande échelle. Les dispositifs d'entrée utilisés sont souvent basés sur des systèmes de tracking qui présentent plusieurs inconvénients dont la fatigue. En comparaison avec les systèmes de types CAVE, l'infrastructure des TEAMViz est moins coûteuse et plus flexible (la disposition de la surface verticale et de la table tactile peut être modifiée facilement). Les interactions sur la table tactile sont plus confortables que l'utilisation d'un système de tracking. De plus, les aspects collaboratifs sont favorisés en utilisant la table tactile en comparaison avec une CAVE. La conception des interactions pour les TEAMViz est différente de celles pour les CAVEs en raison de l'étude de la disposition des surfaces d'affichage et de la disposition des vues sur ces surfaces d'affichage.

Plusieurs travaux de recherche combinent une table tactile avec un dispositif d'affichage vertical. Ces travaux nous ont fortement inspirés dans l'étude de la conception des TEAMViz. Cependant, ces systèmes se différencient des TEAMViz en raison de l'im-

possibilité d'interagir directement ou par pointage avec la vue présentée verticalement ce qui est un élément essentiel dans les TEAMViz permettant la prise en charge d'un grand nombre de degrés de liberté.

Chapitre 2

Du monde physique au monde numérique

2.1 Introduction

Pour l'élaboration de systèmes TEAMViz, nous proposons de combiner plusieurs dispositifs en entrée et plusieurs vues d'un même espace virtuel projetées sur des surfaces d'affichage multiples afin de pallier le manque de dispositifs d'entrée et de techniques d'interaction adaptés à la manipulation d'objets virtuels selon 3, 6 ou 12 degrés de liberté. Dans ce type de combinaison, certains éléments sont essentiels tels que les dimensions d'affichage, les degrés de liberté, la perception, l'espace physique et les tâches. Ces notions spatiales ne sont pas prises en compte dans les modèles et les espaces de conception existants [Bouchet & Nigay, 2004, Bouchet *et al.*, 2004b, Dragicevic & Fekete, 2004, Kobayashi *et al.*, 2005]. De plus, ces modèles ne permettent pas de tenir compte des relations entre les interactions réalisées dans le monde physique et leurs effets sur le monde numérique.

L'espace de conception que nous proposons pour la combinaison de dispositifs d'entrées, de vues et de surfaces multiples se divise en trois phases :

1. la mise en relation des interactions en entrée avec les tâches à réaliser ;
2. la conception logicielle ;
3. la gestion de la concurrence.

Dans ce chapitre nous présentons tout d'abord un modèle appelé modèle physique/numérique qui nous permet de bien situer les trois grandes phases de conception. Ensuite nous décrivons le modèle appelé *mapping* pour la première phase de notre espace de conception. Ce modèle relie les interactions effectuées dans le monde physique

aux tâches ou données numériques que ces interactions modifient en tenant compte des degrés de liberté d'interaction. Après la présentation de ce modèle nous poursuivons par la présentation des contraintes et des étapes de conception de cette phase de conception. Nous concluons ce chapitre par une expérimentation comparant six modes d'interaction réalisés en appliquant le modèle *mapping*.

2.2 Modèle physique/numérique

Dans le but de simplifier les modèles d'architecture existants et de préciser les trois étapes de conception, nous définissons un modèle reliant le monde physique au monde numérique constitué de trois couches (voir figure 2.1). La couche *interaction physique* est constituée des interactions physiques qu'un utilisateur effectue sur un (des) dispositif(s) en entrée ainsi que le(s) dispositif(s) de sortie. La couche intermédiaire *interaction logique* est composée d'éléments appelés composantes logiques (voir chapitre 3) qui permettent de relier les interactions du monde physique aux données numériques du *contrôleur de dialogue* ainsi que de gérer les sorties. Le *contrôleur de dialogue*, qui est la troisième couche de ce modèle, contient les données numériques et gère la cohérence de ces données.

En comparant le modèle physique/numérique au modèle ARCH [UIMS, 1992], le modèle physique/numérique fait abstraction des couches *noyau fonctionnel* et *adaptateur du noyau fonctionnel* et les intègre dans la composante *contrôleur de dialogue*. La couche *interaction logique* se compose de composantes logiques qui contient et fait abstraction des deux couches *boite à outil d'interaction* et *présentation* du modèle ARCH. La couche *interaction physique* du modèle physique/numérique représente les interactions réalisées dans le monde physique.

La modélisation des interactions et la mise en relation des interactions réalisées dans le monde physique avec les données numériques s'effectue sur trois niveaux (voir figure 2.1). Le *mapping* est le premier niveau de conception des interactions. Il permet de relier les interactions réalisées dans le monde physique aux données numériques. Le second niveau de conception est celui des composantes logiques. La conception des composantes logiques dans la couche *interaction logique* permet la mise en relation entre le mapping et la conception logicielle. Le dernier niveau de conception (le plus bas) est celui des machines à états qui permet de gérer la concurrence des interactions en entrée.

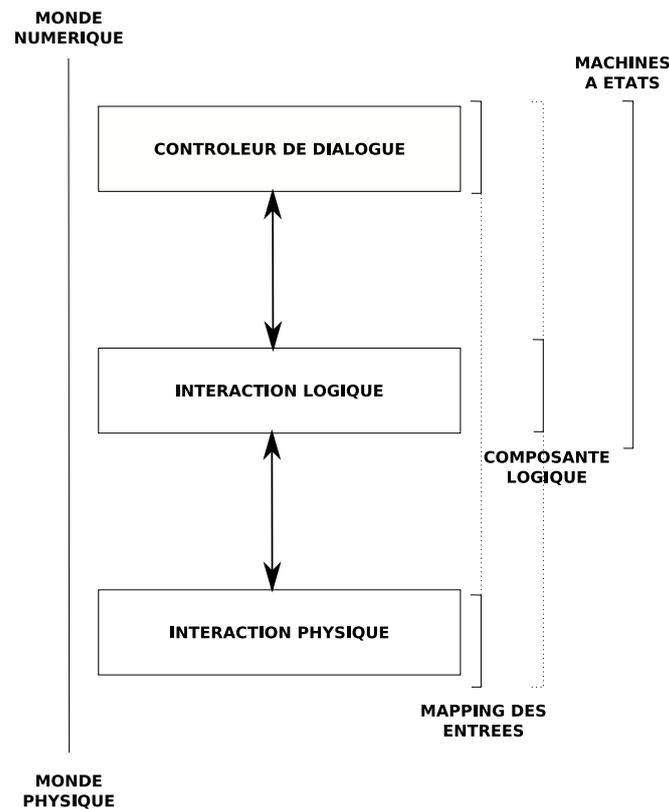


FIGURE 2.1 Modèle physique/numérique et les trois niveaux de conception des interactions : le *mapping* au plus haut niveau, les *composantes logiques* à un niveau central et les *machines à états* au plus bas niveau.

2.3 Mapping

Le modèle *mapping* est composé de trois composantes (voir figure 2.2) : les *dispositifs physiques*, la *table des techniques d'interaction* et les *tâches*. La composante *dispositifs physiques* intègre la liste des dispositifs physiques en entrée disponibles. Chaque dispositif physique contient une liste de techniques d'interaction qui lui sont propres. Ces techniques d'interaction sont représentées dans la composante *table des techniques d'interaction* et sont reliées aux dispositifs d'entrée associés. Les techniques d'interaction de la *table des techniques d'interaction* sont connectées aussi aux tâches ou données numériques sur lesquelles elles interagissent.



FIGURE 2.2 Les trois composantes du modèle *mapping*.

Les degrés de liberté sont le nombre des dimensions indépendantes du mouvement d'un corps. Un degré de liberté peut être utilisé pour décrire les mouvements d'un dispositif d'entrée, le mouvement d'un objet articulé complexe tel que le bras ou la main d'un homme et les mouvements possibles d'un objet virtuel [Bowman et al., 2004, p. 6]. En plus des translations et des rotations, nous considérons le redimensionnement selon un axe prédéfini comme un degré de liberté. Nous caractérisons les degrés de liberté selon ces trois types : translation, rotation et redimensionnement. Dans le modèle *mapping*, les degrés de liberté d'un dispositif en entrée dans le monde physique sont intégrés dans la *table des techniques d'interaction* tandis que ceux du monde numérique le sont au niveau des *tâches*.

Le modèle *mapping* permet la conception de combinaisons de dispositifs multiples à un haut niveau de la conception. Il relie plusieurs dispositifs d'entrée aux tâches à réaliser. Les éléments intéressants de ce modèle sont la mise en relation entre les degrés de liberté dans le monde physique et ceux du monde numérique. De plus ce modèle permet de visualiser l'affectation et l'équivalence des interactions par rapport aux tâches à réaliser. En effet, chaque degré de liberté d'une tâche peut être relié à une (affectation) ou plusieurs (équivalence) techniques d'interaction de la table des techniques d'interaction. Ainsi, la visualisation des complémentarités entre les dispositifs d'entrée est rendue possible.

A titre d'exemple considérons une application affichant des photos sur un écran et permettant leurs translations par manipulation directe en utilisant une souris. La translation de la souris dans le monde physique s'effectue dans le plan horizontal. Les translations des photos sont réalisées dans le plan d'affichage qui est vertical. Le *mapping* correspondant est illustré dans la figure 2.3.

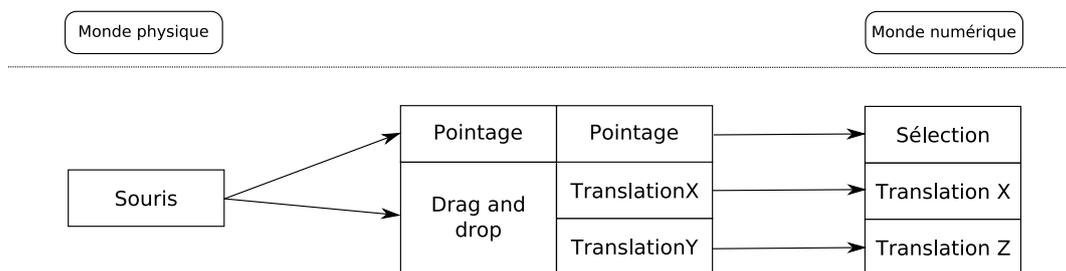


FIGURE 2.3 *Mapping* d'une application de station de travail permettant la translation des photos sur l'écran en utilisant une souris.

Nous souhaitons positionner nos travaux par rapport au modèle d'interaction instrumental [Beaudouin-Lafon, 2000] dans lequel toute technique d'interaction est décrite au moyen d'un instrument (e.g. glissière). Cet instrument possède une facette physique (intégrant le dispositif d'entrée) et une facette logique (niveau logiciel et affichage sur

l'écran). Un utilisateur agit sur un instrument qui renvoi un retour d'information à l'utilisateur et agit sur les objets d'intérêt de ce-dernier. Dans ce modèle, Beaudouin-Lafon décrit trois propriétés des instruments :

- Indirection : temporelle et spatiale. L'indirection temporelle est définie par le temps entre l'action sur l'instrument et son effet sur l'objet d'intérêt (e.g. le degré d'indirection temporelle est important lors de l'utilisation d'un bouton de validation). L'indirection spatiale est définie par la distance séparant l'instrument de l'objet d'intérêt (e.g. le degré d'indirection spatiale est petit pour un manipulateur graphique attaché à l'objet sur lequel il agit) ;
- Intégration : entre les degrés de liberté de la facette logique d'un instrument et les degrés de liberté de sa facette physique (e.g. l'utilisation de la glissière avec la souris possède un niveau d'intégrité de 1/2) ;
- Compatibilité : entre les actions physiques de l'utilisateur sur l'instrument et l'effet sur l'objet d'intérêt (e.g. le degré de compatibilité de l'utilisation d'une entrée textuelle pour définir la taille de police d'un texte est petit).

La notion d'instrument se réduit à la facette dispositif uniquement dans le cadre d'interactions directes par le touché sur les objets d'intérêts sur les tables tactiles. En raison de l'importance de ce type d'interaction sur les tables tactiles, nous séparons les dispositifs d'entrée (intégrés dans la composante *dispositifs physiques* du *mapping*) des techniques d'interaction (intégrées dans *la table des techniques d'interaction*) et n'utiliserons pas la notion d'instruments.

2.4 Conception par *mapping*

Dans notre travail, nous différencions trois types d'interaction : *interaction directe**, *interaction partiellement directe** et *interaction indirecte**. Nous définissons l'interaction directe sur un objet virtuel comme une technique d'interaction nécessitant une action de l'utilisateur selon un degré de liberté dans le monde physique (sur un dispositif physique d'entrée) transposable sur un ou des degré(s) de liberté de manipulation d'un objet virtuel. Dans ce type de technique d'interaction, l'utilisateur agit directement sur une représentation de l'objet virtuel. Des exemples de techniques d'interaction directes sont le déplacement de la souris selon deux axes impliquant le déplacement de l'objet virtuel selon deux axes dans le repère du monde virtuel. Les techniques d'interaction partiellement directes sur un objet virtuel nécessitent aussi une action de l'utilisateur selon des degrés de liberté dans le monde physique. Ces techniques d'interactions ne sont pas réalisées directement sur l'objet virtuel (*A*) mais sont réalisées sur un autre objet virtuel (*B*) (i.e. widget) qui augmente l'objet virtuel (*A*) et dont la manipulation affecte

l'objet virtuel initial (A). Un exemple d'un tel type de techniques d'interaction est le contrôle avec une souris d'un manipulateur (e.g. une glissière) rattaché à un objet virtuel permettant ainsi sa rotation autour d'un axe prédéfini. Les techniques d'interaction directes et partiellement directes que nous définissons se distinguent par le degré d'indirection spatiale défini par Beaudouin-Lafon [2000]. En opposition à ces deux types de techniques d'interaction, nous définissons les techniques d'interaction indirectes d'un objet virtuel comme des techniques d'interaction ne faisant pas appel à des manipulations selon des degrés de liberté dans le monde physique. Des exemples de tel techniques d'interaction sont l'utilisation du clavier pour la définition de la distance de translation d'un objet virtuel selon un certain axe ou l'utilisation d'un bouton virtuel pour modifier l'orientation d'un objet virtuel. Les techniques d'interaction indirectes se distinguent des deux autres types de techniques d'interaction par le degré de compatibilité défini par Beaudouin-Lafon [2000].

La combinaison de plusieurs surfaces, de plusieurs vues ainsi que de plusieurs dispositifs en entrée nous suggère de définir deux types de relations possibles entre les entrées et les sorties. Le premier type de relation est l'association d'un dispositif en entrée à une vue et/ou à un dispositif de sortie. Un exemple d'une telle association est la souris qui est associée à un écran au moyen d'un pointeur. Le second type de relation est la dissociation entre un dispositif en entrée et les sorties. Une telle relation existe lors de l'utilisation d'un dispositif en entrée (qui n'est pas un dispositif de pointage) et dont les effets ne sont pas dépendants de la disposition des vues. Un exemple de dissociation d'un dispositif en entrée et d'une vue est la combinaison d'un dispositif de type Space-Mouse avec plusieurs vues en sortie telle que la translation selon un axe du dispositif en entrée entraîne la translation d'un objet virtuel selon un axe du repère du monde virtuel dans le modèle de données et donc répercutée dans toutes les vues (voir le modèle MVC [Krasner & Pope, 1988]).

Dans cette section, nous exposons un espace de conception pour la combinaison d'interactions réalisées sur des dispositifs en entrée distincts et combinant des vues et des surfaces multiples. Cet espace de conception se base sur le modèle *mapping*. Nous illustrons tout d'abord les contraintes liées à un tel type de combinaison. Ensuite nous exposons la méthodologie de conception au moyen de ses différentes étapes et des règles à respecter pour éliminer les différentes combinaisons impossibles suivi d'un exemple d'application. Nous concluons par les choix à retenir et une méthode d'expérimentation permettant la comparaison des différentes combinaisons.

2.4.1 Contraintes

La combinaison de plusieurs dispositifs en entrée nécessite de lister et prendre en considération les contraintes de ces dispositifs. Ces contraintes portent essentiellement sur les degrés de liberté et sont :

- séparation ou non des degrés de liberté ;
- isotonicité (e.g. souris) vs isométrie (e.g. SpaceMouse ou joystick) ;
- dans le cas d'un dispositif isotonique, le rapport monde physique / monde numérique (e.g. pour une table tactile ce rapport est de 1) ;
- snapping (capture de valeurs continues mais discrétisées comme par exemple la molette de la souris) ;
- précision pour la sélection et pour la manipulation (pouvant être différentes).

De plus, la combinaison de plusieurs vues et de plusieurs surfaces d'affichage de l'information en sortie nous incite à lister et étudier les contraintes liées à ce type de combinaison lors de la phase de conception. Les contraintes des vues sont :

- dimension de présentation de l'information sur ces vues (X, Y et/ou Z) ;
- occlusion (taille et nombre des objets virtuels). La sélection d'un objet totalement masqué par d'autres est impossible. Néanmoins, le "drag and drop" supporte le masquage dans le cas d'une trajectoire ;
- résolution ;
- taille physique ;
- disposition et orientation dans le monde physique ;
- relation spatiale existant entre les vues multiples dans le monde physique ;
- attention de l'utilisateur.

Il existe aussi des contraintes liées aux entrées et aux sorties simultanément. Une des contraintes majeures est le couplage fort entre un dispositif d'entrée et un dispositif de sortie comme par exemple dans le cas des surfaces tactiles. Cette contrainte nécessite de préserver l'affichage de l'information selon les interactions effectuées par un utilisateur de façon couplée (e.g. une photo numérique déplacée sur une surface tactile avec un doigt doit être positionnée toujours au-dessous de ce doigt). Une autre contrainte d'entrée/sortie dans le cadre de combinaisons de surfaces d'affichage multiples et de dispositifs d'entrée multiples est l'association ou la dissociation d'un dispositif physique en entrée à une vue ou à une surface. Le choix de l'association ou la dissociation d'un dispositif à une vue ou à une surface d'affichage est crucial à cause des retours d'information sur des vues multiples entraînant parfois une désorientation des utilisateurs. Pour illustrer ce type de problème, considérons une application combinant deux vues perpendiculaires d'un même environnement virtuel et permettant de déplacer un objet virtuel avec un dispositif SpaceMouse associé à l'une des vues. Les translations effectuées phy-

siquement avec le SpaceMouse correspondent aux translations de l'objet virtuel sur la vue associée mais sont totalement différentes sur la seconde vue en raison d'une rotation de 90 degrés entre les vues.

La dernière contrainte est liée au recouvrement des fonctionnalités. Les fonctionnalités doivent être définies au préalable par le concepteur, et le système final doit permettre de réaliser toutes les fonctionnalités au moyen d'au moins une technique d'interaction.

2.4.2 Méthodologie de conception

Suite à la définition des contraintes, nous présentons une méthodologie de conception en utilisant le modèle *mapping* pour la combinaison d'interactions sur des dispositifs en entrée, des vues multiples et des surfaces d'affichage multiples. Cette méthodologie de conception se décompose en cinq étapes (voir figure 2.4) : *définition des tâches*, *choix des vues*, *choix des dispositifs d'entrée*, *définition de la table des techniques d'interaction* et *la génération et filtrage de la combinatoire**. Nous illustrons ces différentes étapes par un exemple générique de conception de la combinaison d'interactions effectuées avec deux dispositifs d'entrée sur deux vues disposées perpendiculairement dans le monde physique. Les tâches souhaitées sont la sélection et la translation 3D d'un objet virtuel.



FIGURE 2.4 La conception par *mapping* en cinq étapes.

Définition des tâches

La première étape de la conception par *mapping* consiste à définir les tâches et donc la composante *tâche* du *mapping*. Les tâches sont caractérisées selon les degrés de liberté de manipulation ainsi que la sélection d'un objet virtuel lorsque plusieurs objets virtuels peuvent être désignés. La figure 2.5.a illustre l'état du *mapping* à ce stade de la conception.

Choix des vues

Après la définition des tâches, un choix sur le nombre de vues, leurs dispositions selon les surfaces d'affichage, le type de vue (subjectif ou objectif) et les dimensions d'affichage doit être réalisé. En s'inspirant des travaux de Baldonado *et al.* [2000] et de Roberts [1998], cette sélection doit tenir compte des critères suivants :

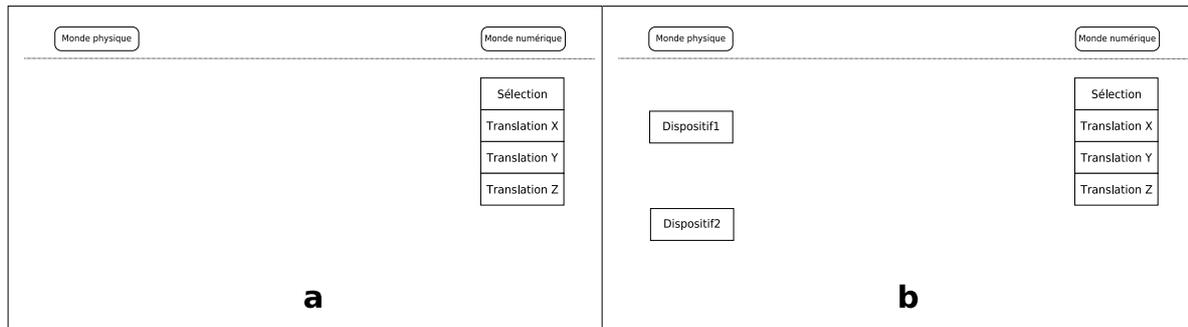


FIGURE 2.5 Etapes initiales de la conception par *mapping* : a- Etape 1 de la conception (définition des tâches). b- Etape 3 de la conception (choix des dispositifs d'entrée).

Critère de vues 1 Un nombre élevé de vues implique une plus grande complexité (pour l'utilisateur) de compréhension de la structure du monde virtuel mais permet d'éviter le problème d'occlusion lorsque les vues sont complémentaires.

Critère de vues 2 La présentation d'une vue globale de la scène virtuelle facilite la représentation mentale du monde numérique par les utilisateurs.

Critère de vues 3 La présentation de dimensions différentes selon les vues accroît la complémentarité et la diversité des vues.

Critère de vues 4 La préservation des dimensions entre le monde physique et le monde numérique au moyen des vues facilite la compréhension de la structure du monde numérique par les utilisateurs. Ce critère se traduit par le choix des dimensions d'une vue selon sa disposition dans le monde physique. Par exemple la dimension verticale d'une vue présentée sur un écran vertical correspond à la dimension verticale dans le monde physique. Cette préservation est généralement respectée dans les systèmes de réalité virtuelle tel que les CAVEs [Cruz-Neira *et al.*, 1992].

Critère de vues 5 La réduction des transformations numériques entre les différentes vues d'une même scène virtuelle facilite la compréhension de la relation entre ces vues. Les translations et les facteurs d'échelle sont tolérables mais les rotations mentales sont des transformations nécessitant un effort cognitif important de la part de l'utilisateur.

Choix des dispositifs d'entrée

Le choix des dispositifs d'entrée doit être motivé par leur complémentarité, le nombre de dispositifs, la facilité et l'intuitivité de l'interaction. Deux dispositifs d'entrée peuvent être complémentaires selon plusieurs critères tels que les degrés de liberté, la rapidité

et la précision. Par exemple, un dispositif permettant de réaliser des interactions rapides mais non précises est complémentaire à un dispositif permettant de réaliser des interactions précises mais lentes. Le nombre de dispositifs d'entrée doit être réduit pour minimiser les passages d'un dispositif à l'autre. Pour la facilité des interactions, il est souhaitable de réduire le nombre de degrés de liberté lorsque c'est possible [Bowman *et al.*, 2004, p. 180-181]. Le choix des dispositifs d'entrée définit la composante dispositif physique du *mapping* (voir figure 2.5.b).

Définition de la table des techniques d'interaction

La quatrième étape de la conception consiste à réaliser une première définition de la composante *table des techniques d'interaction* du *mapping*. Chaque dispositif d'entrée fournissant des données continues possède des caractéristiques d'*interaction directe*. La *table des techniques d'interaction* est donc définie par ces types d'interaction (e.g. déplacement de la souris par translation X et translation Y dans le monde physique). Chaque dispositif physique de la composante *dispositifs physiques* du *mapping* est relié directement aux techniques d'interaction correspondantes. La figure 2.6.a illustre l'état du *mapping* à ce stade de la conception.

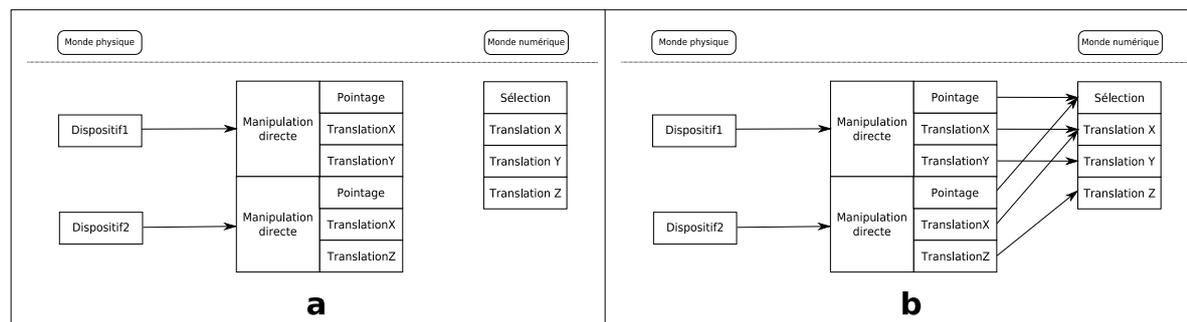


FIGURE 2.6 Etapes finales de la conception par *mapping* : a- Etape 4 de la conception (définition de la *table des techniques d'interaction*). b- Etat final du *mapping* de l'une des possibilités de la combinatoire.

Génération et filtrage de la combinatoire

Les trois composantes du *mapping* étant définies, elles produisent un grand nombre de possibilités de liaisons entre la composante *table des techniques d'interaction* et la composante *tâche*. La cinquième étape de la conception par *mapping* consiste à filtrer les associations indésirables de la combinatoire des correspondances entre les techniques d'interaction et les tâches. L'étude de la combinatoire et l'élimination des possibilités s'effectuent selon les règles suivantes qui sont présentées par ordre d'importance décroissante :

Mapping1 Respecter les habitudes des utilisateurs lorsqu'elles sont exprimées. Cette règle dépend de la relation entre un dispositif en entrée et une vue, et le cas échéant, de la disposition de la vue sur le monde numérique.

Mapping2 Respecter la correspondance entre le type de degré de liberté dans le monde physique et dans le monde numérique. Ainsi les rotations réalisées dans le monde physique sur un dispositif en entrée doivent correspondre en priorité à des rotations dans le monde numérique. Cette règle préserve l'intuitivité des interactions.

Mapping3 Réduire les transformations entre les degrés de liberté d'un dispositif d'entrée dans le monde physique et les degrés de liberté de l'objet virtuel affiché dans la vue associée au dispositif d'entrée. Cette règle s'applique essentiellement aux rotations qui demandent un effort important à l'utilisateur pour réaliser la correspondance entre les degrés de liberté du monde physique et ceux du monde numérique. La translation et le facteur d'échelle (s'il est identique selon tous les axes) sont acceptables. Le respect idéal de cette règle mène à réaliser une correspondance entre les degrés de liberté du dispositif d'entrée dans le monde physique et ceux du monde numérique selon la vue associée.

Mapping4 Assigner une technique d'interaction (de la composante *table des techniques d'interaction*) à une seule tâche (de la composante *tâche*).

Mapping5 Respecter le recouvrement des tâches. Pour chaque tâche de la composante *tâche* doit correspondre au moins une technique d'interaction de la composante *table des techniques d'interaction*. Lorsque la *table des techniques d'interaction* ne permet pas le recouvrement de toutes les tâches, de nouvelles techniques d'interaction *partiellement directes* et/ou *indirectes* (e.g. utilisation d'une glissière ou de boutons graphiques) doivent être ajoutées selon les règles listées ci-dessous.

La règle Mapping3 peut être en opposition avec les habitudes des utilisateurs (règle Mapping1). Dans le cas de conflit, préférer le respect des habitudes des utilisateurs ou réaliser des expériences de comparaison en tenant compte de l'efficacité, de l'intuitivité et du facteur d'apprentissage.

Les règles d'ajout de techniques d'interaction partiellement directes ou indirectes sont :

Règle d'ajout de TI 1 Préférer l'ajout d'une technique d'interaction partiellement directe à une technique d'interaction indirecte sauf dans le cas d'une surpopulation d'objets virtuels où l'ajout d'objets visuels augmente la complexité de la visualisation et l'occlusion. Le but de cette règle est de renforcer l'intuitivité du système.

Règle d'ajout de TI 2 Pour un dispositif en entrée, ajouter des techniques d'interaction partiellement indirectes sur la vue associée (si elle existe). Les effets des degrés de liberté en entrée pour réaliser la tâche doivent être visibles sur les dimensions en sortie qui sont représentées dans cette vue. Cette règle permet de renforcer le couplage dispositif d'entrée/dispositif de sortie.

Règle d'ajout de TI 3 Préférer des techniques d'interaction partiellement directes dont les degrés de liberté dans le monde physique correspondent aux degrés de liberté nécessaires pour l'accomplissement de la tâche dans le monde numérique (e.g. rotation horizontale dans le monde physique correspond à une rotation dans le monde virtuel).

Certaines de ces règles sont déduites de manières informelles par expérience personnelle et par discussions avec des utilisateurs tandis que d'autres sont basées sur la littérature. Elles permettent d'éliminer rapidement lors de la conception certaines combinaisons qui ne les respectent pas. Il reste cependant généralement plusieurs solutions possibles après cette phase. La figure 2.6.b représente l'état final du *mapping* de l'une des combinaisons possibles de notre exemple.

2.4.3 Spécification d'un prototype d'interfaces selon la méthodologie de conception

Nous spécifions un prototype d'une interface en développant les différentes étapes de conception pour illustrer la méthodologie de conception par *mapping*. Les conditions initiales de la conception de cette interface sont la connaissance des dispositifs d'entrée et de sortie ainsi que des fonctionnalités souhaitées. Les dispositifs d'entrée disponibles sont une table tactile et une souris tandis que les dispositifs de sortie sont deux vidéo-projecteurs. Les fonctionnalités souhaitées de cette interface sont la sélection, la translation en 3D, le redimensionnement en 3D et la rotation horizontale d'objets virtuels 3D. Nous appelons cette interface $TT + S$ à cause de la combinaison d'une table tactile et d'une souris en entrée.

La première étape de la conception par *mapping* est la définition des tâches. Dans $TT + S$, les tâches souhaitées sont la sélection d'un objet virtuel, sa translation et son redimensionnement selon les trois dimensions ainsi que sa rotation selon l'axe vertical. Ainsi les tâches de $TT + S$ sont représentées dans la figure 2.7.a.

La seconde étape de la conception par *mapping* est le choix des vues. L'utilisation d'une table tactile comme dispositif d'entrée dans $TT + S$ incite à utiliser une vue horizontale projetée sur cette table. Nous appelons cette vue la vue (A). Nous proposons

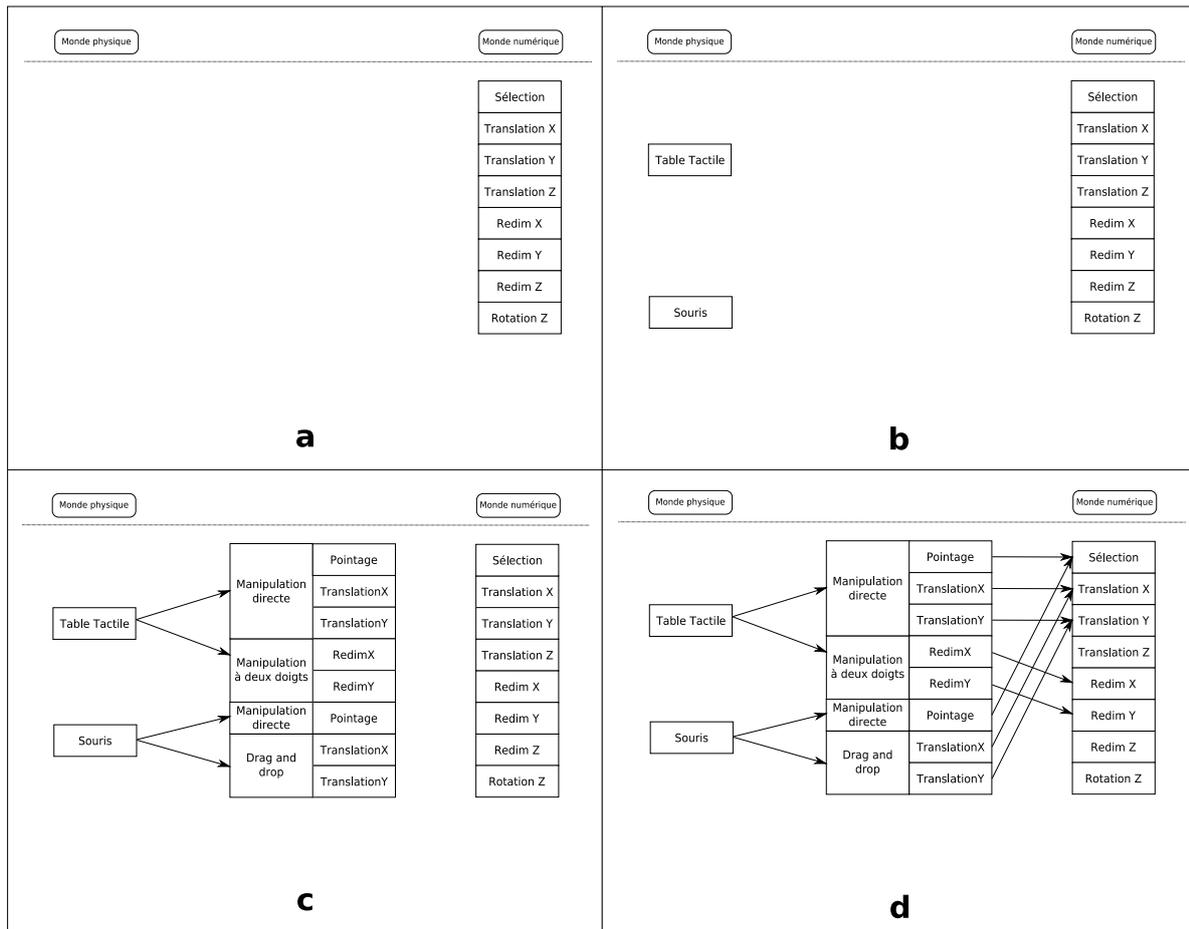


FIGURE 2.7 Exemple de conception par *mapping* : a- Etape 1 de la conception (définition des tâches). b- Etape 3 de la conception (choix des dispositifs d'entrée). c- Etape 4 de la conception (définition de la *table des techniques d'interaction*). d- Une des possibilités de combinatoire ne respectant pas le Mapping1.

l'utilisation d'une vue verticale complémentaire appelée vue (*B*). Le choix de deux vues soutient le premier critère (Critère de vues 1) du choix du nombre de vues qui est en équilibre entre la complexité et la gestion de l'occlusion. De plus, pour respecter le Critère de vues 2, nous choisissons que la vue (*A*) représente une vue globale de l'environnement virtuel. Etudions à présent la disposition des représentations du monde virtuelle dans ces vues. Si nous affichons une représentation du monde virtuel selon les mêmes dimensions sur les deux vues, le Critère de vues 3 n'est pas respecté. La disposition des deux vues orientées physiquement différemment mais virtuellement selon les mêmes dimensions n'est pas souhaitable pour l'utilisateur. En effet, dans ce cas la redondance est importante entre les deux vues et la complémentarité est réduite. Cette disposition des dimensions virtuelles est plus appropriée dans le cas de deux vues physiquement orientées identiquement. Une seconde possibilité est d'orienter la représentation du monde virtuel verticalement sur la vue (*A*) et horizontalement sur la

vue (B). Ce type de disposition des vues est en contradiction avec le Critère de vues 4. Le choix le plus approprié est l'orientation de la représentation du monde virtuel horizontalement sur la vue (A) et verticalement sur la vue (B). Les transformations numériques entre la vue (A) et la vue (B) sont un facteur d'échelle, des translations et possiblement une rotation horizontale ce qui respecte partiellement le cinquième critère.

Les dispositifs d'entrée disponibles sont la table tactile et la souris dans $TT + S$. Il faut choisir quelles sont les associations de chaque dispositif à une vue qui à mettre en place. Un système de pointage doit nécessairement être associé à au moins une vue. La table tactile semble plus appropriée à la vue horizontale. Pour cette raison nous choisissons l'association de la table tactile à la vue horizontale et la souris à l'autre vue dans le but d'accroître la complémentarité des dispositifs. Le choix des dispositifs d'entrée est représenté dans la figure 2.7.b.

La quatrième étape de la conception par *mapping* est la définition de la *table des techniques d'interaction*. Les techniques d'interaction qui sont intégrées dans la table des techniques d'interaction à ce stade de la conception sont les techniques d'interaction directes des dispositifs d'entrée (définies à la deuxième étape de la conception par *mapping*). La sélection et la translation selon les axes X et Y avec la table tactile s'effectuent par manipulation directe. De plus, la technique d'interaction à deux doigts permettant un redimensionnement d'un objet virtuel sur la table tactile est aussi une *technique d'interaction directe*. La souris est associée à une translation en X et une translation en Y dans le monde physique. Nous établissons donc la *table des techniques d'interaction* et ses connexions à la composante *dispositifs d'entrée* du *mapping* illustrée dans la figure 2.7.c.

L'étude de la combinatoire des relations connectant les techniques d'interaction aux tâches élémentaires est la dernière étape de la conception par *mapping*. Les techniques d'interaction de sélection de la table tactile et de la souris doivent être reliées à la tâche sélection de la composante *tâche* du *mapping*. Pour respecter les règles Mapping1, Mapping2 et Mapping3 nous affectons la translation selon l'axe X (respectivement l'axe Y) dans le monde physique en utilisant la table tactile à la translation selon l'axe X (respectivement l'axe Y) de la composante *tâche*. De même, nous affectons les redimensionnements selon les axes X et Y de la *table des techniques d'interaction* aux redimensionnements selon les mêmes axes de la composante *tâche*. Pour les translations de la souris, nous connectons la translation selon l'axe X (respectivement l'axe Y) à la translation selon l'axe X (respectivement l'axe Y) de la composante *tâche* du *mapping* (voir figure 2.7.d). Ce choix respecte les règles Mapping2 et Mapping3 mais pas le Mapping1 en raison de l'habitude des utilisateurs de réaliser des translations avec la souris

dans le plan de projection de la vue correspondante. Ainsi, la translation selon l'axe Y de la souris est reliée à la translation selon l'axe Z de la composante *tâche* (voir figure 2.8.a).

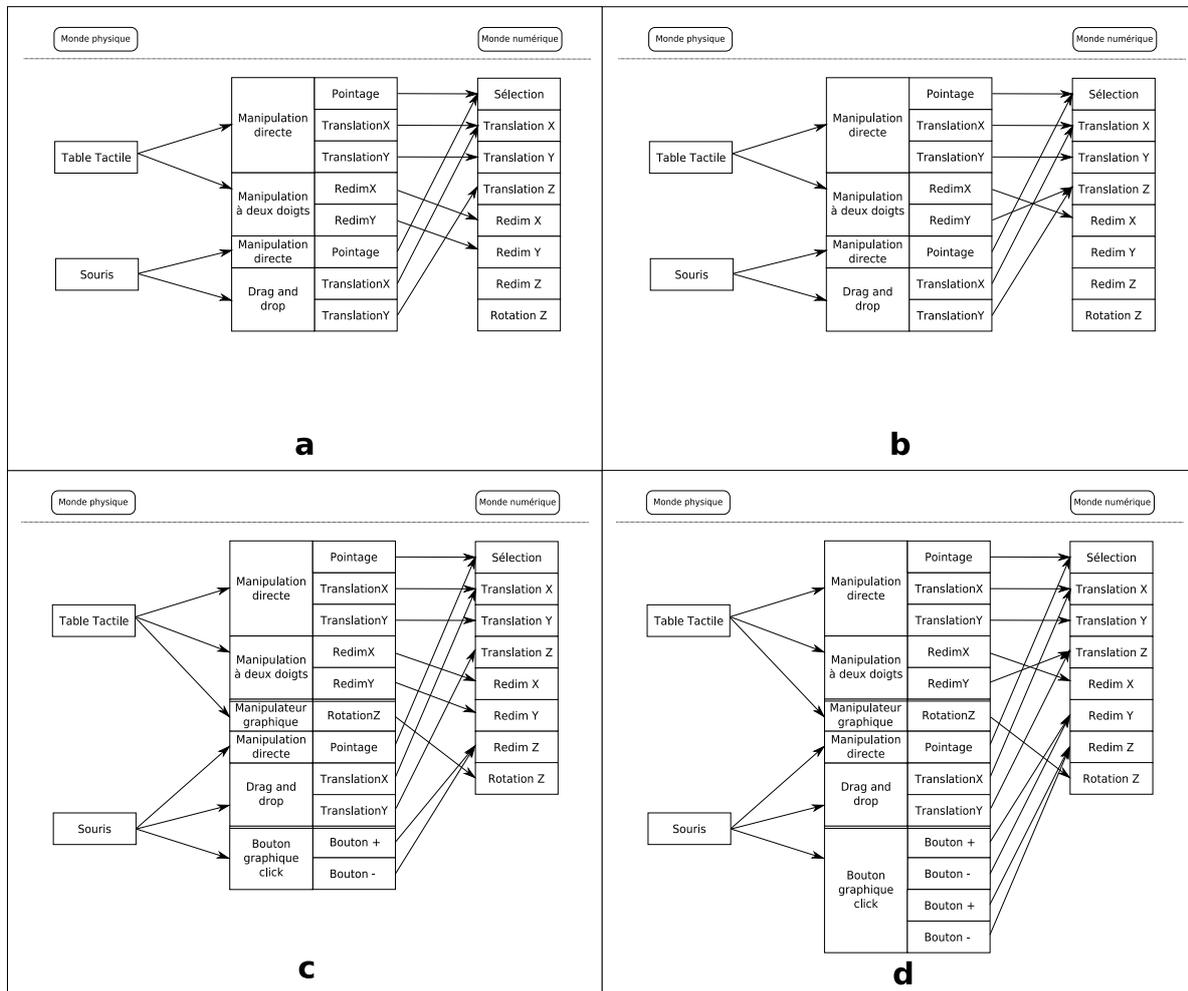


FIGURE 2.8 Combinaisons possibles de l'exemple de conception par *mapping* : a- Une possibilité de combinatoire respectant le Mapping1, Mapping2, Mapping4 et partiellement le Mapping3. b- Une possibilité de combinatoire respectant le Mapping1, Mapping4 et partiellement le Mapping2 et Mapping3. c- Solution a avec un ajout de techniques d'interaction pour respecter le Mapping5. d- Solution b avec un ajout de techniques d'interaction pour respecter le Mapping5.

Nous pouvons imaginer que la manipulation à deux doigts sur une surface tactile soit associée à la translation selon l'axe perpendiculaire à la surface simulant le déplacement en profondeur. Ainsi, le redimensionnement avec deux doigts selon l'axe Y peut être associé à la translation selon l'axe Z (voir figure 2.8.b). Ce choix respecte la règle Mapping1 mais pas les règles Mapping2 et Mapping3.

Les résultats de la conception par *mapping* des figures 2.8.a et 2.8.b respectent la règle Mapping4. Néanmoins, la règle Mapping5 n'est pas respectée dans la combinaison

de la figure 2.8.a (respectivement 2.8.b) en raison du non recouvrement des tâches de redimensionnement et de rotation selon l'axe Z (respectivement le redimensionnement selon l'axe Y et Z et la rotation selon l'axe Z). Un des choix possibles pour respecter le recouvrement des tâches de la combinaison de la figure 2.8.a (respectivement 2.8.b) est représenté dans la figure 2.8.c (respectivement 2.8.d). Le manipulateur graphique permettant à l'utilisateur de réaliser une rotation de l'objet virtuel avec une action de rotation dans le monde physique sur la table tactile est une technique d'interaction partiellement directe. Cette technique d'interaction respecte les trois règles d'ajout de techniques d'interaction. En opposition, les boutons graphiques permettant le redimensionnement selon l'axe Z (et selon l'axe Y) d'un objet virtuel en cliquant sur ceux-ci sont des techniques d'interaction indirectes. Ces techniques d'interaction ne respectent aucune des règles d'ajout de techniques d'interaction.

Les deux combinaisons des figures 2.8.c et 2.8.d respectent le recouvrement des tâches. D'autres techniques d'interaction peuvent être utilisées pour satisfaire le recouvrement des tâches impliquant d'autres *solutions**. Néanmoins ces deux combinaisons sont possibles et l'une des différences entre elles est le niveau de respect des règles de combinatoire. En effet, la combinaison de la figure 2.8.c respecte toutes les règles sauf la règle Mapping3 qui est satisfaite partiellement. La combinaison de la figure 2.8.d respecte totalement les règles Mapping1, Mapping4 et Mapping5 et partiellement les règles Mapping2 et Mapping3.

2.4.4 Choix retenus et comparaison

Les solutions de la conception par *mapping* peuvent être multiples. Un choix doit être donc réalisé selon les critères suivants :

- efficacité (temps, précision, taux d'erreur) ;
- compromis entre utilisabilité et apprentissage (intuitivité, préférence utilisateur, etc.).

Pour établir ce choix, une évaluation de comparaison entre les différentes solutions doit être effectuée. L'évaluation doit porter sur l'ensemble des tâches définies par le concepteur d'une façon globale (toutes les tâches combinées dans une même évaluation) ou d'une façon individuelle. Dans le but de respecter les deux critères précédents, nous suggérons que l'évaluation se déroule en deux étapes. Dans une première étape, un ou plusieurs utilisateurs réalisent l'expérience en enregistrant le temps, la précision et les différents stades de déroulement de l'expérience (e.g. position de l'objet virtuel, son orientation, etc.). Cette étape permet d'analyser et de comparer l'efficacité des différents choix de conception d'une façon objective (premier critère). La seconde étape consiste

à remplir un questionnaire dans lequel les utilisateurs expriment d'une façon subjective l'intuitivité des interactions ainsi que leurs préférences. Cette seconde étape permet de faciliter la réponse au second critère.

Après l'analyse et la comparaison des différents choix de conception, le concepteur pourra effectuer des modifications à deux niveaux de la conception. Les modifications en aval s'effectuent en sélectionnant les combinaisons les plus appropriées tout en gardant les dispositifs d'entrée et de sortie ainsi que la disposition des vues. Ces modifications interviennent sur les étapes 4 et/ou 5 de la conception par *mapping*. Le changement de la disposition des vues, des dispositifs d'entrée et/ou de sortie nécessitent des modifications dans les étapes initiales de la conception et donc des modifications en amont. Les modifications en amont génèrent une nouvelle combinatoire de solutions nécessitant de nouveaux filtrages à effectuer en aval.

2.5 Expérimentation

L'exemple $TT + S$ présenté dans la section 2.4.3 est assez générique et mérite d'aller jusqu'à la phase d'évaluation. Ainsi, nous développons dans cette section une expérimentation en nous basant sur la conception par *mapping*. Le but de cette expérimentation est de combiner plusieurs dispositifs en entrée pour une tâche dite de "docking" (soit en français amarrage ou accostage) et de réaliser une comparaison entre les différentes solutions. Nous présentons la conception par *mapping* de l'expérimentation puis les conditions de l'expérimentation. Nous concluons cette section par l'analyse des résultats.

2.5.1 Conception de l'expérimentation

Nous concevons une expérience en nous basant sur la méthodologie de conception par *mapping*. L'expérimentation doit permettre de comparer plusieurs combinaisons de dispositifs en entrée ainsi que plusieurs combinaisons de techniques d'interaction. Cette comparaison s'effectue avec les mêmes conditions de choix de tâches et de vues.

La tâche dite de "docking" de l'expérience consiste à déplacer et à redimensionner un objet virtuel de façon à ce qu'il soit bien encastré dans un autre objet virtuel. La tâche se décompose en sept tâches élémentaires : la sélection, les trois translations et les trois redimensionnements (selon les trois dimensions). La composante *tâche* du *mapping* est illustrée dans la figure 2.9.

Le choix du nombre de vues, de leurs dispositions dans le monde numérique et dans le monde physique est celui de l'exemple de conception par *mapping* $TT + S$ présenté

Sélection
Translation X
Translation Y
Translation Z
Redim X
Redim Y
Redim Z

FIGURE 2.9 La composante *tâche* de la conception par *mapping* de l'expérimentation.

dans la section 2.4.3. Nous choisissons donc de combiner une vue 2D représentant les dimensions horizontales de l'environnement virtuel avec une vue 3D. La vue 2D est une vue globale de la scène virtuelle qui est présentée sur une surface horizontale dans le monde physique. La vue 3D est affichée sur une surface verticale (écran ou projection) et les dimensions numériques correspondent aux dimensions dans le monde physique. Les deux vues tentent de respecter au mieux les critères de choix et de combinaison des vues.

Pour les dispositifs en entrée, nous choisissons la souris, une table tactile et un dispositif à six degrés de liberté. La souris est retenue pour son utilisation répandue et donc l'absence de temps d'apprentissage par les participants de l'expérimentation. La table tactile est choisie pour ses interactions intuitives qui évitent également une période d'apprentissage. Pour le dernier dispositif, nous avons choisi le SpaceTraveler (voir figure 2.10) qui couvre la totalité des degrés de liberté nécessaires pour réaliser la tâche. C'est un dispositif d'entrée à six degrés de liberté de la famille des dispositifs de type SpaceMouse produit par 3Dconnexion [3Dconnexion].



FIGURE 2.10 Dispositif SpaceTraveler. Source : [3Dconnexion].

Nous restreignons le nombre de dispositifs d'entrée à deux au maximum pour un ensemble de composantes donné afin de limiter les temps de passage d'un dispositif à un autre. Nous associons la table tactile à la vue 2D à cause de la disposition horizontale

de la vue 2D dans le monde physique. La souris et le SpaceTraveler sont associés à la vue 3D présentée verticalement. Le choix de l'association de la souris à la vue 3D est justifié par la présentation verticale de cette vue dans le monde physique ce qui facilite la perception du curseur et respecte l'habitude des utilisateurs. Le SpaceTraveler est associé à la vue 3D en raison de l'affichage de toutes les dimensions nécessaires dans cette vue pour le retour d'information sur les interactions en entrée.

Six possibilités de combinaison de ces trois dispositifs en entrée existent en se limitant à un maximum de deux dispositifs par combinaison. Nous choisissons d'éliminer la possibilité de combinaison du SpaceTraveler et de la souris à cause de leurs associations à la même vue bien que cette combinaison soit possible avec notre modèle et notre méthodologie de conception. Nous étudions donc chaque dispositif séparément, la combinaison de la souris avec la table tactile et la combinaison du SpaceTraveler avec la table tactile. Les cinq Ensembles de Dispositifs d'Entrée (EDE^*) sont :

- EDE T : table tactile uniquement ;
- EDE S : souris uniquement ;
- EDE St : dispositif SpaceTraveler uniquement ;
- EDE TS : combinaison de la table tactile avec la souris ;
- EDE TSt : combinaison de la table tactile avec le SpaceTraveler.

Pour chaque dispositif d'entrée nous définissons la *table des techniques d'interaction*. Pour la table tactile, les techniques d'interaction directes sont la manipulation directe pour le pointage, la translation selon les dimensions physiques horizontales et l'utilisation de deux doigts pour réaliser des redimensionnements selon les dimensions horizontales. La figure 2.11.a montre la *table des techniques d'interaction* de la table tactile. Pour la souris, la manipulation directe pour le pointage, les déplacements dans le plan physique horizontal et l'utilisation de la molette forment des *techniques d'interaction directes* (voir figure 2.11.b). Les six degrés de liberté du SpaceTraveler forment des techniques d'interaction directes ainsi que le rajout d'un pointeur pour la sélection. La figure 2.11.c illustre la *table des techniques d'interaction* obtenues.

L'étape suivante de la conception par *mapping* est l'étude de la combinatoire du *mapping* entre la *table des techniques d'interaction* et les *tâches*. Cette étude doit être réalisée pour chacune des cinq possibilités de combinaison des trois dispositifs d'entrée.

Pour l'EDE T , la figure 2.12 présente une première tentative de mise en relation entre la *table des techniques d'interaction* et la composante *tâche* du *mapping*. Cette combinaison ne respecte pas la règle de recouvrement des tâches (Mapping5) en raison de l'absence de connexions reliant la translation et le redimensionnement selon l'axe Z de la composante *tâche* à des techniques d'interaction. Pour résoudre ce problème nous choisissons d'ajouter deux glissières sur la vue 2D de la table tactile pour réaliser des

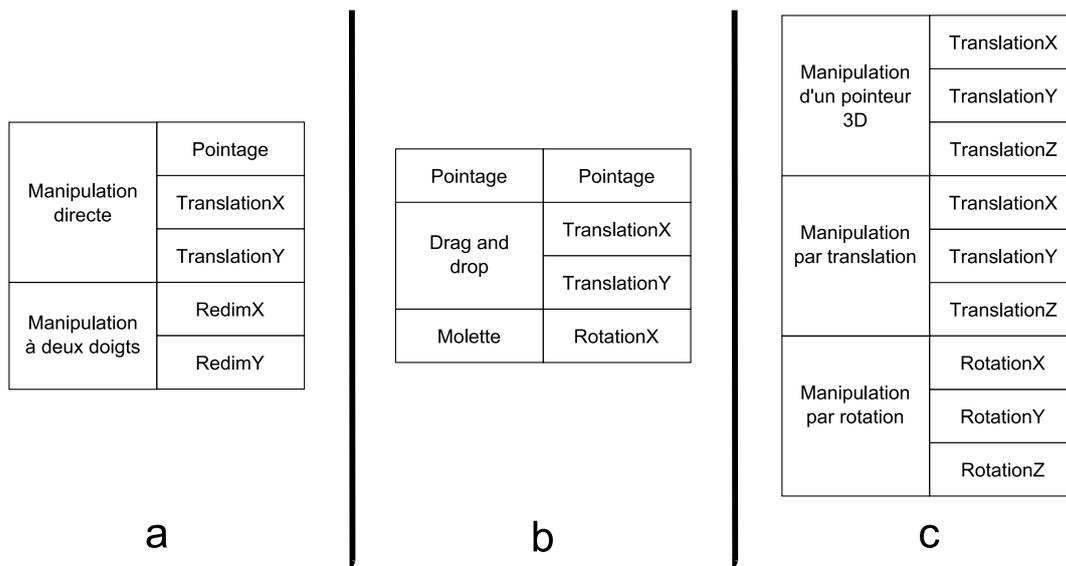


FIGURE 2.11 a. Table des techniques d'interaction de la table tactile. b. Table des techniques d'interaction de la souris. c. Table des techniques d'interaction du SpaceTraveler.

interactions partiellement directes. Cet ajout ne respecte pas la règle (Règle d'ajout de TI 2) sur l'ajout d'une interaction partiellement directe pour un dispositif d'entrée et dont l'effet est visible sur la vue associée. Le seul retour d'information des interactions effectuées sur les glissières par la vue 2D est leurs niveaux. La figure 2.13 illustre une solution de la conception de l'EDE T appelée $SEDE^* T$.

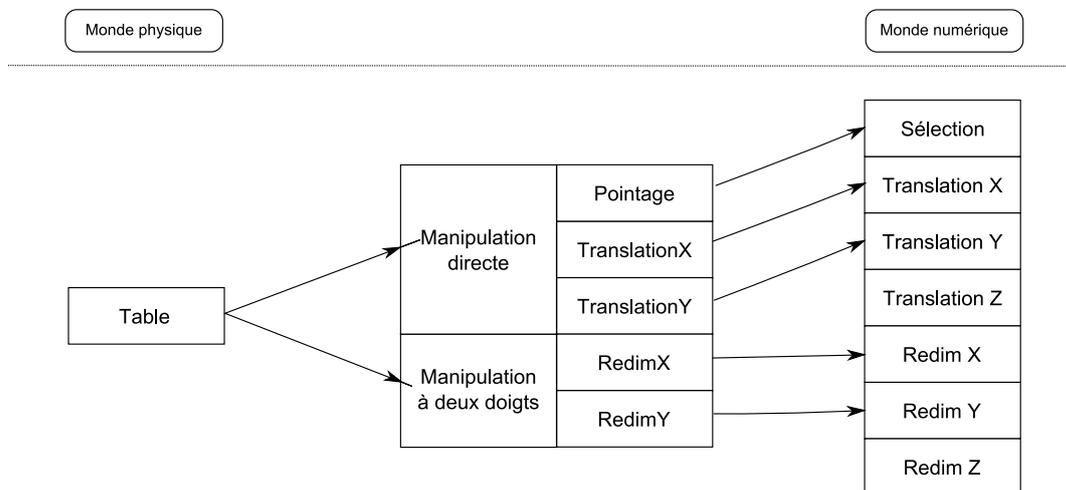


FIGURE 2.12 Mapping initial de l'EDE T (table uniquement).

La figure 2.14 présente une première mise en relation entre la *table des techniques d'interaction* et la composante *tâche* du *mapping* pour l'EDE S . Cette combinaison ne respecte pas la règle de recouvrement des tâches (Mapping5) pour les tâches de redimensionnement. Nous ajoutons des boutons graphiques pour le redimensionnement selon les axes X et Z et l'utilisation d'un click et la molette de la souris pour le redimen-

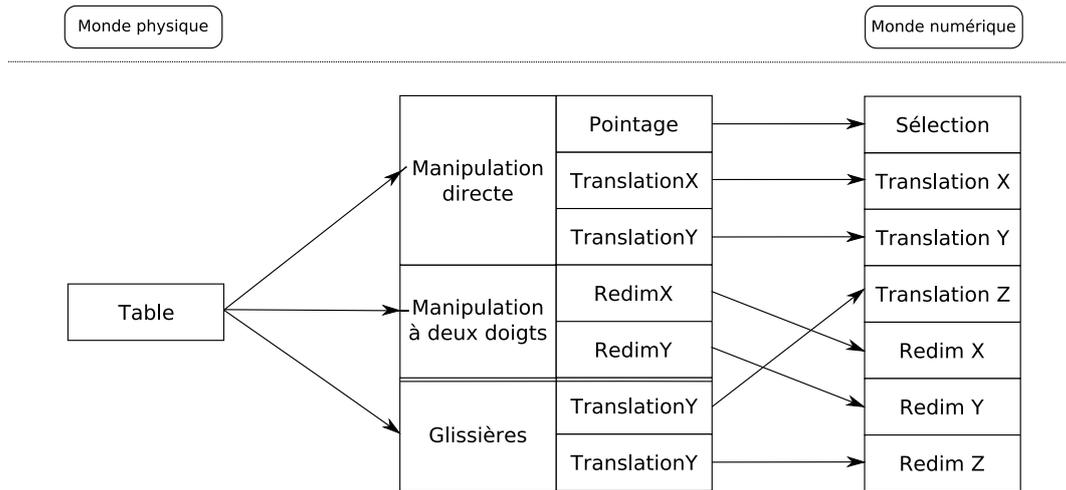


FIGURE 2.13 Solution du *mapping* de l'EDE T (table tactile uniquement).

sionnement selon l'axe Y. Les boutons graphiques forment une technique d'interaction indirecte. Il est préférable de choisir des manipulateurs graphiques dont la position est modifiée pour le redimensionnement d'objets virtuels afin de respecter la règle de préférence d'ajout de techniques partiellement directes aux techniques indirectes (Règle d'ajout de TI 1). Puisque nous souhaitons tester si l'utilisation d'un bouton graphique est pénalisante, nous choisissons d'enfreindre cette règle. La figure 2.15 montre une solution de la conception par *mapping* de l'EDE S que nous appelons SEDE S .

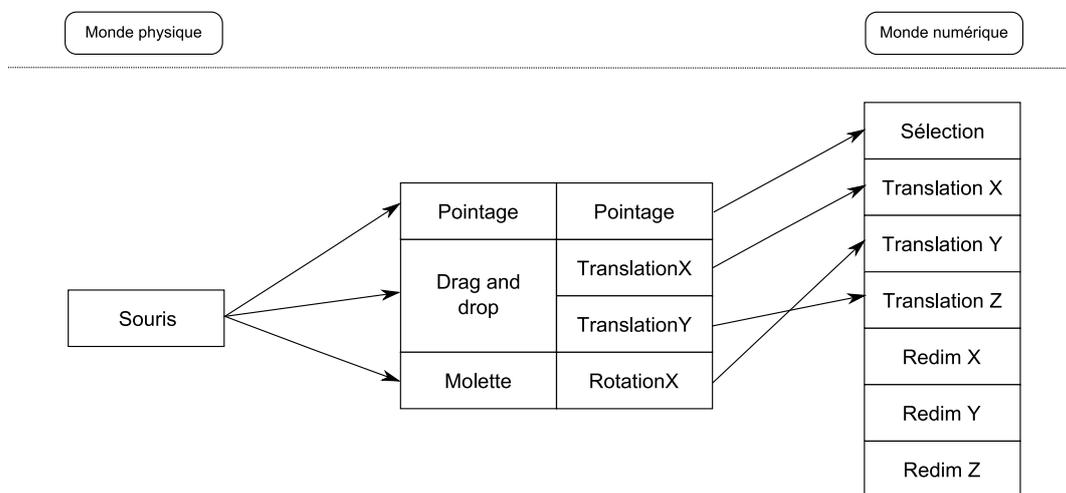


FIGURE 2.14 *Mapping* initial de l'EDE S (souris uniquement).

Pour l'EDE St , une solution respectant au mieux les règles de combinatoire est présentée dans la figure 2.16. Cette solution, appelée SEDE St , ne respecte pas la règle de préservation des types de degrés de liberté entre le monde physique et le monde numérique (Mapping2) à cause de l'utilisation de rotations physiques pour les redimensionnements numériques. En revanche, la règle de recouvrement des tâches (Mapping5)

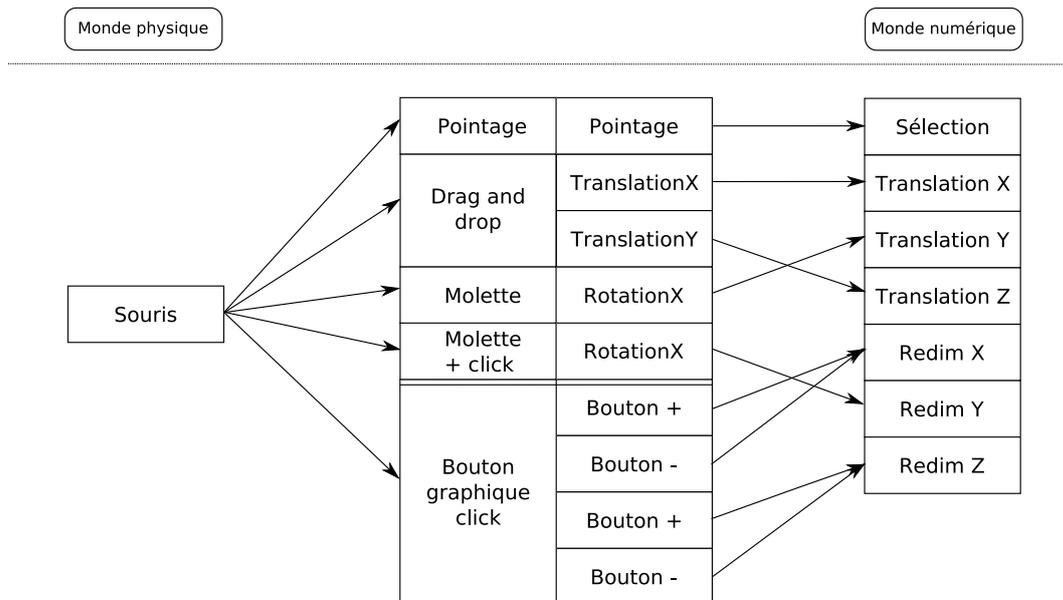


FIGURE 2.15 Solution du *mapping* de l'EDE S (souris uniquement).

est satisfaite.

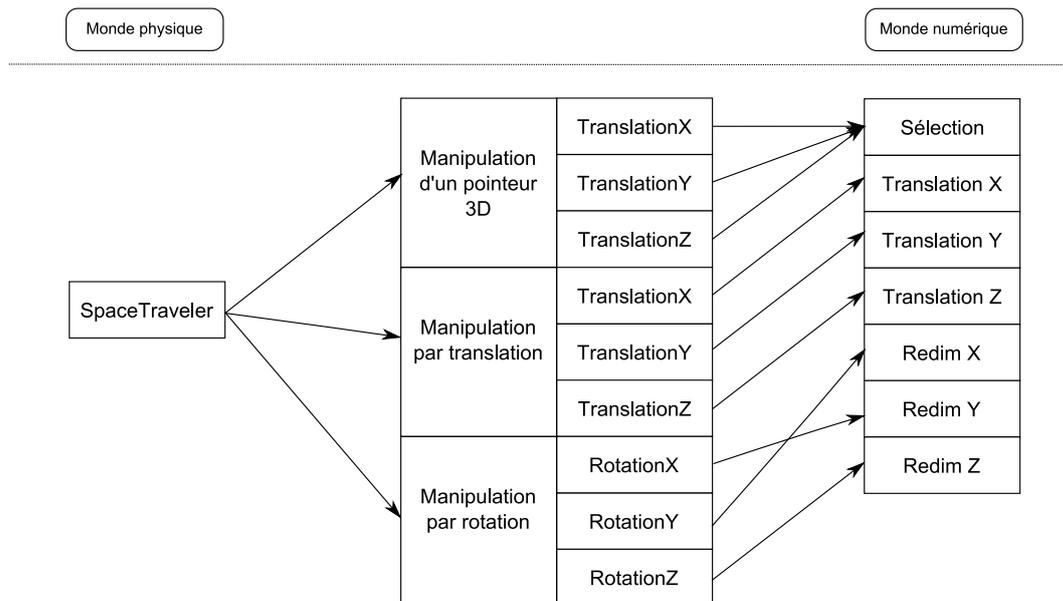


FIGURE 2.16 Solution du *mapping* de l'EDE St (SpaceTraveler uniquement).

L'utilisation de la molette pour la réalisation des translations ne respecte pas la règle de préservation des types de degrés de liberté entre le monde physique et le monde numérique (Mapping2). Etant donné que pour la combinaison de la table tactile avec la souris les tâches de translation et de redimensionnement selon l'axe Y sont réalisables au moyen de la table tactile, l'utilisation de la molette est abandonnée. La combinaison des techniques d'interaction directes de la table tactile et de celles de la souris ne

respectent pas la règle de recouvrement (i.e. pour le redimensionnement selon l'axe Z). Nous ajoutons des boutons graphiques comme pour l'EDE S . Nous obtenons donc la solution de la conception appelée SEDE TS et illustrée dans la figure 2.17 pour l'EDE TS .

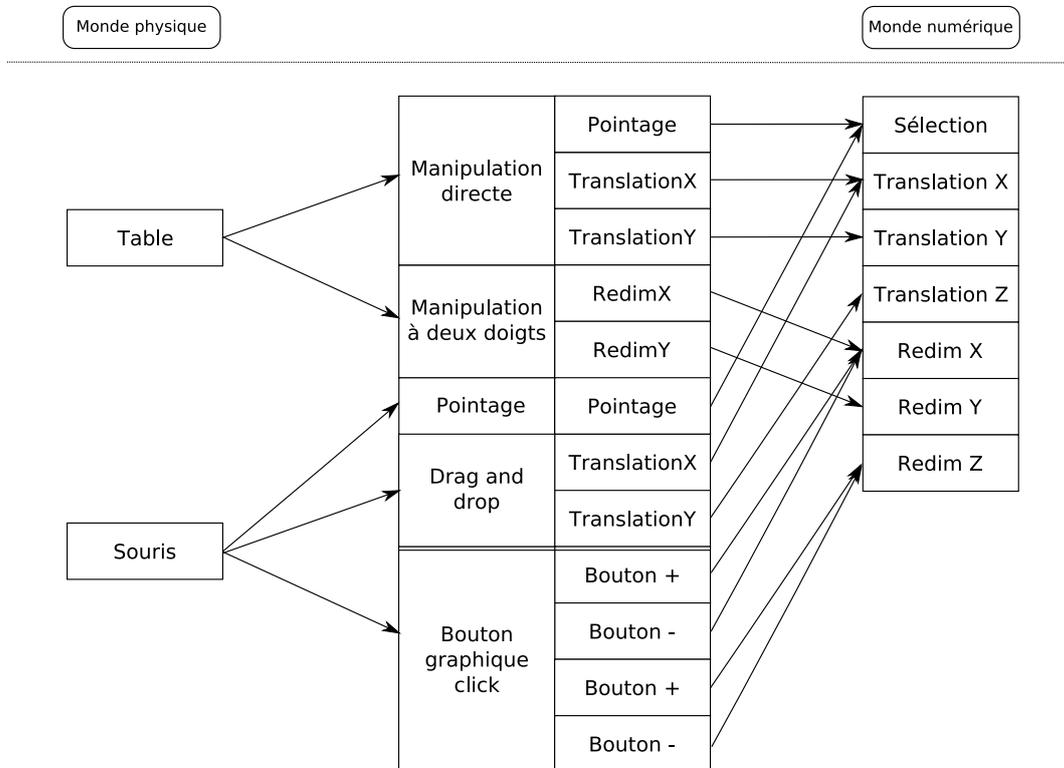


FIGURE 2.17 Solution du *mapping* de l'EDE TS (table tactile + souris).

La dernière possibilité est la combinaison de la table tactile avec le SpaceTraveler. Considérant qu'il n'existe qu'un seul objet virtuel, sa sélection peut être effectuée par un unique dispositif d'entrée. Nous choisissons donc la sélection sur la table tactile afin de réduire le nombre de degrés de liberté de la sélection. Les combinaisons possibles se différencient par le *taux d'équivalence** (possibilité de réaliser une tâche de plusieurs façons différentes) pour l'accomplissement des tâches entre les deux dispositifs en entrée. Nous choisissons d'étudier les combinaisons présentant le taux d'équivalence minimal (aucune tâche ne peut être réalisée de deux façons différentes) et maximal (un nombre maximal de tâche peut être réalisé de plusieurs façons). Nous appelons la solution avec un taux d'équivalence minimal, qui est présentée dans la figure 2.18, SEDE TSt_1 . La figure 2.19 illustre le *mapping* d'une solution avec un taux d'équivalence maximal appelée SEDE TSt_2 .

En résumé, plusieurs choix d'EDE existent. Notre méthode de conception nous amène à comparer les six solutions suivantes :

- SEDE T : table uniquement ;

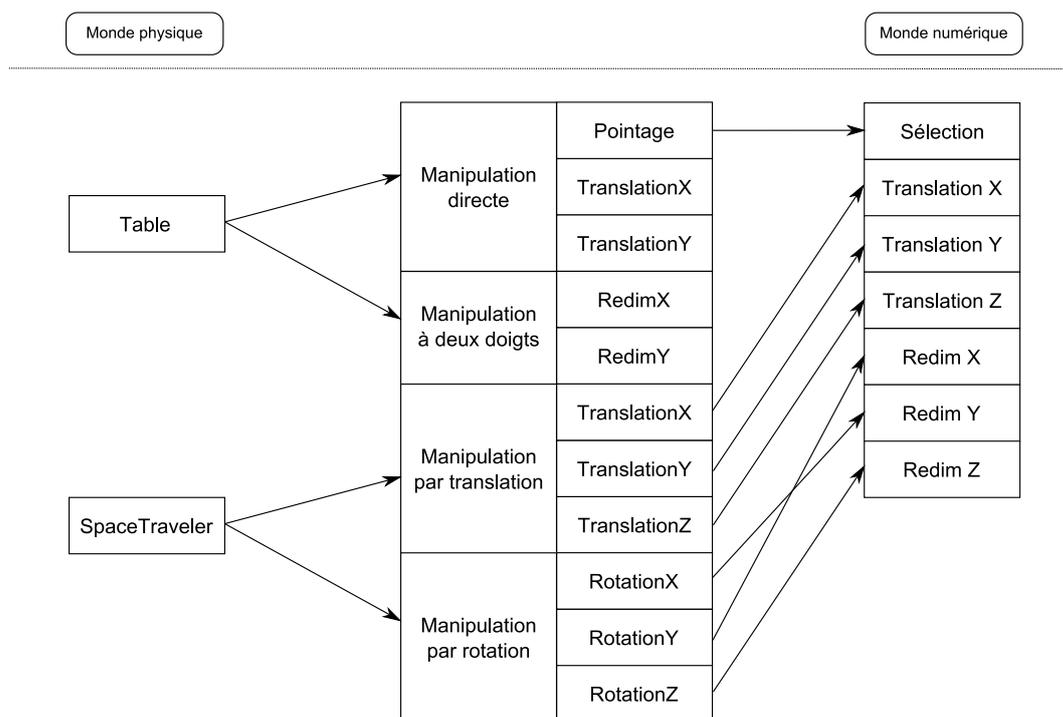


FIGURE 2.18 Première solution du *mapping* de l'EDE TSt (table + SpaceTraveler) appelée SEDE TSt_1 .

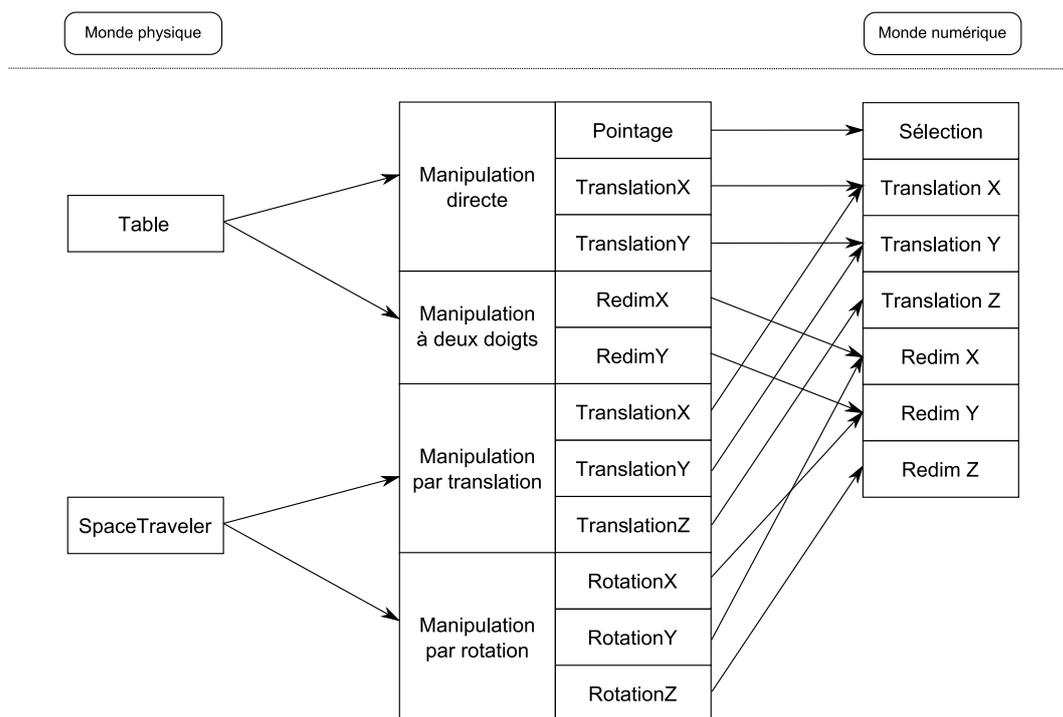


FIGURE 2.19 Seconde solution du *mapping* de l'EDE TSt (table + SpaceTraveler) appelée SEDE TSt_2 .

- SEDE S : souris uniquement ;
- SEDE St : SpaceTraveler uniquement ;
- SEDE TS : table et souris ;

- SEDE TSt_1 : table pour la sélection et SpaceTraveler pour les manipulations ;
- SEDE TSt_2 : table pour la sélection et les manipulations horizontales et le SpaceTraveler pour les manipulations.

2.5.2 Conditions de l'expérimentation

Le but de l'expérimentation est d'examiner les différentes solutions obtenues par la conception par *mapping* pour une tâche de "docking". Les six SEDE sont donc comparées selon les performances et les préférences des utilisateurs. Cette comparaison permet d'éliminer des solutions obtenues grâce à la conception par *mapping* mais qui soulèvent certains problèmes pour les utilisateurs.

Chaque utilisateur est assis devant la table tactile et en face de la surface d'affichage verticale (voir figure 2.20). Il dispose d'une phase d'apprentissage avant la phase d'expérimentation. Durant cette phase d'apprentissage, l'utilisateur teste librement les six SEDE pour la sélection d'une fenêtre 3D virtuelle, son déplacement et son redimensionnement selon les trois dimensions (voir figure 2.21). Cette fenêtre est affichée en 2D sur la table tactile ainsi qu'une icône de caméra représentant le point de vue de la vue 3D présentée verticalement.

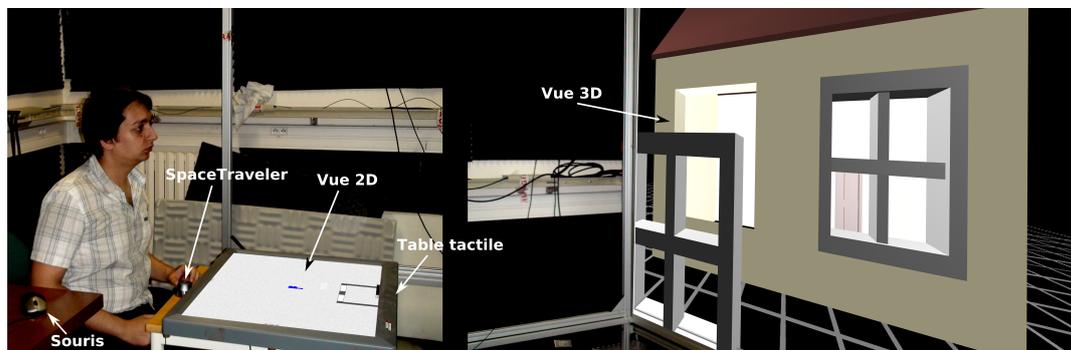


FIGURE 2.20 Utilisateur dans les conditions de l'expérimentation.

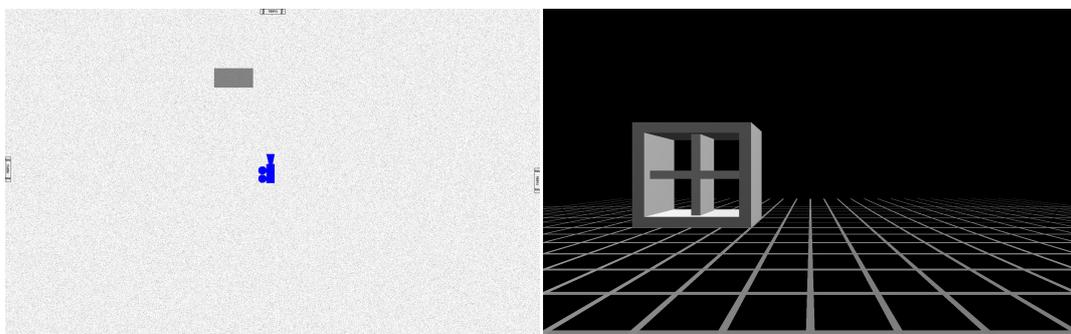


FIGURE 2.21 Vue 2D (à gauche) et vue 3D (à droite) présentées lors de la phase d'apprentissage.

La scène virtuelle de l'expérience est constituée d'une maison avec deux fenêtres, une ouverture et une porte (voir figure 2.22). Une représentation du point de vue 3D est présentée sur la vue 2D par une icône de caméra. Devant cette camera virtuelle, une fenêtre bien positionnée est présente sur le côté droit du mur. Sur le côté gauche, une ouverture de la même taille que la fenêtre sur le côté droit existe dans le mur. La tâche à réaliser par les utilisateurs consiste à positionner et à redimensionner la seconde fenêtre selon les trois axes pour qu'elle corresponde à la position et aux dimensions de l'ouverture. Chaque utilisateur doit réaliser la tâche selon les six SEDE et avec trois conditions initiales différentes pour chaque SEDE : position initiale différente de la position souhaitée uniquement, dimensions initiales différentes des dimensions souhaitées uniquement et position et dimensions initiales différentes de la position et des dimensions souhaitées. L'ordre des conditions diffère d'une SEDE à l'autre pour un même utilisateur. De plus, l'ordre des SEDE est différent d'un utilisateur à l'autre. Ces permutations sont réalisées pour éviter de biaiser les résultats.

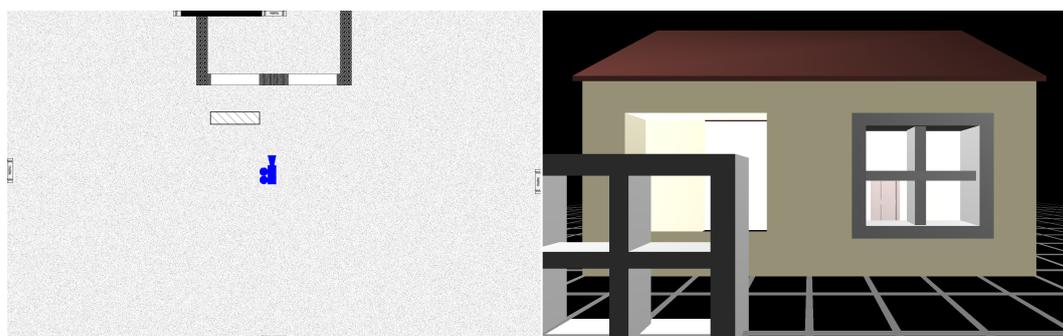


FIGURE 2.22 Vue 2D (à gauche) et vue 3D (à droite) présentées lors de la phase d'évaluation.

Les utilisateurs remplissent un questionnaire durant et après l'expérimentation afin d'obtenir des comparaisons subjectives des solutions. Dans un premier temps, les utilisateurs répondent à des questions concernant leurs expériences avec des environnements virtuels, des tables tactiles et la fréquence à laquelle ils jouent aux jeux vidéos. Pendant l'expérience, et plus précisément après l'accomplissement des tâches en utilisant une solution donnée, chaque utilisateur note sur une échelle de Likert [1932] allant de 1 à 7 (1 étant la note la plus négative et 7 la note la plus positive) la facilité, la satisfaction, l'intuitivité et l'efficacité de la solution utilisée. A la fin de l'expérience, les utilisateurs indiquent leur solution préférée ainsi que leur disposition et combinaison préférée des dispositifs d'entrée (et/ou de sortie).

Le nombre total de participants à l'expérimentation a été de 12 personnes (8 hommes et 4 femmes) âgés entre 22 et 24 ans. Les participants possédaient des formations différentes et leurs familiarités avec les surfaces tactiles et les jeux vidéos étaient très

hétérogènes.

2.5.3 Résultats et analyses

Nous présentons et analysons dans un premier temps les résultats relatifs à la sélection de l'objet virtuel. Ensuite nous étudions les résultats de la manipulation et nous concluons avec l'étude de l'ensemble sélection/manipulation d'un objet virtuel.

Analyse de la sélection

SEDE de sélection. La sélection de l'objet virtuel (i.e. la fenêtre virtuelle) peut s'effectuer avec les trois dispositifs d'entrées. Les six SEDE se réduisent à quatre pour la sélection. En effet, les SEDE T , TSt_1 et TSt_2 ne permettent la sélection de l'objet qu'au moyen de la table tactile et par manipulation directe. Nous obtenons donc les quatre SEDE pour la sélection illustrées dans la figure 2.23.

Analyse objective de la sélection (données de log). Le temps moyen de sélection est très élevé pour la SEDE St en comparaison avec les autres SEDE (voir figure 2.24). Il est environ 4.5 fois le temps moyen de sélection de la SEDE la plus rapide pour la sélection (SEDE S). La sélection avec les autres SEDE s'effectue selon deux degrés

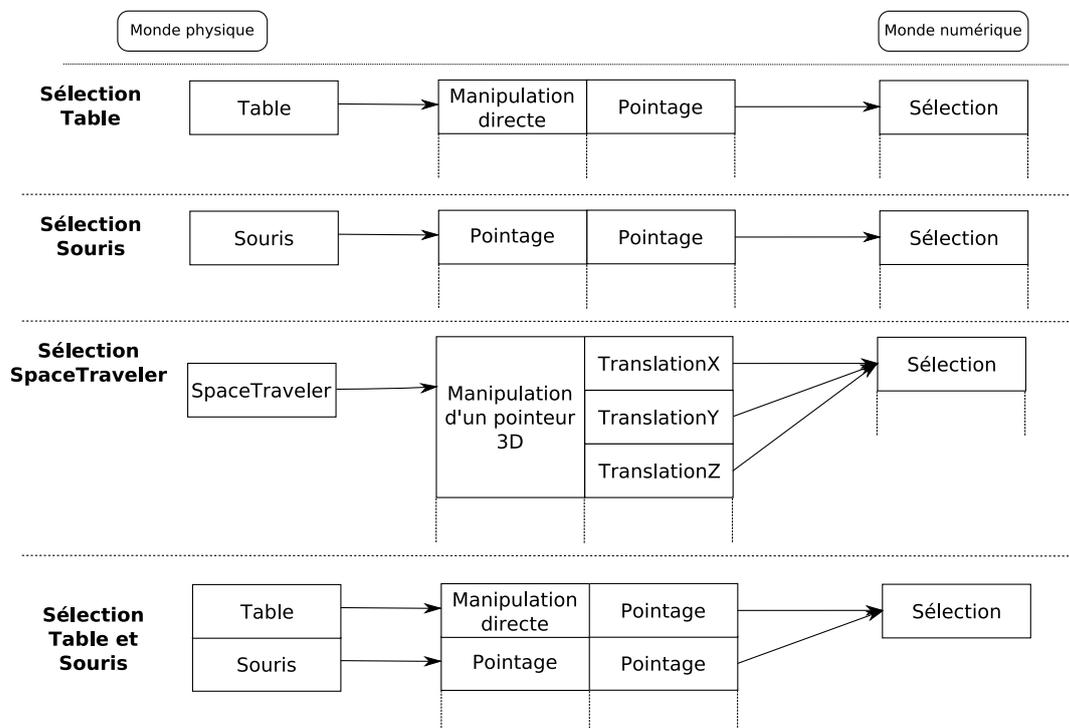


FIGURE 2.23 Les quatre possibilités de sélection de l'expérience.

de liberté uniquement (manipulation directe selon les axes X et Y pour la table et par pointage selon les axes X et Z avec la souris). L'ajout d'un degré de liberté pour la sélection explique ce temps moyen important pour la sélection avec la SEDE *St*. L'aspect isométrique du SpaceTraveler peut aussi expliquer cette différence de temps avec les autres SEDE.

Le temps moyen de sélection par manipulation directe au moyen de la table tactile est le double du temps moyen de sélection par pointage avec la souris. Cette différence de temps peut s'expliquer par :

- la différence des vues : le rapport taille objet sélectionné / vue est supérieur dans la vue 3D associée à la souris que dans la vue 2D associée à la table (voir figure 2.22) ;
- la différence des dispositifs d'entrée : le rapport de déplacement physique / déplacement numérique est inférieur lors du pointage par la souris que par manipulation directe au moyen de la table.

Analyse subjective de la sélection (préférences utilisateurs). Dans le questionnaire auquel les participants ont été amené à répondre, une question a été posée sur la préférence du dispositif de sélection.

75% des participants préfèrent la sélection avec la table tactile. Cinq personnes sur les neuf qui ont effectué ce choix le justifient (entre autres) par sa rapidité de sélection. Un participant a choisi la table pour la sélection pour son côté "haptique". Les deux personnes qui ont choisi la souris comme le dispositif de sélection préféré justifient leur choix par l'intuitivité et la facilité de la sélection. Ces deux participants considèrent que la table est aussi un bon dispositif pour la sélection mais l'un d'entre eux souligne la lenteur et l'imprécision de la sélection sur la table. L'unique participant qui a choisi le SpaceTraveler comme dispositif préféré pour la sélection justifie sa réponse par le côté

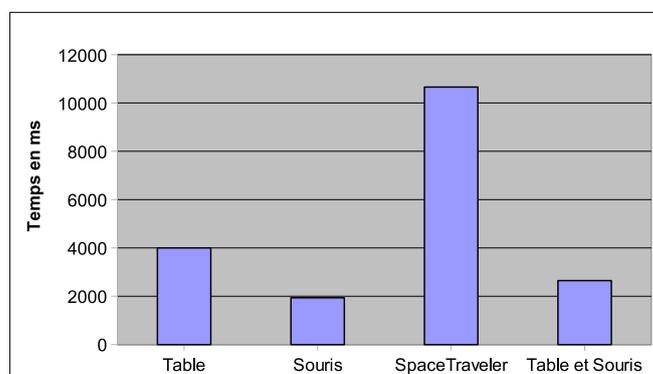


FIGURE 2.24 Temps moyen de sélection en ms selon les quatre SEDE de sélection.

"fun" de ce mode de sélection. Une autre personne a souligné le fait que si l'occlusion était importante, il aurait choisi aussi le SpaceTraveler comme dispositif préféré pour la sélection.

En résumé, la table tactile est le dispositif préféré par la majorité (75%) des participants pour la sélection pour son côté rapide (selon les participants). Le SpaceTraveler est le dispositif le moins retenu pour la sélection en raison de son temps important de sélection.

Conclusion sur la sélection. La comparaison des temps nécessaires pour la sélection d'un objet virtuel montre qu'ils sont nettement inférieurs lors de l'utilisation de dispositifs d'entrée à deux dimensions que lors de l'utilisation de dispositifs à trois dimensions. Cet aspect est soutenu par les préférences des participants (92%) qui privilégient les dispositifs à deux dimensions aux dispositifs à trois dimensions pour la sélection. **Les dispositifs d'entrée à deux dimensions sont plus appropriés pour les tâches de sélection.**

Suite aux conclusions précédentes, nous suggérons, pour la modélisation des interactions sur des tâches de sélection, l'utilisation de dispositifs d'entrée bidimensionnels et non des dispositifs d'entrée à plusieurs degrés de liberté. Pour lutter contre les phénomènes d'occlusion, trois possibilités existent :

- la possibilité de sélection au moyen de plusieurs vues ;
- la navigation ;
- la création de techniques d'interaction permettant la sélection en évitant les occlusions (e.g. lentille "see through" ou lentille permettant de voir et de sélectionner des objets cachés par d'autres objets virtuels).

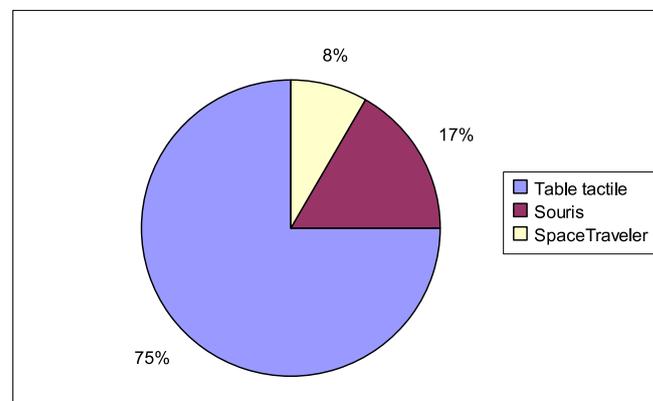


FIGURE 2.25 Préférence des utilisateurs pour la SEDE de sélection.

Analyse de la manipulation

Après avoir étudié la sélection selon les différentes solutions de conception obtenues, nous comparons les six SEDE pour la manipulation d'un objet virtuel. Nous discutons des similarités entre certaines des solutions. Les résultats objectifs puis subjectifs sont présentés ensuite. Nous concluons cette section par des recommandations pour optimiser certaines solutions obtenues.

SEDE de manipulation. Il existe six SEDE pour la réalisation des tâches de translation et de redimensionnement de l'objet virtuel selon les trois dimensions. Les résultats de la conception par *mapping* de la SEDE *St* (voir figure 2.16) et de la SEDE *TSt₁* (voir figure 2.18) pour la manipulation d'un objet virtuel sont semblables. Avec la SEDE *St* les utilisateurs déplacent en 3D un pointeur dans la vue 3D pour réaliser la sélection tandis que dans la SEDE *TSt₁* la sélection est effectuée au moyen de la table tactile. Dans ces deux solutions, les translations (respectivement les redimensionnements) de l'objet virtuel sont effectuées identiquement en translatant le SpaceTraveler (respectivement en modifiant l'orientation du SpaceTraveler). A cause de la différence de la technique d'interaction pour la sélection entre ces deux SEDE, nous étudions la manipulation de l'objet virtuel avec ces deux SEDE de façon séparée.

Analyse objective de la manipulation (données de log). Le temps moyen de manipulation le moins élevé est celui des interactions effectuées avec la souris uniquement (voir figures 2.26 et 2.27). Le temps de manipulation en utilisant le SpaceTraveler est inférieur dans la SEDE *St* à celui des SEDE *TSt₁* et *TSt₂*. Cela s'explique par le temps nécessaire pour changer de dispositif entre la table et le SpaceTraveler. Ce changement de dispositif intervient une fois uniquement (après la sélection) pour la SEDE *TSt₁* et au moins une fois pour la SEDE *TSt₂*. De plus, la lenteur du retour d'information au niveau de la vue 3D lors du redimensionnement pour les SEDE *TSt₁* et *TSt₂*

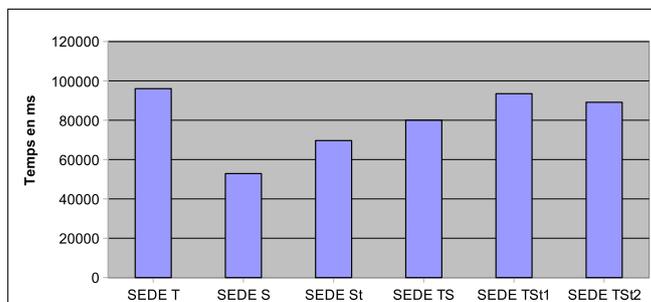


FIGURE 2.26 Temps moyen de manipulation en ms selon les six SEDE d'interaction.

peut expliquer les mauvaises performances du SpaceTraveler. Le temps moyen de manipulation le plus élevé est celui de la SEDE impliquant des interactions qui utilisent la table tactile uniquement.

Les données de l'expérience avec le participant 7 ont été retirées des résultats finaux en raison de problèmes de perception 3D. Les figures 2.28 et 2.29 montrent que la précision en utilisant la SEDE S est nettement plus importante que les autres SEDE. Le redimensionnement en utilisant la SEDE TSt_1 est le moins précis. Cela s'explique par deux participants qui ont été imprécis en n'utilisant pas la bonne vue pour le contrôle visuel d'une transformation (voir figures 2.30 et 2.31). A la fin de la tâche, l'un de ces

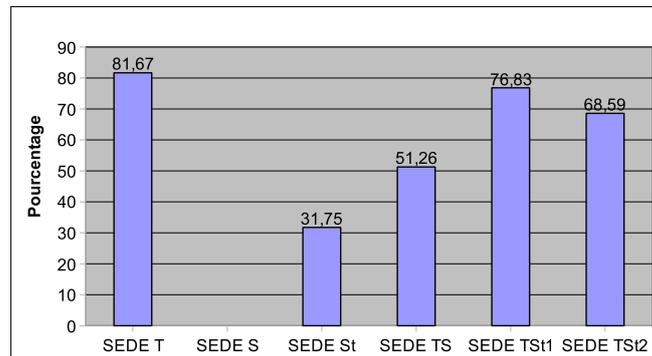


FIGURE 2.27 Pourcentage de la différence de temps entre une SEDE de manipulation et la SEDE de manipulation ayant le temps moyen de manipulation le moins élevé (SEDE S).

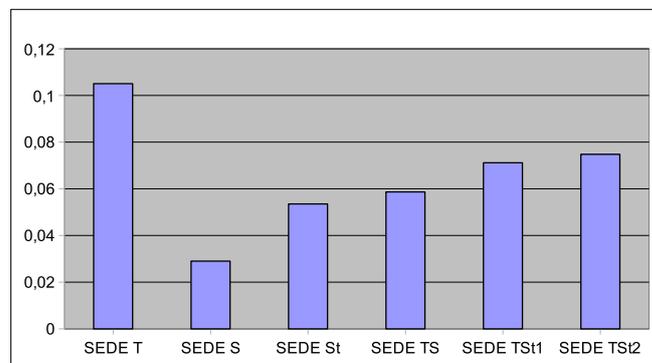


FIGURE 2.28 Précision moyenne de position pour chaque SEDE (en unité de mesure du repère du monde virtuel).

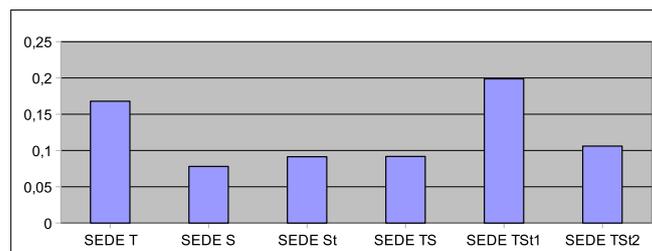


FIGURE 2.29 Précision moyenne de dimension pour chaque SEDE (en unité de mesure du repère du monde virtuel).

deux participants a déplacé l'objet virtuel selon l'axe Y (en profondeur) en se référant à la vue 3D alors que le meilleur retour d'information pour ce type de manipulation se situe sur la vue 2D. L'autre participant a commis une erreur similaire en utilisant la vue de dessus en 2D pour le contrôle de la hauteur. En omettant la donnée de la précision de ce participant sur cette tâche, la précision de redimensionnement avec la SEDE TSt_1 est meilleure que la SEDE T . La SEDE T est environ 4 fois plus imprécise pour le positionnement en comparaison avec la SEDE la plus précise qui est la SEDE S (voir figure 2.28).

Analyse subjective de la manipulation (préférences utilisateurs). Les participants (43%) ayant choisi la SEDE S comme SEDE préférée pour la manipulation (voir figure 2.32) justifient leur choix par leur habitude d'utiliser la souris et la précision de ce dispositif. Deux des quatre participants qui ont choisi la SEDE TSt_2 justifient leurs choix par la facilité et la vitesse pour effectuer des grandes translations dans le plan horizontal sur la table tactile et l'intuitivité du SpaceTraveler. Le troisième participant qui a choisi cette SEDE le justifie par le côté intuitif alors que la dernière personne a préféré cette SEDE pour la possibilité d'effectuer les redimensionnements par différents

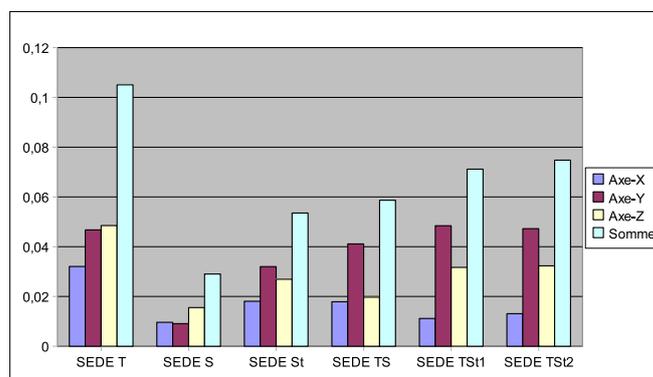


FIGURE 2.30 Précision moyenne de position pour chaque SEDE de manipulation selon chaque axe (en unité de mesure du repère du monde virtuel).

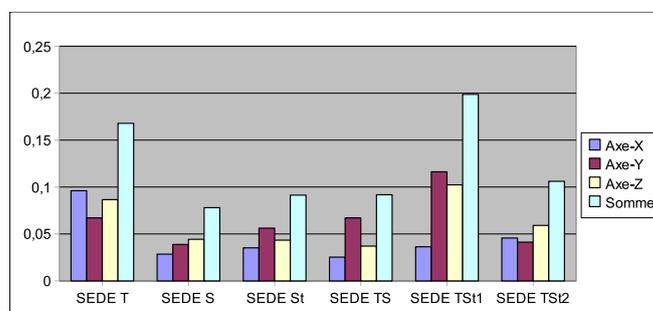


FIGURE 2.31 Précision moyenne de dimension pour chaque SEDE de manipulation selon chaque axe (en unité de mesure du repère du monde virtuel).

moyens. Soulignons que la SEDE T n'a été choisi par aucun des participants comme solution préférée pour la manipulation d'un objet virtuel.

En résumé, les avis sont partagés sur la préférence de la SEDE de manipulation entre la SEDE S et la SEDE TSt_2 . Les participants ayant choisi la SEDE S le justifient par leur habitude d'utiliser la souris et par sa précision. Les utilisateurs qui ont privilégié la SEDE TSt_2 le justifient par l'intuitivité de cette solution et par le bon partage des fonctionnalités entre les dispositifs (translation horizontale grossière sur la table et les six degrés de liberté avec le SpaceTraveler). L'un des participants ayant choisi la SEDE S comme solution préférée a souligné qu'avec plus d'apprentissage il aurait préféré la SEDE TSt_1 . Un autre participant ayant choisi la SEDE S met au même niveau la SEDE TSt_2 . Plusieurs participants ont signalé une sensibilité trop importante du SpaceTraveler rendant son utilisation difficile.

Conclusion sur la manipulation. La précision et le temps de manipulation favorisent la SEDE S qui est préférée par 43% des participants. Néanmoins, 33% des participants ont préféré la SEDE TSt_2 qui permet la réalisation de la translation et du redimensionnement dans le plan horizontal sur la table et qui offre toutes les fonctionnalités (sauf la sélection) en utilisant le SpaceTraveler. De plus, en sommant les SEDE utilisant le SpaceTraveler nous obtenons un résultat de 49% des participants qui préfèrent une SEDE utilisant ce dispositif. L'apprentissage pourrait aider les participants à mieux utiliser le SpaceTraveler. La table tactile est peu adaptée pour des manipulations précises surtout pour le redimensionnement horizontal de l'objet. **Pour la manipulation d'objets virtuels, la souris reste le dispositif le plus précis, rapide et préféré des participants mais avec l'apprentissage, et en rendant le dispositif à six degrés de liberté moins sensible, la combinaison d'un dispositif de type SpaceMouse avec une table tactile est très prometteuse.**

Suite aux conclusions précédentes, nous suggérons pour la modélisation des inter-

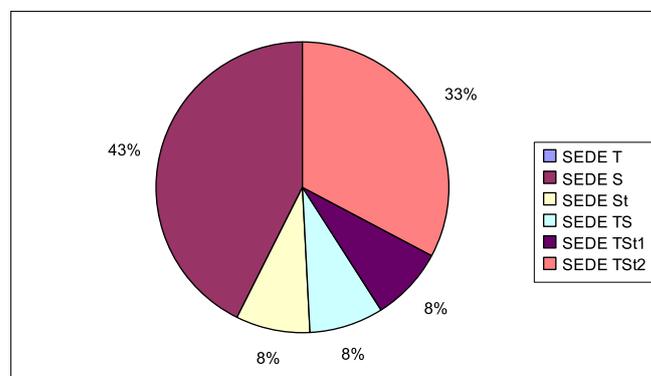


FIGURE 2.32 Préférence des utilisateurs pour la SEDE de manipulation.

actions pour des tâches de manipulation d'un objet virtuel l'utilisation de la souris ou la combinaison d'un dispositif à plusieurs degrés de liberté avec une table tactile en respectant les conditions suivantes :

- donner un temps suffisant pour l'appropriation des dispositifs d'entrée ;
- rendre le dispositif à six degrés de liberté peu sensible ou offrir aux utilisateurs la possibilité de régler la sensibilité du dispositif.

Analyse de l'intégralité sélection/manipulation

Dans cette section, nous présentons les résultats de l'ensemble de la sélection et de la manipulation. Nous analysons les résultats objectifs ensuite les résultats subjectifs de l'intégralité de l'expérimentation.

Analyse objective de l'intégralité sélection/manipulation (données de log).

Le temps nécessaire pour la sélection d'un objet virtuel dans une tâche de docking est beaucoup moins important que celui pour accomplir la tâche correspondante. Cette différence de temps justifie la ressemblance entre les figures 2.33 et la figure 2.26. Comme indiqué dans les parties précédentes, la SEDE utilisant la souris uniquement possède le temps moyen d'interaction le plus faible par rapport aux autres solutions.

Analyse subjective de l'intégralité sélection/manipulation (préférences utilisateurs). Chaque participant a été amené à noter dans le questionnaire sur une échelle de Likert [1932] de 1 à 7 (1 étant la note la plus négative et 7 la note la plus positive) la facilité, la satisfaction, l'intuitivité et l'efficacité de chaque SEDE. Les sommes des points affectés par tous les participants sont illustrées dans les figure 2.34.a (pour la facilité d'interaction), 2.34.b (pour la satisfaction d'interaction), 2.34.c (pour l'intuitivité d'interaction) et 2.34.d (pour l'efficacité d'interaction). La somme globale de tous ces points est illustrée dans la figure 2.35.

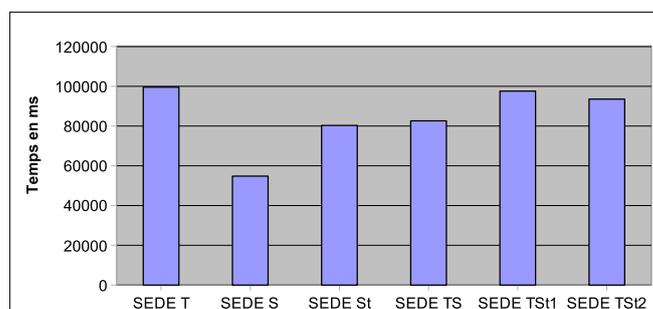


FIGURE 2.33 Temps moyen d'interaction global en ms selon les six SEDE.

Dans le cadre de cette expérience, la SEDE utilisant la souris uniquement est la solution que les participants trouvent la plus efficace (figure 2.34.d), la plus facile (figure 2.34.a) et leur niveau de satisfaction pour cette solution est le plus élevé (figure 2.34.b). Les participants n'ont pas trouvé cette SEDE la plus intuitive (figure 2.34.c). L'utilisation plus fréquente de la souris que les autres dispositifs (SpaceTraveler et table tactile) par les participants pourrait expliquer ces résultats (i.e. l'un des participants l'a explicitement formulé ainsi). Dans la situation inverse se trouve la SEDE *T*. Cette solution a été considérée comme la plus intuitive par les participants (figure 2.34.c) mais ils ne l'ont trouvée ni facile, ni efficace (figures 2.34.a et 2.34.d). De plus, les participants sont les moins satisfaits de cette SEDE (figure 2.34.b). Les participants soulignent la difficulté de redimensionnement à deux doigts et l'imprécision de la table tactile. La solution combinant la table et la souris (SEDE *TS*) a été moins bien notée que la SEDE *St* et les SEDE *TSt*₁ et *TSt*₂ qui sont quasiment au même niveau (figure 2.35). Les dif-

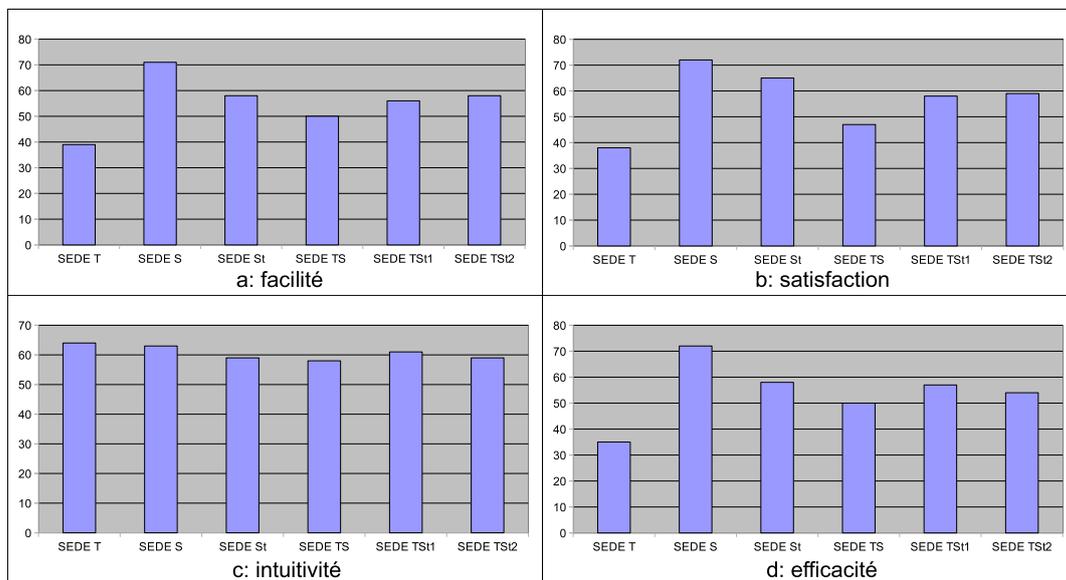


FIGURE 2.34 Notation des quatre métriques subjectives selon chaque SEDE.

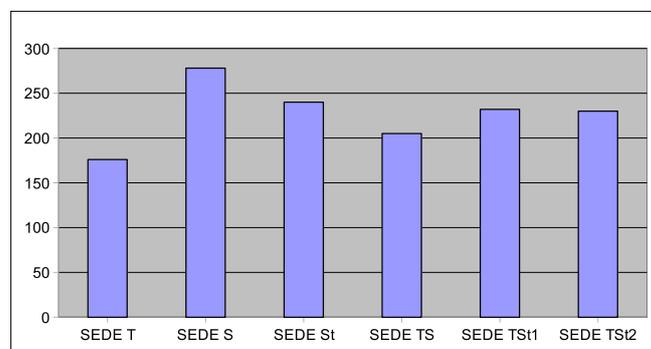


FIGURE 2.35 La somme de tous les points affectés par les participants à chaque SEDE.

ficultés que les participants ont rencontrés durant l'utilisation de la table pour effectuer les interactions avec deux doigts pour le redimensionnement horizontal expliquent ces résultats.

En résumé, la SEDE utilisant la souris obtient la meilleure notation des participants, suivi des trois SEDE utilisant le SpaceTraveler, puis la SEDE combinant la table et la souris. La SEDE la moins bien notée par les participants est celle utilisant la table tactile uniquement comme dispositif d'entrée.

59% des participants préfèrent de façon globale la SEDE *S* (souris uniquement) par rapport aux autres SEDE (voir figure 2.36). Les participants justifient leurs choix par leur habitude d'utilisation de la souris et par la facilité, la précision et la rapidité des interactions avec ce dispositif. Aucun participant n'a choisi la SEDE *T* ou la SEDE *TSt₁* comme solution préférée. 25% des participants ont préféré la SEDE *TSt₂*. Nous soulignons que ces participants jouent souvent aux jeux vidéos. Un participant qui a choisi la SEDE *S* met au même niveau la SEDE *TSt₂*. Ces données montrent que la souris reste un dispositif d'entrée que les participants préfèrent en raison de leurs habitudes à utiliser ce dispositif.

Conclusion sur l'intégralité sélection/manipulation. Dans le cadre de l'expérimentation, la souris est le dispositif nécessitant le moins de temps d'interaction pour accomplir la tâche. De plus, la préférence des utilisateurs pour ce dispositif est visible par leurs notations des différentes solutions. Même si la SEDE *S* est la solution préférée des participants, la SEDE *T* a été mieux notée que la SEDE *S* pour son intuitivité. En reprenant la conclusion de la partie précédente, **la souris reste le dispositif le plus précis, rapide et préféré des participants mais avec l'apprentissage, et en le rendant moins sensible, un dispositif de type SpaceMouse combiné avec une table tactile offre une excellente solution.**

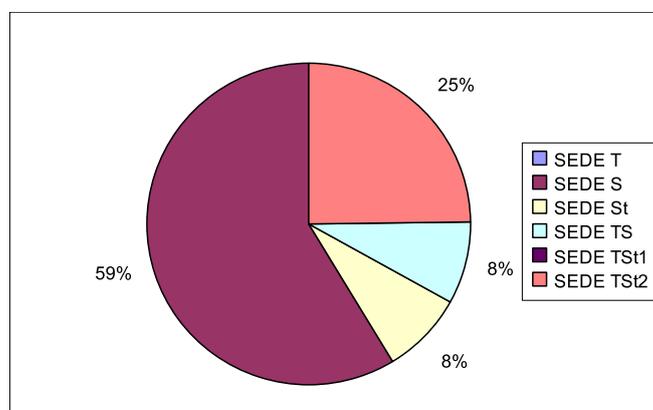


FIGURE 2.36 Préférence des utilisateurs de la SEDE d'interaction pour l'ensemble de l'expérience.

2.5.4 Conclusion de l'expérimentation

Six combinaisons de dispositifs d'entrée et de correspondance entre ces dispositifs et les degrés de liberté d'une tâche de docking ont été développées avec le modèle de *mapping*. Il existe deux niveaux de combinatoire possibles dans la conception par *mapping* : le premier se situe au niveau de la combinaison de dispositifs en entrée tandis que le second se situe au niveau de la combinatoire des liaisons reliant la *table des techniques d'interaction* à la composante *tâche* du *mapping*.

Les résultats de l'expérimentation montrent que l'utilisation d'un dispositif à six degrés de liberté pour des tâches spatiales nécessitant six degrés de liberté n'est pas la plus idéale. La souris reste un dispositif d'entrée préféré par les utilisateurs mais la combinaison d'un dispositif d'entrée à six degrés de liberté avec une table tactile est prometteuse en raison des complémentarités entre ces deux types de dispositifs.

La différence entre la SEDE TSt_1 et la SEDE TSt_2 se situe au niveau de la combinatoire des connexions entre la *table des techniques d'interaction* et la composante *tâche* du *mapping*. La comparaison des résultats obtenues par ces deux solutions montre que la préférence des utilisateurs pour la SEDE TSt_2 (table tactile et SpaceTraveler) est supérieure à celle pour la SEDE TSt_1 (qui combine les mêmes dispositifs d'entrée). Cette différence justifie l'importance du taux d'équivalence des interactions dans le choix des combinaisons entre techniques d'interaction pour la réalisation d'une même tâche.

2.6 Conclusion

La conception par *mapping* permet de combiner plusieurs dispositifs en entrée, plusieurs vues et plusieurs surfaces d'affichage. L'élément intéressant de ce modèle est la prise en compte de l'association d'un dispositif en entrée à une surface d'affichage, ainsi que les correspondances entre l'espace physique et le(s) espace(s) numérique(s) en entrée et en sortie :

1. en entrée, l'association des degrés de liberté entre le monde physique et le monde numérique ;
2. en sortie, la disposition des vues dans le monde physique et leurs corrélations avec les dimensions du monde numérique ;
3. la relation entre les vues.

La conception par *mapping* génère une combinatoire de taille importante mais les règles définies permettent d'éliminer les combinaisons ne respectant pas certaines

contraintes. L'expérimentation avec des utilisateurs permet de valider et de comparer les solutions obtenues.

La modélisation en utilisant le modèle *mapping* se situe à un haut niveau de la conception des interactions et des interfaces. Pour rapprocher le modèle *mapping* de la conception logicielle nous définissons dans le chapitre suivant les composantes logiques ainsi que l'espace de conception qui les utilise.

Chapitre 3

Architecture logicielle de la combinatoire des composantes logiques

3.1 Introduction

Dans le domaine de la multimodalité, la conception des entrées est souvent découplée de la conception des sorties. Or dans certaines techniques d'interaction, il existe un couplage fort entre l'entrée et la sortie (e.g. manipulation directe d'un objet virtuel sur un écran tactile). Lorsque le couplage fort existe sur plusieurs dimensions (comme dans les TEAMViz) il est encore plus difficile de réaliser la conception avec une architecture découplant les entrées des sorties. Afin de pouvoir combiner des interactions à couplage fort, nous définissons la notion de composante logique qui relie les entrées et les sorties d'une interface multimodale. Une composante logique est un ensemble de composants logiciels (fichiers ou objets). La conception par composante logique permet de faciliter la conception logicielle des TEAMViz.

Ce chapitre présente dans un premier temps la notion de composante logique et la relation entre une composante logique et une modalité. Ensuite, nous définissons un espace de conception ainsi que la transposition de la conception par *mapping* à la conception par composantes logiques. La combinatoire des connexions entre les dispositifs physiques et les composantes logiques d'une part, et les composantes logiques et les tâches d'autre part, est de taille importante. Comme pour la conception par *mapping*, nous énonçons des règles de conception par composantes logiques afin d'éliminer des combinaisons indésirables. Pour conclure ce chapitre, nous illustrons la conception par composantes logiques sur un exemple d'application.

3.2 Les composantes logiques

Dans cette section nous définissons la notion de composante logique puis nous la comparons aux modalités. Nous présentons ensuite les différents types de composantes logiques qui peuvent exister.

3.2.1 Définition

Les composantes logiques se situent dans la couche *interaction logique* du modèle physique/numérique présenté dans la figure 2.1. Chaque composante relie au moins un dispositif physique d'entrée et/ou de sortie au *contrôleur de dialogue*. La figure 3.1 illustre trois composantes logiques de la couche *interaction logique* du modèle physique/numérique ainsi que leurs connexions aux dispositifs physiques présents dans la couche *interaction physique* et à un élément de la couche *contrôleur de dialogue*.

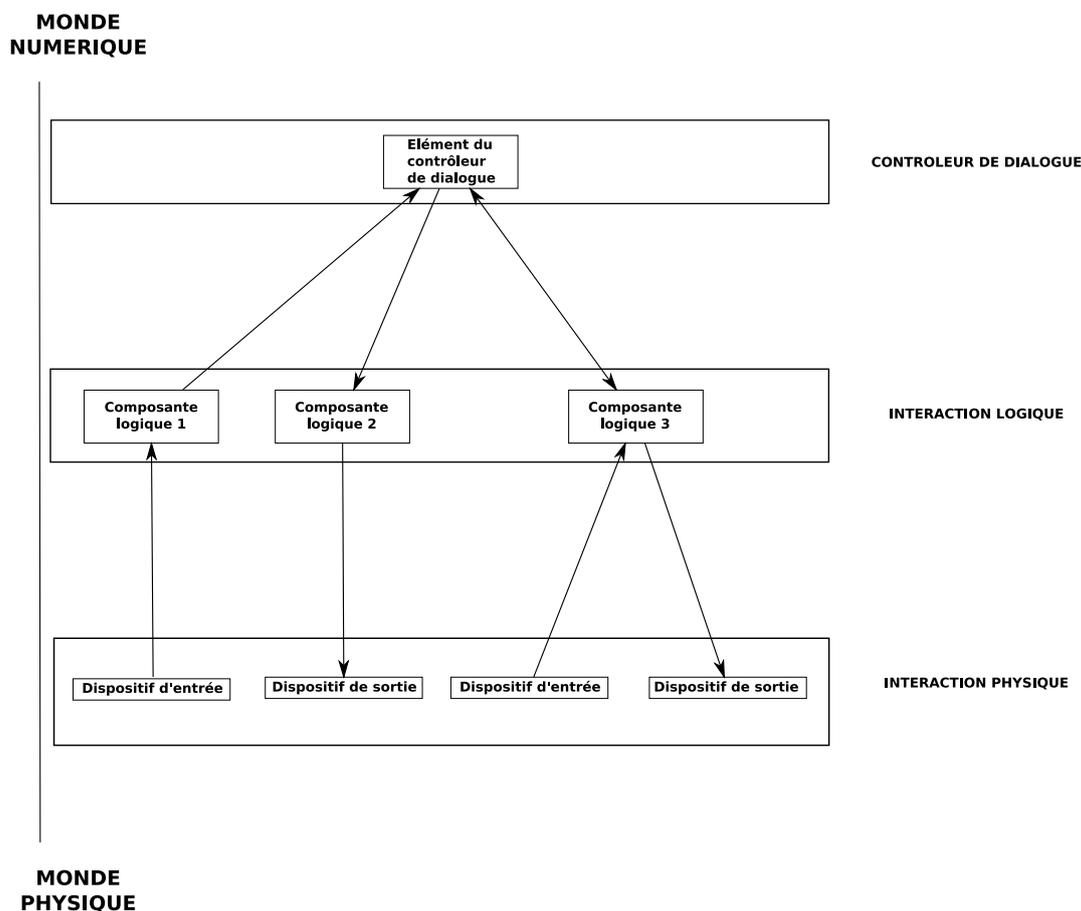


FIGURE 3.1 Trois composantes logiques dans le modèle physique/numérique et leurs connexions aux couches *interaction physique* et *contrôleur de dialogue*.

Chaque dispositif physique de la couche *interaction physique* est connecté à une ou plusieurs composantes logiques. En effet, considérons le cas simple d'un système gérant

les manipulations directes sur un écran tactile de deux formes géométriques qui sont un cercle et un carré. Le couplage fort entre les entrées et les sorties est nécessaire dans ce type de système en raison de la co-localisation de l'écran (dispositif en sortie) et de la surface tactile (dispositif en entrée). Dans ce système, la conception nécessite de distinguer deux composantes logiques CL_{cercle} et $CL_{carré}$ liées respectivement aux deux formes géométriques et connectées à la surface tactile en entrée et à l'écran en sortie (voir figure 3.2). Ainsi, un même dispositif physique peut être connecté à plusieurs composantes logiques.

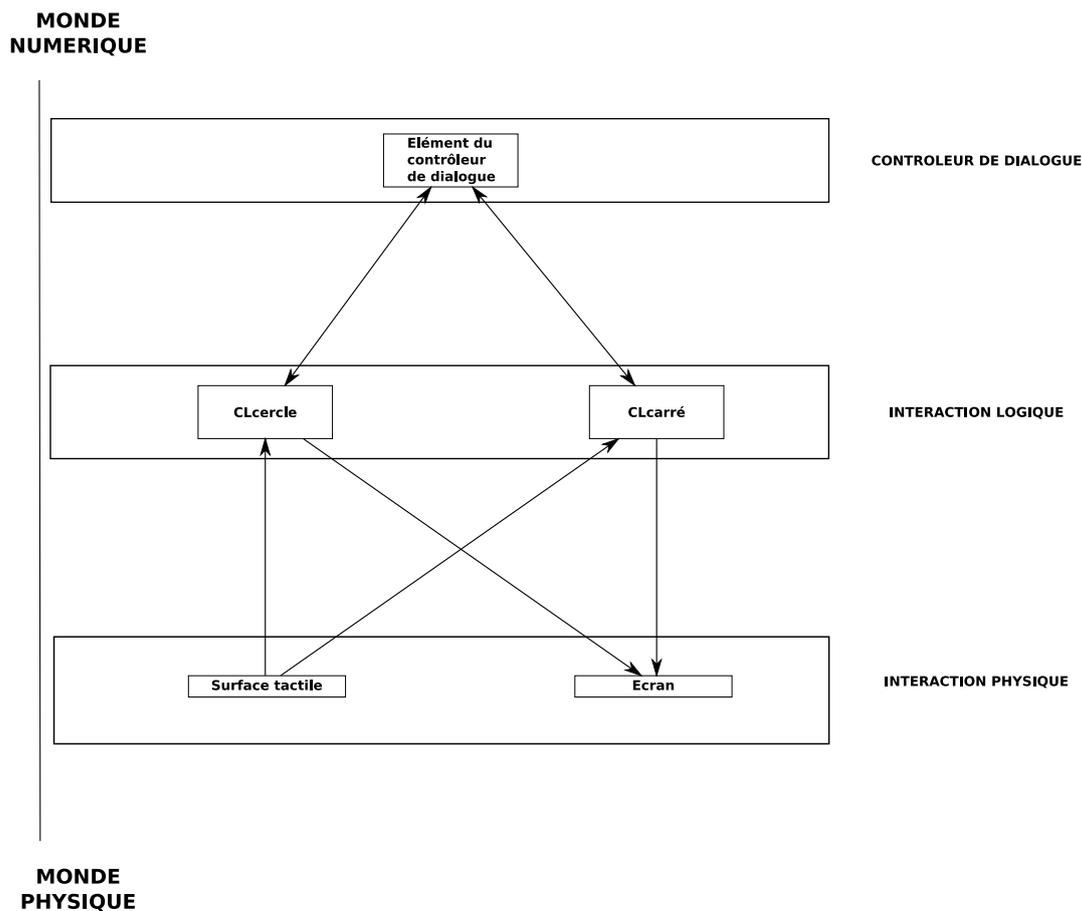


FIGURE 3.2 Deux composantes logiques CL_{cercle} et $CL_{carré}$ reliées à un écran tactile.

L'utilisation de composantes logiques facilite le développement de systèmes dans lesquels le couplage fort entre les entrées et les sorties est requis. Elle permet de combiner ces dispositifs avec d'autres dispositifs physiques d'entrées et/ou de sorties. Les composantes logiques permettent également d'abstraire les dispositifs physiques au moyen d'entités reconnues qui sont plus adéquates pour le concepteur (e.g. abstraction de l'utilisation de la souris pour un déplacement d'un objet virtuel par une composante logique liée à cet objet).

3.2.2 Composantes logiques et modalités

La variété des dispositifs physiques d'entrée et de sortie ainsi que les variations du degré du couplage entre les entrées et les sorties d'un système nous mènent à différencier trois types de composantes logiques selon le type de dispositifs physiques auxquels elles sont connectées. Un dispositif est le dispositif physique qui acquiert une information (dispositif en entrée) ou qui présente l'information (dispositif en sortie). A chaque dispositif correspond un *langage d'interaction défini par un ensemble d'expressions bien formées qui transportent une signification* [Nigay & Coutaz, 1995]. Dans la suite nous considérons une modalité comme un couple (dispositif, langage d'interaction) comme définie par Nigay et Coutaz [1995]. Un exemple d'une modalité est le couple (microphone, pseudo langage naturel).

La composante logique d'entrée intègre le langage d'interaction du dispositif d'entrée auquel elle est connectée et assure la liaison entre ce dispositif et le *contrôleur de dialogue*. Elle est connectée uniquement à un dispositif (d'entrée) de la couche *interaction physique* du modèle physique/numérique. Le couple (dispositif physique d'entrée, composante logique d'entrée) intègre une modalité en entrée. La composante logique 1 de la figure 3.1 est un exemple d'une composante logique d'entrée (e.g. composante logique liée à un microphone).

La composante logique de sortie assure la liaison connectant le *contrôleur de dialogue* à l'unique dispositif de sortie de la couche *interaction physique* auquel elle est connectée. Elle intègre le langage d'interaction de ce dispositif en sortie. Le couple (dispositif physique de sortie, composante logique de sortie) intègre ainsi une modalité en sortie. Un exemple de composante logique de sortie est la composante logique 2 de la figure 3.1 (e.g. composante logique liée à la visualisation d'un cube virtuel non interactif).

Les deux types de composantes logiques précédents associés aux dispositifs physiques intègrent les modalités en entrée et en sortie. Elles séparent totalement les entrées des sorties et sont donc utilisées lorsque le couplage fort entrée/sortie n'est pas requis. Appelé composante logique d'entrée/sortie, le troisième type de composante logique permet par contre de préserver le couplage fort d'entrée/sortie. Constituée d'un langage d'interaction en entrée et d'un langage d'interaction en sortie, chaque composante logique d'entrée/sortie est connectée à un dispositif en entrée et à un dispositif en sortie de la couche *interaction physique* du modèle physique/numérique. La composante 3 de la figure 3.1 est une composante logique d'entrée/sortie. La composante logique d'entrée/sortie augmente le pouvoir d'expressivité du concepteur qui utilise des composantes logiques d'entrée ou de sortie séparément. Une composante logique liée à un cube virtuel manipulable en utilisant une souris est un exemple d'une composante logique

d'entrée/sortie.

3.2.3 Composante logique spatiale

Dans nos travaux, nous nous intéressons particulièrement à la combinaison d'interactions effectuées au moyen de deux vues. La première vue est une vue 2D projetée sur une table tactile tandis que la seconde est une vue 3D présentée sur un écran (ou surface d'affichage) vertical. Dans le cadre de la combinaison d'interactions au moyen d'une vue 2D et d'une vue 3D, les notions d'espaces, de dimensions et de degrés de liberté sont fondamentales. Ainsi, nous proposons d'augmenter la notion de composante logique par des données sur l'espace numérique. Cette nouvelle composante logique appelée composante logique spatiale, intègre les degrés de liberté de manipulation qui sont définis selon le dispositif en entrée, la composante logique en elle-même et/ou les dimensions qui sont liées à la visualisation de l'information en sortie.

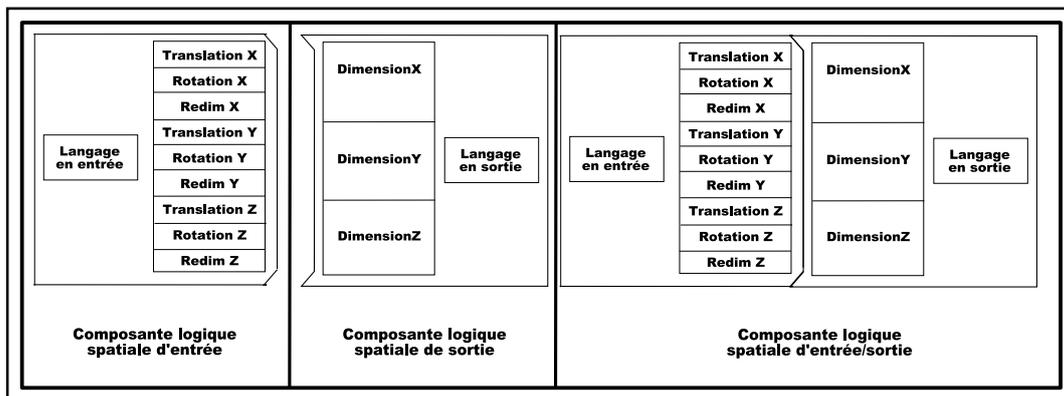


FIGURE 3.3 Types de composantes logiques spatiales. Gauche : composante logique spatiale d'entrée. Centre : composante logique spatiale de sortie. Droite : composante logique spatiale d'entrée/sortie.

Une composante logique spatiale peut être définie selon les trois types de composantes logiques présentés auparavant (voir figure 3.3). Une composante logique spatiale d'entrée intègre donc les degrés de liberté de manipulation numériques tandis qu'une composante logique spatiale de sortie intègre les dimensions de présentation de l'information. L'élément intéressant est la composante logique spatiale d'entrée/sortie qui inclue les degrés de liberté de manipulation en entrée et les dimensions de présentation de l'information. L'intégration de ces deux notions implique l'existence de relations entre elles dans une même composante logique ou avec des composantes logiques différentes connectées à un même élément du contrôleur du dialogue. En effet, le but de cette architecture est par exemple d'avoir un retour d'information sur la rotation d'un objet géométrique qui doit être visible sur l'un des dispositifs de sortie. Ce retour d'in-

formation implique une relation forte entre degrés de liberté en entrée et dimensions en sortie.

3.2.4 Types de composantes logiques

Nous différencions une composante logique non spatiale (CL^*) d'une composante logique spatiale (CLS^*). Nous définissons aussi trois types de composantes logiques : d'entrée (E), de sortie (S) et d'entrée/sortie (ES). En plus, nous séparons les composantes logiques selon leurs fonctionnalités. La distinction d'une composante logique représentant un objet virtuel d'une autre qui représente une vue est due à l'hétérogénéité des tâches à accomplir et à la façon de présenter l'information entre ces deux notions. Nous déterminons aussi une troisième fonctionnalité qui est la représentation d'une vue sur une autre vue (e.g. icône d'une caméra affichée sur une vue correspondant au point de vue d'une seconde vue). Elle permet, si la composante logique l'autorise, de naviguer dans une vue toute en interagissant sur une autre vue. Ainsi, nous identifions trois fonctionnalités essentielles : un Objet virtuel (Obj), une Vue (V) et une Représentation d'une Vue dans une autre vue (RV).

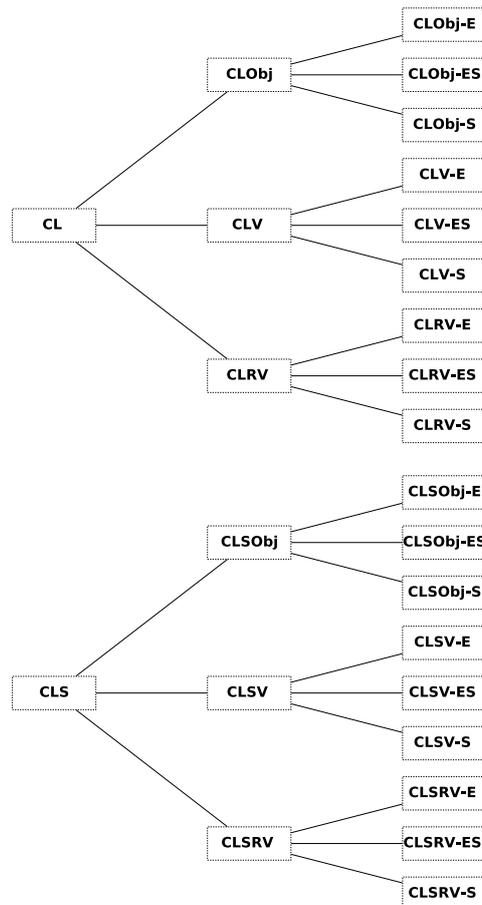


FIGURE 3.4 Liste des différents types de composantes logiques.

Nous listons l'ensemble des types de composantes logiques sous une forme hiérarchique dans la figure 3.4. Cette hiérarchie est constituée de trois niveaux qui représentent les trois niveaux de séparation des composantes logiques : spatialisation ou non, type et fonctionnalité.

3.3 Conception par composantes logiques

Ce modèle ne clarifie pas en soit la conception. Dans cette section nous présentons d'abord les étapes de la conception par composantes logiques ainsi que le passage de la conception par *mapping* à la conception par composantes logiques. Nous définissons ensuite des règles que les solutions possibles devraient respecter, éliminant des solutions indésirables, dans le but de faciliter la conception.

3.3.1 Définition

La conception par composantes logiques connecte les dispositifs physiques au contrôleur du dialogue. Dans ce but, cette conception intervient sur les trois couches du modèle physique/numérique ainsi que les interconnexions existant entre elles. Elle se décompose en cinq étapes :

- définition des dispositifs d'entrée et de sortie ;
- définition des éléments du contrôleur de dialogue ;
- choix du type des composantes logiques (spatiales ou non) ;
- définition des composantes logiques liées aux vues (i.e. de type CLV, CLRV, CLSV et/ou CLSRV) et leurs mises en relation avec des éléments du contrôleur de dialogue ;
- définition des composantes logiques pour les objets virtuels (i.e. de type CLObj et/ou CLSObj) ;
- validation des solutions et élimination de celles qui sont indésirables.

Le schéma général de la conception par composantes logiques est représenté dans la figure 3.5.

3.3.2 Du Mapping aux composantes logiques

Certaines étapes de la conception par composantes logiques sont liées à la conception par *mapping* et d'autres sont spécifiques à la conception par composantes logiques. Dans cette partie nous présentons les étapes de conception par composantes logiques et de leurs relations avec la conception par *mapping* décrite au chapitre 2.

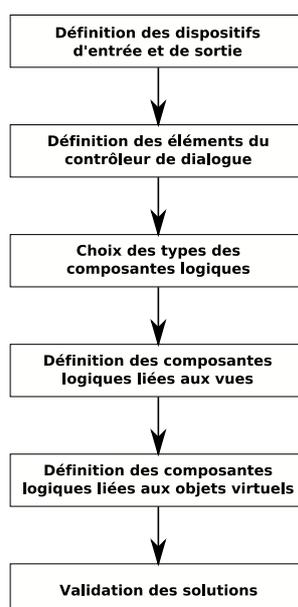


FIGURE 3.5 Les étapes de la conception par composantes logiques.

Définition des dispositifs d'entrée et de sortie. Les dispositifs d'entrée de la conception par composantes logiques sont identiques aux dispositifs d'entrée définis dans la conception par *mapping*. La définition des dispositifs de sortie est liée à la définition des vues et à la disposition des vues et des surfaces d'affichage. Chaque surface d'affichage nécessite un dispositif d'affichage de sortie (e.g. écran, projecteur, etc.). Ainsi, la première étape de la conception par composantes logiques définit les dispositifs d'entrée et les dispositifs de sortie de la même façon que la conception par *mapping*.

Définition des éléments du contrôleur de dialogue. Le rôle du *contrôleur de dialogue* est le séquençage des tâches, les modes (si le système en a) et la gestion de la cohérence des données en gérant les interactions en entrée et la présentation de l'information en sortie. Nous définissons le *contrôleur de dialogue* comme un ensemble d'éléments dont chacun intègre et administre une partie indépendante des données (e.g. un élément du contrôleur de dialogue peut gérer les données d'un objet virtuel). Nous différencions deux types d'éléments du contrôleur de dialogue : les éléments de type Contrôleur de Dialogue d'Initialisation (*CDI**) qui ne sont pas modifiables dynamiquement et ceux de type Contrôleur de Dialogue d'Initialisation/Tâche (*CDIT**) qui sont modifiables dynamiquement.

Les deux types d'éléments du contrôleur de dialogue sont nécessaires pour la gestion de la cohérence des données entre les différentes composantes logiques dans la phase d'initialisation. En plus de cette phase d'initialisation, la gestion de la cohérence des données doit être effectuée durant la réalisation des tâches ce qui est spécifique aux

éléments de type *CDIT*. La définition des éléments du contrôleur de dialogue de ce type s'effectue donc en fonction de la tâche à réaliser, qui est déterminée au niveau de la composante *tâche* de la conception par *mapping*.

Pour chaque objet virtuel et pour chaque vue un élément du contrôleur de dialogue doit être défini (de type *CDI* ou *CDIT*). Chaque élément du contrôleur de dialogue intègre les données géométriques correspondantes. Par exemple, un élément du contrôleur de dialogue représentant un objet virtuel dont la position, les dimensions et l'orientation horizontale sont nécessaires est illustré dans la figure 3.6.

Objet 3D					
Px	Py	Pz	Dx	Dy	Dz
			Rz		

FIGURE 3.6 Exemple d'un élément du contrôleur de dialogue correspondant à un objet virtuel.

Choix des types des composantes logiques. Le choix des types de composantes logiques (spatiales ou non) doit être réalisé selon le type de dispositif d'entrée et/ou de sortie auxquels elles sont connectées ainsi que la modalité d'affichage de l'information. Si un dispositif ne permet pas d'interagir avec des degrés de liberté (e.g. translation, rotation) dans le monde physique alors les composantes logiques connectées à ce dispositif sont des composantes logiques non spatiales. Par exemple, des composantes logiques connectées à un microphone ne sont pas des composantes logiques spatiales. Le type des composantes logiques connectées à des dispositifs de sortie est défini selon la modalité de présentation de l'information. Si la modalité de présentation intègre les dimensions géométriques alors les composantes logiques correspondantes sont spatiales. Par exemple une composante logique correspondant à un objet virtuel représenté géométriquement est une composante logique spatiale.

Définition des composantes logiques liées aux vues. La quatrième étape de la conception par composantes logiques est la définition des composantes logiques de vues et de représentation de vues. Les composantes logiques liées aux vues sont déterminées selon les dimensions d'affichage des vues qui sont définies durant la conception par *mapping*. Par exemple, dans les TEAMViz une composante logique pour la vue 2D projetée sur la table tactile doit être modélisée ainsi qu'une seconde composante logique pour la vue 3D présentée verticalement. La définition des composantes logiques de représentation de vues (e.g. une icône de caméra sur la vue 2D représentant le point de vue 3D dans un TEAMViz) est dépendante des dimensions de présentation de l'information des vues sur lesquelles elles sont affichées. Les composantes logiques spatiales intègrent les degrés de liberté numériques en entrée et les dimensions d'affichage en

sortie. Une transposition des degrés de liberté du monde physique aux degrés de liberté du monde numérique par rapport à la vue correspondante doit être réalisée. Les règles de définition et de conception des composantes logiques de vues et de représentation de vues sont exposées dans la section suivante 3.3.3.

Définition des composantes logiques liées aux objets virtuels. Cette étape consiste à définir les composantes logiques correspondant aux objets virtuels. La définition des composantes logiques liées aux objets virtuels dépend elle aussi des composantes logiques correspondant aux vues (surtout pour les composantes logiques de sortie et d'entrée/sortie). Par exemple, un objet virtuel présenté sur une vue 2D est modélisé par une composante logique spatiale intégrant ces deux dimensions. Ainsi que pour les composantes logiques spatiales de vues et de représentation de vues, une transposition entre les degrés de liberté physiques et numériques selon la vue correspondante est nécessaire pour les composantes logiques liées aux objets virtuels.

Validation des solutions. Un nombre important de solutions possibles est généré par les cinq étapes précédentes de la conception par composantes logiques. Certaines de ces solutions ne sont pas applicables ou posent des problèmes de conception. Le but de cette dernière étape de la conception par composantes logiques est de vérifier selon une liste de règles si les solutions sont valables et d'éliminer celles qui ne le sont pas. Dans la section suivante nous présentons les règles de conception par composantes logiques.

3.3.3 Règles de conception de composantes logiques spatiales

Le choix du nombre de composantes logiques nécessaires pour la conception et de leurs types (entrée, sortie ou entrée/sortie) doit être défini avec attention. En plus, la combinatoire des connexions reliant la couche *interaction physique* à la couche *interaction logique* d'une part et celles reliant la couche *interaction logique* et la couche *contrôleur de dialogue* de l'autre est souvent de taille importante. Nous définissons ici des règles pour la conception des composantes logiques et de leurs connexions dans le but de définir les composantes logiques adéquates, les connexions acceptables, et d'éliminer des solutions indésirables. Ces règles permettent d'écartier des erreurs et incohérences associées à la définition des composantes logiques tout en laissant une certaine flexibilité de conception surtout pour la conception des composantes logiques liées aux vues. Nous identifions deux types de règles : règles de génération des composantes logiques et de flexibilité des connexions d'une part et les règles de vérification d'autre part.

Afin de permettre une présentation formelle et synthétique des règles, nous définissons les notations utilisées dans les règles dans la table 3.1 et utilisons les opérateurs mathématiques standards. Nous construisons ces représentations pour expliciter les contraintes logiques correspondant à des expressions en langage naturel et ainsi faciliter la conception par composantes logiques aux futurs concepteurs. Certaines de ces règles sont simples mais il est nécessaire de les exprimer pour la complétude d'un système de génération et de filtrage d'interfaces implémenté sur un moteur de résolution de contraintes. Le moteur de résolution de contraintes sort cependant du cadre du sujet de cette thèse.

Règles de génération des composantes logiques et de flexibilité des connexions

Le but des règles de cette section est , d'une part, de définir les composantes logiques nécessaires pour la conception et, d'autre part, de définir la flexibilité des connexions possibles.

Règle CL1. Pour toute vue il existe une composante logique de vue de sortie ou d'entrée/sortie même si elle n'est pas interactive. Cette composante logique est connectée à un élément du contrôleur de dialogue de vue.

$$\forall v \in VUE, \exists l \in CL, \exists d \in CD, \\ (d, l) \in CON \wedge \text{fonc}(d) = V \wedge \text{fonc}(l) = V \wedge (\text{typ}(l) = S \vee \text{typ}(l) = ES)$$

Exemple : Deux points de vue distincts sur un environnement 3D affichés sur deux écrans distincts nécessitent deux composantes logiques différentes associées respectivement à chacune des vues.

Règle CL2. La représentation d'information d'une vue (v) sur une vue (autre ou la même) est modélisée par une composante logique de représentation de vue de sortie ou d'entrée/sortie connectée à l'élément du contrôleur de dialogue correspondant à la vue (v).

$$\forall l \in CL, \forall v \in VUE, \forall d \in CD, \\ (\text{fonc}(l) = RV \wedge \text{rep}(l) = v \wedge \text{rep}(d) = v \wedge (\text{typ}(l) = S \vee \text{typ}(l) = ES)) \Rightarrow (d, l) \in CON$$

Type	Notation	Correspondance
Ensembles	CD	Ensemble des éléments du contrôleur de dialogue
	CL	Ensemble des composantes logiques
	DE	Ensemble des dispositifs d'entrée
	DS	Ensemble des dispositifs de sortie
	CON	Ensemble des connexions reliant des composantes logiques à des dispositifs ou à des éléments du contrôleur de dialogue
	VUE	Ensemble des vues
	OV	Ensemble des objets virtuels
Ensembles de la conception par <i>mapping</i>	$LIAIM$	Ensemble des couples (dispositif d'entrée, dispositif de sortie) liés
	$TTIM$	Ensembles des techniques d'interaction du <i>mapping</i>
	$TACHEM$	Ensemble des tâches élémentaires du <i>mapping</i>
	$CONM$	Ensemble des connexions du <i>mapping</i>
Typages	CDI	Type initialisation de l'élément du contrôleur de dialogue
	$CDIT$	Type initialisation/tâche de l'élément du contrôleur de dialogue
	E	Type de composante logique d'entrée
	S	Type de composante logique de sortie
	ES	Type de composante logique d'entrée/sortie
	Obj	Fonctionnalité d'objets virtuels
	V	Fonctionnalité de vue
	RV	Fonctionnalité d'une représentation d'une vue
Typages de la conception par <i>mapping</i>	$SELECT$	Type sélection de la tâche élémentaire
Opérateurs	$typ(a)$	Renvoie le type de a
	$fonc(a)$	Renvoie la fonctionnalité de a
	$disp(v)$	Renvoie l'ensemble des dispositifs de sortie de la vue v
	$rep(a)$	Renvoie l'élément que représente a
	$pres(a)$	Renvoie la vue sur laquelle a est présentée

TABLE 3.1 Notations des éléments nécessaires pour les règles de conception par composantes logiques.

Exemple : Considérons une vue 2D affichée sur un écran sur laquelle une représentation de la position du point de vue d'une seconde vue 3D est présentée. Cette représentation est connectée à l'élément du contrôleur de dialogue qui correspond à la vue 3D.

Règle CL3. Il existe une composante logique de sortie ou d'entrée/sortie d'objets virtuels pour chaque représentation d'un objet virtuel sur chaque vue.

$$\begin{aligned} & \forall o \in OV, \forall v \in VUE, \forall s \in DS, (pres(o) = v \wedge s \in disp(v)) \\ \Rightarrow & (\exists l \in CL, rep(l) = o \wedge (l, s) \in CON \wedge (typ(l) = S \vee typ(l) = ES)) \end{aligned}$$

Exemple : Considérons une sphère virtuelle présentée sur deux vues 3D distinctes. Il doit exister deux composantes logiques pour représenter cette même sphère chacune gérant son affichage sur la vue correspondante.

Règle CL4. Toute composante logique connectée à un élément du contrôleur de dialogue non modifiable dynamiquement est une composante logique de sortie uniquement.

$$\forall l \in CL, \forall d \in CD, ((d, l) \in CON \wedge typ(d) = CDI) \Rightarrow typ(l) = S$$

Exemple : Une composante logique correspondant à un objet virtuel statique (dont les attributs ne sont pas modifiables dynamiquement) présenté au moyen d'une vue n'est pas connectée à un dispositif d'entrée et donc elle constitue une composante logique de sortie.

Règle CL5. Il peut exister une composante logique d'objet connectée à un élément du contrôleur de dialogue correspondant à une vue.

$$\neg(\forall d \in CD, (fonc(d) = V) \Rightarrow (\forall l \in CL, (l, d) \in CON, (fonc(l) = V \vee fonc(l) = RV)))$$

Exemple : Considérons une technique d'interaction permettant la navigation dans une vue en déplaçant un objet virtuel (e.g. une sphère virtuelle) de cette même vue. L'élément du contrôleur de dialogue correspondant à la vue est connecté à une composante logique de vue et une composante logique d'objets virtuels.

Règle CL5bis. Il peut exister une composante logique de vue ou de représentation de vue connectée à un élément du contrôleur de dialogue correspondant à un objet virtuel.

$$\neg(\forall d \in CD, (fonc(d) = Obj) \Rightarrow (\forall l \in CL, (l, d) \in CON, fonc(l) = Obj))$$

Exemple : Considérons une technique d'interaction qui positionne un objet virtuel durant la navigation dans une vue 3D. L'élément du contrôleur de dialogue correspondant à l'objet virtuel est connecté à une composante logique d'objets virtuels et une composante logique de vue.

Règles de vérification

Le nombre de connexions reliant les dispositifs aux composantes logiques et les composantes logiques aux éléments du contrôleur de dialogue est important. Ce nombre important de connexions possibles engendre un nombre de solutions encore plus conséquent. Cependant, toutes ces solutions ne sont pas valides. Nous définissons dans cette section des règles que les solutions doivent respecter afin d'éliminer celles qui sont invalides.

Règle CL6. Toute composante logique d'entrée est connectée à un dispositif d'entrée et à aucun dispositif de sortie.

$$\forall l \in CL, typ(l) = E \Rightarrow ((\exists e \in DE, (e, l) \in CON) \wedge (\forall s \in DS, (l, s) \notin CON))$$

Exemple : Une composante logique gérant les entrées d'un microphone est connectée au microphone et à aucun autre dispositif en sortie.

Règle CL6bis. Toute composante logique de sortie est connectée à un dispositif de sortie et à aucun dispositif d'entrée.

$$\forall l \in CL, typ(l) = S \Rightarrow ((\exists s \in DS, (l, s) \in CON) \wedge (\forall e \in DE, (e, l) \notin CON))$$

Exemple : Une composante logique gérant uniquement la présentation d'un objet virtuel sur une vue est connectée à un écran mais à aucun dispositif en entrée.

Règle CL6ter. Toute composante logique d'entrée/sortie est connectée à un dispositif d'entrée et à un dispositif de sortie.

$$\forall l \in CL, typ(l) = ES \Rightarrow ((\exists e \in DE, (e, l) \in CON) \wedge (\exists s \in DS, (l, s) \in CON))$$

Exemple : Une composante logique correspondant à un objet virtuel affiché sur une table tactile est connectée à la surface tactile en entrée et au projecteur en sortie.

Règle CL7. Tout dispositif d'entrée lié indirectement à aucun dispositif en sortie est connecté à au moins une composante logique d'entrée. Il est connecté uniquement à des composantes logiques d'entrée.

$$\forall e \in DE, \nexists s \in DS, (e, s) \notin LIAIM \Rightarrow ((\exists l_1 \in CL, (e, l_1) \in CON, typ(l_1) = E) \wedge (\forall l_2 \in CL, (e, l_2) \in CON \Rightarrow typ(l_2) = E))$$

Exemple : Un microphone est connecté à des composantes logiques d'entrée uniquement.

Règle CL8. Toute composante logique de vue correspondant à une vue est connectée au(x) dispositif(s) physique(s) de sortie affichant cette vue.

$$\forall l \in CL, \forall v \in VUE, \forall s \in DS, (s \in disp(v), rep(l) = v \Rightarrow (l, s) \in CON)$$

Exemple : Considérons une vue 2D présentée sur un écran. La composante logique de vue correspondant à cette vue 2D est connectée à l'écran.

Règle CL9. Une composante logique de représentation de vue de sortie ou d'entrée/sortie qui représente une vue (v_1) affichée sur une vue (v_2) est connectée au(x) dispositif(s) de sortie de la vue (v_2).

$$\forall l \in CL, \forall v_1, v_2 \in VUE, \forall s \in DS, (fonc(l) = RV \wedge rep(l) = v_1 \wedge pres(l) = v_2 \wedge s \in disp(v_2)) \Rightarrow (l, s) \in CON$$

Exemple : Considérons une vue 2D affichée sur un écran horizontal, une vue 3D présentée sur un écran vertical et qu'une représentation de la position du point de vue de la vue 3D est présentée sur la vue 2D. Cette représentation est alors connectée à l'écran horizontal.

Règle CL10. Tout élément du contrôleur de dialogue modifiable dynamiquement est connecté à au moins une composante logique d'entrée ou d'entrée/sortie.

$$\forall d \in CD, (typ(d) = CDIT) \Rightarrow (\exists l \in CL, (l, d) \in CON \wedge (typ(l) = E \vee typ(l) = ES))$$

Exemple : Un élément du contrôleur de dialogue représentant un objet virtuel dont la position est modifiable en utilisant la souris est connecté à une composante logique d'entrée/sortie reliée à la souris.

Règle CL11. Toute connexion reliant la *table des techniques d'interaction* et la composante *tâche* de la conception par *mapping* (sauf celles pour la sélection) correspond à une connexion entre une composante logique d'entrée ou d'entrée/sortie et un élément du contrôleur de dialogue.

$$\begin{aligned} \forall ti \in TTIM, \forall t \in TACHEM, (func(t) \neq SELECT \wedge (ti, t) \in CONM) \\ \Rightarrow (\exists l \in CL, \exists d \in CD, (l, d) \in CON \wedge (typ(l) = E \vee typ(l) = ES)) \end{aligned}$$

Exemple : Considérons qu'il existe une connexion reliant la translation X d'une *table des techniques d'interaction* à la translationX de la composante *tâche* du *mapping* correspondant à la translation de la souris pour le déplacement d'un objet virtuel. Cette connexion se transpose à la conception par composante logique par une connexion reliant une composante logique de l'objet virtuel à un élément du contrôleur de dialogue représentant l'objet virtuel. Rappelons que la conception par *mapping* est une conception à haut niveau tandis que la conception par composantes logiques se situe à un niveau proche de la conception logicielle.

Règle 12. Chaque élément du contrôleur de dialogue de vue est connecté à au moins une composante logique de vue ou de représentation de vue de sortie ou d'entrée/sortie.

$$\forall d \in CD, \text{fonc}(d) = V$$

$$\Rightarrow \exists l \in CL, (\text{fonc}(l) = V \vee \text{fonc}(l) = RV) \wedge (\text{typ}(l) = S \vee \text{typ}(l) = ES)$$

Exemple : Considérons une vue 3D fixe d'un environnement virtuel. L'élément du contrôleur de dialogue représentant cette vue est connecté à une composante logique de vue en sortie correspondant à cette vue.

3.4 Conception d'un simulateur de char d'assaut par composantes logiques

Pour illustrer les différentes étapes de conception par composantes logiques, nous proposons d'étudier la conception d'un simulateur d'un char d'assaut (tank). La modélisation de ce simulateur montre la mise en œuvre des règles pour générer l'espace des combinatoires possibles par la conception des composantes logiques puis la validation de certaines des solutions obtenues par des règles de vérification.

Le simulateur que nous proposons de modéliser présente deux points de vue : le premier point de vue est celui du conducteur du tank affiché sur un écran et le second point de vue est celui de la personne située dans la tourelle affiché sur un écran distant (voir figure 3.7). Nous considérons que les interactions sont réalisées uniquement en utilisant une souris liée à la vue du conducteur du tank. Une glissière et un potentiomètre numérique rotatif sont présentés sur la vue du conducteur permettant la modification de l'orientation de la tourelle. Nous illustrons uniquement la conception par composantes logiques d'une telle application. Le nombre d'utilisateurs n'est donc pas réellement important.

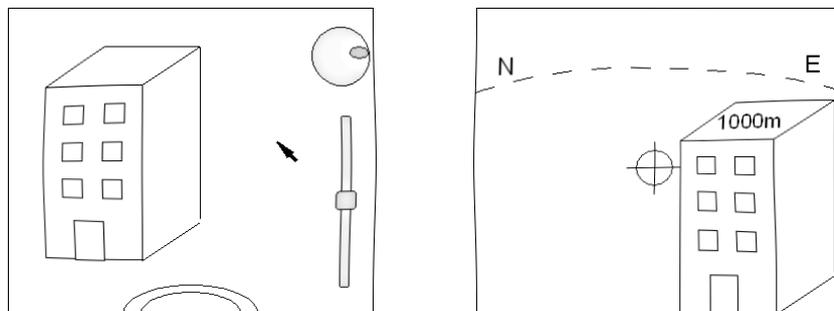


FIGURE 3.7 Sketch des vues du simulateur de tank. Gauche : vue du conducteur. Droite : vue de la tourelle.

Définition des dispositifs d'entrée et de sortie. Les dispositifs physiques utilisés pour ce simulateur sont deux écrans comme dispositifs de sorties et une *Souris* comme dispositif d'entrée. Nous appelons l'écran affichant le point de vue du conducteur *EcranCond* et le second écran *EcranTour* pour les différencier. L'ensemble de ces dispositifs physiques est représenté dans la figure 3.8. Ces dispositifs physiques se situent dans la couche *interaction physique* du modèle physique/numérique.

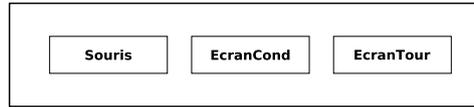


FIGURE 3.8 Dispositifs physiques utilisés pour le simulateur de tank.

Définition des éléments du contrôleur de dialogue. Nous considérons que la scène virtuelle est composée de bâtiments virtuels pour simplifier l'exemple. Les données numériques des bâtiments sont leurs positions et leurs dimensions qui sont statiques. Les tâches que nous imaginons pour le simulateur sont :

- les translations horizontales dans l'axe de vue du conducteur du corps et de la tourelle du tank ;
- la rotation horizontale du corps du tank ;
- les rotations horizontale et verticale de la tourelle.

En tenant compte qu'une vue affiche le point de vue du conducteur, que la seconde affiche le point de vue de la tourelle et qu'elles sont dynamiquement modifiables, un élément du contrôleur de dialogue de type *CDIT* est créé pour chacune des vues. Les éléments du contrôleur de dialogue correspondants aux bâtiments virtuels statiques sont de type *CDI*. Nous appelons l'élément du contrôleur de dialogue représentant la vue du conducteur *CDCond*, celle de la vue de la tourelle *CDTour* et ceux des bâtiments *CDBat_i*. L'ensemble des éléments du contrôleur de dialogue est illustré dans la figure 3.9.

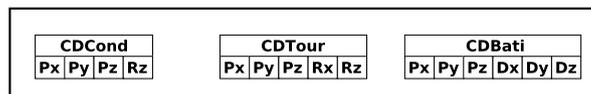


FIGURE 3.9 Éléments du contrôleur de dialogue utilisés pour le simulateur de tank.

Choix des types des composantes logiques. Le choix du type de composantes logiques spatiales est déterminé par l'importance des dimensions et des degrés de liberté pour ce simulateur de tank. De plus, les tâches élémentaires que nous avons décrites

s'articulent autour des degrés de liberté de navigation. Nous choisissons donc les composantes logiques spatiales.

Définition des composantes logiques liées aux vues. En appliquant la règle CL1, nous concevons deux composantes logiques de vue correspondant à chacune des vues. Nous appelons la composante logique correspondant à la vue du conducteur *CLSVCond* et celle correspondant à la vue de la tourelle *CLSVTour*. *CLSVCond* est connectée à *CDCCond* tandis que *CLSVTour* est reliée à *CDTour*. Les connexions sont réalisées selon les dimensions de présentation de l'information ainsi que les degrés de liberté en entrée.

CDCCond et *CDTour* sont deux éléments du contrôleur de dialogue modifiables dynamiquement. A ce stade de la conception, le choix entre composantes logiques de sortie ou d'entrée/sortie ne peut être réalisé pour *CLSVCond* et *CLSVTour*. La figure 3.10 illustre les deux types de composantes logiques possibles pour *CLSVTour*.

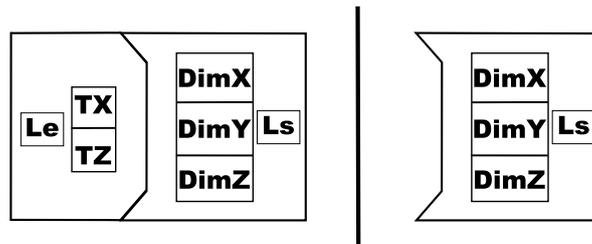


FIGURE 3.10 Types possibles pour la composante logique de la vue de la tourelle. Gauche : composante logique d'entrée/sortie. Droite : composante logique de sortie.

La glissière (pour la rotation verticale partielle de la tourelle) et le potentiomètre numérique circulaire (pour la rotation horizontale de la tourelle) peuvent être définies par des composantes logiques d'objets virtuels. Cependant, elles présentent des informations concernant l'orientation de la tourelle sur la vue du conducteur. La règle CL2 impose de définir deux composantes logiques de représentation de vue, une pour la glissière (nommée *CLSRVGlis*) et l'autre pour le potentiomètre (nommée *CLSRVPote*). Ces deux composantes logiques sont connectées au *CDTour* de la couche *contrôleur de dialogue*. L'ensemble des composantes logiques des vues ainsi que les différentes connexions sont illustrées de manière simplifiée dans la figure 3.11.

Définition des composantes logiques liées aux objets virtuels. Puisque les bâtiments virtuels restent statiques dans le repère absolu du monde virtuel, la règle CL4 impose que les composantes logiques liées aux bâtiments soient des composantes

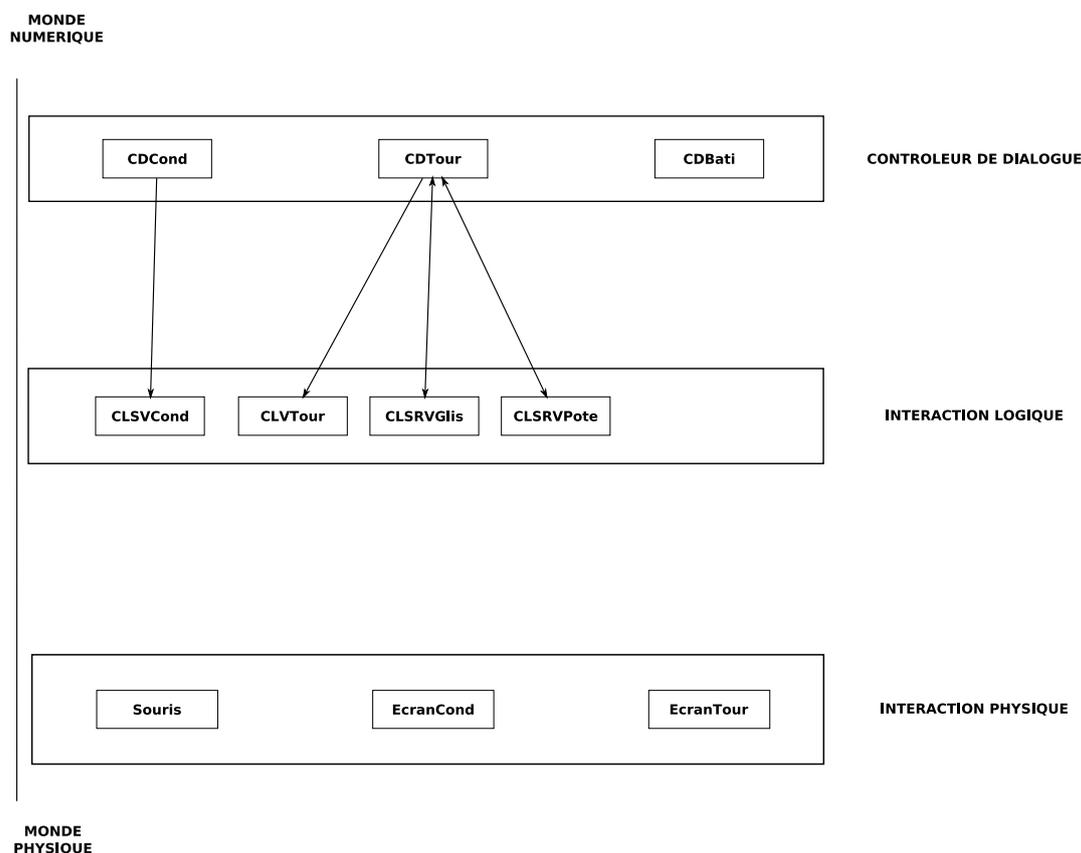


FIGURE 3.11 Définition des composantes logiques liées aux vues pour le simulateur de tank.

logiques de sortie. La règle CL3 impose qu'une composante logique représentant un bâtiment soit définie pour chacune des deux vues. Chacune de ces composantes logiques est connectée à $CDBat_i$ et au dispositif physique de sortie affichant la vue correspondante. Nous appelons la composante logique gérant l'affichage d'un bâtiment virtuel sur la vue du conducteur $CLSOBJCond$ et celle affichant ce bâtiment dans la vue de la tourelle $CLSOBJTour$. La figure 3.12 illustre la totalité des composantes logiques et leurs connexions déjà définies de manière simplifiée.

Validation des solutions. Le choix du type de $CLSVCond$ et $CLSVTour$ entre sortie et entrée/sortie ainsi que le nombre de connexions possibles reliant des composantes logiques à des dispositifs physiques et des éléments du contrôleur de dialogue génère un nombre important de solutions. Nous choisissons quatre solutions (appelée $Sol1$, $Sol2$, $Sol3$ et $Sol4$) de l'ensemble de ces solutions et étudions leurs validités.

Dans la solution $Sol1$, illustrée de façon simplifiée dans la figure 3.13, $CLSVCond$ est connectée à $EcranTour$ et $CLSVTour$ est connectée à $EcranCond$. Cette solution respecte toutes les règles sauf la règle CL8. En effet, $CLSVCond$ n'est pas reliée au dispositif de sortie qui présente la vue de cette composante logique. De même, $CLSVTour$

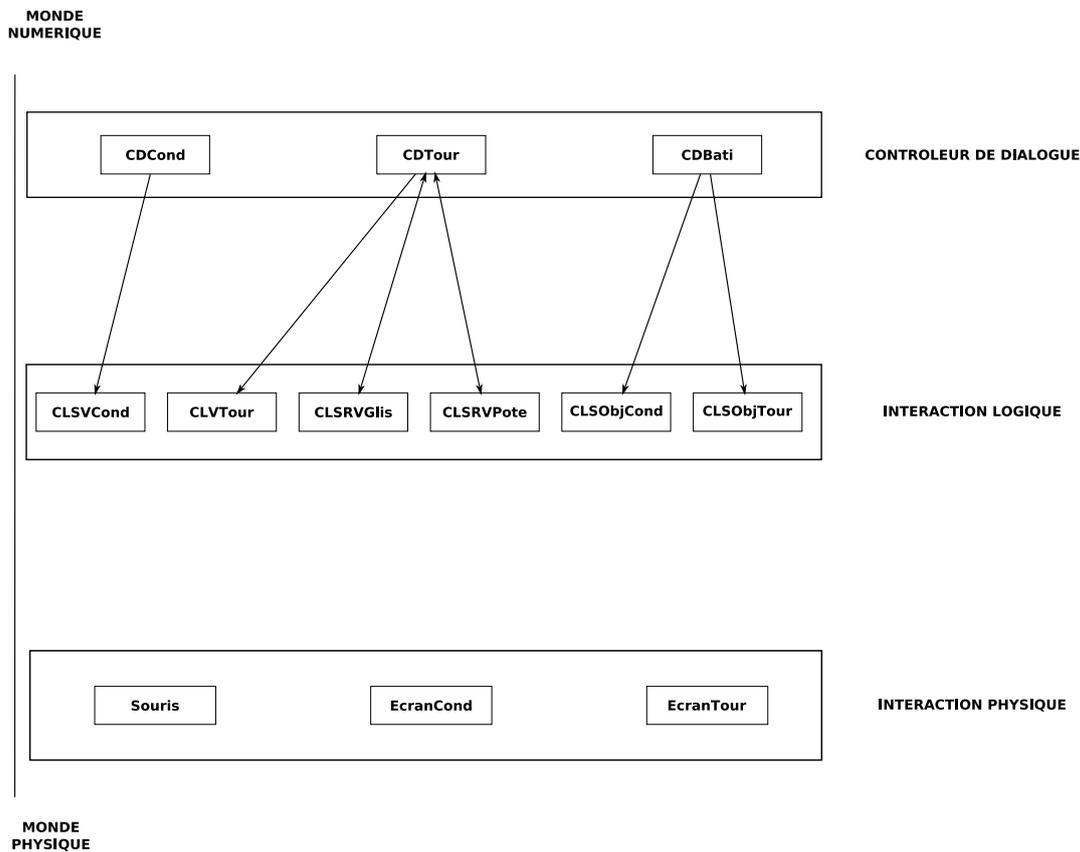


FIGURE 3.12 Ensemble des composantes logiques pour le simulateur de tank.

n'est pas connectée à *EcranTour* qui présente la vue de la tourelle. Ainsi cette solution n'est pas valide.

CLSVTour et *CLSVCond* sont adéquatement connectées aux dispositifs physiques dans la solution *Sol2* (voir figure 3.14). Dans cette solution, *CLSVTour* est une composante logique d'entrée/sortie reliée dans les deux sens au *CDTour*. Cependant, *Sol2* ne respecte pas la règle CL6ter du fait que *CLSVTour* est connectée à aucun dispositif en entrée. Ainsi *Sol2* n'est pas une solution valable.

Dans les solutions *Sol3* (voir figure 3.15) et *Sol4* (voir figure 3.16), *CLSVTour* est une composante logique de sortie uniquement. Ces deux solutions respectent les règles CL6, CL6bis et CL6ter du fait que :

- *CLSVCond*, *CLSRVGlis* et *CLSRVPote* sont des composantes logiques d'entrée/sorties connectées à la *Souris* en entrée et à *EcranCond* en sortie ;
- *CLSVTour*, *CLSOBJCond* et *CLSOBJTour* sont des composantes logiques de sorties reliées uniquement à des dispositifs en sortie uniquement.

Puisque chaque dispositif d'entrée est relié à une vue alors la règle CL7 est satisfaite. La règle CL8 est respectée puisque *CLSVTour* (respectivement *CLSVCond*) est connectée au dispositif *EcranTour* (respectivement *EcranCond*) qui présente la vue de la tourelle

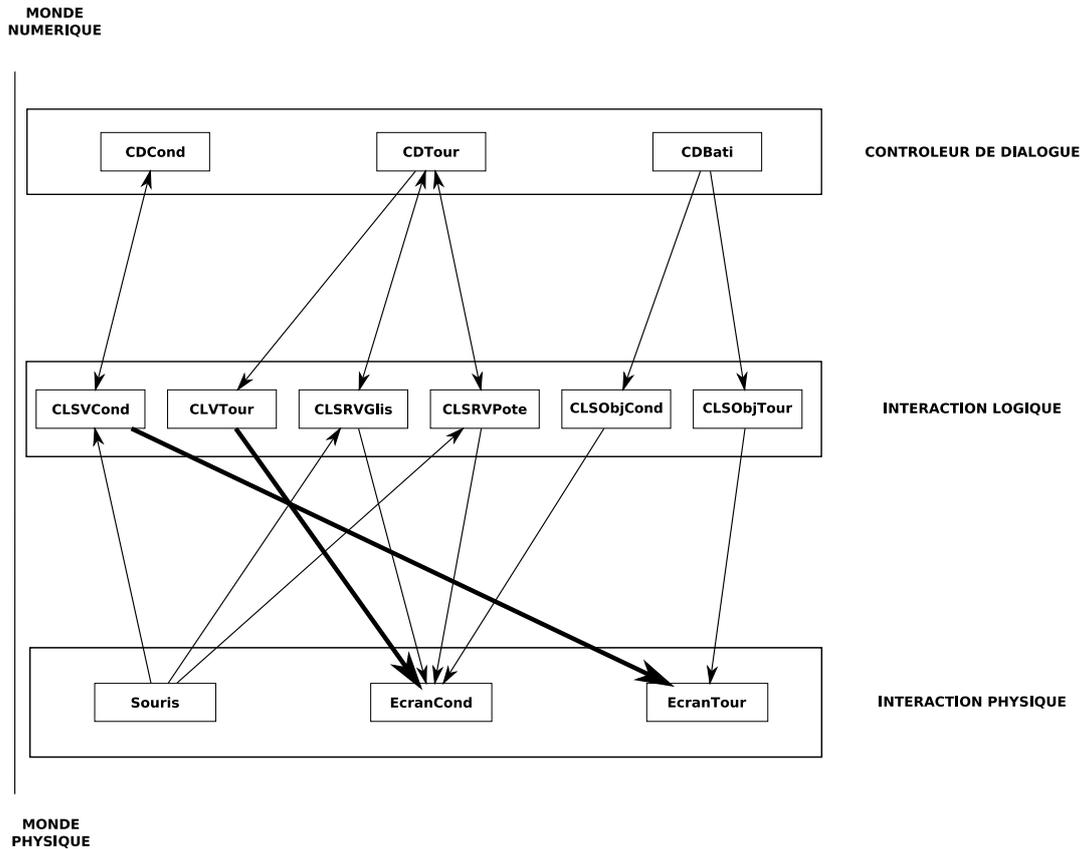


FIGURE 3.13 Composantes logiques et leurs connexions pour la solution *Sol1*.

(respectivement de la tourelle). La règle CL9 est aussi respectée du fait que *CLSRVGlis* et *CLSRVPote* sont reliées au dispositif *EcranCond* sur lequel elles sont présentées. Du fait que *CLSVCond* est connectée au *CDCCond* et que *CLSRVGlis* est reliée au *CDTour*, les règles 10 et 12 sont satisfaites.

L'unique différence entre les solutions *Sol3* et *Sol4* est la connexion reliant la translation selon l'axe X de *CLSVCond* à la rotation autour de l'axe Z de *CDTour*. En effet, *Sol3* correspond au cas dans lequel la rotation selon l'axe Z du corps du tank n'affecte pas la rotation selon l'axe Z de la tourelle contrairement à la solution *Sol4*. Cette différence doit exister au niveau de la conception par *mapping* et plus précisément dans les connexions entre la *table des techniques d'interaction* et la composante *tâche*. Deux solutions de la conception par *mapping* correspondent chacune à une des solutions *Sol3* et *Sol4* de la conception par composantes logiques. Ainsi, *Sol3* ou *Sol4* respectent la règle CL11 selon la solution par *mapping* retenue.

La modélisation des systèmes par composantes logiques génère un nombre important de combinaisons possibles qui doivent être validées, selon une liste de règles, dans cette dernière étape de conception. Quelques unes de ces combinaisons peuvent ne pas correspondre à des solutions obtenues à l'aide de la conception par *mapping* effectuée

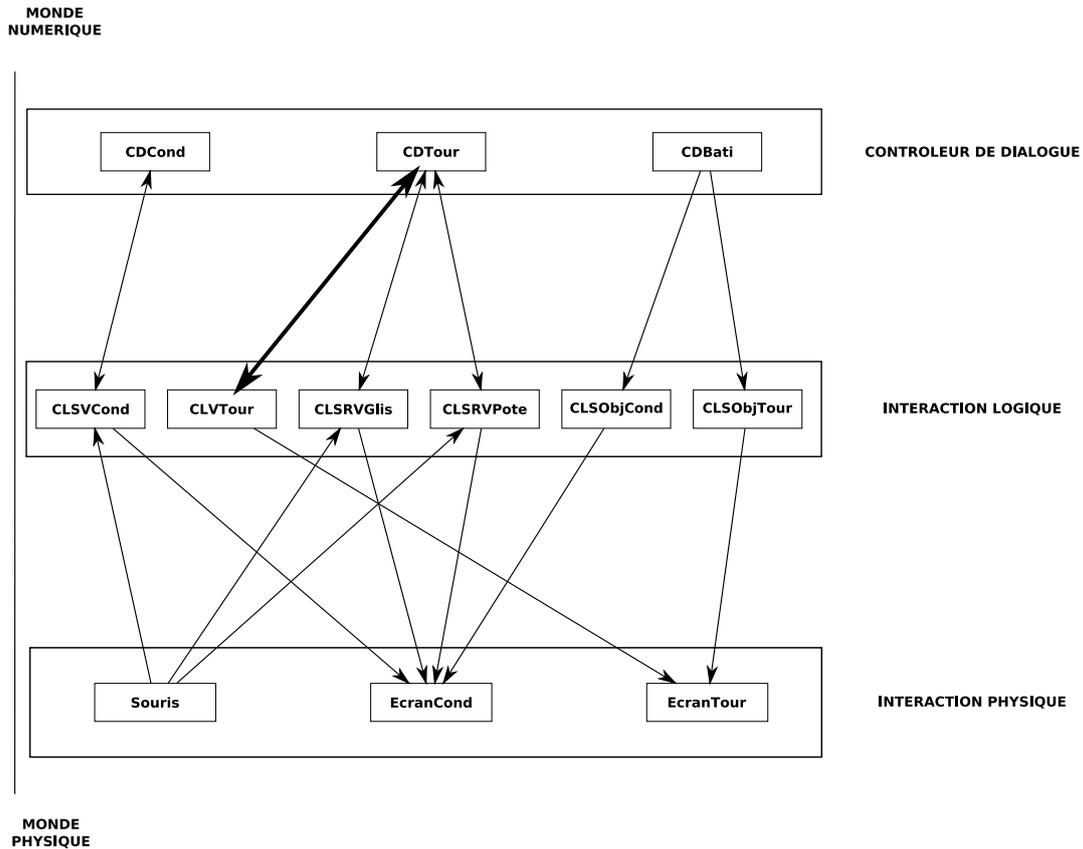


FIGURE 3.14 Composantes logiques et leurs connexions pour la solution Sol2.

auparavant (i.e. la règle CL11 insatisfaite). Cependant, elles peuvent se révéler intéressantes et dans ce cas, un retour à la conception par *mapping* est nécessaire pour générer les nouvelles solutions correspondantes.

3.5 Conclusion

La conception par composantes logiques permet une modélisation de systèmes combinant des vues multiples, des surfaces d'affichage multiples et des dispositifs en entrée multiples à un niveau de conception très proche de la conception logicielle. Elle intègre la mise en relation des interactions réalisées en entrée avec la présentation de l'information pour la manipulation d'un objet virtuel ainsi que pour la navigation. La modélisation par composantes logiques offre une certaine flexibilité permettant la conception d'interactions inter-dispositifs tout en respectant un certain nombre de règles. Elle offre la possibilité de modéliser des interactions "inter-tâches" comme le déplacement d'un objet d'une scène virtuelle en naviguant dans cette scène. Elle permet également de concevoir des interactions concurrentes pour la réalisation d'une tâche. Néanmoins, elle ne conçoit pas la gestion de la concurrence entre les différentes interactions en entrée.

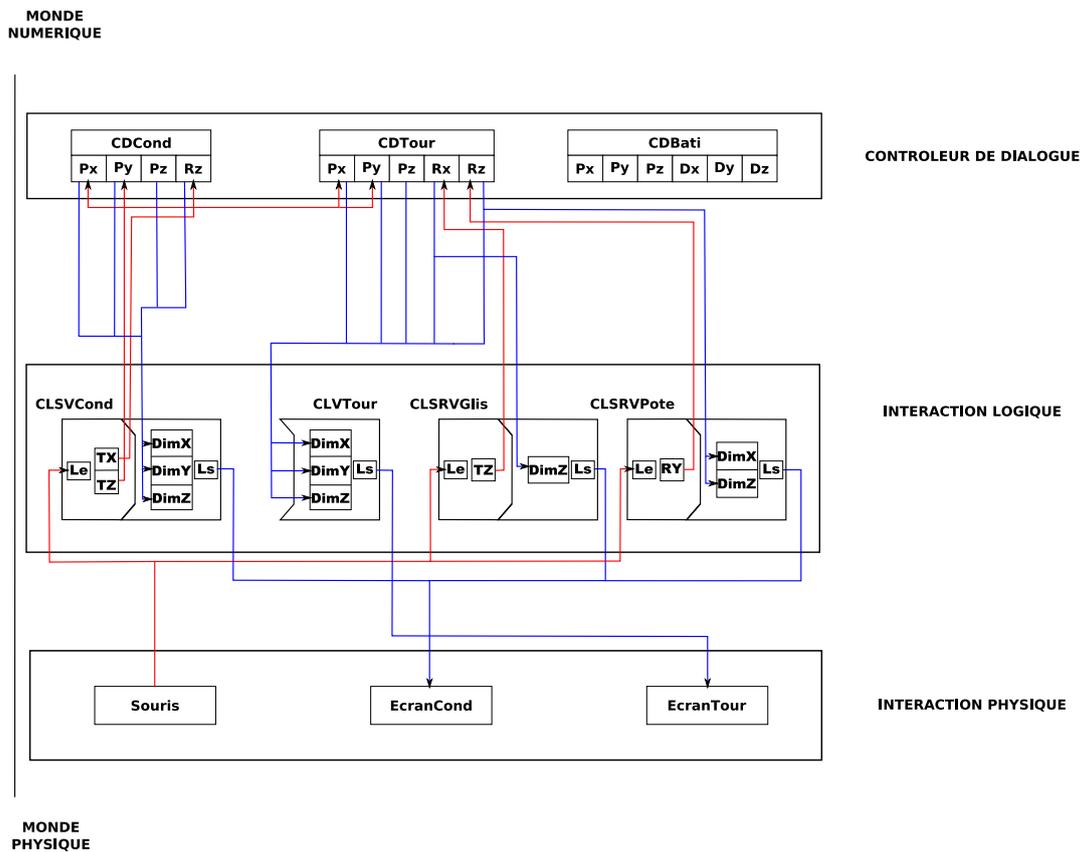


FIGURE 3.15 Composantes logiques et leurs connexions pour la solution Sol3.

Nous proposons une modélisation de la gestion de concurrence en utilisant des machines à états dans le chapitre suivant.

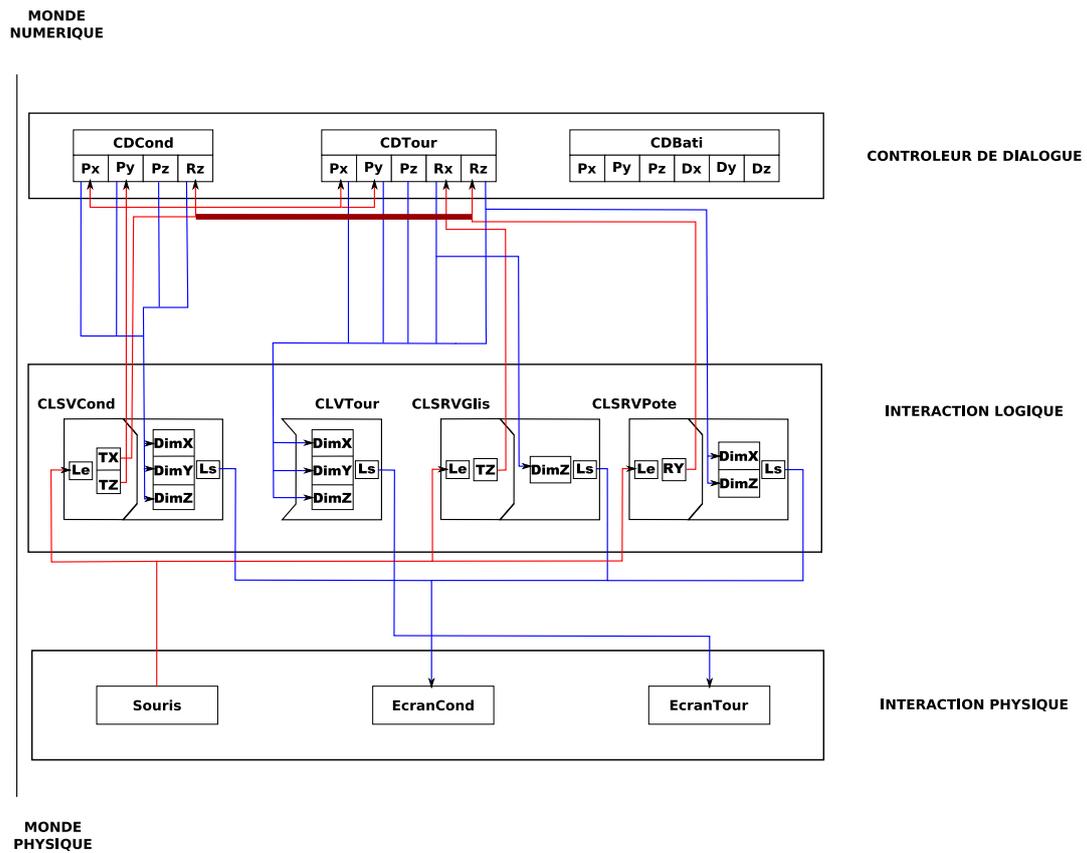


FIGURE 3.16 Composantes logiques et leurs connexions pour la solution Sol4.

Chapitre 4

Contribution à la gestion de la concurrence par un modèle de machines à états

4.1 Introduction

La combinaison d'interactions effectuées au moyen de deux vues distinctes d'un même monde géométrique (qui existe dans les TEAMViz) requiert une gestion de la concurrence et de la cohérence des données. Dans ce chapitre nous étudions dans un premier temps les problèmes liés à ce type de combinaison d'interaction ainsi que quelques systèmes de gestion de la cohérence des données. Ensuite nous présentons une architecture logicielle centralisée pour la gestion de la concurrence des interactions en entrée et de la cohérence des données. L'utilisation des machines à états pour la gestion de la concurrence et leur relation avec les composantes logiques sont décrites par la suite. Finalement nous exposons une expérimentation comparant deux moyens d'utiliser les machines à états pour la gestion de la concurrence dans un système de type TEAMViz.

4.2 Enjeux

La combinaison des interactions réalisées simultanément en utilisant des vues distinctes (mais présentant une même information) exige d'une part la gestion de la présentation de l'information en sortie et d'autre part la gestion des interactions en entrée. La présentation de l'information sur des vues multiples nécessite la gestion de la cohérence des données affichées sur ces vues. Par exemple, si une vue présente une maison virtuelle et une autre vue affiche cette même maison semi-transparente enrichie par les

canalisations d'eau alors les formes géométriques des deux représentations de la maison doivent être identiques. En plus, une transposition de façon synchronisée de toute modification d'une information sur une vue à toutes les vues présentant cette information est nécessaire. La gestion des interactions en entrée exige la gestion de la concurrence des interactions. En effet, considérons une chaise dans une maison virtuelle présentée de manière identique sur deux vues. La modification simultanée de la position de cette chaise dans ces deux vues dans des sens opposés entraînerait une incohérence des données affichées suite à l'interaction.

Pour illustrer les problèmes d'interactions concurrentes multi-vues, nous réalisons des tests élémentaires¹ sur un serveur, nommé VERSE [Verse] (serveur et protocole de communication sous licence libre développé par Eskil Steenberg et Emil Brink), dédié à la synchronisation d'un modèle 3D édité par plusieurs modeleurs 3D distincts. Ce serveur offre la possibilité de synchroniser des éditeurs de modèles 3D de types différents mais nous utilisons deux instances distinctes d'un même modeleur appelé Blender [Blender] (logiciel libre développé principalement par Blender Foundation et Blender Institute dirigé par Ton Roosendaal). Dans nos tests, les deux instances de Blender, que nous nommons B_1 et B_2 , sont lancées sur des machines distinctes et sont connectées au même serveur VERSE. Chacune de ces instances est reliée à un écran, un clavier et une souris. A l'initialisation, un cube virtuel est exporté de B_1 vers le serveur puis B_2 est synchronisée avec le serveur. Cette synchronisation consiste à acquérir une copie des données géométriques présentes dans le serveur. Ainsi les données sur le serveur, sur B_1 et sur B_2 sont identiques et cohérentes. Les éditions réalisées dans B_1 sur le cube virtuel sont immédiatement transposées dans B_2 permettant une gestion de la cohérence des données entre les différents éléments de cette architecture logicielle. L'édition simultanée du cube virtuel au moyen de B_1 et B_2 est possible (permettant une plus grande flexibilité) mais présente des problèmes de concurrence. En effet, l'édition simultanée d'un même vertex (i.e. sommet en 3D d'une forme géométrique) dans B_1 et B_2 montre que chacune des deux instances affiche la position du vertex selon les interactions réalisées au moyen de cette instance du modeleur. Le couplage entrée/sortie est ainsi préservé mais la cohérence des données n'est plus valide entre les deux instances car chacune affiche une position différente du vertex. Ces tests illustrent l'utilité d'une architecture logicielle centralisée pour la gestion de la cohérence des données entre différentes applications graphiques partageant les mêmes données. Cependant, l'utilisation du serveur VERSE et des instances de Blender présente des problèmes de gestion de la concurrence

1. Tests réalisés durant une visite d'un mois (en septembre 2008) financée par le projet européen INTUITION au Department of Media Technology de l'université technologique de Helsinki (TKK). Ce travail a été effectué en collaboration avec Mikko Multimäki et Tapio Takala.

et de la cohérence des données. La gestion de la concurrence doit être effectuée au niveau de l'application centrale ainsi qu'au niveau des applications graphiques (i.e. en bloquant les interactions impossibles à un instant donné à cause d'une modification des données par un autre composant).

Dans la cas d'un collecticiel distant (e.g. Google Docs [Google Docs]), une synchronisation automatique et très fréquente suffit dans une très grande majorité des cas. Dans le cadre d'un collecticiel co-localisé et donc fortement concurrent, la fréquence des collisions est nécessairement plus grande. Nos observations conduisent à ne pas retenir cette solution.

Plusieurs travaux de recherche étudient la gestion de la cohérence des données dans des systèmes intégrant plusieurs applications. Thompson *et al.* [2003] présentent un système centralisé permettant la synchronisation (ou comme les auteurs l'appellent l'*orchestration*) des interactions réalisées en utilisant des dispositifs multiples. Dans un autre travail de recherche, Greenhalgh *et al.* [2000] décrivent la gestion de la cohérence des données d'un système d'environnement virtuel collaboratif appelé Massive-3. L'élément intéressant de cette gestion de la cohérence des données est la gestion de la concurrence par la notion de permission. Chaque application souhaitant modifier une donnée doit avoir la permission sur ces données. Les auteurs présentent trois approches pour la gestion du système de permission (voir figure 4.1). La première approche consiste à envoyer une demande de permission de la part d'une application client désirant cette permission à une application centrale qui la lui renvoie. Ensuite le client effectue les modifications des données et renvoie à l'application centrale ces modifications qui sont retransmises à toutes les applications utilisant ces données. Dans la deuxième approche, une demande de modification de données est envoyée du client vers l'application centrale qui met à jour sa version des données et renvoie les modifications à toutes les applications utilisant ces données (dont le client). Le client met à jour ensuite sa propre version des données. La dernière approche consiste à transmettre une demande de permission ainsi qu'une demande de modification des données lors du premier envoi à l'application centrale. Ensuite l'application centrale renvoie les permissions et informe toutes les applications des modifications réalisées. Enfin, le client modifie sa version des données lorsqu'il possède la permission et renvoie ces modifications à l'application centrale qui à son tour renvoie ces données aux applications utilisant ces données. L'architecture logicielle centralisée et la gestion de la concurrence utilisant un système de permission nous ont fortement inspiré dans notre conception logicielle.

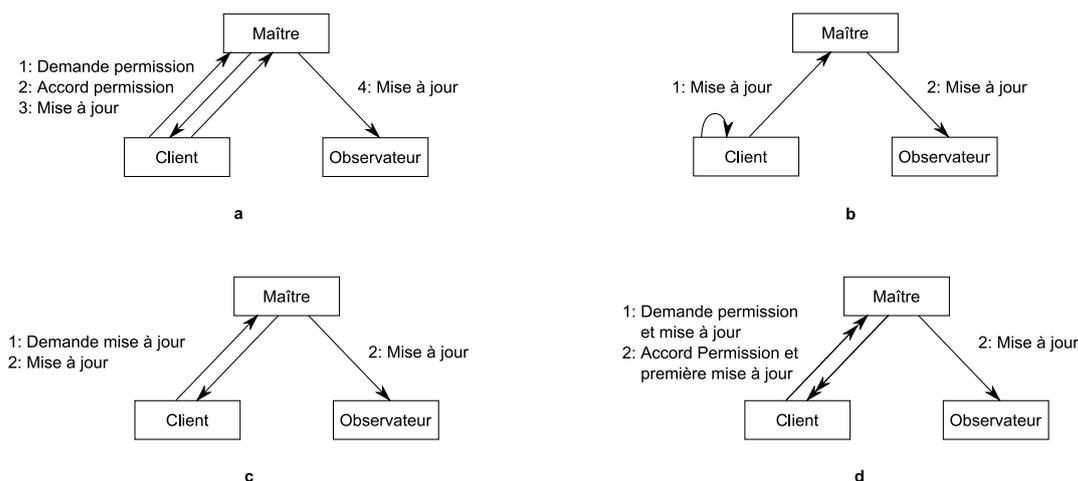


FIGURE 4.1 Approches de gestion des permissions [Greenhalgh *et al.*, 2000]. a-b : demande de permission. c : demande de mise à jour. d : demande de permission et mise à jour.

4.3 Architecture logicielle

Nous proposons une architecture logicielle centralisée pour faciliter la gestion de la cohérence des données. Les objectifs de l'application centrale sont multiples :

- gestion d'une version cohérente des données ;
- gestion de la concurrence des interactions sur les différents dispositifs en entrée ;
- gestion de la cohérence de l'affichage de l'information sur les différentes vues.

La conception par composants logiques permet de bien déterminer les différentes composantes à intégrer au niveau de l'application centrale. En effet, la seule composante connectée aux différentes composantes logiques est le contrôleur de dialogue. Ainsi, tout élément du contrôleur de dialogue défini durant la conception par composants logiques (voir chapitre 3) est nécessairement intégré au niveau de l'application centrale.

Les applications que nous appelons applications d'IHM sont reliées aux dispositifs d'entrée et/ou de sortie et à l'application centrale. Une application IHM regroupe les composantes logiques d'entrée/sortie (voir chapitre 3) connectées aux mêmes dispositifs d'entrée et de sortie. Si d'autres composantes logiques sont reliées aux mêmes dispositifs de sortie, elles sont alors intégrées dans la même application logicielle. Si un dispositif est connecté à une (des) composante(s) logique(s) d'entrée uniquement alors cette composante logique peut être intégrée au niveau de l'application centrale ou au niveau d'une nouvelle application. Les composantes logiques liées à un même dispositif de sortie déconnecté de toute composante logique d'entrée/sortie sont regroupées dans une application logicielle. La figure 4.2 illustre l'architecture logicielle centralisée composée d'une application centrale contenant les éléments du contrôleur de dialogue et de deux applications IHM contenant les composantes logiques.

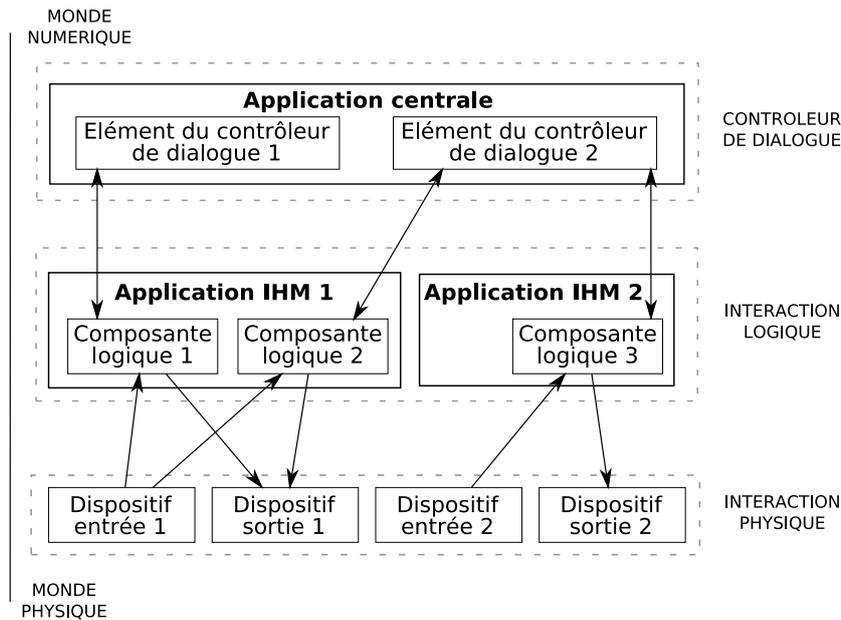


FIGURE 4.2 Architecture logicielle centralisée avec l'intégration des éléments du contrôleur de dialogue dans l'application centrale et des composants logiques dans les applications IHM.

Dans le but de gérer la cohérence des données, l'application centrale contient la version cohérente des données. Une copie des données présentes au niveau de l'application centrale est transmise aux différentes applications (intégrant des composants logiques de sortie et/ou d'entrée/sortie) durant une phase d'initialisation. Cette copie des données au niveau des applications d'IHM permet un couplage fort d'entrée/sortie entre certains dispositifs d'entrée et de sortie. La gestion de la concurrence est réalisée par un système de permissions décrit dans la section suivante.

Nous avons développé un système de type TEAMViz que nous appelons *MC** (Management Cabin ou en français cabine de commande). Il combine donc des interactions réalisées au moyen d'une vue 2D d'un environnement virtuel présentée sur une table tactile et d'une vue 3D complémentaire du même environnement virtuel. L'architecture logicielle de ce système est composée de trois composantes principales (voir figure 4.3) :

- une composante IHM 2D connectée à une table tactile et à un vidéo projecteur. Elle gère les interactions en entrée et la présentation de l'information de la vue 2D ;
- une composante IHM 3D connectée à des dispositifs d'entrée liés à la vue 3D et à un vidéo projecteur. Elle gère les interactions en entrée et la présentation de l'information de la vue 3D ;
- une composante centrale qui gère la cohérence des données. Cette application gère la communication entre les deux autres composantes IHM.

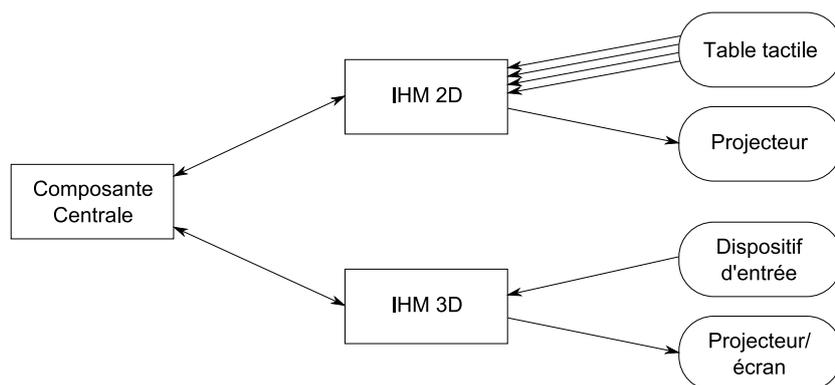


FIGURE 4.3 Architecture logicielle de MC et ses connexions avec les dispositifs physiques d'entrée et de sortie.

L'intérêt de MC par rapport à l'utilisation d'un serveur VERSE et des instances de Blender est l'intégration facile des interactions d'une table tactile. Néanmoins, les tests réalisés avec le serveur VERSE et les deux instances de Blender ont permis d'identifier les problèmes essentiels à gérer pour le développement de MC dont la gestion de la concurrence des interactions et de la cohérence des données.

4.4 Gestion de la concurrence par machines à états

L'architecture logicielle centralisée et la séparation des différentes interfaces utilisateurs permet de réaliser un système basé sur des permissions pour gérer la concurrence des interactions en entrée et la cohérence des données. En s'inspirant des trois approches de Greenhalgh *et al.* [2000], nous avons développé les trois types de gestion des mises à jour et des permissions suivants : gestion par demande de modifications des données, gestion par demande de permission et gestion par demande de permission et mises à jour. Pour réaliser la gestion des permissions et des mises à jour nous nous sommes inspiré des machines à états étudiées dans le domaine de l'IHM pour la modélisation des interactions [Appert & Beaudouin-Lafon, 2006, Blanch & Beaudouin-Lafon, 2006]. Nous proposons l'utilisation des machines à états sur deux niveaux :

- le premier niveau se situe dans les composantes IHM de l'architecture logicielle et plus précisément dans chaque composante logique d'entrée/sortie de la conception par composantes logiques (voir chapitre 3) ;
- le second niveau se situe dans la composante logicielle centrale (couche contrôleur de dialogue du modèle physique/numérique) et plus précisément dans les éléments du contrôleur de dialogue de la conception par composantes logiques (voir chapitre 3).

Dans les parties suivantes nous présentons les différentes approches du système de permission. Nous présentons ensuite les deux types de machines à états pour la gestion

des permissions et des mises à jour. Enfin, nous terminons cette section avec la présentation des différents niveaux de granularité possibles pour la gestion la concurrence.

4.4.1 Gestion de la concurrence par un système de permission

Chacune des approches de gestion de permission (gestion par demande de modifications des données, gestion par demande de permission et gestion par demande de permission et mises à jour) est spécifique à un type d'interfaces. Par exemple, une application connectée uniquement à un dispositif d'entrée ne contient pas une copie des données donc l'approche de demande de modification des données semble préférable.

L'approche par demande de permission et de mises à jour est appliquée dans le cas d'interfaces intégrant la notion de "fantômes". Un fantôme, dont la représentation graphique est semblable à celle des données que l'utilisateur souhaite modifier, est manipulable par l'utilisateur mais les données ne subissent les modifications qu'après la validation du système. Un exemple d'un fantôme est l'icône située sous le pointeur de la souris lorsqu'un utilisateur déplace un fichier d'un répertoire à un autre par manipulation directe. Au début de la manipulation d'un fantôme, l'application distante met à jour les informations concernant le fantôme et envoie une demande de permission et de mise à jour des données (voir le message 1 de la figure 4.4). L'application centrale après acceptation met à jour sa version des données et renvoie les informations concernant la modification des permissions et des données à toutes les applications dont la première application (voir messages 2 et 3 de la figure 4.4). La première application met à jour ensuite sa version des données. Enfin, toutes les autres applications mettent également à jour leurs versions des données et bloquent l'utilisation du fantôme représentant ces données s'il existe. La manipulation d'un fantôme dans une application est ainsi impossible lorsque les données qu'il représente sont manipulées dans une autre application.

Les interfaces connectées à des dispositifs d'entrée couplés à des dispositifs de sortie et n'intégrant pas des fantômes requièrent une approche à base de demande de permission. Lorsqu'un utilisateur sélectionne un objet virtuel dans une interface de ce type, l'application distante envoie un message de demande de permission à l'application centrale (voir message 1 de la figure 4.5) tout en bloquant la manipulation de l'objet virtuel par l'utilisateur. L'application centrale envoie ensuite un message d'information sur la modification des permissions aux différentes applications (voir messages 2 et 3 de la figure 4.5). Les autres applications peuvent ainsi présenter à l'utilisateur l'information sur l'interface avec laquelle est manipulé l'objet virtuel et bloquer le fantôme le représentant s'il existe. Lorsque la première application reçoit la permission, l'objet virtuel est débloqué et l'utilisateur peut ainsi le manipuler. Durant la manipulation, la copie

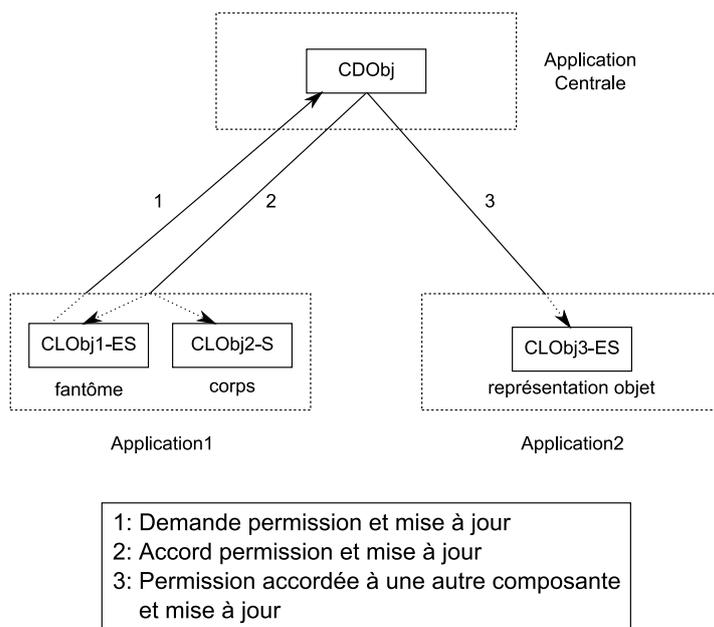


FIGURE 4.4 Gestion des permissions par demande de permission et mise à jour dans le cas d'un système intégrant des fantômes.

locale des données est modifiée et des messages de modification des données (voir message 4 de la figure 4.5) sont envoyés à l'application centrale qui à son tour modifie sa version des données et renvoie cette information aux autres applications (voir message 5 de la figure 4.5).

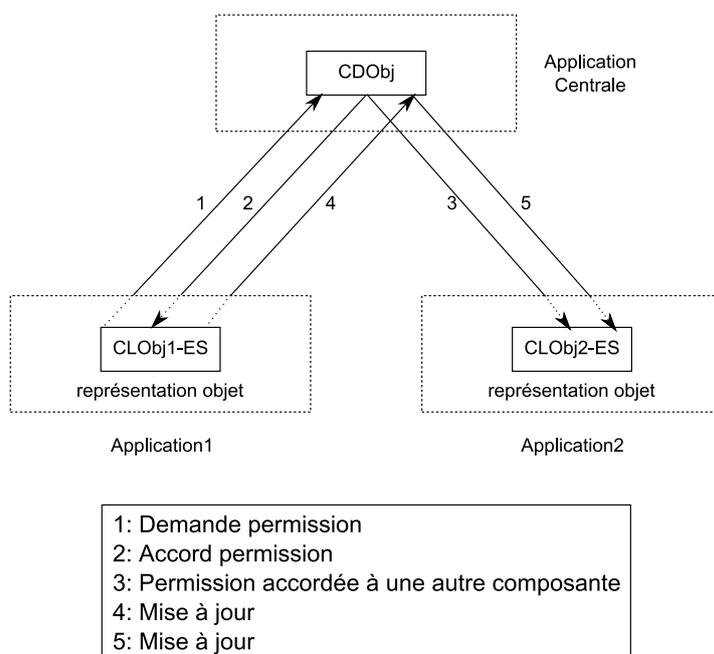


FIGURE 4.5 Gestion des permissions par demande de permission dans le cas d'un système n'intégrant pas de fantômes.

Les trois approches du système de permission permettent la gestion de la concurrence des interactions en entrée et le maintien de la cohérence des données. Néanmoins,

une gestion des permissions doit être réalisée au niveau de l'application centrale ainsi qu'une gestion des blocages ou non des interactions au niveau des autres applications. Afin de gérer les blocages par accès à des ressources partagées, nous avons développé des machines à états au niveau de l'application centrale et au niveau des applications clientes.

4.4.2 Machines à états des composantes logiques

Les machines à états ainsi que les réseaux de Pétri sont utilisés dans le domaine de l'IHM pour modéliser et formaliser les interactions en entrée [Appert & Beaudouin-Lafon, 2006, Schyn *et al.*, 2003]. Nous concevons des machines à états au niveau de chaque application IHM, et plus précisément au niveau des composantes logiques d'entrée/sortie, pour la gestion des blocages et des permissions des interactions. Un autre élément intéressant de ces machines à états est la possibilité d'identifier l'application avec laquelle la manipulation de l'objet virtuel ou la navigation est réalisée, et ainsi de permettre la présentation de l'information sur l'application active dans les autres applications. Un exemple d'une telle présentation d'information est le rendu d'un objet virtuel avec une couleur rouge dans une application IHM pour illustrer que cet objet est manipulé par une autre application IHM.

La modélisation des machines à états au niveau des composantes logiques d'entrée/sortie s'effectue en deux phases qui sont la définition des états et ensuite la spécification des transitions entre ces états. Le nombre d'états (*nbEtats*) d'une machine à états d'une composante logique est déterminé par le nombre des autres composantes logiques d'entrée/sortie (*nbCL*) connectées à l'élément du contrôleur de dialogue auquel elle est reliée. Le nombre d'états est le suivant :

$$nbEtats = 3 + 2 * nbCL$$

Les trois états de l'équation correspondent aux états de repos, en attente de permission et à l'état de permission obtenue. Les $2 * nbCL$ états correspondent aux états de blocage par une autre composante logique et aux états d'attente de permission avec un blocage par une autre composante. La figure 4.6 illustre une architecture avec deux composantes logiques *CL1* et *CL2* connectées à un même élément de contrôleur de dialogue *CD*. Les états de la machine à états de la composante logique *CL1* sont illustrés dans la figure 4.7.

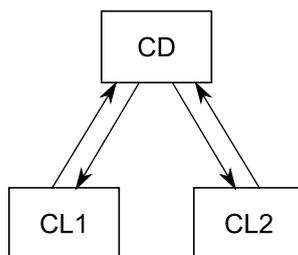


FIGURE 4.6 Architecture avec deux composantes logiques $CL1$ et $CL2$ connectées à un même élément du contrôleur de dialogue CD .

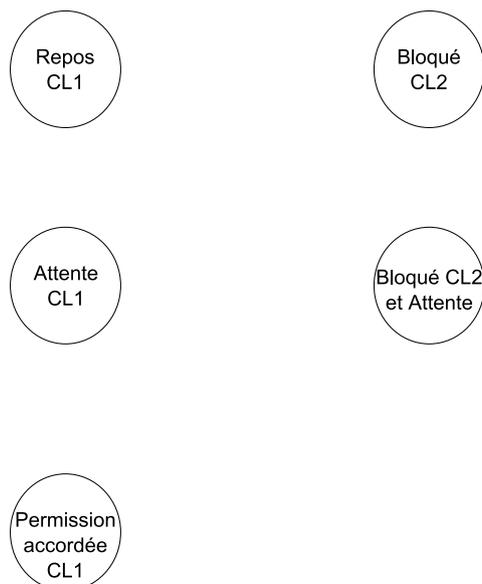


FIGURE 4.7 Etats d'une machine à états d'une composante logique d'entrée/sortie $CL1$ dans le cas de deux composantes logiques $CL1$ et $CL2$ reliées au même élément du contrôleur de dialogue (voir architecture de la figure 4.6).

Nous définissons deux types de transitions possibles entre les différents états des machines à états. Le premier type correspond aux transitions internes à l'application intégrant la composante logique (e.g. sélection d'un objet virtuel). Ces transitions sont :

- la *sélection*. Elle relie l'état de repos à l'état d'attente de la permission et les états de blocage aux états de blocage en attente correspondants ;
- la *dé-sélection*. Elle relie les états d'attente et de permission obtenue à l'état de repos ainsi que les états de blocage en attente aux états de blocage correspondants ;
- la *modification des données*. Elle relie à lui-même l'état de permission obtenue.

Dans le cas d'utilisation de fantômes, cette transition existe aussi au niveau de l'état en attente.

Le second type de transitions correspond aux transitions externes. Ce sont les messages reçus de l'application centrale. Les transitions externes possibles sont :

- l'*accord de permission*. Elle relie les états d'attente à l'état de permission accordée ;
- le *blocage*. Elle relie les états de blocage aux autres états de blocage, les états de

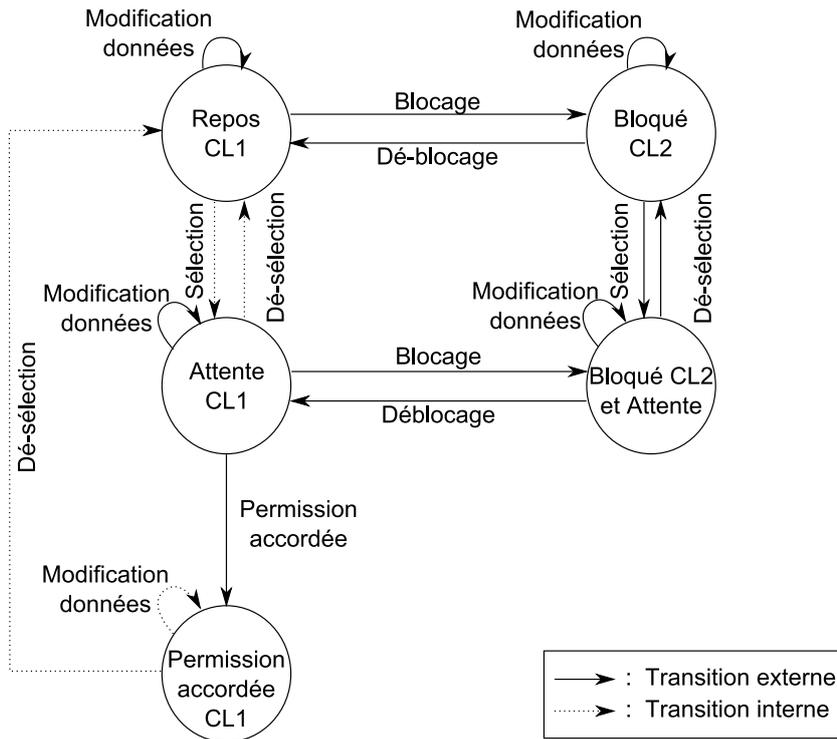


FIGURE 4.8 Machine à états de la composante logique d'entrée/sortie *CL1* de l'architecture de la figure 4.6.

- blocage en attente aux autres états de blocage en attente, l'état de repos à tous les états de blocage et l'état en attente aux états de blocage en attente ;
- le *dé-blocage*. Elle relie les états de blocage à l'état de repos et les états de blocage en attente à l'état d'attente ;
- la *modification des données*. Ces transitions relient chacun des états à lui-même sauf pour l'état de permission accordée. Les transitions de modification des données au niveau de l'état de repos et d'attente correspondent à des modifications réalisées par des composantes logiques d'entrée.

La figure 4.8 illustre la machine à états de la composante logique *CL1* de la figure 4.6. Nous remarquons qu'aucun état ne possède deux transitions sur lui-même pour les modifications internes et externes.

La présentation de l'information sur l'application utilisée à un instant donné (pour la manipulation d'un objet virtuel ou la navigation) au niveau des autres applications est nécessaire. La gestion de cette présentation au niveau de composantes logiques de sortie peut être réalisée par la modélisation de machines à états. Cependant, une variable suffit pour représenter ce type d'information et ainsi il n'est pas nécessaire de modéliser des machines à états de présentation de l'information au niveau des composantes logiques de sortie.

4.4.3 Machines à états du contrôleur de dialogue

Les machines à états des éléments du contrôleur de dialogue gèrent la cohérence des données ainsi que la concurrence des interactions effectuées au niveau des différentes applications. La modélisation des machines à états au niveau des éléments du contrôleur de dialogue est réalisée sur deux étapes : la définition des états et la définition des transitions.

Le nombre d'états $nbEtats$ d'une machine à états d'un élément du contrôleur de dialogue dépend du nombre de composantes logiques d'entrée/sortie $nbCL$ connectées à cet élément :

$$nbEtats = 1 + nbCL^2$$

Un des états modélisés représente l'état de repos dans lequel la permission est située au niveau du contrôleur de dialogue. Chaque composante logique d'entrée/sortie est représentée par un état qui indique qu'elle possède la permission. Finalement, pour chacun de ces $nbCL$ états correspondant à une composante logique CLi , il existe un nombre $nbCL - 1$ d'états qui représentent les états dans lesquels la permission est accordée à cette composante logique CLi et au moins une autre composante logique est mise en attente. Ces états permettent d'identifier la première composante logique mise en attente. S'il existe d'autres composantes logiques mises en attente alors elles sont gérées par une liste d'attente ce qui permet de réduire le nombre d'états des machines à états. La figure 4.9 illustre les états de la machine à états de l'élément du contrôleur de dialogue CD connecté à deux composantes logiques d'entrée/sortie $CL1$ et $CL2$ présentées dans la figure 4.6.

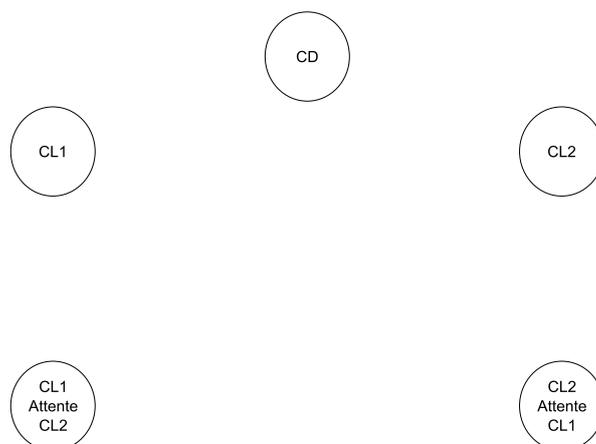


FIGURE 4.9 Etats de la machine à états de l'élément du contrôleur de dialogue CD de la figure 4.6.

Nous identifions trois types de transitions possibles entre les états des machines à

états au niveau du contrôleur de dialogue :

- la *sélection*. Elle relie l'état de repos aux états de permission accordée à une composante logique ainsi que les états de permission accordée aux états associés de permission accordée avec mise en attente ;
- la *dé-sélection*. Elle relie premièrement les états de permission accordée à l'état de repos. Deuxièmement, elle relie chacun des états de permission accordée avec mise en attente à l'état de permission accordée correspondant (de la même composante logique). Troisièmement, elle relie aussi chacun des états de permission accordée avec mise en attente à l'état de permission accordée de la composante logique mise en attente dans le premier état. Les deux derniers cas de transition de dé-sélection sont spécifiques aux files d'attente non vides dans le cas de machines à états à au moins dix états (cas d'au moins trois composantes logiques connectées à cet élément du contrôleur de dialogue). Elle relie ainsi chacun des états de permission accordée avec mise en attente aux autres états de permission accordée avec mise en attente de la même composante logique (cas de dé-sélection de la première composante logique mise en attente). Finalement, elle relie chacun des états de permission accordée avec mise en attente aux états de permission accordée avec mise en attente de la première composante logique déjà en attente dans le premier état (la nouvelle composante logique en attente est différente de celle qui possède la permission dans le premier état). Ce cas correspond au transfert de permission d'une composante logique à une autre et pour laquelle une troisième composante logique est dans la file d'attente ;
- la *modification des données*. Elle relie tout état à lui-même. La transition de modification des données au niveau de l'état de repos correspond à des modifications réalisées par des composantes logiques d'entrée.

La figure 4.10 illustre les états ainsi que les transitions d'une machine à états d'un élément du contrôleur de dialogue *CD* connecté à deux composantes logiques d'entrée/sortie *CL1* et *CL2* (voir figure 4.6).

Les machines à états de chacune des composantes logiques d'entrée/sortie et de chacun des éléments du contrôleur de dialogue permettent de gérer la concurrence ainsi que la cohérence des données. Néanmoins, il existe d'autres niveaux de gestion de la concurrence comme la gestion de la concurrence au niveau des degrés de liberté de manipulation d'un objet virtuel. Nous présentons dans la section suivante les différents niveaux de gestion de la concurrence ainsi que les modifications que cela entraîne dans la modélisation des machines à états.

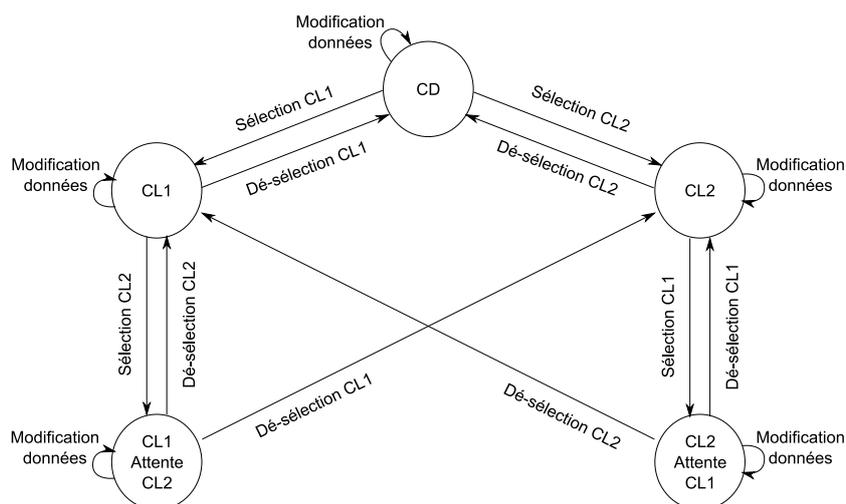


FIGURE 4.10 Machine à états de l'élément du contrôleur de dialogue *CD* de l'architecture de la figure 4.6.

4.4.4 Granularité de la gestion de la concurrence

Nous identifions plusieurs niveaux de granularité de gestion de la concurrence des interactions en entrée. En effet, la gestion de la concurrence des interactions dans un système combinant plusieurs applications peut s'effectuer soit en bloquant toutes les interactions en entrée d'une application, soit en bloquant uniquement la manipulation d'un objet virtuel au moyen de cette application lorsqu'un utilisateur le manipule dans une autre application. Une granularité grossière de la gestion de la concurrence restreint les interactions des utilisateurs en bloquant des interactions alors qu'elles pourraient être disponibles avec une approche de granularité plus fine (e.g. au niveau d'une partie de la manipulation des données et non au niveau des applications).

Le niveau de granularité le moins fin existe lorsque la gestion des permissions s'effectue au niveau des applications c'est à dire lorsqu'une application est utilisée par un utilisateur en entrée et les autres applications sont alors totalement bloquées. Dans cette approche et à un instant donné, une seule application est considérée comme une interface en entrée et en sortie et les autres deviennent des systèmes de présentation de l'information uniquement. Cette approche est très restrictive au niveau de l'interactivité du système.

Un second niveau de granularité est celui de la gestion de la concurrence au niveau de chaque objet virtuel et chaque vue connectée à des composantes logiques d'entrée et/ou d'entrée/sortie. Ce niveau de granularité permet de manipuler simultanément des objets virtuels distincts au moyen de plusieurs applications autorisant ainsi une plus grande flexibilité des interactions. Les machines à états sont modélisées au niveau de chaque composante logique d'entrée/sortie et au niveau de chaque élément du contrôleur de

dialogue comme décrit dans les deux parties précédentes. Cependant, la manipulation simultanée d'un même objet virtuel au moyen de plusieurs applications est impossible avec ce niveau de granularité.

Pinho *et al.* [2002] partagent la manipulation d'un même objet virtuel entre deux utilisateurs selon ses degrés de liberté disponibles en interaction. Dans le but d'affiner la granularité et en s'inspirant de ces travaux, nous proposons de gérer la concurrence des interactions au niveau des degrés de liberté. Ainsi, la manipulation simultanée d'un objet virtuel est possible au moyen de deux applications distinctes en partageant le contrôle des degrés de liberté. Ce niveau de granularité est justifié par la nécessité de contrôler un objet virtuel en combinant simultanément plusieurs degrés de liberté (e.g. déplacer un objet selon une trajectoire prédéfinie). Certains degrés de liberté manipulés au moyen d'une application sont alors en concurrence avec d'autres degrés de liberté manipulés dans une autre application. Cette concurrence existe dans le cas d'un couplage fort entre l'entrée et la sortie.

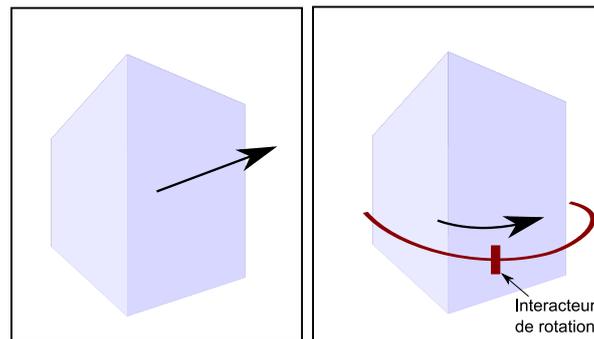


FIGURE 4.11 Exemple de degrés de liberté concurrents. Gauche : translation d'un objet virtuel. Droite : rotation du même objet virtuel affiché sur une seconde vue.

Pour illustrer cette concurrence, considérons deux applications graphiques affichant la même scène virtuelle sur deux tables tactiles. Une application permet la translation selon les axes X et Y d'un objet virtuel tandis que la seconde permet sa rotation par manipulation directe d'un interacteur attaché graphiquement à l'objet virtuel (voir figure 4.11). La translation de l'objet sur la première application implique la translation du même objet virtuel dans la seconde application. Ainsi, il est difficile (voire impossible) de réaliser la rotation en préservant le couplage entrée/sortie et la liaison du manipulateur à l'objet virtuel. Nous définissons la notion de concurrence entre degrés de liberté lorsque la présentation des effets de l'interaction avec le premier degré de liberté entraîne une discontinuité et/ou une gêne pour la réalisation d'une interaction utilisant le second degré de liberté. Cette discontinuité existe lorsque le couplage fort entrée/sortie n'est plus préservé. Par exemple, considérons deux vues 2D dont l'une (appelée V_1) présente

le plan XY et la seconde (appelée V_2) présente le plan XZ. La manipulation directe pour la translation d'un objet virtuel dans V_1 selon l'axe X entraîne une discontinuité de la manipulation directe de cet objet dans V_2 pour la translation selon l'axe Z (à cause de la translation d'objet selon l'axe X dans la seconde vue). Ainsi, la translation selon l'axe X réalisée par manipulation directe dans V_1 est concurrente à la translation selon l'axe Z par manipulation directe dans V_2 . La concurrence des degrés de liberté est liée aux techniques d'interaction utilisées et au couplage des entrées/sorties.

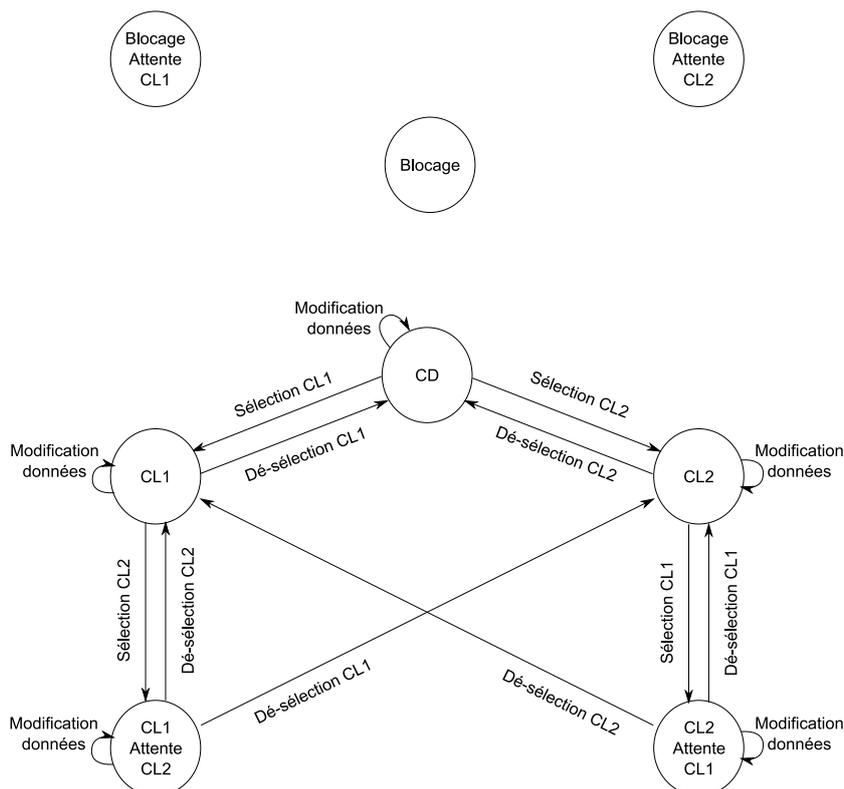


FIGURE 4.12 Exemple des états de machine à états d'un degré de liberté d'un élément du contrôleur de dialogue.

L'affinement de la granularité implique une modification de la modélisation des machines à états qui doivent se situer au niveau des degrés de liberté de manipulation des objets virtuels et/ou des vues. Les machines à états des degrés de liberté des composantes logiques sont identiques à celles modélisées au niveau des composantes logiques. Cependant, la concurrence entre degrés de liberté implique une modification des états des machines à états au niveau des degrés de liberté des éléments du contrôleur de dialogue. En effet, la modélisation de ces machines à états nécessite des états supplémentaires par rapport aux machines à états des éléments du contrôleur de dialogue pour bloquer les degrés de liberté concurrents. Le nombre d'états supplémentaires est égal au nombre de composantes logiques d'entrée/sortie connectées à cet élément plus un. Un état représente le cas dans lequel le degré de liberté est bloqué par le contrôleur de dia-

logue. Les autres états représentent le blocage d'un degré de liberté par le contrôleur de dialogue et la mise en attente d'une des applications souhaitant l'utiliser. La figure 4.12 illustre les états des machines à états au niveau de chaque degré de liberté d'un élément du contrôleur de dialogue connecté à deux composantes logiques d'entrée/sortie (voir architecture de la figure 4.6). Les transitions illustrées dans cette figure sont définies selon la modélisation des machines à états d'un élément du contrôleur de dialogue. Les transitions supplémentaires sont les suivantes :

- le *blocage*. Elle relie l'état de repos à l'état de blocage par le contrôleur de dialogue. Cette transition intervient lors de la permission d'utiliser un degré de liberté concurrent au degré de liberté de cette machine à états ;
- le *dé-blocage*. Elle relie l'état de blocage à l'état de repos. Elle relie aussi chacun des états de blocage avec mise en attente à l'état de permission accordée et aux états de permission accordée avec mise en attente dont la composante logique possédant la permission est mise en attente dans l'état précédent ;
- la *sélection*. Elle relie l'état de blocage aux états de blocage avec mise en attente ;
- la *dé-sélection*. Elle relie chacun des états de blocage avec mise en attente à l'état de blocage et aux autres états de blocage avec mise en attente.

La figure 4.13 illustre les états et les transitions des machines à états au niveau de chaque degré de liberté d'un élément du contrôleur de dialogue connecté à deux composantes logiques d'entrée/sortie *CL1* et *CL2* (voir architecture de la figure 4.6).

Chacune des composantes logiques d'entrée/sortie n'intègre pas obligatoirement tous les degrés de liberté. Une étape de simplification des machines à états au niveau des degrés de liberté est donc nécessaire. Cette étape consiste à éliminer les états de permission accordée et de permission accordée avec mise en attente des composantes logiques qui n'intègrent pas ce degré de liberté. De plus, tous les états dans lesquels une composante logique mise en attente n'intègre pas le degré de liberté de cette machine à états sont supprimés. Les transitions connectées à l'un des états enlevés sont elles aussi supprimées. Finalement, si le degré de liberté n'est en concurrence avec aucun autre degré de liberté, les états de blocage et de blocage avec mise en attente sont supprimés ainsi que les transitions qui y sont liées. La figure 4.14 illustre deux cas de machines à états pour la translation selon l'axe X. Cette translation est réalisable uniquement au moyen de la composante logique *CL1*. Dans le premier cas (figure 4.14 gauche) ce degré de liberté est en concurrence avec un autre degré de liberté tandis que dans le second cas (figure 4.14 droite) ce degré de liberté n'est en concurrence avec aucun autre.

Les niveaux de granularité de la gestion des interactions couvrent un champ allant du blocage de toutes les interactions dans une application au blocage partiel de certains degrés de liberté de manipulation d'un objet virtuel ou de navigation. Afin d'évaluer

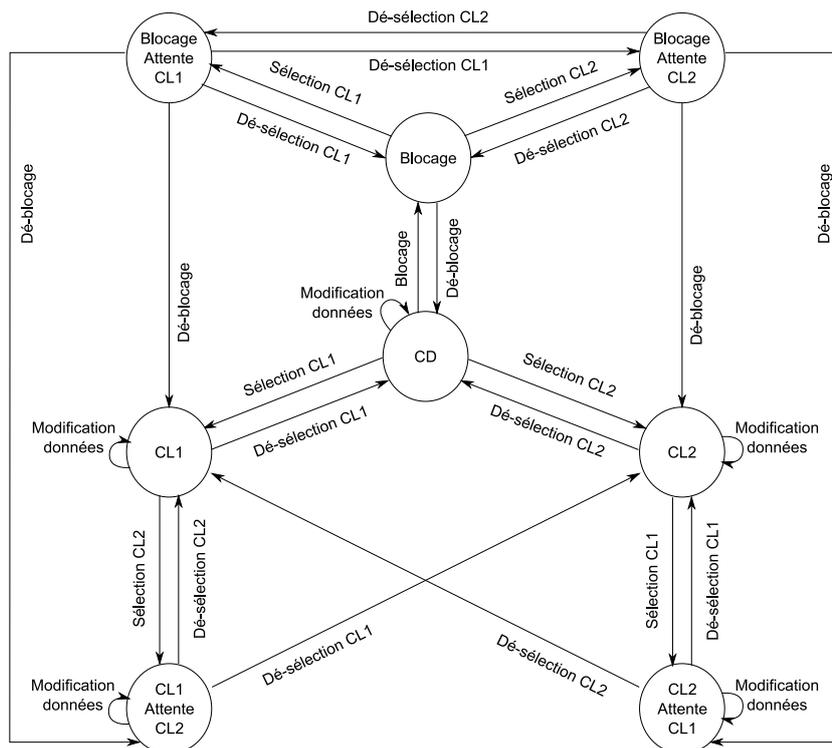


FIGURE 4.13 Exemple d'une machine à états au niveau d'un degré de liberté d'un élément du contrôleur de dialogue.

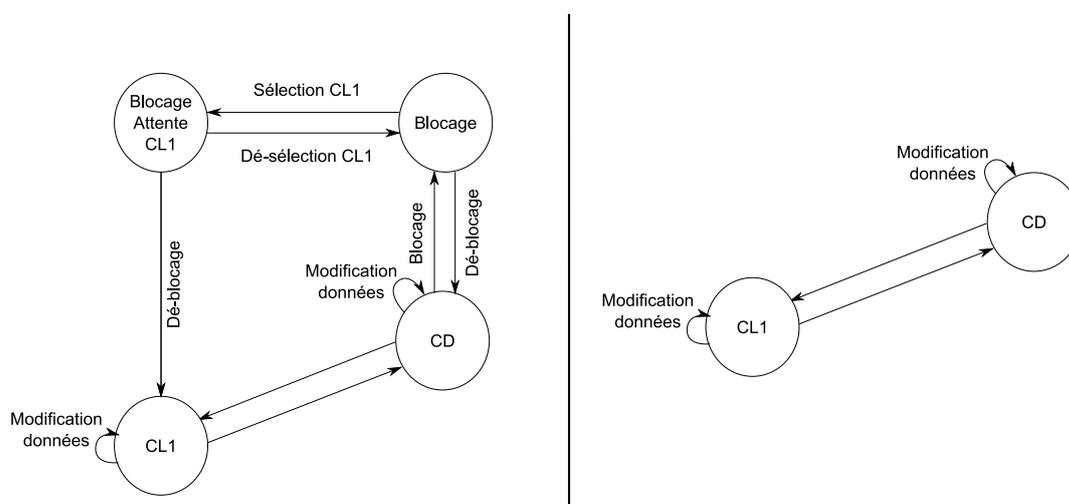


FIGURE 4.14 Exemple de deux machines à états au niveau d'un degré de liberté d'un élément du contrôleur de dialogue après simplification. Gauche : degré de liberté concurrent avec un autre. Droite : le degré de liberté n'est en concurrence avec aucun autre.

la facilité et l'utilité de l'interaction par plusieurs applications et selon plusieurs degrés de liberté en fonction des utilisateurs, nous présentons dans la section suivante une expérimentation comparant des niveaux de granularité différents. Cette expérience est effectuée sur la manipulation d'un objet virtuel et sur la navigation selon plusieurs degrés de liberté.

4.5 Expérimentation

Nous présentons dans cette section une expérimentation étudiant les performances et les préférences des utilisateurs dans le but de comparer différents niveaux de granularité. Les deux degrés de granularité que nous examinons sont la gestion de la concurrence au niveau des composantes logiques (que nous appelons mode *exclusif*) et au niveau des degrés de liberté (que nous appelons mode *coopératif*). Nous réalisons cette comparaison pour une tâche coopérative bi-utilisateurs de positionnement d'un objet 3D (docking) et pour une tâche coopérative bi-utilisateurs de navigation pour le positionnement et l'orientation du point de vue 3D. Nous posons comme première hypothèse que les temps nécessaires pour la réalisation des tâches sont inférieurs avec un niveau de granularité plus fin. Nous posons comme seconde hypothèse que les utilisateurs préfèrent le degré de granularité le plus fin. Dans cette section nous décrivons la conception et les conditions dans lesquelles s'est déroulée cette expérimentation. Nous analysons ensuite les résultats obtenus.

4.5.1 Conception de l'expérimentation

Nous nous basons sur l'architecture de MC présentée dans la figure 4.3 pour réaliser cette expérimentation. La souris est utilisée comme dispositif d'entrée de la composante IHM 3D. Une vue 2D horizontale d'un environnement virtuel est présentée sur la table tactile ainsi qu'une représentation du point de vue 3D. La vue 3D est projetée sur un écran vertical situé en face des utilisateurs.

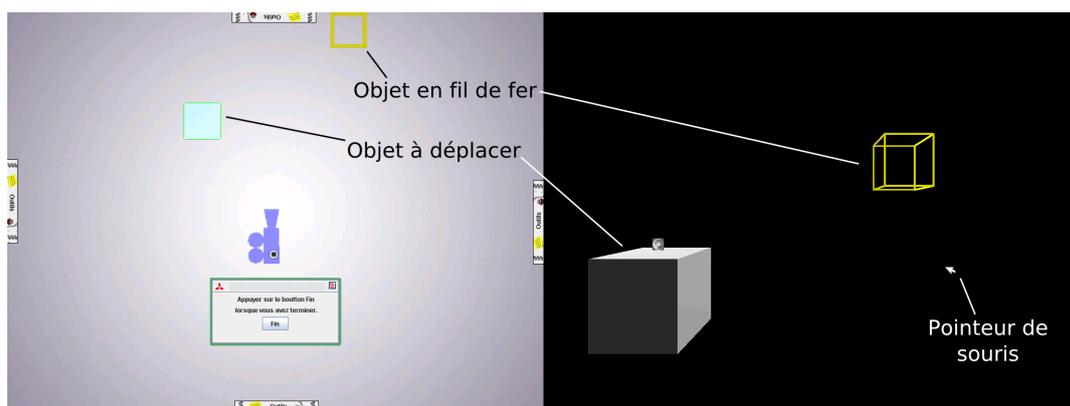


FIGURE 4.15 Vues initiales de la tâche de positionnement d'un objet virtuel. Gauche : vue 2D présentée sur la table tactile. Droite : vue 3D projetée sur un écran vertical face aux utilisateurs.

Nous définissons deux tâches à réaliser par chaque couple d'utilisateurs : une tâche de manipulation d'un objet virtuel et une tâche de navigation. La tâche de manipulation

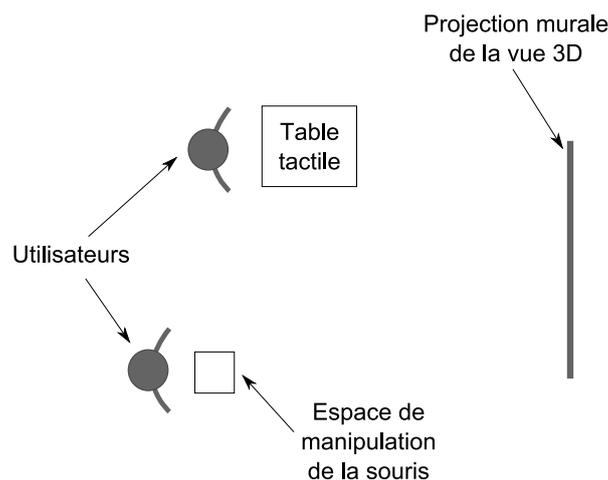


FIGURE 4.16 Plan de l'expérience avec un utilisateur interagissant avec la table tactile et un autre avec la souris.

d'un objet virtuel consiste à positionner un cube virtuel dans un autre cube virtuel rendu en fil de fer (i.e. dont les arêtes sont rendues visibles uniquement). La figure 4.15 illustre les vues 2D et 3D initiales pour la tâche de positionnement du cube virtuel. Un utilisateur peut réaliser des translations horizontales avec la table tactile tandis qu'un autre utilisateur déplace une souris sur une surface située à côté de la table pour effectuer des translations dans le plan de projection de la vue 3D (voir figure 4.16). Les utilisateurs peuvent manipuler la souris ou interagir avec la table tactile avec leur main dominante (i.e. gauche pour les gauchers). L'élément du contrôleur de dialogue représentant le cube virtuel à positionner est connecté à deux composantes logiques d'entrée/sortie (voir figure 4.18). Les degrés de liberté de translation accessibles aux utilisateurs pour la tâche de positionnement d'un objet virtuel sont : les axes X et Y du repère du monde virtuel sur la table tactile et les axes X' et Z' du plan de projection 3D avec la souris. Ces quatre dimensions définissent une base vectorielle de l'espace 3D car le plan de projection de la vue 3D n'est pas parallèle au plan XY. Pour la tâche de positionnement, ce plan de projection est constamment parallèle au plan XZ du repère du monde virtuel (voir figure 4.17) mais nous considérons que ce n'est pas obligatoirement le cas pour la modélisation des composantes logiques et la gestion de la concurrence. Ces quatre dimensions permettent donc bien aux utilisateurs de parcourir les trois dimensions de l'espace 3D.

La tâche de navigation consiste à positionner le point de vue 3D à l'intérieur d'une sphère jaune rendue en transparence et de l'orienter de façon à positionner un cylindre au centre de la vue 3D (voir la figure 4.19 qui illustre les vues 2D et 3D initiales). Le positionnement du point de vue 3D dans la sphère précise la position finale souhaitée tandis que le positionnement du cylindre au centre de la vue 3D définit les angles de rotations souhaités. Pour cette tâche, les degrés de liberté partagés entre les utilisateurs

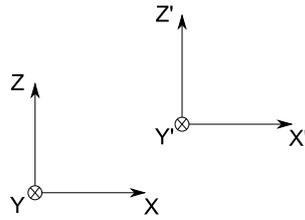


FIGURE 4.17 Différences entre le repère du monde virtuel et le repère du plan de projection pour la tâche de positionnement d'un cube virtuel.

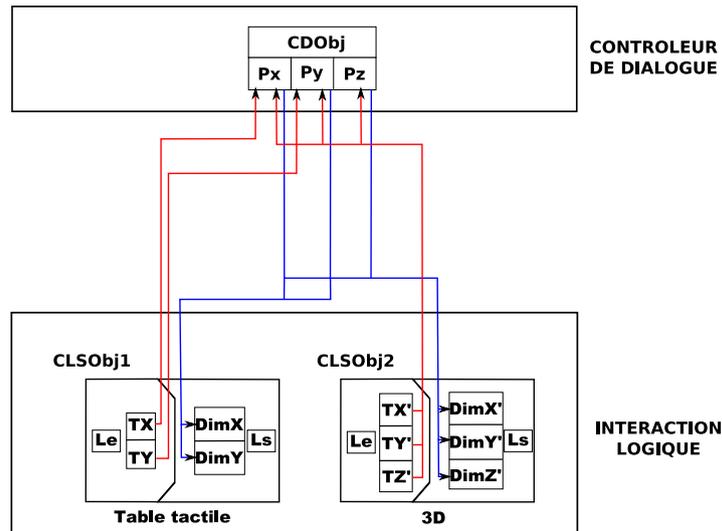


FIGURE 4.18 Composantes logiques d'entrée/sortie et l'élément du contrôleur de dialogue nécessaires pour la tâche de positionnement d'un objet virtuel.

sont : les translations (TX et TY) et la rotation (RZ) dans le plan XY du repère du monde virtuel sur la table tactile et les rotations autour des axes X' (RX') et Z' (RZ') du repère du point de vue 3D avec la souris. Le plan de projection X'Z' n'est pas nécessairement parallèle au plan XZ du repère du monde virtuel (voir figure 4.20). Néanmoins, nous limitons la rotation angulaire selon l'axe X' pour réduire l'angle entre l'axe Z du repère du monde virtuel et l'axe Z' de la projection. L'utilisateur interagissant avec la table tactile peut réaliser des translations horizontales et des changements d'orientation avec un suivi du mouvement (modification de l'orientation angulaire horizontale selon la tangente à la trajectoire) en manipulant la représentation du point de vue 3D sur la table tactile (i.e. icône de caméra). Pour cette tâche de navigation, l'élément du contrôleur de dialogue nécessitant la gestion de la concurrence est l'élément représentant la vue 3D. Il est connecté à une composante logique de vue représentant la vue 3D et à une composante de représentation de vue sur la vue 2D. Ces deux composantes logiques et leurs connexions à l'élément du contrôleur de dialogue sont illustrées dans la figure 4.21.

La gestion de la concurrence au niveau des composantes logiques (mode *exclusif*) est

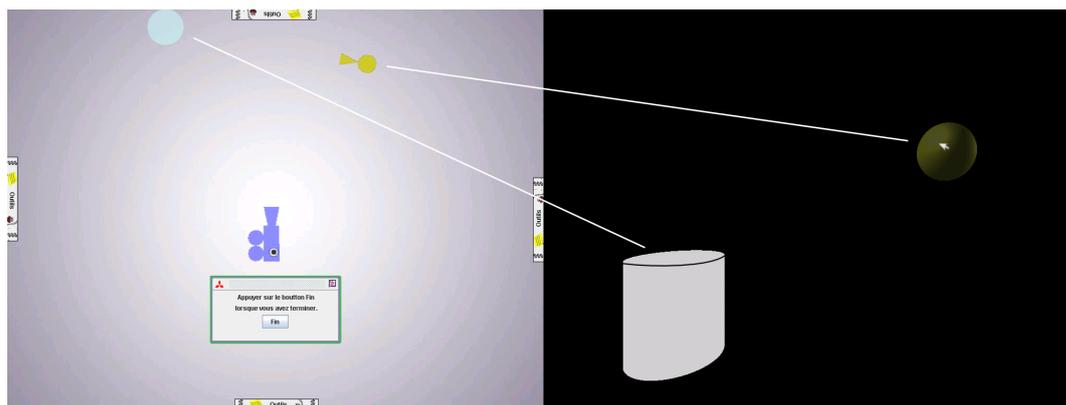


FIGURE 4.19 Vues initiales de la tâche de navigation. Gauche : vue 2D présentée sur la table tactile. Droite : vue 3D projetée sur un écran vertical.

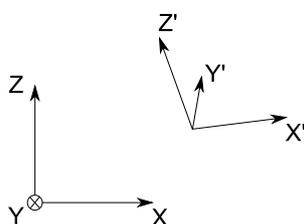


FIGURE 4.20 Différences entre le repère du monde virtuel et le repère du plan de projection pour la tâche de navigation.

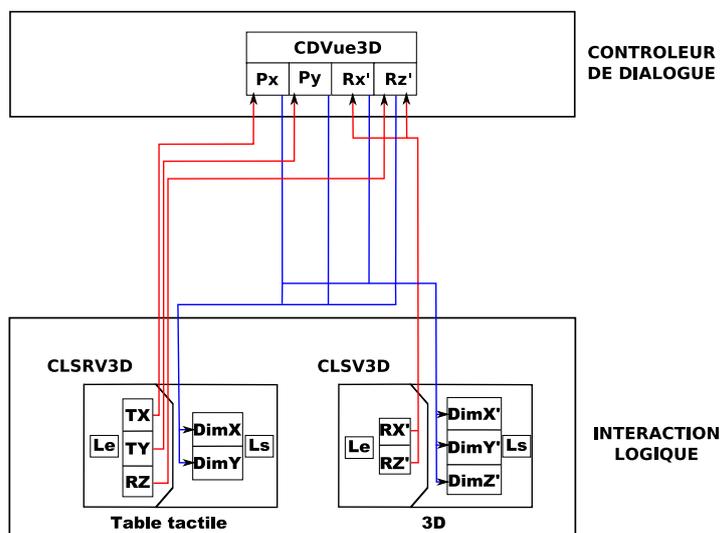


FIGURE 4.21 Composantes logiques d'entrée/sortie et l'élément du contrôleur de dialogue nécessaires pour la tâche de navigation.

réalisée en modélisant une machine à états pour chacune des composantes logiques et pour chacun des éléments du contrôleur de dialogue. En appliquant les différentes étapes de conception des machines à états (pour la tâche de positionnement) au niveau des composantes logiques et au niveau des éléments du contrôleur de dialogue, nous obtenons les trois machines à états de la figure 4.22. Elles correspondent respectivement à la machine à états de la composante logique *CLSO_{obj1}*, de l'élément *CDO_{obj}* du contrôleur

de dialogue et de la composante logique *CLSOBJ2*. Les machines à états modélisées pour la tâche de navigation sont identiques à celles pour la tâche de positionnement mais sont modélisées respectivement au niveau de la composante logique *CLSRV3D*, de l'élément du contrôleur de dialogue *CDVUE3D* et de la composante logique *CLSV3D*.

La gestion de la concurrence au niveau des degrés de liberté (mode *coopératif*) est réalisée en modélisant une machine à états pour chaque degré de liberté des composantes logiques et des éléments du contrôleurs du dialogue présentés précédemment. Les machines à états des degrés de liberté des composantes logiques sont identiques à celles des composantes logiques sauf qu'elles sont conçues pour chaque degré de liberté. Pour la conception des machines à états au niveau des éléments du contrôleur de dialogue, nous n'identifions aucune concurrence entre les degrés de liberté pour le positionnement du cube virtuel. Les machines à états pour les translations selon les axes X et Y sont identiques et sont représentées dans la partie gauche de la figure 4.23. La machine à états correspondant à la translation selon l'axe Z est présentée dans la partie droite de la figure 4.23.

Pour la tâche de navigation, une concurrence existe entre la rotation horizontale avec la souris et les translations selon les axes X et Y avec la table tactile. Les machines à états correspondant aux degrés de liberté des translations selon les axes X et Y sont identiques et sont illustrées en haut de la figure 4.24. La partie au centre de la figure 4.24 (respectivement en bas de la figure 4.24) illustre la machine à états correspondant à la rotation autour de l'axe Z' (respectivement autour de l'axe X').

4.5.2 Conditions de l'expérimentation

Durant chaque expérience deux utilisateurs coopèrent pour réaliser les tâches. L'un des utilisateurs effectue les interactions en utilisant la table tactile tandis que le second réalise les interactions en manipulant la souris (voir figures 4.16 et 4.25). Le rôle des utilisateurs est interverti pendant l'expérience. Le choix de deux utilisateurs et non pas d'un seul est justifié par la volonté de comparer les niveaux de granularité et donc de vérifier si dans le mode *coopératif* les interactions sont réalisées simultanément ou non en utilisant la table tactile et la souris. Nous pensons qu'un utilisateur non expérimenté manipulerait chacun des dispositifs séparément ce qui aurait limité la comparaison des deux niveaux de granularité. Douze couples d'utilisateurs (10 femmes et 14 hommes) ont participé à l'expérimentation. Les utilisateurs possèdent des niveaux d'expérience variés sur les environnements virtuels et les jeux vidéos en 3D.

Chaque couple d'utilisateurs est amené à effectuer chacune des deux tâches selon les deux modes (*coopératif* et *exclusif*) et selon deux conditions finales différentes (position

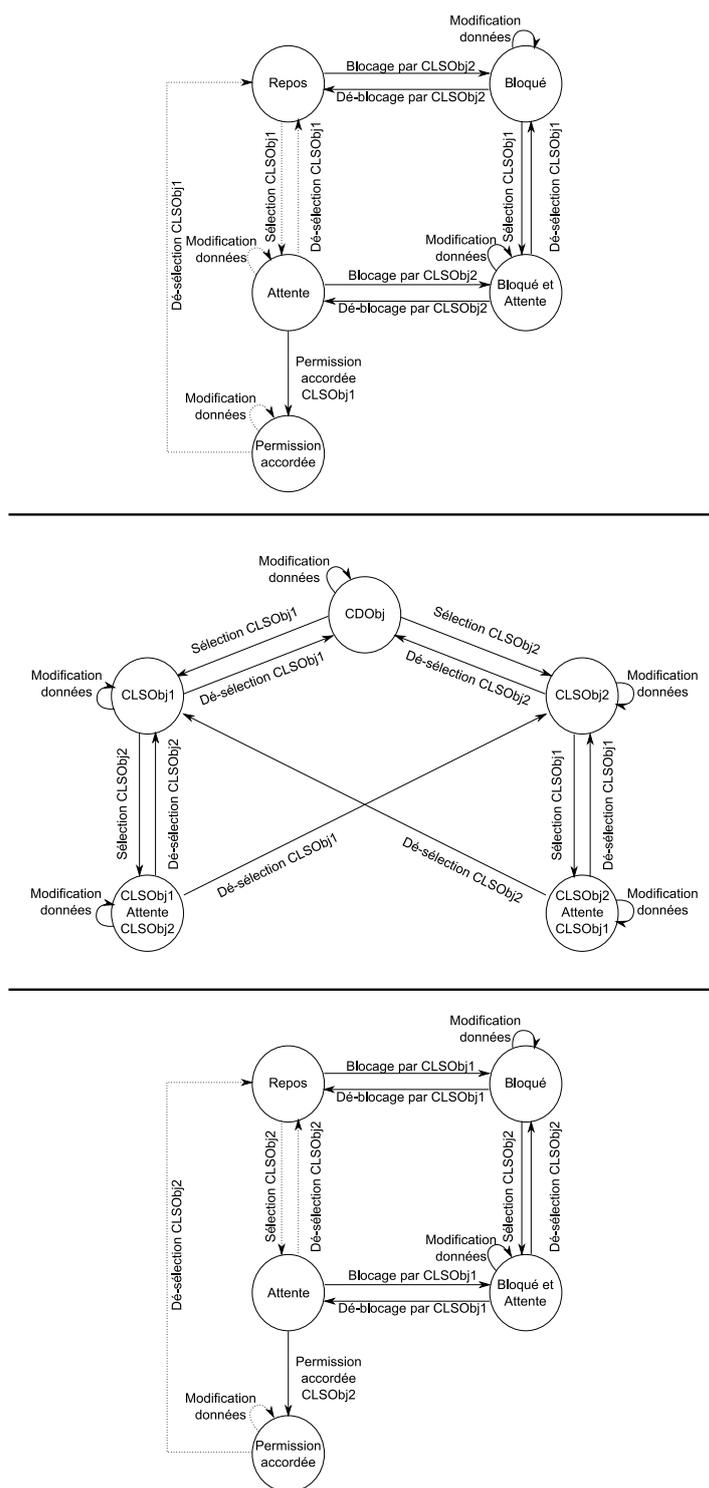


FIGURE 4.22 Machines à états développées au niveau des composantes logiques et de l'élément du contrôleur de dialogue pour le mode *exclusif*. Haut : machine à états au niveau de la composante logique fantôme de la représentation du cube en 2D. Centre : machine à états développée au niveau de l'élément du contrôleur de dialogue correspondant au cube virtuel. Bas : machine à états au niveau de la composante logique de la représentation du cube en 3D.

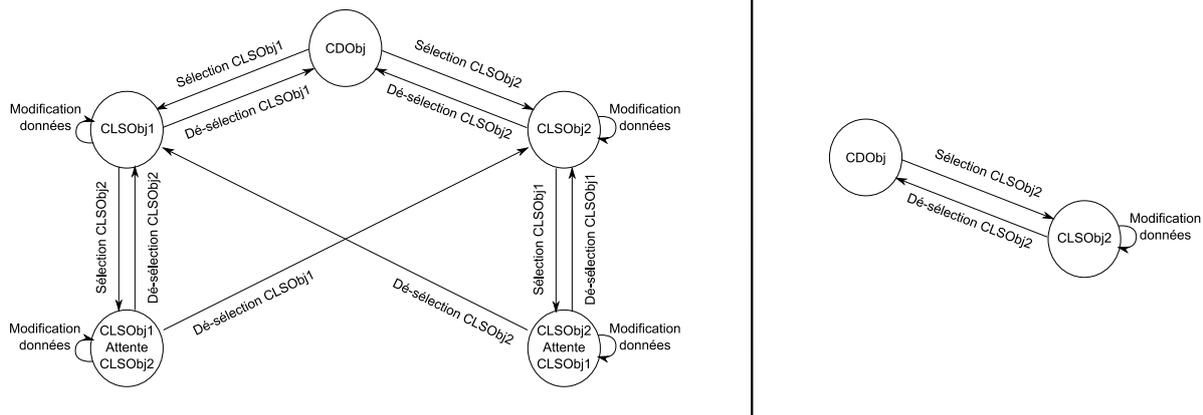


FIGURE 4.23 Machines à états développées au niveau des degrés de liberté de l'élément du contrôleur de dialogue représentant le cube virtuel pour la tâche de positionnement d'un objet virtuel. Gauche : machine à états pour la translation selon l'axe X et selon l'axe Y. Droite : machine à états pour la translation selon l'axe Z.

à atteindre du cube pour la tâche de positionnement d'un objet virtuel et position et orientation souhaitées du point de vue 3D pour la tâche de navigation). Avant la réalisation des tâches, une fiche expliquant les possibilités d'interactions selon les modes et les tâches est présentée aux utilisateurs. L'ordre des passages est inversé entre chaque mode et l'ordre des tâches est aussi inversé entre chaque couple d'utilisateurs pour éviter l'introduction d'un biais.

A la fin de l'expérimentation, les utilisateurs répondent à un questionnaire pour collecter des données subjectives afin de comparer les deux niveaux de granularité pour chaque tâche. Les utilisateurs sont priés de préciser le niveau de facilité des interactions pour chacune des tâches et pour chacun des modes sur une échelle de Likert [1932] à cinq niveaux (très bonne, bonne, moyenne, mauvaise et très mauvaise). Ils doivent préciser leur mode préféré pour chacune des tâches aussi sur une échelle de Likert [1932] à cinq niveaux (définitivement le mode *exclusif*, moyennement le mode *exclusif*, pas de préférence, moyennement le mode *coopératif* et totalement le mode *coopératif*). Finalement, les utilisateurs doivent indiquer le niveau d'importance de l'interaction simultanée sur le même objet et pour la navigation selon trois niveaux (indispensable, souhaitable et pas du tout nécessaire) afin d'évaluer leur appréciation de la concurrence entre les modalités.

4.5.3 Résultats et analyses de l'expérimentation

Dans cette partie nous décrivons les résultats de l'expérimentation. Dans un premier temps nous exposons les résultats objectifs calculés à partir des valeurs enregistrées dans

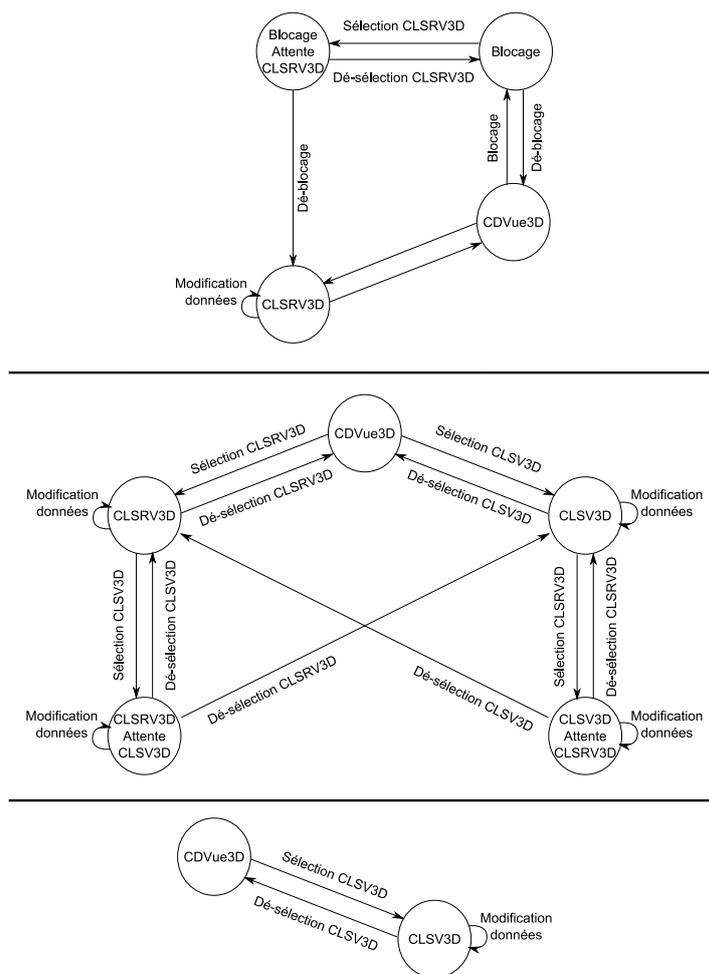


FIGURE 4.24 Machines à états développées au niveau des degrés de liberté de l'élément du contrôleur de dialogue représentant le point de vue 3D pour la tâche de navigation. Haut : machine à états pour la translation selon l'axe X et selon l'axe Y. Centre : machine à états pour la rotation autour de l'axe Z'. Bas : machine à états pour la rotation autour de l'axe X'.



FIGURE 4.25 Deux utilisatrices durant l'expérience.

les fichiers de log. Ensuite nous présentons les résultats subjectifs des questionnaires.

Résultats objectifs et analyse

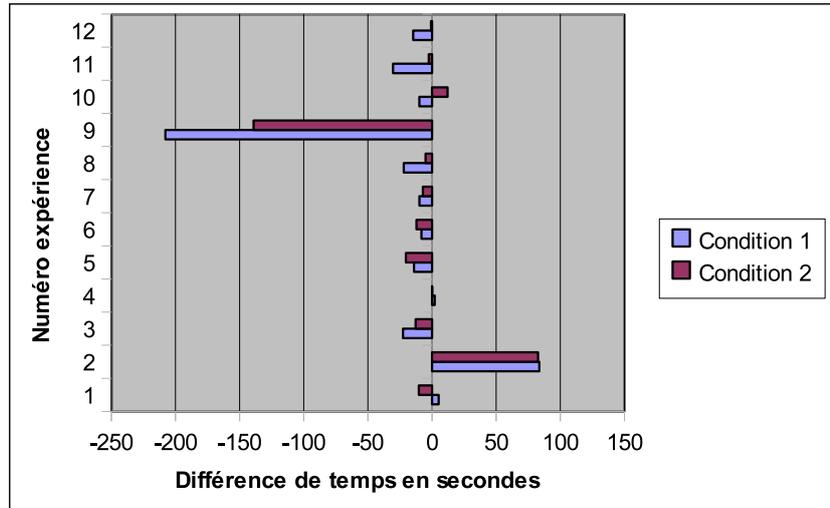


FIGURE 4.26 Différence de temps (en secondes) entre le mode *exclusif* et le mode *coopératif* pour effectuer la tâche de positionnement d'un objet virtuel.

Nous analysons tout d'abord la précision de la position et de l'orientation finale, le temps nécessaire pour la réalisation des tâches de positionnement et de navigation et le nombre de tentatives d'interactions simultanées. Les valeurs de la précision ne révèlent pas de différences significatives entre le mode *coopératif* et le mode *exclusif* pour ces tâches. La différence de temps entre les deux modes dans la tâche de positionnement (respectivement de navigation) est illustrée dans la figure 4.26 (respectivement 4.27). Ces résultats montrent que dans la majorité des cas, le temps nécessaire pour réaliser la tâche est inférieur dans le mode *exclusif* à celui du mode *coopératif*.

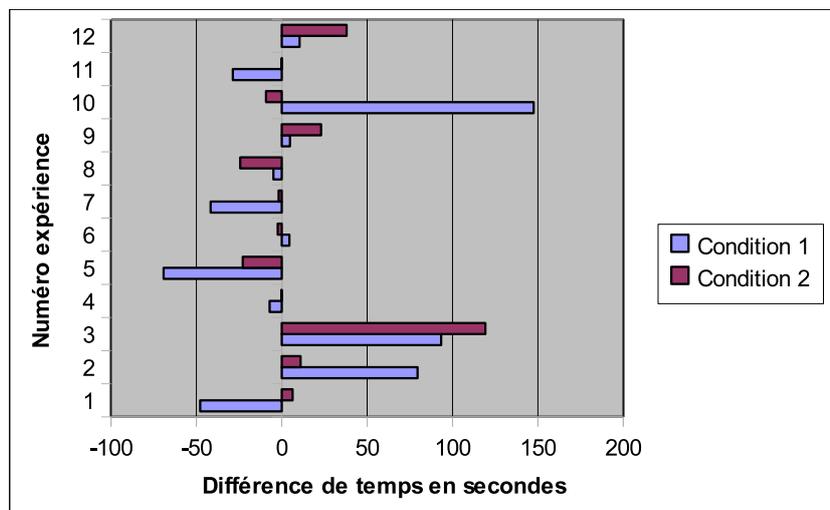


FIGURE 4.27 Différence de temps (en secondes) entre le mode *exclusif* et le mode *coopératif* pour la réalisation de la tâche de navigation.

Le nombre d'interactions réalisées simultanément avec la table tactile et la souris pour la tâche de positionnement du cube virtuel (respectivement de navigation) dans le mode *coopératif* est illustré dans la figure 4.28 (respectivement 4.29). Ces résultats représentent le nombre de demandes de permissions accordées par un élément du contrôleur de dialogue à une composante logique n'ayant aucune des permissions et dont l'une des permissions est accordée à une autre composante logique. Le nombre d'interactions réalisées simultanément varie selon les couples d'utilisateurs. Certains utilisateurs n'ont effectué aucune interaction simultanée tandis que d'autres en ont réalisé plusieurs pour accomplir la tâche. Néanmoins, la majorité des utilisateurs ont réalisé au moins une interaction simultanée avec la table tactile et la souris.

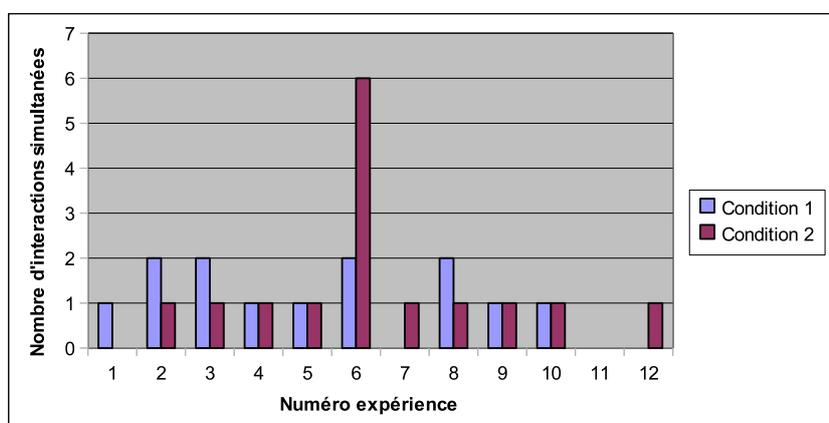


FIGURE 4.28 Nombre d'interactions réalisées simultanément par les utilisateurs pour la réalisation de la tâche de positionnement d'un objet virtuel.

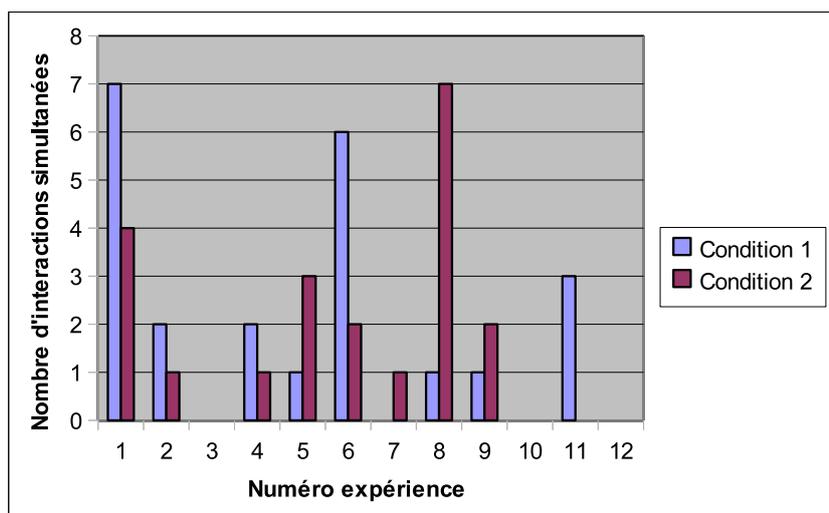


FIGURE 4.29 Nombre d'interactions réalisées simultanément par les utilisateurs pour l'accomplissement de la tâche de navigation.

Les temps de réalisation des tâches de positionnement d'un objet virtuel et de navigation réfutent notre hypothèse qui stipule que les performances des utilisateurs sont

meilleures avec un niveau de granularité plus fin. Nous estimons que ces résultats sont dus à la simplicité des tâches. Dans le cas de tâches plus complexes les résultats pourraient être différents. Les résultats confirment cependant que la majorité des utilisateurs ont réalisé des interactions simultanées avec la table tactile et la souris dans le mode *coopératif*. De plus, certains utilisateurs en mode *exclusif* ont tenté d'utiliser des interactions simultanées interdites. Ces résultats montrent que les utilisateurs souhaitent réaliser des interactions simultanément sans totalement être bloqués par l'autre utilisateur.

Résultats subjectifs et analyse

Les résultats obtenus du questionnaire sur la notation des utilisateurs de la facilité des interactions sont illustrés dans la figure 4.30. Ces résultats montrent que la majorité des utilisateurs considèrent les interactions faciles à réaliser. Le taux d'utilisateurs qui considèrent les interactions faciles est légèrement inférieur pour le mode *coopératif* à celui pour le mode *exclusif* (4% pour la tâche de manipulation d'un objet virtuel et 13% pour la tâche de navigation). Les interactions sont moins faciles pour la tâche de navigation que pour la tâche de manipulation d'un objet virtuel selon les utilisateurs.

Les utilisateurs notent leurs préférences pour le mode *exclusif* ou *coopératif* sur une échelle de Likert [1932] à cinq niveaux selon les deux tâches (voir figure 4.31). Les résultats obtenus sont différents selon les tâches. Pour la tâche de manipulation d'un objet virtuel (voir figure 4.31 gauche), 67% des utilisateurs préfèrent le mode *coopératif* contre 12% qui préfèrent le mode *exclusif*. Pour la tâche de navigation (voir figure 4.31 droite), 34% des utilisateurs préfèrent le mode *coopératif* tandis que 45% préfèrent le mode *exclusif*. Ces résultats montrent une différence importante de la préférence des utilisateurs pour le niveau de granularité de la gestion de la concurrence selon le type de tâche. Ce type de préférences aide à la conception des interfaces en orientant les choix de conception.

Dans la dernière partie du questionnaire, les utilisateurs sont amenés à noter l'importance qu'ils accordent à disposer de la possibilité de réaliser des interactions simultanément avec la table tactile et la souris. Les résultats montrent que la majorité des utilisateurs considèrent que la réalisation d'interactions simultanées est au moins souhaitable (voir figure 4.32), et 17% considèrent qu'elle n'est pas du tout nécessaire pour la tâche de manipulation d'objet virtuel contre 25% pour la navigation.

Les résultats subjectifs de l'expérimentation montrent une différence des préférences des utilisateurs pour les niveaux de granularité selon les tâches. Nous pensons que cette

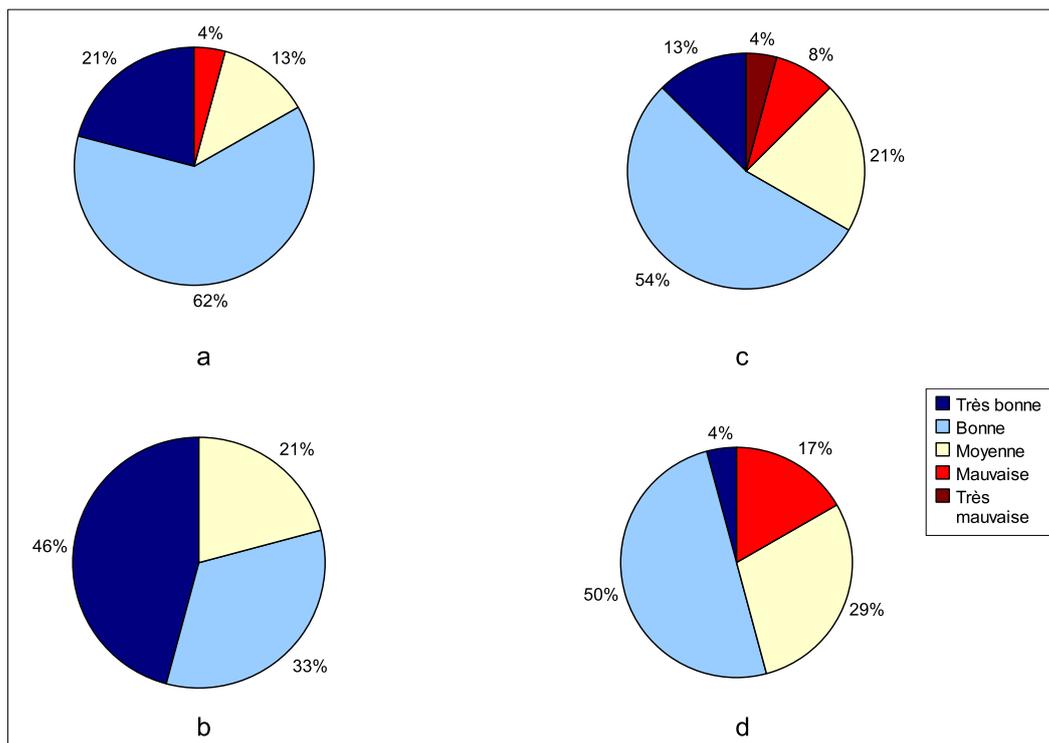


FIGURE 4.30 Facilité des interactions selon les utilisateurs pour chacune des tâches et pour chaque niveau de granularité. a : tâche de positionnement d'un objet virtuel avec le mode *exclusif*. b : tâche de positionnement d'un objet virtuel avec le mode *coopératif*. c : tâche de navigation avec le mode *exclusif*. d : tâche de navigation avec le mode *coopératif*.

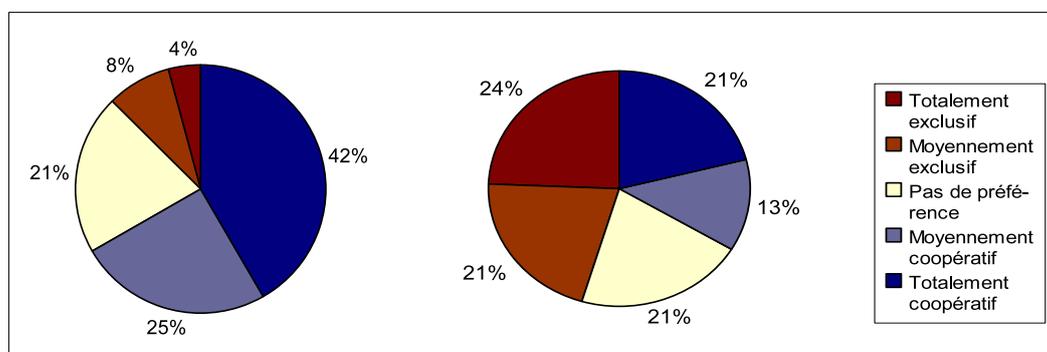


FIGURE 4.31 Préférence des utilisateurs pour le niveau de granularité de la gestion de la concurrence. Gauche : tâche de positionnement d'un objet virtuel. Droite : tâche de navigation.

différence est due à la gestion de la combinaison des interactions ce qui explique la différence des résultats pour la facilité des interactions selon les tâches. Enfin, notre hypothèse sur la préférence des utilisateurs pour une granularité fine de la gestion de la concurrence est vérifiée par la volonté des utilisateurs d'avoir la possibilité de réaliser des interactions simultanées sur un même élément (objet virtuel ou point de vue) au moyen de deux vues distinctes (79% pour la tâche de positionnement d'un objet virtuel

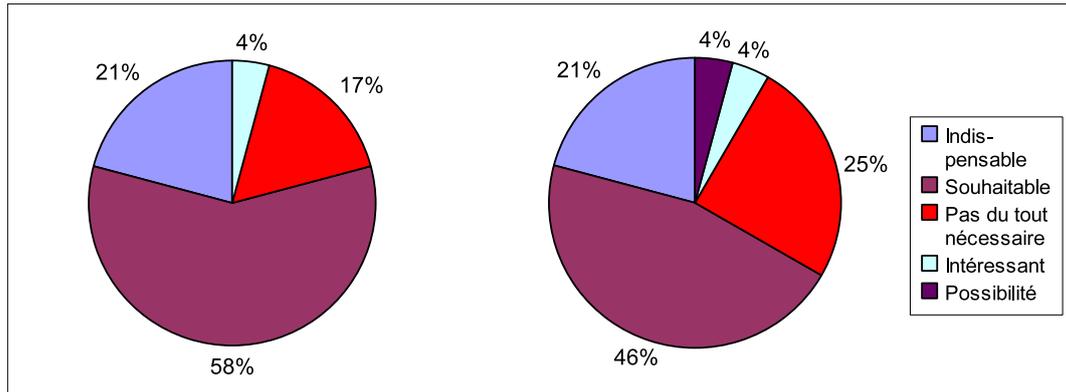


FIGURE 4.32 Importance de la possibilité de réalisation d'interactions simultanément sur un même élément au moyen de deux vues. Gauche : tâche de positionnement d'un objet virtuel. Droite : tâche de navigation par déplacement du point de vue.

et 71% pour la tâche de navigation).

4.5.4 Conclusion sur l'expérimentation

Le développement de machines à états permet de gérer la concurrence à plusieurs niveaux de granularité. L'expérimentation avait pour but de vérifier si un niveau de granularité fin de la gestion de la concurrence accroît les performances des utilisateurs et si les préférences des utilisateurs sont meilleures par rapport à un niveau de granularité grossier. Les résultats de l'expérimentation montrent que les performances ne sont pas meilleures avec un niveau de granularité plus fin ce qui réfute notre première hypothèse. Néanmoins, les utilisateurs préfèrent ce niveau de granularité (justifiant notre seconde hypothèse) à cause de la possibilité d'interagir avec deux représentations du même élément (objet virtuel ou point de vue) de façon simultanée. Nous soulignons qu'avec un niveau de granularité fin, les utilisateurs peuvent réaliser des interactions de la même façon qu'avec un niveau de granularité plus élevé mais le contraire est impossible.

4.6 Conclusion

La troisième étape de notre espace de conception a porté sur la gestion de la concurrence des interactions. Cette gestion de la concurrence intervient à la suite de la conception de la combinaison des interactions par *mapping* et de la conception à un niveau proche de la conception logicielle avec la conception par composantes logiques. Pour gérer la concurrence et la cohérence des données, nous avons proposé un système basé sur la gestion des permissions en modélisant des machines à états. Nous avons défini

plusieurs niveaux de gestion de la concurrence. L'expérimentation que nous avons réalisée pour comparer deux de ces niveaux montre que les utilisateurs préfèrent une gestion fine de la concurrence leur permettant une plus grande flexibilité dans la réalisation des interactions.

Après avoir présenté notre espace de conception avec ces trois étapes, il convient d'étudier la spécificité et la genericité de notre approche. Pour cela, nous étudions dans le chapitre suivant la modification de notre système de type TEAMViz présenté dans ce chapitre en changeant le repère de la vue 2D de la table tactile ainsi que l'extension de ce système à un système de simulation acoustique.

Chapitre 5

Evolutivité des systèmes et conséquences sur les trois phases de conception

5.1 Introduction

Notre espace de conception, présenté dans les chapitres 2, 3 et 4, est dédié à la conception d'interactions et de systèmes combinant des dispositifs multiples en entrée ainsi que des vues multiples présentées sur des surfaces d'affichage distinctes. Il est constitué de trois étapes allant de la conception de haut niveau des interactions à la gestion de concurrence dans les couches basses. Compte tenu du parti pris de centrer la conception des modalités autour des degrés de liberté spatiaux, de traiter des modalités en entrée et en sortie mais de ne s'attacher à aucun domaine applicatif spécifique, il convient d'étudier en détail le degré de généralité de notre approche. Ceci permettra d'apprécier l'extension rapide de notre approche à d'autres domaines d'application et la facilité de modification de certains paramètres de conception.

L'étude de la généralité de notre approche est réalisée en modifiant et en étendant le système MC de type TEAMViz (voir chapitre 4) conçu avec notre espace de conception à d'autres domaines applicatifs. Les tâches réalisables dans MC sont la manipulation d'un objet virtuel ainsi que la navigation dans l'environnement virtuel (voir figure 5.1).

Dans ce chapitre nous présentons deux cas d'étude de modification de MC. Dans le premier, nous modifions le repère du point de vue associé à la vue 2D et étudions son effet sur les trois étapes de conception. Dans le second cas nous étendons MC à un système de rendu visuel et audio des réflexions sonores dans une scène virtuelle 3D.

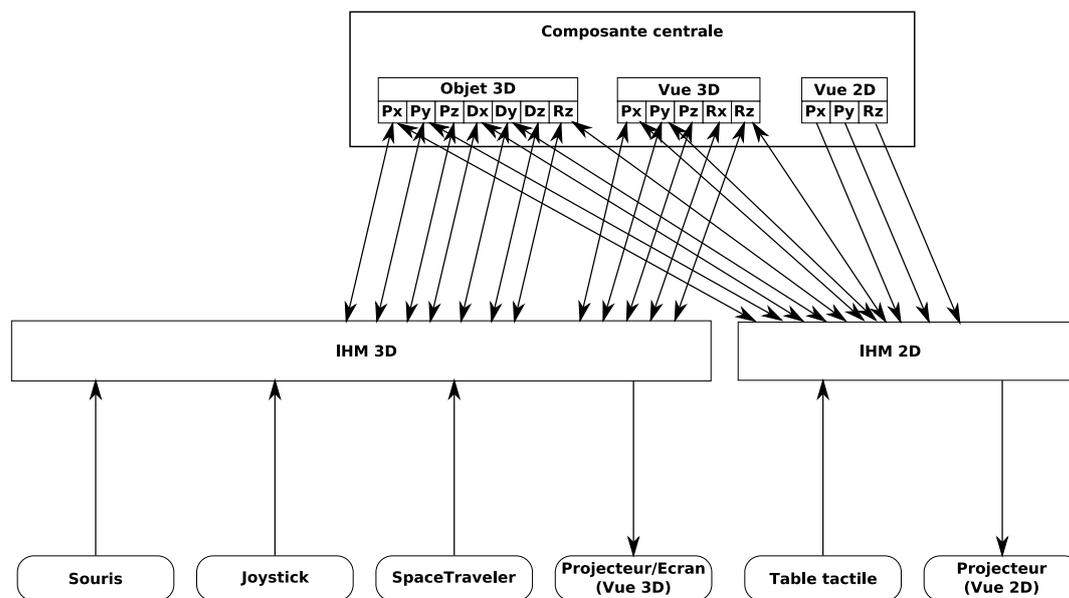


FIGURE 5.1 Architecture et tâches réalisables dans MC.

Nous proposons enfin des recommandations pour l'extension ou la modification d'un système reposant sur MC.

5.2 Modification du repère du point de vue 2D dans MC

Certains changements ou ajouts au système MC (présenté au chapitre 4) conçu dans notre espace de conception nécessitent des adaptations importantes lors des trois étapes de conception. Nous choisissons de changer le repère du point de vue associé à la vue 2D présentée sur la table tactile en réalisant une rotation du repère de 90 degrés autour de l'axe X. Ainsi, les axes X et Z de l'environnement virtuel sont projetés sur la table tactile au lieu des axes X et Y ce qui implique le changement des axes des degrés de liberté d'interaction dans le monde virtuel en utilisant la table. Ces modifications ont pour but de faciliter la réalisation de tâches d'empilement d'objets virtuels ou de décoration de murs en offrant la possibilité de réaliser des translations selon l'axe Z dans le monde virtuel avec la table tactile alors que la version initiale ne le permettait pas. Nous étudions ici les changements nécessaires au niveau des trois phases de conception relatifs à cette modification des dimensions d'affichage.

Dans MC, la vue 2D est une vue 2D horizontale de l'environnement virtuel. Les tâches élémentaires de manipulation d'un objet virtuel sont les translations et les re-dimensionnements selon les trois dimensions du repère du monde virtuel. La rotation horizontale d'un objet virtuel ainsi que la navigation dans l'environnement virtuel sont aussi possibles mais ne sont pas développées dans cette section. La figure 5.2 illustre la

conception par *mapping* de la combinaison des interactions dans MC pour la manipulation d'un objet virtuel dans le système de projection des axes X et Y (sur la table tactile) initial.

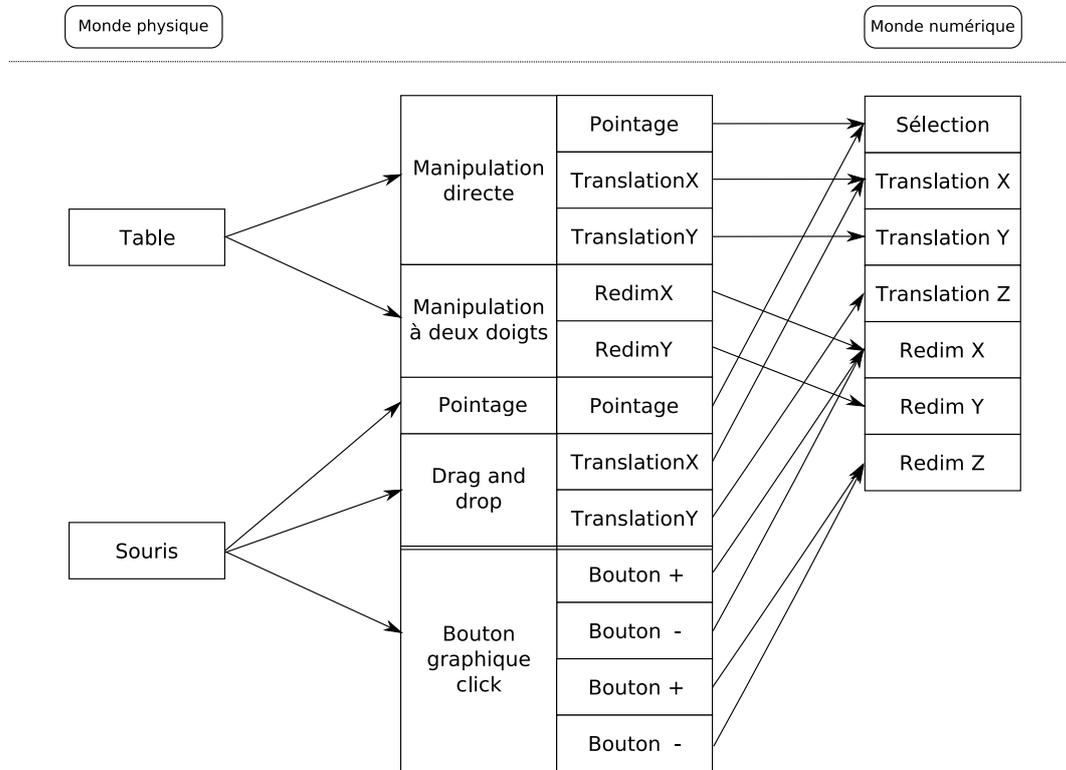


FIGURE 5.2 Conception par *mapping* des interactions pour la manipulation d'un objet virtuel dans le système MC initial avec projection des axes X et Y du repère du monde virtuel sur la table tactile.

Le changement de repère du point de vue associé à la vue 2D que nous réalisons nécessite une adaptation des connexions reliant la *table des techniques d'interaction* à la composante *tâche* du *mapping*. En effet, les connexions reliant les techniques d'interaction de la table tactile aux tâches élémentaires ne respectent plus la première et la troisième règle (Mapping1 et Mapping3). La translation (respectivement le redimensionnement) selon l'axe Y sur la table tactile (dans le monde réel) doit être reliée à la translation (respectivement au redimensionnement) selon l'axe Z des tâches élémentaires. Ces modifications respectent les règles Mapping1 et Mapping3 mais la règle de recouvrement des tâches (Mapping5) n'est plus satisfaite. Nous proposons d'utiliser la molette pour la translation selon l'axe Y et la molette avec un click sur le bouton droit de la souris pour réaliser le redimensionnement selon le même axe. Le résultat des modifications apportées à la conception par *mapping* est illustré dans la figure 5.3.

Le changement de repère de la vue 2D nécessite aussi des modifications dans la conception par composantes logiques mais sur trois niveaux. Le premier niveau est celui du repère de(s) composante(s) logique(s) et de l'élément du contrôleur de dialogue

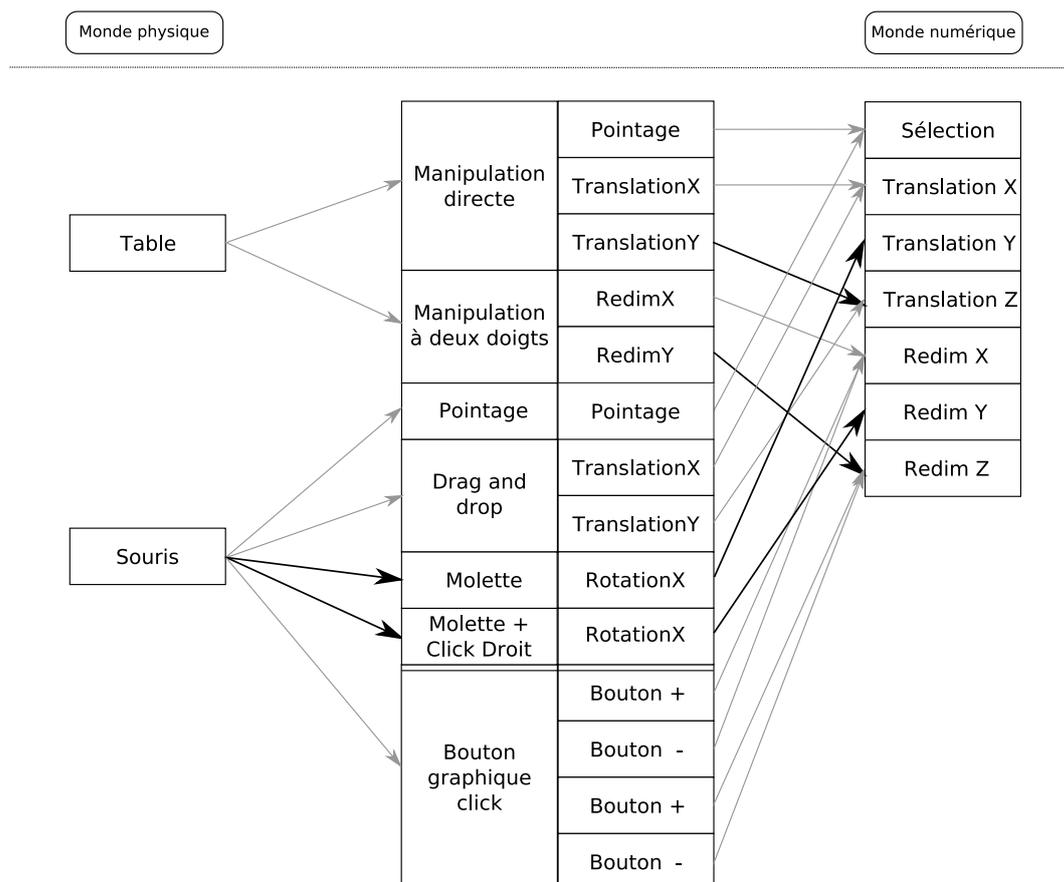


FIGURE 5.3 Conception par *mapping* des interactions dans MC après changement de repère de la vue 2D.

correspondant à la vue 2D. Le second est celui du repère du plan de projection pour l’affichage des composantes logiques connectées au dispositif de sortie 2D. La modification des connexions reliant les techniques d’interaction de la table tactile aux tâches élémentaires de la conception par *mapping* implique le troisième niveau de modifications. Ces changements s’adressent aux degrés de liberté en entrée des composantes logiques reliées à la table tactile ainsi qu’aux connexions de ces composantes logiques à un élément du contrôleur de dialogue. La figure 5.4 illustre le deuxième et le troisième niveau de modifications de la composante logique liée à la représentation de l’objet virtuel sur la vue 2D et ses connexions à l’élément du contrôleur de dialogue correspondant.

La modification des axes des degrés de liberté en entrée des composantes logiques ainsi que leurs connexions à l’élément du contrôleur de dialogue correspondant nécessite une nouvelle étude de la gestion de la concurrence si celle-ci est effectuée au niveau des degrés de liberté (et non au niveau des composantes logiques et des éléments du contrôleur de dialogue). Une re-modélisation des machines à états des degrés de liberté des composantes logiques modifiées doit être réalisée ainsi que celles correspondant aux degrés de liberté de l’élément du contrôleur de dialogue connectés à ces composantes

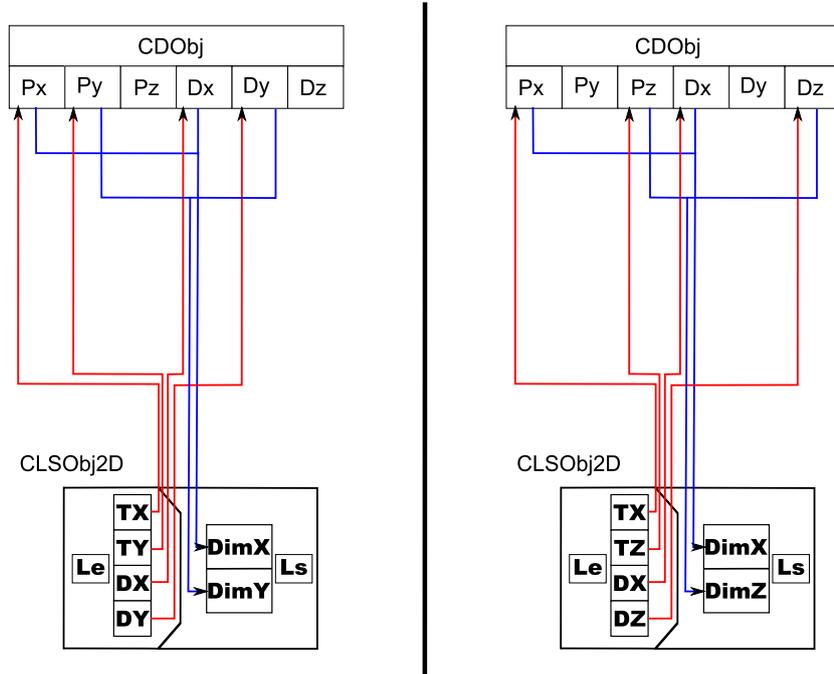


FIGURE 5.4 Modification de la composante logique de MC correspondant à la représentation 2D d'un objet virtuel et de ses connexions à l'élément du contrôleur de dialogue. Gauche : avant le changement de repère de la vue 2D. Droite : après le changement de repère de la vue 2D.

logiques. Si l'une des composantes logiques reliées à un degré de liberté d'un élément du contrôleur de dialogue a été altérée, alors la gestion de la concurrence doit être re-modélisée au niveau de tous les degrés de liberté de cet élément. Le nombre total d'états (avant simplification) reste identique mais la simplification des états change ce qui oblige à redéfinir les machines à états finales en étudiant la concurrence des degrés de liberté.

L'exemple de modification du système MC de cette section illustre que la rotation du repère de présentation de l'information sur l'une des vues implique des adaptations majeures au niveau de chaque phase de la conception. Ce nombre important de modifications est dû à une re-modélisation totale des interactions, du système final et une nouvelle gestion de la concurrence. Le remplacement d'un dispositif d'entrée par un autre dispositif d'entrée présentant des caractéristiques différentes implique la redéfinition des composantes *dispositifs d'entrée* et *table des techniques d'interaction* de la conception par *mapping*. Les changements au niveau des connexions reliant la *table des techniques d'interaction* et la composante *tâche* du *mapping*, au niveau de la conception par composantes logiques et au niveau de la gestion de la concurrence sont identiques à ceux pour la modification du repère d'une vue.

Le nombre important de modifications nécessaires dans les trois étapes de la conception de l'exemple illustré précédemment suggère d'automatiser certaines d'entre elles.

Cette automatisation des modifications pourrait intervenir au niveau du passage de la conception par *mapping* à la conception par composantes logiques et plus précisément dans la définition des dimensions et/ou des degrés de liberté des composantes logiques. La modification des machines à états pourrait être aussi automatisée surtout si la gestion de la concurrence n'est pas effectuée au niveau des degrés de liberté. Néanmoins, la totalité des modifications ne peut pas être automatisée comme par exemple : la définition de la *table des techniques d'interaction* d'un nouveau dispositif d'entrée au niveau de la conception par *mapping* et la définition des degrés de liberté concurrents pour la gestion de la concurrence au niveau des degrés de liberté. L'automatisation des modifications nécessite un travail supplémentaire qui n'est pas entrepris dans cette thèse.

5.3 Extension à un environnement de simulation acoustique

Nous étendons MC à un système de rendu visuel et sonore de l'acoustique des salles afin de poursuivre l'étude de la généricité de notre approche déjà abordée dans la section précédente. Dans cette section, nous présentons dans un premier temps le système de rendu visuel et sonore de l'acoustique des salles. Nous étudions ensuite la combinaison de notre architecture logicielle avec l'architecture logicielle de ce système ainsi que son implication au niveau des trois étapes de conception : conception par *mapping*, conception par composantes logiques et gestion de la concurrence.

5.3.1 Système de rendu visuel et audio interactif de l'acoustique des salles

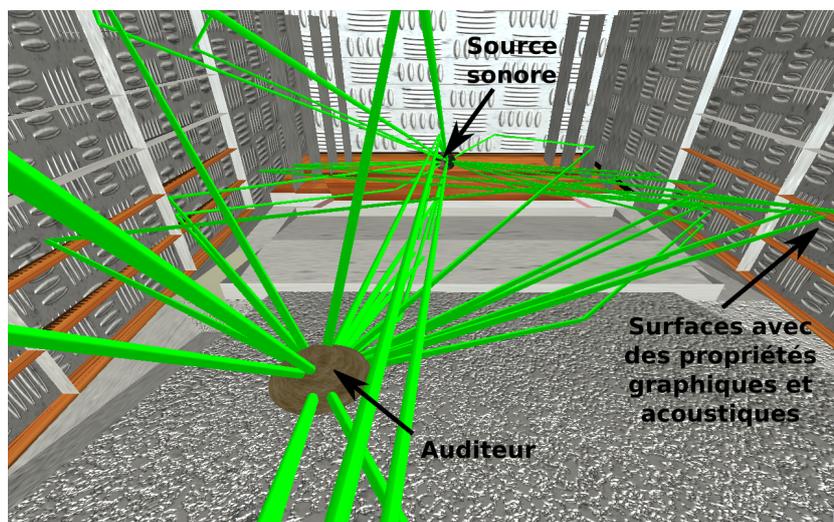


FIGURE 5.5 Vue 3D du système de rendu visuel et audio de l'acoustique des salles.

Le système interactif de visualisation et d'auralisation (i.e. rendu audio) des réflexions sonores dans un environnement virtuel permet de visualiser ces réflexions allant d'une source sonore omni-directionnelle vers un auditeur dans une scène architecturale fermée (e.g. une salle d'opéra). La source sonore est présentée graphiquement par une sphère tandis qu'un auditeur est représenté par une tête ou un cône (voir figure 5.5). L'architecture est représentée par un ensemble de triangles ayant des propriétés acoustiques variables utilisés pour le calcul du rendu audio et rendus graphiquement dans la vue 3D. Ce système permet ainsi l'auralisation tridimensionnelle du son issu d'une source sonore pour un auditeur virtuel en fonction des propriétés acoustiques et géométriques de la scène virtuelle. L'architecture logicielle de ce système (voir figure 5.6) est constituée de trois composantes :

- *Audio2graphical* est une application de rendu graphique tridimensionnel et de gestion des interactions en entrée. Elle contient initialement les données géométriques et les propriétés acoustiques de la scène virtuelle ;
- *EVERTims* est un simulateur de réflexions sonores en temps réel d'un son allant d'une (de) source(s) sonore(s) en direction d'un auditeur ;
- *auralisation* est une application pour le rendu sonore tridimensionnel du son allant de la (des) source(s) sonore(s) vers un auditeur qui utilise les résultats du calcul des réflexions sonores par EVERTims.

EVERTims calcule les réflexions sonores entre une source sonore et un auditeur par une méthode combinant le calcul de sources images [Allen & Berkley, 1979] et celui de "beam-tracing" (tracé de cônes) [Funkhouser *et al.*, 1998]. Nous renvoyons le lecteur à la lecture de [Ajaj *et al.*, 2008] sur la description du système et la partie graphique ainsi que [Noisternig *et al.*, 2008] sur la description du système de calcul acoustique et de l'auralisation associée.

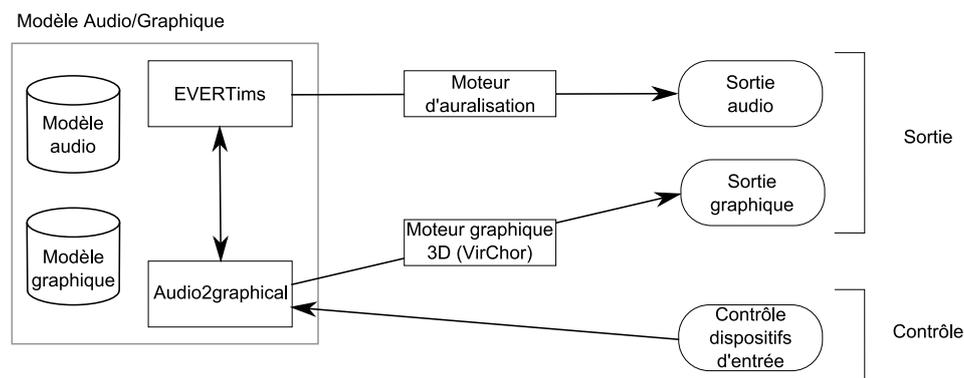


FIGURE 5.6 Architecture du système de rendu visuel et audio de l'acoustique des salles.

Le système décrit dans cette section permet une visualisation interactive et l'auralisation des réflexions sonores dans un espace virtuel fermé en temps réel. Il permet de

naviguer dans l'espace virtuel et de manipuler le(s) auditeur(s) ainsi que la position de la (des) source(s) sonore(s). La navigation dans la scène virtuel s'effectue en utilisant un joystick et/ou une souris pour les rotations. La souris permet aussi la manipulation des sources sonores et des auditeurs virtuels. Le déplacement et l'orientation des auditeurs est réalisable en utilisant le SpaceTraveler [3Dconnexion]. L'auditeur et le point de vue peuvent être fusionnés et dans ce cas la navigation déplace l'auditeur virtuel et le point de vue simultanément. A chaque modification de la position d'une source sonore ou d'un auditeur, le simulateur de réflexions sonores calcule les nouvelles réflexions et renvoie l'information à l'application de gestion des données (Audio2graphical) qui calcule la présentation de l'information graphique et au moteur d'auralisation pour le rendu audio 3D. Ces modèles de données audio/graphiques mis à jour sont ensuite utilisés par le moteur graphique 3D (VirChor [VirChor]) pour le rendu graphique.

En résumé, les tâches offertes aux utilisateurs sont : la translation selon les trois dimensions (TX, TY, TZ) des sources sonores, la translation et la rotation selon les trois dimensions (TX, TY, TZ, RX, RY, RZ) des auditeurs et la translation et la rotation du point de vue 3D selon les trois dimensions (TX, TY, TZ, RX, RY, RZ). Les positions et les orientations des réflexions sonores sont calculées automatiquement par le simulateur et ne peuvent pas être modifiées par les utilisateurs. Les faces de l'architecture (e.g. mur, cloison sol, etc.) sont fixes et ne sont pas modifiables dynamiquement ni par le système ni par les utilisateurs.

L'augmentation du système de rendu visuel et audio des réflexions sonores dans une scène virtuelle par une vue 2D supplémentaire de cette scène présentée sur une table tactile semble intéressante en raison de l'aspect collaboratif de la table. Cette extension permet de réunir des spécialistes de plusieurs domaines (e.g. architectes, acousticiens, etc.) autour de la table en leur offrant une vue 2D globale de la scène virtuelle. En plus, elle offre la possibilité de partager les interactions tactiles sur la table.

5.3.2 Extension selon les étapes de conception

L'extension de MC au système de rendu visuel et sonore de l'acoustique des salles permet de l'adapter à un nouveau cas d'application en ajoutant de nouvelles fonctionnalités. Nous avons réalisé cette extension en combinant l'architecture logicielle de MC avec celle du système de rendu visuel et audio de l'acoustique des salles. Cette combinaison a consisté à remplacer la composante IHM 3D de MC, présentée au chapitre 4, par l'application Audio2graphical. Nous étudions par la suite cette extension selon les trois étapes de notre espace de conception afin d'examiner les spécificités et les généralités de notre approche.

Conception par *mapping*

Dans MC, la conception par *mapping* est réalisée pour la combinaison d'une vue 2D projetée sur une table tactile et une vue 3D d'un même environnement virtuel présentée verticalement. Pour les dispositifs d'entrées et la *table des techniques d'interaction*, la conception par *mapping* de MC intègre les interactions effectuées en utilisant la table tactile, un SpaceTraveler, un joystick et une souris. La disposition et la détermination des vues ainsi que la définition des dispositifs d'entrées est la même pour MC et pour l'extension audiovisuelle. Notre approche s'applique donc à cette extension.

L'élément modifié est la composante tâche de la conception par *mapping*. Dans le système de rendu visuel et audio pour l'acoustique des salles, les tâches élémentaires sont : la translation 3D des sources sonores et la translation et la rotation 3D des auditeurs (les points d'écoute) et du point de vue pour la navigation. Dans le rendu visuel, les sources sonores et les auditeurs sont considérés comme des objets graphiques uniquement mais possèdent des propriétés non graphiques différentes utilisées pour le calcul des réflexions sonores et du rendu audio. De plus, les tâches élémentaires effectuées sur les auditeurs sont différentes de celles effectuées sur les sources sonores (translation et rotation pour les auditeurs et translation uniquement pour les sources sonores). Ainsi, dans la conception par *mapping* de l'extension de MC, les tâches élémentaires des source sonores, des auditeurs et du point de vue 3D sont séparées. Les tâches élémentaires du point de vue sont identiques dans la conception par *mapping* de MC et de son extension. Pour les sources sonores et les auditeurs, les tâches élémentaires sont basées sur celles effectuées sur les objets virtuels de MC (e.g. cube virtuel). Les redimensionnements 3D et leurs connexions sont supprimés pour ces deux types d'objets graphiques spéciaux. La rotation selon l'axe Z est supprimée pour les sources sonores mais des rotations selon les axes X et Y sont ajoutées pour les auditeurs. Ces changements impliquent des modifications au niveau des connexions reliant la *table des techniques d'interaction* à la composante *tâche*. Nous connectons la rotation selon l'axe X (respectivement l'axe Y et l'axe Z) du SpaceTraveler à la rotation selon l'axe X (respectivement l'axe Y et l'axe Z) de l'auditeur dans la composante *tâche*. Ces liaisons respectent les règles de combinaison des connexions reliant la *table des techniques interaction* à la composante *tâche* du *mapping*.

Conception par composantes logiques

Puisque les sources sonores et les auditeurs possèdent des représentations graphiques, les composantes logiques d'entrée/sortie pour les sources sonores et les auditeurs sont

identiques à celles pour les objets virtuels. Les redimensionnements sont supprimés des composantes logiques d'entrée/sortie et des éléments du contrôleur de dialogue correspondant aux auditeurs et aux sources sonores. En plus, une suppression de la rotation est réalisée sur les composantes logiques d'entrée/sortie et les éléments du contrôleur de dialogue correspondant aux sources sonores. Les rotations selon les axes X et Y sont ajoutées à la composante logique d'entrée/sortie reliée au SpaceTraveler ainsi qu'aux éléments du contrôleur de dialogue correspondant à chaque auditeur.

Dans le système de rendu visuel et audio de l'acoustique des salles, la modification dynamique de la scène virtuelle est impossible. Les composantes logiques d'entrée/sortie représentant les éléments géométriques de la scène sont donc transformées en composantes logiques de sortie et les éléments du contrôleur de dialogue correspondants deviennent de type CDI (initialisation) et non de type CDIT (initialisation tâche). Les positions et les orientations des rayons sonores sont modifiées dynamiquement selon les calculs réalisés par le simulateur et ne sont pas modifiables directement par l'utilisateur. Ainsi, un élément du contrôleur de dialogue et deux composantes logiques de sortie (une pour chacune des deux vues) ont été modélisés pour ces rayons sonores.

Pour résumer, certaines des modifications apportées au niveau des composantes logiques sont des restrictions. Ces restrictions se traduisent par la transformation de certaines composantes logiques d'entrée/sortie en composantes logiques de sortie ainsi que la transformation de certains éléments du contrôleur de dialogue de type CDIT en éléments du contrôleur de dialogue de type CDI. Certaines de ces restrictions se trouvent au niveau des degrés de liberté et ne sont pas des blocages de toutes les interactions. Dans le cadre de ces restrictions, des degrés de liberté sont supprimés des composantes logiques d'entrée/sortie et de certains des éléments du contrôleur de dialogue. Parmi les autres modifications réalisées, des degrés de liberté ont été ajoutés au niveau des composantes logiques d'entrée/sortie et des éléments du contrôleur de dialogue correspondants.

Gestion de la concurrence

Au niveau de l'application IHM 2D, la modélisation des machines à états pour les auditeurs et les sources sonores est basée sur la modélisation de celles conçues pour la manipulation des objets virtuels dans MC. Les machines à états correspondant à des degrés de liberté inutilisées (e.g. les redimensionnements) sont supprimées. L'impossibilité de modifier l'environnement virtuel implique le blocage de la manipulation des objets virtuels de la scène virtuelle (e.g. sol, plafond, etc.). Les machines à états relatives aux degrés de liberté de ces objets sont donc supprimées.

Aucune machine à états n'existe pour la gestion des blocages des interactions dans l'application Audio2graphical. La modélisation de telles machines à états est donc nécessaire au niveau de chaque degré de liberté des sources sonores, des auditeurs et du point de vue 3D. Cette modélisation est réalisée selon les étapes de conception des machines à états au niveau de chaque degré de liberté des composantes logiques (voir les sections 4.4.2 et 4.4.4). Par exemple, une machine à états est conçue pour la translation selon l'axe X d'une source sonore en utilisant la souris.

Dans l'application centrale du système MC initial (voir les section 4.3 et 4.4.4 du chapitre 4), des machines à états ont été modélisées au niveau des degrés liberté de chaque élément du contrôleur de dialogue lié à un objet virtuel. Basé sur la modélisation de ces machines à états, nous concevons des machines à états au niveau des degrés de liberté des éléments du contrôleur de dialogue correspondant aux sources sonores et aux auditeurs. Les machines à états relatives aux des degrés de liberté inutilisés sont supprimées et une étude de la concurrence des degrés de liberté est réalisée. Les machines à états sont ensuite modifiées en fonction de la concurrence entre les degrés de liberté : certaines machines à état nécessitent l'extension de machines à états existantes par des états de "blocage" et de "blocage avec mise en attente" ainsi que les transitions correspondantes. Le blocage des interactions pour la modification de la scène virtuelle (e.g. déplacement du plafond) implique la suppression des machines à états au niveau des degrés de liberté des éléments du contrôleur de dialogue correspondant à ces objets géométriques.

En résumé, l'extension de notre architecture à un système de rendu visuel et audio de l'acoustique des salles nécessite des adaptations mineures au niveau des trois étapes de notre espace de conception. Ce nombre restreint de changements est dû au recouvrement d'un nombre important des tâches élémentaires dans MC ainsi qu'à l'interactivité limitée dans le système de rendu visuel et audio de l'acoustique des salles (e.g. impossibilité de déplacer les murs virtuels). Au niveau d'Audio2graphical, nous avons modélisé des machines à états pour la gestion des blocages des interactions.

Les deux cas d'étude de modification de MC décrits dans ce chapitre présentent des caractéristiques différentes. Ces différences de caractéristiques impliquent des modifications distinctes au niveau de chacune des trois phases de notre espace conception. Dans la section suivante nous décrivons de façon plus générale les adaptations nécessaires au niveau de chacune des étapes de conceptions selon les changements apportés à un système déjà conçu avec notre espace de conception. Cette description s'appuie sur les deux cas particuliers présentés dans les sections 5.2 et 5.3 et étend ces modifications à un cas général en fonction de ses caractéristiques.

5.4 Conseils d'extension et de modification d'un système

Le changement de repère du point de vue associé à l'une des vues d'un système liée à un dispositif d'entrée nécessite des adaptations majeures au niveau des trois étapes de conception. A l'opposé, l'extension d'un système à un autre système possédant les mêmes caractéristiques (mêmes dimensions de présentation de l'information et mêmes dispositifs d'entrée) ne requiert que des modifications mineures. Nous présentons dans cette section quels sont les changements d'un système qui altèrent les trois phases de conception et ceux qui sont spécifiques à une seule de ces phases.

La rotation du repère d'une vue liée à un dispositif en entrée et le remplacement d'un dispositif en entrée par un autre dispositif sont des modifications influençant les trois phases de conception. Les changements intervenant au niveau des trois étapes de conception sont :

- l'ajout, la suppression ou le remplacement d'un dispositif en entrée ;
- l'ajout, la suppression ou la modification des dimensions d'une vue liée à un dispositif d'entrée ;
- l'ajout de nouvelles techniques d'interaction affectant des tâches élémentaires déconnectées des autres techniques d'interaction du même dispositif ;
- l'ajout ou la suppression d'une connexion reliant une technique d'interaction à une tâche élémentaire du *mapping* non couverte par une autre technique d'interaction du même dispositif d'entrée ;
- l'ajout et la suppression d'une tâche élémentaire.

Les adaptations nécessaires, en raison de la suppression d'une tâche élémentaire, au niveau des deux dernières étapes de la conception sont dépendantes des modifications dans la conception par *mapping*. Si cette suppression ne nécessite que la suppression des connexions liées à cette tâche alors les modifications sont mineures au niveau de la conception par composantes logiques (suppression de connexion) et de la gestion de la concurrence (suppression de machines à états et étude de la concurrence des degrés de liberté). Dans la cas contraire, des changements importants sont nécessaires dans les deux dernières étapes de notre espace de conception. Ainsi, les connexions reliant les composantes logiques aux éléments du contrôleur de dialogue sont modifiées et certaines composantes logiques peuvent être ajoutées ou supprimées. Au niveau de la gestion de la concurrence, des machines à états doivent être modélisées (respectivement supprimées) pour les nouvelles composantes logiques (respectivement pour les composantes logiques supprimées) ou pour leurs degrés de liberté selon le niveau de granularité de gestion de la concurrence. De plus, les machines à états des éléments du contrôleur de dialogue ou de leurs degrés de liberté doivent être changer selon les modifications des composantes

logiques.

L'altération de certaines caractéristiques d'un système déjà conçu avec notre espace de conception interviennent uniquement dans la conception par *mapping*. Ces changements sont liés à l'ajout, la suppression et le remplacement d'une technique d'interaction dans le cas où d'autres techniques d'interaction du même dispositif d'entrée sont connectées à la même tâche élémentaire. Seules la *table des techniques d'interaction* et les connexions la reliant aux tâches élémentaires sont modifiées. Certaines modifications des techniques d'interaction nécessitent une étude de la concurrence des degrés de liberté dans le cas d'une granularité fine de la gestion de la concurrence des interactions.

Les adaptations spécifiques à la conception par composantes logiques sont liées à la présentation de l'information. En effet, l'ajout ou la suppression d'éléments présentés sur une vue et non modifiables directement au moyen de cette vue n'affectent ni la conception par *mapping* ni la gestion de la concurrence. En plus, l'ajout, la suppression, le remplacement ou la modification des dimensions d'affichage d'une vue déconnectée de tout dispositif d'entrée est spécifique à la conception par composantes logiques. Ces évolutions impliquent l'ajout, la suppression ou la modification des composantes logiques de sortie liées à la vue ainsi que leurs connexions aux éléments du contrôleur de dialogue.

Certains changements des caractéristiques d'un système conçu avec notre espace de conception affectent exclusivement la dernière phase de conception. En effet, l'accroissement ou la diminution du degré de granularité de la gestion de la concurrence ne requiert des modifications qu'au niveau de la dernière étape de notre espace de conception. De plus, l'étude de la concurrence des degrés de liberté pour une granularité fine est réalisée uniquement à ce stade de la conception.

En résumé, plusieurs évolutions du système n'affectent que certaines étapes de la conception tandis que d'autres altèrent toutes les trois phases. Le tableau 5.1 résume à quelle(s) étape(s) de la conception interviennent ces modifications. Nous déduisons de cette étude les règles de spécificité et de généralité des phases de conception suivantes (les mots entre parenthèses précédés par ★ représentent des éléments de la figure 5.7 à des fins d'illustration) :

- l'ajout, la suppression, le remplacement ou la modification d'un dispositif d'entrée et/ou d'une vue liée à un dispositif d'entrée implique dans la plupart des cas la modification des connexions reliant la *table des techniques d'interaction* aux tâches élémentaires du *mapping* (★ CM1) ;
- l'ajout d'une tâche élémentaire nécessite l'ajout de connexions la reliant à la *table des techniques d'interaction* (★ CM1). Elle implique aussi son ajout au niveau d'un élément du contrôleur de dialogue (★ Px) ainsi que la modélisation de la

Modification du système	Conception par <i>mapping</i>	Conception par composantes logiques	Gestion de la concurrence
Ajout, suppression ou remplacement d'un dispositif en entrée	✓	✓	✓
Ajout, suppression ou modification des dimensions de présentation d'une vue liée à un dispositif en entrée	✓	✓	✓
Ajout d'une tâche élémentaire	✓	✓	✓
Ajout de nouvelles techniques d'interaction affectant des tâches élémentaires déconnectées des autres techniques d'interaction du même dispositif	✓	✓	✓
Ajout, suppression ou modification des connexions reliant des techniques d'interaction d'un dispositif en entrée à des tâches élémentaires non couvertes par ce dispositif	✓	✓	✓
Remplacement d'une technique d'interaction par une autre technique d'interaction utilisant le même dispositif en entrée et connectée à la même tâche	✓		Modifications potentielles
Ajout, suppression ou modification d'une vue ou de présentation de l'information déconnectée de tout dispositif d'entrée		✓	
Modification de la granularité de la gestion de la concurrence			✓

TABLE 5.1 Phases de conception touchées par la modification des caractéristiques d'un système.

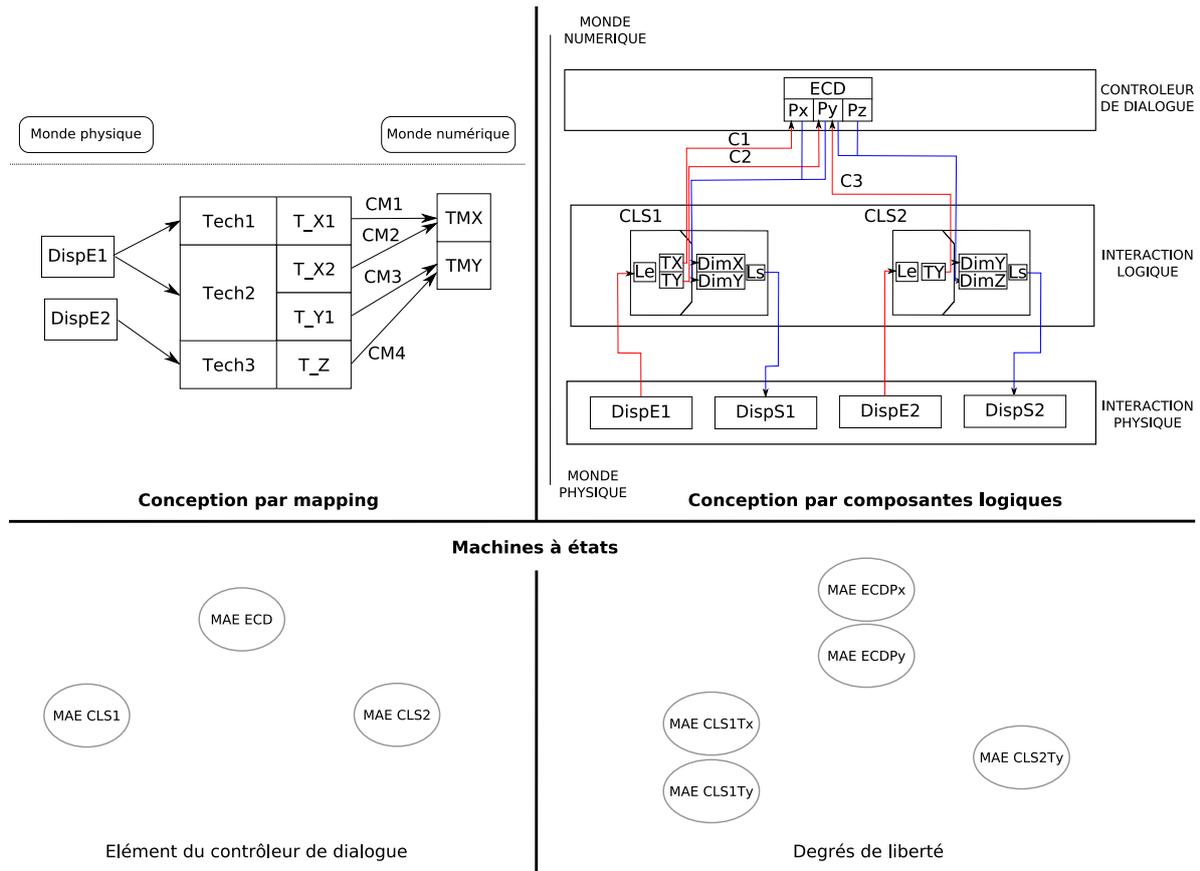


FIGURE 5.7 Exemple récapitulatif des trois phases de conception. En haut à gauche : une solution de la conception par *mapping*. En haut à droite : conception par composantes logiques correspondante. En bas : liste des machines à états pour une gestion de la concurrence au niveau des composantes logiques et des éléments du contrôleur de dialogue (en bas à gauche) et au niveau de leurs degrés de liberté (en bas à droite).

machine à états du degré de liberté correspondant dans le cas d'une granularité fine (★ MAE ECDPx);

- la suppression d'une tâche élémentaire (★ TMX) nécessite la suppression des connexions la reliant à des techniques d'interaction (★ CM1 et CM2) et potentiellement la modification des autres connexions au niveau de la conception par *mapping* (★ CM3 et CM4). Elle nécessite aussi sa suppression au niveau de l'élément du contrôleur de dialogue correspondant (★ Px). Dans le cas d'une granularité fine, cet élément du contrôleur de dialogue perd la machine à états correspondant à ce degré de liberté (★ MAE ECDPx);
- tout changement (suppression, ajout, modification) des connexions reliant la *table des techniques d'interaction* à la composante *tâche* du *mapping* (★ CM4) nécessite des modifications du même ordre au niveau des connexions reliant le degré de liberté de la composante logique au degré de liberté de l'élément du contrôleur de dialogue correspondant à cette tâche élémentaire (★ C3). L'exception à cette règle

- est le cas d'ajout ou de remplacement d'une technique d'interaction (★ Tech1) par une autre technique d'interaction connectée à un même dispositif d'entrée et aux mêmes tâches élémentaires (★ Tech2) ;
- la modification des connexions reliant (en entrée) des composantes logiques d'entrée ou d'entrée/sortie à des éléments du contrôleur de dialogue (★ C1) nécessite des adaptations des machines à états des degrés de liberté de ces composantes logiques et de ces éléments du contrôleur de dialogue dans le cas d'une granularité fine (★ MAE CLS1Tx et MAE ECDPx) ;
 - la suppression de toutes les connexions reliant (en entrée) une composante logique à un élément du contrôleur de dialogue (★ C1 et C2) nécessite la modification des machines à états de cette composante logique et de l'élément du contrôleur de dialogue correspondant dans le cas d'une gestion de la concurrence au niveau des composantes logiques et des éléments du contrôleur de dialogue (★ MAE CLS1 et MAE ECD) ;
 - l'ajout d'une connexion reliant (en entrée) une composante logique à un élément du contrôleur de dialogue alors qu'aucune autre connexion en entrée n'existe déjà entre eux (★ C3) nécessite la modification des machines à états de cette composante logique et celle de l'élément du contrôleur de dialogue correspondant dans le cas d'une gestion de la concurrence au niveau des composantes logiques et des éléments du contrôleur de dialogue (★ MAE CLS2 et MAE ECD) ;
 - l'ajout, la suppression ou le remplacement d'une technique d'interaction d'un dispositif en entrée liée à une tâche couverte par ce dispositif en entrée (★ Tech1) nécessite des modifications au niveau de la conception par *mapping*. Dans le cas d'une granularité fine, certaines modifications requièrent une étude de la concurrence des degrés de liberté et une modification des machines à états qui leurs sont associées ;
 - les modifications liées aux vues et à la présentation de l'information sur une vue déconnectée de tout dispositif d'entrée n'affectent que la conception par composantes logiques ;
 - la modification de la granularité de la gestion de la concurrence n'affecte que la dernière étape de conception.

5.5 Conclusion

Certaines évolutions d'un système conçu avec notre espace de conception affectent les trois étapes de conception tandis que d'autres modifient une étape précise. L'extension d'un système à un autre environnement avec notre approche est simple si les deux

systèmes présentent les informations selon les mêmes dimensions et intègrent les mêmes dispositifs en entrée. La suppression de tâches élémentaires est simple à réaliser si elle n'implique que la suppression des connexions les reliant à la *table des techniques d'interaction* dans la conception par *mapping* et aux composantes logiques dans la seconde phase de conception. Ainsi, la conception d'un système couvrant un large nombre de tâches élémentaires, de dispositifs en entrée et de techniques d'interaction permet son extension simple à un autre environnement possédant un sous-ensemble de ces éléments. MC est donc un système générique dont l'extension à d'autres cas d'étude et d'autres applications que l'architecture est simple.

Dans les chapitres précédents, nous nous sommes focalisé principalement sur la conception d'interactions pour la *manipulation d'objets virtuels*. Dans le chapitre suivant nous appliquons notre espace de conception à la *navigation* dans un environnement virtuel présenté en 2D sur une table tactile et en 3D verticalement.

Chapitre 6

Navigation multi-surfaces dans une scène 3D

6.1 Introduction

La navigation dans un environnement virtuel est le *déplacement dans cet environnement* [Bowman *et al.*, 2004, p. 136]. Elle s'effectue par la translation et la rotation du point de vue dans l'environnement virtuel qui représente le point virtuel depuis lequel l'utilisateur visualise la scène. Le point de vue est aussi appelé caméra dans le domaine de l'informatique graphique. Outre la translation de l'origine dans l'espace (x, y, z) et l'orientation du point de vue (voir pitch, yaw et roll de la figure 6.1), il se caractérise par un angle d'ouverture et par deux plans de coupe (proche et lointain).

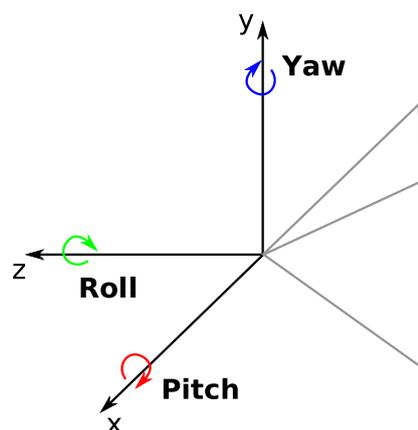


FIGURE 6.1 Les trois angles de rotation d'un point de vue.

Nous avons présenté dans le chapitre 1 plusieurs dispositifs et techniques d'interaction qui ont été développés pour la navigation dans un environnement tridimensionnel. Sur le plan cognitif, deux types de référentiels existent : les référentiels égocentriques

(i.e. la vue à la première personne) et les référentiels allocentriques (i.e. le référentiel du monde) [Bowman *et al.*, 2004, p. 232-234]. Un référentiel égocentrique offre des informations sur l'environnement virtuel relatives à la position de l'utilisateur dans l'environnement (e.g. jeux de tir à la première personne). Au contraire un référentiel allocentrique fournit des informations sur l'environnement depuis un point de vue extérieur à l'utilisateur (e.g. cartographie).

Dans le but de faciliter la navigation dans une scène virtuelle 3D, nous avons étudié la navigation dans un système de type TEAMViz. La navigation dans la scène virtuelle s'effectue en interagissant avec la table tactile et/ou en utilisant un joystick. Tout d'abord, nous présentons deux types de dispositions qui peuvent exister entre deux vues présentées sur des surfaces distinctes : la disposition des points de vue dans le monde virtuel et la disposition des surfaces de projection des vues dans le monde physique. Dans notre système, la prise en considération de la disposition des points de vue dans le monde virtuel correspond à une visualisation de la scène virtuelle dans la vue 2D dans un référentiel allocentrique. Cependant, le couplage de la disposition des points de vue dans le monde virtuel avec la disposition des surfaces de projection dans le monde physique correspond à une visualisation de la scène virtuelle dans la vue 2D dans un référentiel égocentrique. Nous exposons ensuite plusieurs techniques d'interaction ainsi que les différences de conception entre ces techniques. Nous concluons enfin ce chapitre par une expérimentation comparant deux techniques de navigation pour étudier les avantages respectifs des techniques intégrant des types différents de disposition des vues.

6.2 Disposition des vues

Nous différencions deux types de disposition de deux vues d'un même environnement virtuel présentées sur des surfaces distinctes. Le premier type de disposition est la disposition des points de vue dans le monde virtuel. Le second est la disposition des surfaces de projection des vues dans le monde réel. Cette différenciation des types de dispositions est importante en raison de la projection des vues distinctes d'une même scène virtuelle sur des surfaces d'affichage disposer différemment dans le monde physique. Nous définissons comme suit ces deux types de disposition.

6.2.1 Disposition des points de vue dans le monde virtuel

La disposition des points de vue de deux vues dans le monde virtuel est composée de l'ensemble des transformations nécessaires dans le monde virtuel pour atteindre le point

de vue de la seconde vue à partir de celui de la première vue. Ce type de disposition ne prend pas en considération la disposition dans le monde physique des surfaces d'affichage vues. Une visualisation possible de cette disposition consiste à présenter des informations d'un premier point de vue sur une seconde vue. Un exemple classique de ce type de visualisation est la représentation de la position d'un point de vue par une icône dans la seconde vue.

La visualisation de la disposition de deux points de vue dans le monde virtuel a été réalisée dans plusieurs environnements combinant une vue 2D présentée sur un dispositif tabletop et une (des) vue(s) 3D en perspective d'un même environnement virtuel affichée(s) verticalement. Forlines *et al.* [2006] présentent sur la vue 2D une icône de caméra pour illustrer le point de vue 3D affiché sur l'un des deux écrans verticaux (voir figure 6.2). En plus, la barre de menu utilisée dans la vue 2D est affichée en perspective sur les deux vues 3D pour faciliter la compréhension de la relation entre les vues à l'utilisateur. Finalement, le point de contact d'un doigt sur la vue 2D est affichée sur la vue 3D par un point (voir point B de la figure 6.2). La combinaison de ces visualisations de la disposition des points de vue dans le monde virtuel facilite la compréhension par l'utilisateur des relations existant entre les vues 3D et la vue 2D.



FIGURE 6.2 Présentation de la disposition des points de vue dans le monde virtuel dans le système développé par Forelines *et al.* [2006]. La vue projetée sur la table tactile (gauche) présente une icône en forme de caméra représentant le point de vue de l'une des vues 3D affichées verticalement (centre). Le point de contact avec la table tactile (A) est affiché sur les deux autres vues ainsi que la barre de menu (B). Source : [Forlines *et al.*, 2006].

Plusieurs techniques de navigation dans un environnement semblable à celui utilisé par Forelines *et al.* [2006] intègrent la disposition des points de vues dans le monde virtuel. Fjeld *et al.* [1999b, 1999c] ont développé quatre techniques d'interaction pour la navigation dans BUILD-IT [Fjeld *et al.*, 1998] qui est une interface combinant un dispositif tabletop tangible avec une vue complémentaire 3D présentée sur un mur. Les techniques GroundCatcher et FrameCatcher permettent de naviguer dans la vue 2D projetée horizontalement, tandis que les techniques Camera et Window permettent de naviguer dans la vue 3D verticale en interagissant avec le dispositif tabletop. La



FIGURE 6.3 Navigation dans BUILD-IT en utilisant des "briques". Gauche : plusieurs utilisateurs autour de la table tangible et une projection en perspective sur un mur. Droite : utilisation d'une "brique" pour interagir avec la table tangible. Source : [Fjeld *et al.*, 1999b].

technique GroundCatcher (en français attrape-sol) consiste à manipuler la totalité de la vue 2D. Dans la technique FrameCatcher (en français attrape-fenêtre), les utilisateurs manipulent un cadre virtuel présenté sur la vue 2D ; lors de la dé-sélection de ce cadre virtuel, une interpolation (animation) du cadre est réalisée de sa position et de sa taille actuelles vers la position et la taille de la vue. Dans la technique Camera, les utilisateurs interagissent avec une représentation du point de vue 3D dans la vue 2D. Pour la dernière technique appelée Window, les utilisateurs manipulent une fenêtre virtuelle représentant le bord de la vue 3D affichée verticalement sur un mur. La technique Camera a été étendue à la manipulation d'objets virtuels supplémentaires appelées EyeCatcher (en français attrape-œil) [Fjeld *et al.*, 1999a] pour permettre la réalisation des trois rotations et du zoom. Kuchar *et al.* [2007] travaillent sur une application pour le patrimoine consistant à visualiser un château en 3D sur un mur et en 2D sur une tablette tactile (vue 2D de type carte). La représentation du point de vue 3D est affichée sur la vue 2D sous la forme d'un avatar. L'espace autour de l'avatar est divisé en quatre parties. Le déplacement dans la vue 3D dans une direction donnée s'effectue par la sélection de l'une des quatre parties, située autour de la représentation du point de vue, correspondant à la direction souhaitée. Ces travaux de recherche se focalisent sur la navigation dans un environnement virtuel présenté sur deux vues différentes affichées sur des surfaces distinctes. Néanmoins, elles ne prennent pas en considération le positionnement des surfaces de projection de ces deux vues dans l'espace physique.

6.2.2 Disposition des surfaces de projection des vues dans le monde physique

Nous nous intéressons à la combinaison de deux vues d'un même environnement virtuel affichées sur des surfaces distinctes ce qui nous amène à identifier le second

type de disposition des vues qui est la disposition des surfaces de projection des vues dans le monde physique. Ce type de disposition entre deux surfaces de projection de vues se compose de l'ensemble des transformations nécessaires dans le monde physique pour passer du point de projection de la première vue à celui de la seconde vue. Par exemple, les transformations physiques entre deux surfaces de projections de vues, une horizontale et l'autre verticale, sont une rotation de 90 degrés, une rotation d'un angle α et une translation (voir 6.4). La présentation de l'information correspondant à ce type de disposition consiste à afficher sur une vue la disposition physique de l'écran présentant la seconde vue (e.g. par une icône représentant le second écran). La différence par rapport à la visualisation de la disposition des points de vue dans le monde virtuel est la présentation de l'information sur la position d'un élément du monde réel (la surface d'affichage) et non de la position d'un point de vue dans le monde numérique.

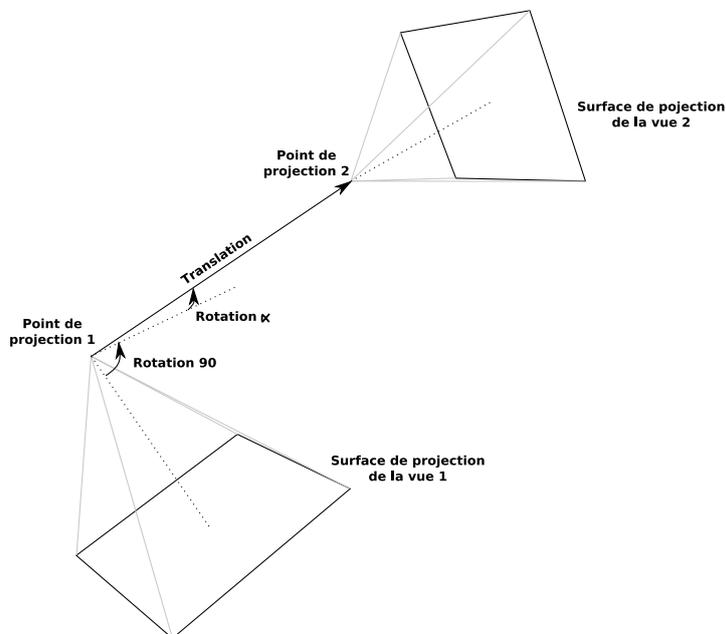


FIGURE 6.4 Disposition des surfaces de projection dans le monde physique dont l'une disposée horizontalement et l'autre verticalement.

Wigdor *et al.* [2006] appellent cette présentation de la disposition des surfaces d'affichage la *visual connectivity* (soit en français la connectivité visuelle). Ils visualisent cette *connectivité* en affichant des patrons sur les bords de la surface tactile de la même couleur que le patron affiché en bas de l'écran vertical le plus proche du bords en question (voir figure 6.5). Cependant, ce travail n'est pas appliqué à la présentation de deux vues d'une même scène virtuelle. Dans la littérature, la disposition dans le monde physique des surfaces de projection de deux vues d'une même scène n'ont pas été étudiées pour un environnement virtuel présenté en 3D sur un écran vertical et en 2D sur une table tactile. Dans ce cas la représentation de la connectivité visuelle se ferait de manière

identique à celle de Wigdor *et al.* [2006].

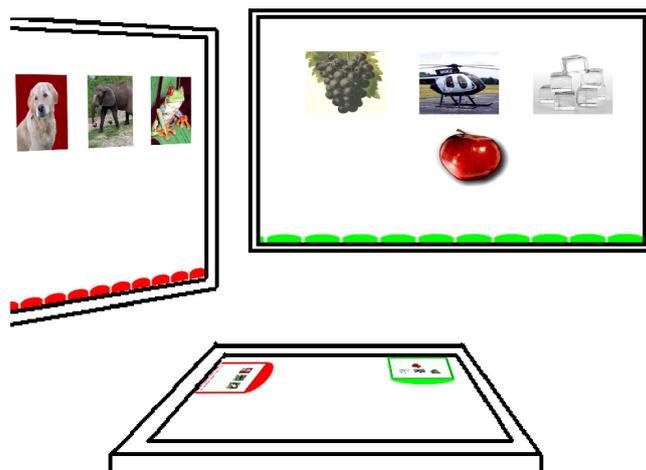


FIGURE 6.5 Présentation de la disposition des surfaces de projection des vues dans le monde physique dans le système développé par Wigdor *et al.*. Source : [Wigdor *et al.*, 2006].

L'enjeu de la présentation de la disposition des surfaces de projection des vues dans le monde physique est de faciliter la compréhension de la connexion entre les dispositifs d'affichage. Nous formulons l'hypothèse qu'une réduction des différences entre la disposition des points de vue de deux vues dans le monde virtuel et la disposition des surfaces de projection de ces vues dans le monde physique entraîne la réduction des transformations mentales nécessaires par l'utilisateur pour comprendre la relation entre ces vues. Cette réduction des transformations mentales (et surtout des rotations mentales) devrait faciliter le travail nécessitant un passage fréquent d'une vue à l'autre. Considérons une vue 2D de dessus d'une scène virtuelle projetée sur une surface horizontale et une vue 3D en perspective de cette même scène projetée sur un écran vertical. Une représentation du point de vue 3D est affichée dans la vue 2D. Les différences entre la disposition des points de vue dans le monde virtuel et la disposition des surfaces de projection de ces vues dans le monde physique sont réduites lorsque la représentation du point de vue 3D sur la table tactile pointe vers la position physique de l'écran vertical. Dans ce cas, les éléments visibles en 2D devant la représentation du point de vue 3D sont visibles en 3D sur l'écran vertical. A l'opposé, ces différences sont accentuées lorsque cette représentation du point de vue 3D dans la vue 2D ne pointe pas vers l'écran vertical.

6.3 Techniques d'interaction pour la navigation

Nous avons développé plusieurs techniques d'interaction, dédiées au système TEAM-Viz. Le point de vue 3D est représenté sur la vue 2D par une icône en forme de caméra.

Cette icône permet de visualiser la position et l'orientation horizontales du point de vue 3D dans la scène virtuelle (disposition des points de vue dans le monde virtuel décrite dans 6.2.1). La manipulation de l'icône de la caméra s'effectue sur la table tactile et peut être augmentée par l'ajout d'un dispositif complémentaire (joystick chargé de contrôler la rotation par exemple). Cependant, nous nous focalisons dans cette section sur les interactions effectuées en utilisant la table tactile exclusivement.

Les quatre premières techniques d'interaction sont appelées :

- *Technique d'Interaction par Manipulation de Caméra (TIMC*)*;
- *Technique d'Interaction de Suivi du Doigt (TISD*)*;
- *Technique d'Interaction d'Orientatation du Doigt (TIOD*)*;
- *Technique d'Interaction de Manipulation à Deux Doigts (TIMDD*)*.

Elles intègrent uniquement la disposition des points de vue dans le monde virtuel. Ces techniques, dédiées à la navigation horizontale (translation et rotation dans le plan horizontal), se différencient principalement par la façon de réaliser la rotation (yaw). La cinquième technique d'interaction, appelée Technique d'Interaction par Couplage des Vues (*TICV**), est aussi consacrée à la navigation horizontale mais couple les deux types de disposition des vues. La Technique d'Interaction de Suivi du Doigt Augmentée (*TISDA**) constitue la dernière technique présentée dans cette section. Elle est identique à la technique de suivi du doigt mais intègre en plus la rotation verticale (pitch).

6.3.1 Technique d'interaction par manipulation de caméra

Dans *TIMC* les utilisateurs déplacent l'icône de caméra sur la table tactile par manipulation directe (voir figure 6.6). Les déplacements de cette icône sont immédiatement transposés sur le point de vue 3D présenté sur un écran vertical. La manipulation directe de la position du point de vue 3D sur la table tactile offre des translations rapides et surtout intuitives de la caméra 3D dans l'espace virtuel.

La rotation horizontale dans *TIMC* est réalisée par l'utilisation d'un manipulateur situé sur l'un des coins de l'icône de caméra (voir figure 6.7). L'utilisation d'un manipulateur est identique à la technique de désignation d'un espace de sélection et de manipulation spécifique à la rotation développée par Hancock *et al.* [2006]

Cette technique sépare la rotation de la translation horizontale qui ne peuvent donc pas être réalisées simultanément. L'avantage de cette séparation est la possibilité d'effectuer la rotation (respectivement la translation) sans modifier la position du point de vue (respectivement l'orientation du point de vue). Cependant, cette séparation ne permet pas d'effectuer des interactions diagonales entre les degrés de liberté (translation

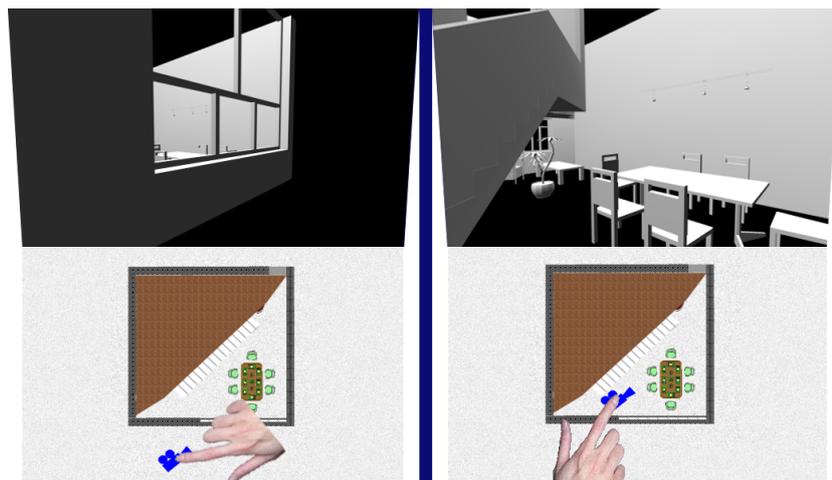


FIGURE 6.6 Translation par manipulation directe d'une icône en forme de caméra représentant le point de vue 3D affichée sur la vue 2D. Bas : vue 2D projetée sur la table tactile. Haut : la vue 3D correspondante affichée sur un écran vertical.

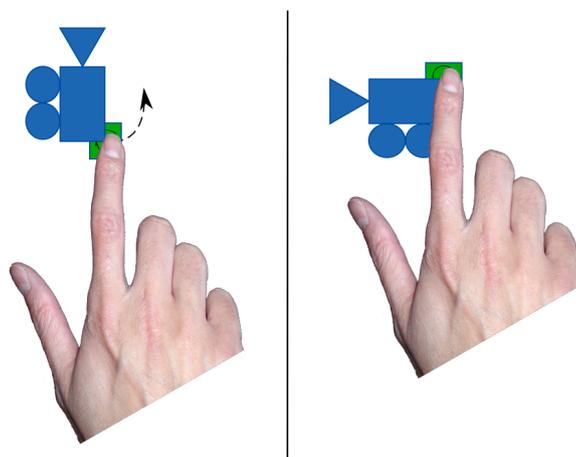


FIGURE 6.7 Icône en forme de caméra représentant le point de vue 3D affichée sur la vue 2D et le manipulateur associé pour la réalisation des rotations dans *TIMC*.

et rotation simultanées) et peut nécessiter des commutations multiples entre les translations et les rotations pour réaliser certaines tâches de navigation (e.g. suivi d'un objet virtuel se déplaçant dans l'environnement virtuel). Un autre inconvénient de cette technique est l'affichage du manipulateur dans le coin de l'icône qui implique une occlusion supplémentaire des objets virtuels situées à cet endroit de la vue 2D.

6.3.2 Technique d'interaction de suivi du doigt

Dans *TISD* l'icône de la caméra suit la trajectoire du doigt de l'utilisateur (voir figure 6.8). L'orientation de l'icône de la caméra, qui représente le point de vue 3D, est calculée en fonction de la tangente à la trajectoire du déplacement. Les mouvements

(en translation et en rotation) de l'icône dans la vue 2D sont immédiatement appliqués au point de vue 3D. *TISD* est similaire à la technique RNT [Beaudouin-Lafon, 2001, Kruger *et al.*, 2005] utilisée par Hancock *et al.* [2006] sur les tables tactiles mais elle est ici appliquée à la navigation dans un environnement virtuel.

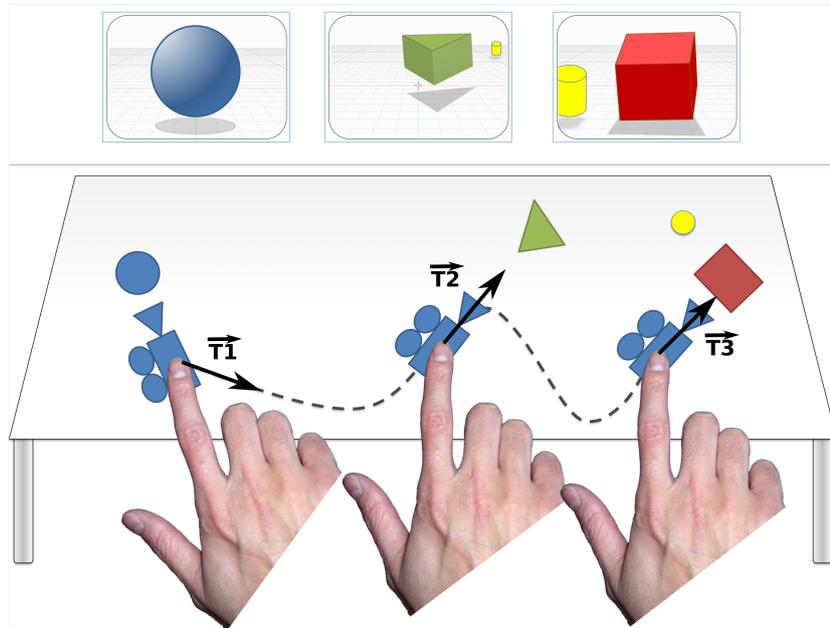


FIGURE 6.8 Technique d'interaction du suivi du doigt sur la table tactile pour la navigation dans un environnement 3D avec orientation de la caméra dans la direction de la tangente à la courbe de déplacement.

L'avantage de *TISD* est la possibilité de naviguer en diagonale entre les degrés de liberté (i.e. translation et rotation simultanée). De plus aucun manipulateur graphique n'est utilisé évitant l'accentuation de l'occlusion sur la vue 2D. Néanmoins, elle présente plusieurs restrictions : la rotation du point de vue ne peut être effectuée sans déplacement et les translations latérales pour lesquels la caméra serait perpendiculaire à la tangente de la trajectoire sont impossibles. Ces restrictions exigent dans certains cas que l'utilisateur effectue des déplacements importants pour un déplacement souhaité restreint. Un exemple d'une telle exigence est de devoir se déplacer sur une demi-ellipse afin de se positionner de façon parallèle au point de vue courant mais à un autre point (voir figure 6.9).

6.3.3 Technique d'interaction d'orientation du doigt

Les translations horizontales dans *TIOD* sont réalisées comme dans *TIMC* en manipulant directement la position de la caméra par le biais d'une icône pouvant être translaturée avec un doigt. Ces deux techniques se différencient exclusivement au niveau de la rotation. Dans *TIOD*, l'orientation horizontale du doigt touchant l'icône de la

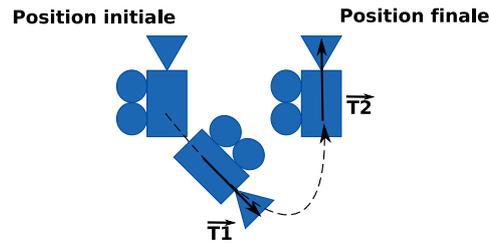


FIGURE 6.9 Déplacement nécessaire pour se positionner parallèlement à la vue 3D courante en un autre point situé perpendiculairement à l'axe de vue en utilisant *TISD*.

caméra définit l'orientation de cette icône et ainsi l'orientation du point de vue 3D dans la scène virtuelle. Cette technique est similaire à la TNT-hand [Liu *et al.*, 2006] mais elle est appliquée à la navigation. Pour l'acquisition de l'information sur l'orientation du doigt de l'utilisateur, nous identifions quatre types d'acquisition selon les dispositifs utilisés :

- acquisition par les capteurs intégrés dans la surface tactile ;
- vision par ordinateur par en-dessous de la surface tactile ;
- vision par ordinateur par au dessus de la surface tactile ;
- augmentation du doigt de l'utilisateur par un dispositif de tracking.

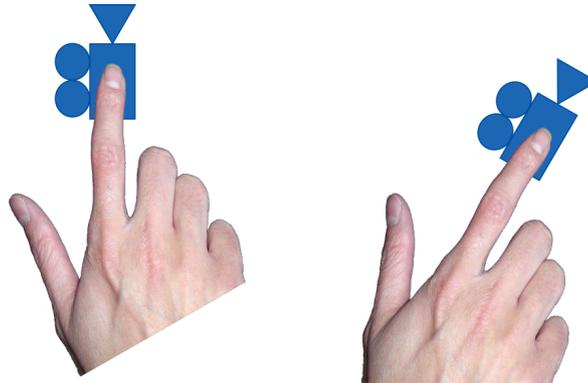


FIGURE 6.10 *TIOD* sur la table tactile pour la navigation dans un environnement 3D par orientation de la caméra en fonction de l'orientation du doigt.

L'avantage principal de *TIOD* est la possibilité de réaliser la translation et la rotation de façon continue ou séparée sans ajouter des manipulateurs graphiques. En effet, l'utilisateur déplace et oriente son doigt simultanément en manipulant l'icône de caméra pour naviguer dans l'environnement virtuel combinant ainsi la translation et la rotation. Pour modifier l'orientation uniquement du point de vue 3D, il suffit que l'utilisateur sélectionne l'icône de caméra avec son doigt et modifie son orientation en tournant son doigt autour du point de contact. Pour réaliser une translation du point de vue 3D sans modifier son orientation, l'utilisateur sélectionne l'icône de caméra avec son doigt et le déplace sans modifier l'orientation de son doigt sur la table tactile. Ainsi, aucun

manipulateur n'est nécessaire pour la rotation, ce qui permet d'éviter l'accroissement de l'occlusion sur la vue 2D.

Les désavantages de *TIOD* se situent au niveau de l'interaction et des techniques d'acquisition de l'orientation du doigt. Au niveau de l'interaction, *TIOD* oblige l'utilisateur à constamment contrôler l'orientation de son doigt durant les interactions ce qui peut entraîner une gêne due aux contraintes mécaniques de la main et du bras (e.g. difficulté de se déplacer loin en gardant le doigt pointé vers soi). Au niveau technique, l'orientation du doigt perpendiculairement à la surface tactile ne permet pas de détecter l'orientation horizontale du doigt. L'utilisateur est donc obligé de garder son doigt de façon horizontale pour que l'orientation en soit correctement détectée. De plus chaque technique d'acquisition de l'orientation du doigt présente des inconvénients :

- les capteurs intégrés dans les surfaces tactiles détectent uniquement la surface du touché et ne peuvent pas prédire dans quel sens pointe le doigt. Dans le cas d'un contact léger, la précision est mauvaise ;
- une grande partie des techniques de vision par ordinateur par en-dessous de la surface tactile ne détectent que les points de contact sur la surface entraînant les mêmes problèmes que ceux pour les capteurs intégrés : le sens du doigt est impossible à prédire et la précision est aussi mauvaise si les contacts sont légers ;
- les techniques de vision par ordinateur par au dessus de la table tactile permettent de visualiser la totalité de la main évitant le problème d'ambiguïté sur l'orientation du doigt mais présente des problèmes d'occlusion (e.g. lorsque une main occulte le doigt ou que le doigt est très incliné verticalement) ;
- l'augmentation du doigt par un dispositif de tracking peut être gênante pour l'utilisateur. Les techniques filaires empêchent les utilisateurs de travailler ensemble tandis que les techniques sans fil sont lourdes car elles forcent de porter des batteries.

6.3.4 Technique d'interaction de manipulation à deux doigts

De plus en plus de techniques d'interaction sur les surfaces tactiles profitent des technologies multi-points (de contact) à plusieurs doigts de la même main ou de mains différentes. La rotation et la translation sur une table tactile en utilisant deux doigts a été étudiée par Hancock *et al.* [2006]. Nous appliquons cette technique à *TIMDD* pour naviguer dans un environnement virtuel en manipulant l'icône de caméra à deux doigts (voir figure 6.11).

Ainsi que pour *TIOD*, l'avantage majeur de *TIMDD* est la possibilité de naviguer en translation et en rotation de façon simultanée tout en rendant facile la réalisation

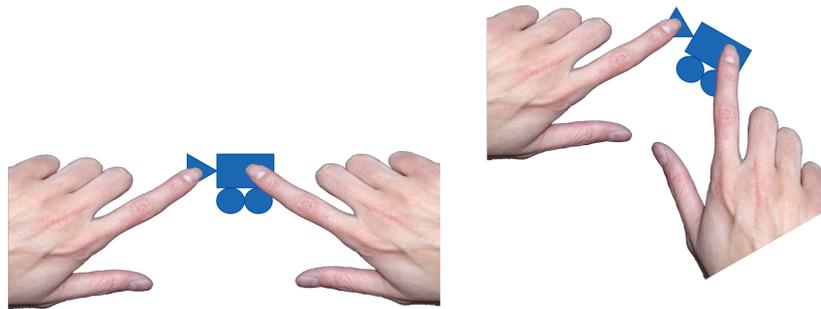


FIGURE 6.11 Technique d'interaction de la manipulation à deux doigts sur la table tactile pour la navigation dans un environnement 3D.

des translations et des rotations de façon disjointe. Néanmoins, l'inconvénient de cette technique est le non respect du couplage entre la position des doigts et la représentation de la caméra. L'icône de caméra reste de taille constante même si la distance entre ses deux doigts varie pour éviter des occlusions supplémentaires de la vue 2D et pour permettre à l'utilisateur de pouvoir la sélectionner facilement. L'utilisation de deux doigts de la même main limite cette variation mais certaines rotations d'angles supérieures à 180 degrés ne sont pas réalisables en une fois sous cette contrainte. L'utilisation de deux doigts de mains différentes mobilise les deux mains alors que l'une d'entre elles pourrait être disponible pour d'autres tâches.

6.3.5 Technique d'interaction de couplage des vues

Contrairement à *TIMC*, *TISD*, *TIOD* et *TIMDD* qui intègrent uniquement la disposition des points de vue dans le monde virtuel, *TICV* intègre en plus la disposition des surfaces de projection dans le monde physique. Dans *TICV* une représentation de l'écran vertical est affichée sur la vue 2D du côté du bord de la table tactile le plus proche de l'écran. Cette représentation de l'écran est illustrée dans la figure 6.12 par le rectangle noir sur le bord supérieur de la vue 2D. En plus de l'intégration de la disposition des surfaces de projection des vues dans le monde physique, un couplage des deux types de disposition des vues est réalisé du fait que l'icône de la caméra reste fixe dans la vue 2D et pointe constamment vers la représentation de l'écran vertical (voir figure 6.12). Durant la navigation, la position et l'orientation du point de vue 2D par rapport au repère du monde virtuel sont modifiées. La position et l'orientation du point de vue 3D est couplée à celles du point de vue 2D. Le déplacement du point de vue 2D dans le monde virtuel implique donc le déplacement du point de vue 3D dans le monde virtuel. Cognitivement, cela semble difficile à intégrer pour l'utilisateur puisque la caméra de la vue 2D est fixe par rapport à lui (il reste au dessus de la table). Ainsi, il a l'impression de déplacer le repère 3D du monde virtuel avec la table tactile.

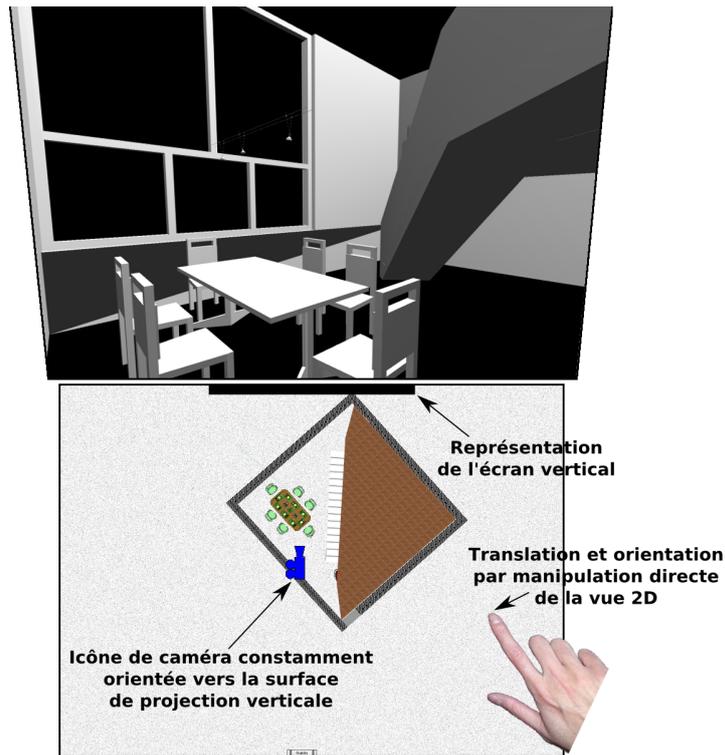


FIGURE 6.12 Technique d'interaction de couplage des vues pour la navigation dans un environnement 3D. Bas : vue 2D projetée sur la table tactile ainsi que la représentation de l'écran vertical (rectangle noir en haut). Haut : la vue 3D correspondante affichée sur un écran vertical.

Les translations sont réalisées par manipulation directe de la projection de la vue 2D dans *TICV*. En effet, un utilisateur sélectionne et déplace n'importe quel point de la projection de la vue 2D ne correspondant ni à un objet virtuel ni à l'icône de caméra. Ces déplacements de la projection de la vue 2D sur la table tactile sont des déplacements du point de vue 2D dans le repère du monde virtuel. La représentation du point de vue 3D dans la vue 2D reste fixe par rapport au point de vue 2D en raison du couplage des deux points de vue. Cela entraîne que les translations obtenues par le déplacement du point de vue 2D dans le monde virtuel sont appliquées à la représentation du point de vue 3D et transposées directement au point de vue 3D présenté sur l'écran vertical. Pour la rotation, un manipulateur est affiché sur le coin de l'icône de caméra dont la sélection entraîne un changement de mode de manipulation de la projection de la vue (mode translation et mode rotation). En mode rotation, les utilisateurs manipulent alors la projection de la vue 2D pour réaliser une rotation de son point de vue autour de l'axe vertical passant par l'icône de caméra. Le centre de l'icône de caméra est le centre de rotation du point de vue 3D ce qui permet d'être cohérent avec la vision égocentrique de l'utilisateur et ainsi limiter son effort cognitif.

Pour éviter l'utilisation de modes pour réaliser les translations et les rotations, la

manipulation à deux doigts de la vue 2D semble être une solution possible mais elle présente certains inconvénients. En effet, la rotation de la vue 2D (implicitement du point de vue 2D) est réalisée dans ce cas en prenant appui sur deux points de la projection de la vue 2D avec deux doigts et en faisant suivre ces points lors des déplacements des doigts. Le centre de rotation peut donc être modifié durant l'interaction à deux doigts pour préserver les points de contact. Cette modification du centre de rotation n'est pas souhaitable du fait que dans *TICV* il se situe au centre de l'icône de caméra.

Cette technique est comparable pour les utilisateurs à la métaphore *scene in hand* (en français la scène dans la main) de Ware et Osborne [1990] qui consiste à manipuler la totalité de la scène virtuelle en utilisant un dispositif d'entrée pour naviguer dans la scène virtuelle. Cependant, dans *TICV* la navigation est réalisée en prenant appui sur la projection d'une vue dans le but de naviguer dans une seconde vue. En comparaison avec le travail de Ware et Osborne [1990], les techniques *TIMC*, *TISD*, *TIOD* et *TIMDD* sont semblables à la métaphore *eye in hand* (en français œil dans la main) qui consiste à manipuler un dispositif d'entrée assimilé au point de vue dans la scène virtuelle.

L'avantage principal de *TICV* en comparaison avec les autres techniques (*TIMC*, *TISD*, *TIOD* et *TIMDD*) est l'orientation fixe de l'icône de caméra sur la vue 2D qui pointe vers la représentation de l'écran vertical et donc vers l'écran vertical physique. En raison de l'alignement de l'utilisateur, de l'icône de caméra et de l'écran vertical, l'utilisateur n'a pas à effectuer de rotation mentale pour comprendre la relation entre les deux vues. Cela revient à orienter une carte dans la direction du déplacement comme cela est fait dans les systèmes de navigation GPS. Notre hypothèse (voir fin de la section 6.2) stipule que la réduction des différences entre la disposition des points de vue dans le monde virtuel et la disposition des surfaces de projection dans le monde physique entraîne la réduction des transformations mentales nécessaires par les utilisateurs pour comprendre la relation entre les vues. Ainsi, la compréhension de la vue 3D par rapport à la vue 2D et le passage d'une vue à l'autre devraient être facilités selon cette hypothèse. Le second avantage de *TICV* est la zone de sélection très large pour la navigation constituée par toute la surface de projection de la vue 2D (sauf la projection d'un objet virtuel ou l'icône de la caméra). Le dernier avantage de cette technique est l'affichage constant et de taille importante sur la vue 2D de la zone de la scène virtuelle située derrière le point de vue 3D.

Les déplacements et les rotations du point de vue 2D dans le monde virtuel et ainsi les déplacements et les rotations de la vue 2D sur la table tactile constituent l'inconvénient majeur de *TICV*. Ces changements défavorisent la conservation d'une représentation

mentale cohérente de l'environnement virtuel. Ainsi, les utilisateurs peuvent se perdre plus facilement dans le monde virtuel en comparaison avec les autres techniques de navigation présentées dans cette section.

6.3.6 Technique d'interaction de suivi du doigt augmentée

TISDA étend *TISD* présentée dans la section 6.3.2 à la rotation autour de l'axe horizontal latéral (voir l'angle pitch de la figure 6.1) qui n'est intégrée dans aucune des techniques *TIMC*, *TISD*, *TIOD*, *TIDDM* et *TICV*. Cette technique s'inspire du comportement de l'être humain dans le monde réel qui incline légèrement la tête vers le bas lorsqu'il marche. Ainsi, dans *TISDA* le point de vue 3D est orienté légèrement vers le bas durant la navigation tout en suivant la trajectoire du mouvement du doigt sur la table tactile (comme dans *TISD*).

L'être humain effectue souvent un pas en arrière lorsqu'il est devant un tableau de grande taille ou lorsqu'il souhaite visualiser un objet en hauteur proche de lui. Inspirée de ce comportement, la modification automatique de l'orientation horizontale du point de vue est bloquée dans *TISDA* lorsque l'utilisateur déplace l'icône de caméra dans le sens opposé vers lequel elle pointe. En plus, dans ce cas, le point de vue 3D est orienté discrètement vers le haut permettant la visualisation d'un objet virtuel situé en hauteur.

En comparaison avec *TISD*, le déplacement vers l'arrière dans l'environnement virtuel est possible sans modifier l'orientation horizontale. Toutefois, les rotations d'axe vertical (yaw) nécessitent le déplacement du point de vue et les déplacements latéraux sont impossibles comme pour *TISD*. L'application de la technique de modification de l'angle de pitch de *TISDA* aux autres techniques d'interaction de ce chapitre serait aussi possible.

Nous avons présenté dans cette section six techniques d'interaction dans des systèmes de type TEAMViz pour la navigation dans un environnement virtuel 3D. Chacune de ces techniques présente des avantages et des inconvénients et peut être combinée avec un dispositif complémentaire (e.g. un joystick). Dans la section suivante, nous comparons la mise en œuvre de ces techniques d'interaction dans notre espace de conception.

6.4 Conception comparative des techniques d'interaction pour la navigation

Dans les quatre premières techniques d'interaction présentées dans la section précédente, l'utilisateur manipule la représentation du point de vue 3D tandis que dans *TICV*

il manipule la projection de la vue 2D pour naviguer dans la vue 3D. Les différences entre les techniques d'interaction impliquent des différences au niveau de leurs conceptions. Nous présentons dans cette section la conception des techniques d'interaction au niveau du *mapping* puis au niveau des composantes logiques. Nous nous focalisons uniquement sur la navigation par translation horizontale et rotation autour de l'axe vertical dans cette section et n'étudions donc pas la conception de *TISDA* qui possède une composante de rotation pitch.

6.4.1 Conception par *mapping*

Les différences entre les techniques d'interaction dans la conception par *mapping* se situent uniquement au niveau de la cinquième étape de conception. Nous présentons d'abord les quatre premières étapes de la conception par *mapping*. Nous décrivons ensuite la cinquième étape pour chacune des techniques d'interaction en les comparant et en présentant le respect ou non des règles d'intégrité (présentées à la section 2.4.2) par chacune des solutions obtenues.

La navigation horizontale dans un environnement se décompose en quatre tâches élémentaires : la sélection, la translation selon l'axe X, la translation selon l'axe Y et la rotation autour de l'axe Z. Nous choisissons de présenter une vue 2D orthographique projetée dans un plan horizontal sur une table tactile horizontale et une vue 3D en perspective de cet environnement sur un écran vertical. Ce choix des vues est identique à celui de l'exemple *TT + M* du chapitre 2. Le dispositif d'entrée utilisé est la table tactile exclusivement.

La *table des techniques d'interaction* de la table tactile est constituée de :

- la manipulation directe pour le pointage, la translation selon l'axe X et la translation selon l'axe Y ;
- la manipulation à deux doigts pour la translation et le redimensionnement selon l'axe X, la translation et le redimensionnement selon l'axe Y et la rotation selon l'axe Z ;
- l'orientation du doigt pour la rotation autour de l'axe Z.

Les redimensionnements selon les axes X et Y ne sont pas utilisés dans les techniques d'interaction présentées dans la section précédente afin de ne pas modifier la taille de l'icône de caméra. L'agrandissement de la taille de l'icône impliquerait des occlusions supplémentaires tandis que la réduction de sa taille compliquerait sa sélection.

TIMC, *TISD* et *TIOD* sont identiques pour la translation mais se différencient au niveau de la rotation. La rotation horizontale est réalisée en utilisant un manipulateur dans *TIMC* impliquant l'ajout d'une technique d'interaction à la *table des techniques*

d'interaction. La solution de *TIMC* respecte toutes les règles de la conception par *mapping* même si une technique d'interaction est ajoutée alors que d'autres techniques ne sont pas utilisées. Dans *TISD*, l'orientation du point de vue est calculée à partir de la trajectoire et elle est donc définie par la séquence des translations selon les axes X et Y et selon un échantillonnage temporel. La connexion simultanée de la translation selon l'axe X (respectivement l'axe Y) de la manipulation directe à la translation selon l'axe X (respectivement l'axe Y) et à la rotation autour de l'axe Z de la composante *tâche* ne respecte pas la règle Mapping4 (règle d'assignation d'une technique d'interaction à une seule tâche élémentaire). En raison du non respect de la règle Mapping4 et de la simultanéité des translations et des rotations, l'utilisateur ne peut pas réaliser une rotation sans effectuer une translation et inversement ce qui peut être gênant pour lui. Finalement, c'est l'orientation physique du doigt qui définit l'orientation numérique du point de vue 3D dans *TIOD*. La solution de *TIOD* respecte toutes les règles de la conception par *mapping* et ne nécessite aucune technique d'interaction supplémentaire. Les solutions du *mapping* de ces trois techniques d'interaction sont illustrées dans la figure 6.13.

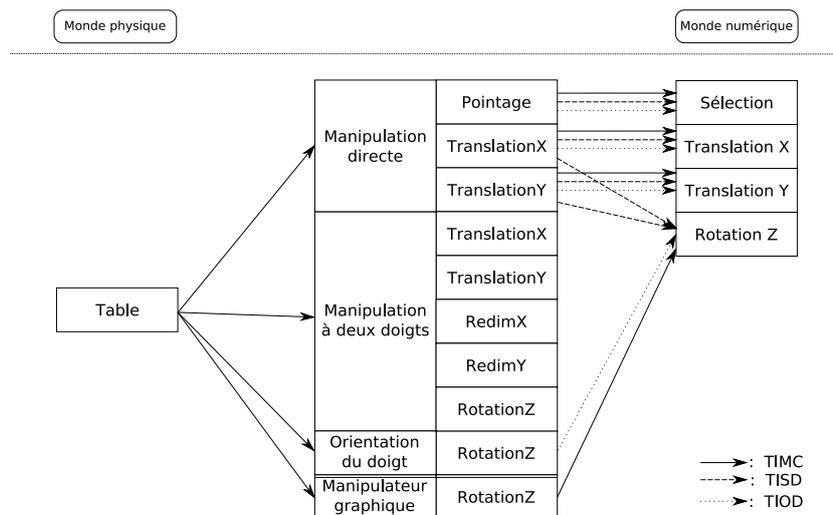


FIGURE 6.13 Solutions de la conception par *mapping* de *TIMC* (flèches pleines), *TISD* (flèches en tirets) et *TIOD* (flèches pointillées).

TIMDD se distingue de *TIMC*, *TISD* et *TIOD* par les connexions reliant la *table des techniques d'interaction* aux tâches élémentaires non seulement pour la rotation selon l'axe Z, mais aussi pour les translations selon les axes X et Y. La navigation dans *TIMDD* s'effectue par la manipulation à deux doigts (de l'icône de caméra) pour la rotation et la translation. La solution de *TIMDD* de la conception par *mapping* illustrée dans la figure 6.14 respecte toutes les règles de conception.

La différence entre *TICV* et les autres techniques d'interaction pour la navigation est la manipulation du point de vue 2D par la manipulation de la totalité de la projec-

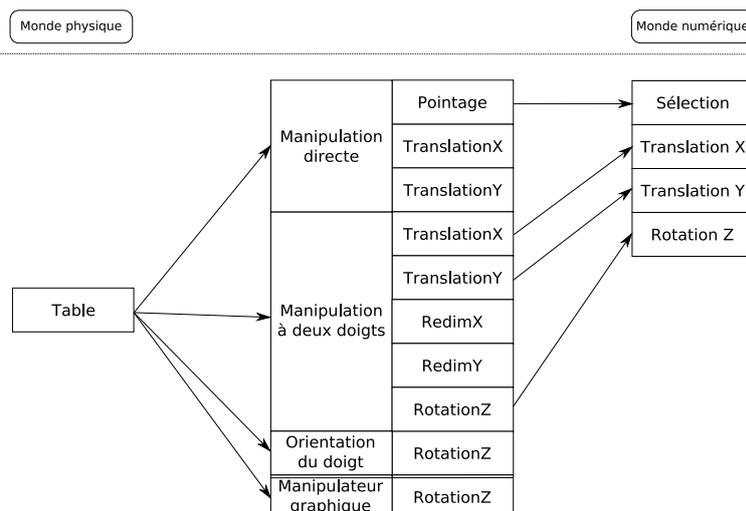


FIGURE 6.14 Solutions de la conception par *mapping* de *TIMDD*.

tion de la vue 2D au lieu de l'icône de caméra qui elle reste fixe dans cette technique. La translation selon l'axe X (respectivement l'axe Y) de la projection de la vue 2D dans le monde physique ne correspond pas à la translation selon l'axe X (respectivement l'axe Y) des points de vue 2D et 3D dans le repère du monde numérique en raison de la rotation possible du point de vue 2D sur la table tactile. Une transposition (par rotation et inversion) des translations selon les axes X et Y de la manipulation directe pour obtenir les translations selon l'axe X et Y de la composante *tâche* du *mapping* est nécessaire. Cette transposition est représentée dans la conception par *mapping* par un carré (voir figure 6.15). Le changement d'échelle entre les interactions sur la vue 2D et le monde numérique n'est pas intégré dans cette transposition mais est implicite au niveau des connexions reliant la *table des techniques d'interaction* à la composante *tâche* du *mapping*. Pour la rotation, l'ajout de la sélection du manipulateur graphique est nécessaire et les translations selon l'axe X et l'axe Y de la *table des techniques d'interaction* sont connectées à la rotation autour de l'axe Z. L'affectation des translations selon l'axe X et Y de la manipulation directe aux translations selon l'axe X et Y et la rotation autour de l'axe Z de la composante *tâche* ne respecte pas la règle Mapping4 de la conception par *mapping*. Cette règle n'est pas satisfaite en raison d'une interaction modale (dans la mesure où l'effet de la translation dans le monde physique dépend du mode dans lequel l'utilisateur se situe) qui est déconseillée en raison des erreurs possibles qu'elle peut engendrer [Norman, 1981].

En résumé, les règles Mapping1, Mapping2, Mapping3 et Mapping5 sont respectées par toutes les techniques d'interaction. Les solutions de *TISD* et *TICV* ne respectent pas la règle Mapping4 et les solutions de *TIMC* et *TICV* nécessitent l'ajout de nouvelles techniques d'interaction à la *table des techniques d'interaction* alors que d'autres

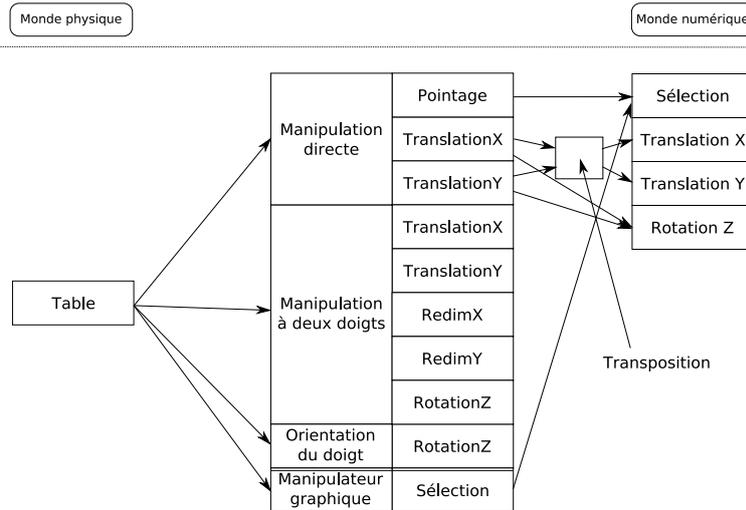


FIGURE 6.15 Solutions de la conception par *mapping* de *TICV*.

techniques ne sont pas utilisées. *TICV* nécessite l'ajout d'une transposition permettant d'obtenir les translations selon les axes X et Y de la composante *tâche* à partir des translations selon les axes X et Y de la manipulation directe.

6.4.2 Conception par composantes logiques

Les différences de conception par composantes logiques existent uniquement entre certaines techniques d'interaction pour la navigation présentées dans la section précédente. Ces différences se situent principalement au niveau des connexions entre les composantes logiques et les éléments du contrôleur de dialogue. Dans cette partie nous décrivons d'abord les cinq premières étapes de la conception par composantes logiques. Ensuite, nous présentons les différences de conception par composantes logiques des systèmes intégrant ces techniques d'interaction.

Les dispositifs de sortie utilisés sont un vidéo-projecteur, que nous appelons *Proj2D*, pour l'affichage de la vue 2D sur la table tactile et un écran (ou projecteur) pour la vue 3D que nous appelons *Ecran3D*. La table tactile est l'unique dispositif d'entrée que nous utilisons. La seconde étape de la conception par composantes logiques consiste à définir les éléments du contrôleur de dialogue. Chaque objet virtuel est représenté par un élément du contrôleur de dialogue de type *CDI* du fait qu'il n'est pas interactif. Un élément du contrôleur de dialogue de type *CDIT* est défini pour la vue 3D (appelé *CDVue3D*). Pour la vue 2D, une différence de typage de l'élément du contrôleur de dialogue correspondant (appelé *CDVue2D*) existe entre les techniques d'interaction. En effet, les utilisateurs manipulent la projection de la vue 2D dans *TICV* ce qui implique que *CDVue2D* est de type *CDIT*. Contrairement à *TICV*, les autres techniques

d'interaction ne permettent pas la manipulation de la vue 2D et dans ce cas $CDVue2D$ est de type CDI .

Pour présenter un environnement virtuel sur des vues multiples et y naviguer, les composantes logiques spatiales sont plus adaptées que les composantes logiques non spatiales. La définition des composantes logiques de vue et de représentation de vue intervient dans la quatrième étape de la conception par composantes logiques. Pour chaque vue nous déterminons une composante logique de vue (appelées $CLSV3D$ pour la vue 3D et $CLSV2D$ pour la vue 2D). Nous identifions aussi une composante logique de représentation de vue appelée $CLSRV3D$ pour la représentation du point de vue 3D sur la vue 2D. Pour les objets virtuels, une composante logique par vue est définie pour chaque objet.

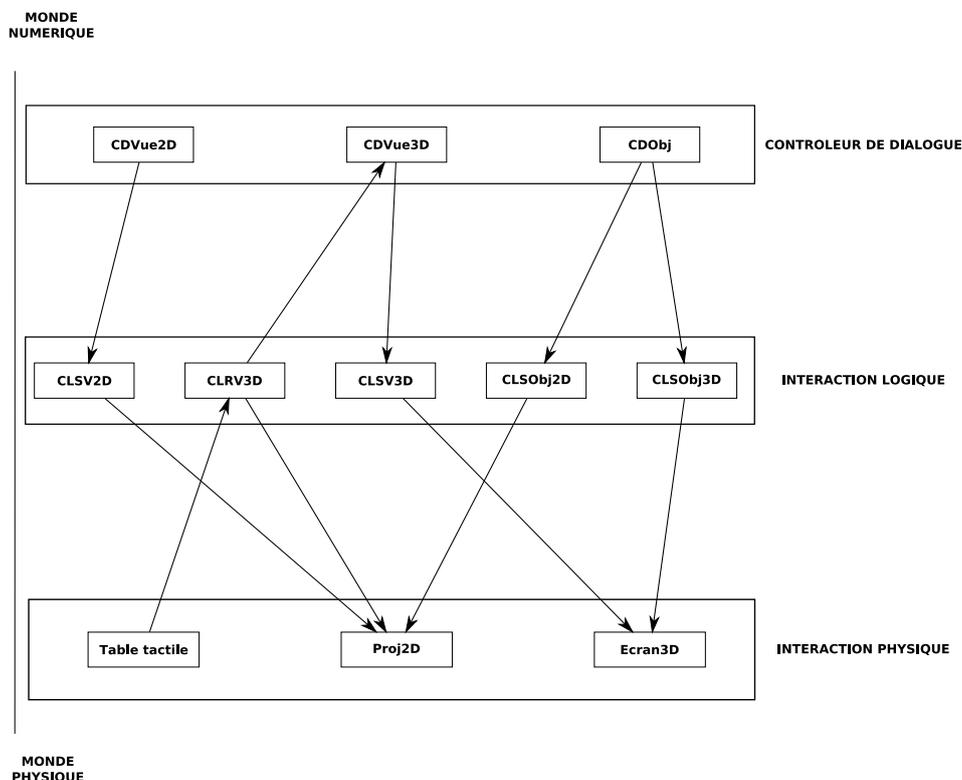


FIGURE 6.16 Solution de la conception par composantes logiques de $TIMC$, $TISD$, $TIOD$ et $TIMDD$.

Dans $TIMC$, $TISD$, $TIOD$ et $TIMDD$ les utilisateurs manipulent la représentation de la vue 3D sur la vue 2D (l'icône de caméra). La vue 2D reste fixe ce qui implique que $CDVue2D$ (de type CDI) est connectée à des composantes logiques de sortie uniquement. $CLSVue2D$ est donc une composante logique de sortie. Pour la vue 3D, $CDVue3D$ est connectée en sortie à $CLSV3D$ et en entrée/sortie à $CLSRV3D$. La figure 6.16 illustre de façon simplifiée la solution de la conception par composantes logiques d'un système intégrant l'une des techniques d'interaction $TIMC$, $TISD$, $TIOD$

ou *TIMDD*. Dans cette figure, nous pouvons visualiser que l'interaction sur la représentation du point de vue 3D (*CLSRV3D*) avec la table tactile modifie les données du point de vue 3D au niveau du contrôleur de dialogue (*CDVue3D*). Ces modifications sont ensuite répercutées sur la vue 3D (*CLSV3D*) présentée sur un écran (*Ecran3D*). Cette solution satisfait toutes les règles de validation de la conception par composantes logiques.

Les utilisateurs manipulent la totalité de la projection de la vue 2D pour naviguer dans la vue 3D en utilisant *TICV*. *CLSV2D* est donc une composante logique d'entrée/sortie connectée à *CDVue2D* (qui est de type *CDIT* dans ce cas) et à *CDVue3D*. *CLSV3D* et *CLSRV3D* sont des composantes logiques de sortie uniquement connectée à *CDVue3D*. De même que pour les autres techniques d'interaction, une composante logique d'objet par vue est définie pour chaque objet virtuel. La solution de la conception par composantes logiques (voir figure 6.17) respecte toutes les règles.

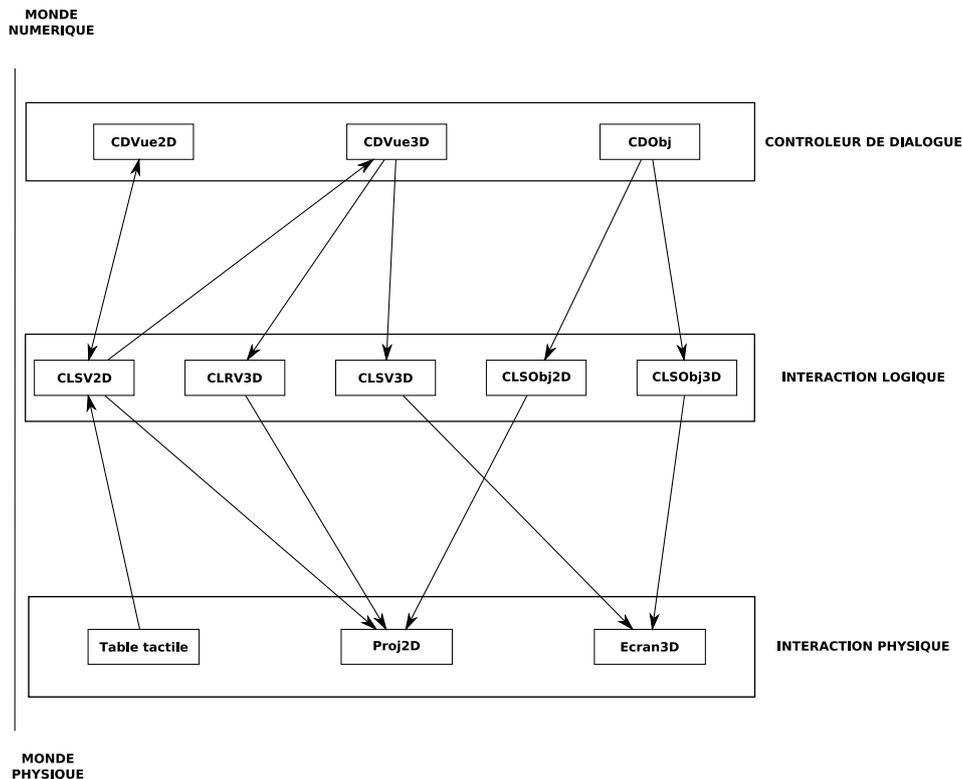


FIGURE 6.17 Solution de la conception par composantes logiques de *TICV*.

En résumé, les différences entre les deux solutions de la conception par composantes logiques se situent aux niveaux des connexions entre les composantes logiques et les éléments du contrôleur de dialogue, du type de *CDVue2D* et de la connexion de la table tactile aux composantes logiques. La définition d'un dispositif supplémentaire de type joystick est possible en ajoutant une composante logique d'entrée connectée à

CDVue3D. Cette composante logique serait aussi connectée à *CDVue2D* dans le cas de *TICV*. Ainsi, si une tâche de navigation nécessite la translation verticale (en Z) ou la rotation verticale (pitch), elle pourrait être réalisée par ce dispositif complémentaire.

L'étude des différences de conception entre les techniques d'interaction ainsi que l'intégration ou non des types de disposition des vues montre une séparation en deux types de techniques : les techniques de manipulation de la représentation de caméra d'une part et la technique de manipulation de la projection d'une vue de l'autre. Dans la section suivante, nous présentons une expérimentation comparant ces deux types de techniques de navigation.

6.5 Expérimentation

TICV se différencie des autres techniques présentées dans ce chapitre par l'intégration de la disposition des surfaces de projection dans le monde physique et son couplage à la disposition des points de vue dans le monde virtuel. Ce couplage nécessite la manipulation de la totalité de la projection de la vue 2D pour naviguer dans la vue 3D. Le déplacement et l'orientation de la projection de la vue 2D ainsi que l'orientation fixe de l'icône de caméra est comparable à l'utilisation d'une carte de type *forward-up* (en français devant en haut) [Darken & Cevik, 1999]. Dans ce type de cartes, les objets virtuels sont présentés dans le référentiel égocentrique. Contrairement à la *TICV*, les autres techniques d'interaction sont comparables à l'usage d'une carte de type *north-up* (en français le nord en haut) [Darken & Cevik, 1999] où la vue 2D reste fixe et c'est la représentation du point de vue qui est déplacée. Les objets virtuels sont présentés dans ce type de cartes dans le référentiel du monde virtuel (référentiel allocentrique).

La différence principale entre *TIMC*, *TISD*, *TIOD* et *TIMDD* se situe au niveau de la technique utilisée pour réaliser la rotation du point de vue autour de l'axe vertical (yaw). Cependant, dans ces techniques d'interaction l'utilisateur manipule la représentation du point de vue 3D tandis qu'il manipule la totalité de la projection de la vue 2D dans *TICV*. Cette différence entre *TICV* et les autres techniques est plus importante que les différences de techniques de rotation. Dans le but d'examiner la différence entre *TICV* et les autres techniques d'interaction, et pour éviter toute influence de la technique de rotation de *TICV* et des autres techniques, nous comparons deux techniques d'interaction qui étendent, respectivement *TICV* et *TIMC*, à l'utilisation du joystick pour la rotation autour de l'axe vertical. La première technique appelée Technique d'Interaction de Couplage des Vues avec Joystick (*TICVJ**) combine la technique *TICV* avec l'utilisation du joystick. Dans *TICVJ*, la translation s'effectue par déplacement

de la projection de la vue 2D (comme pour *TICV*) mais la rotation est réalisée exclusivement avec le joystick. Dans la seconde technique appelée Technique d'Interaction par Manipulation de Caméra avec Joystick (*TIMCJ**), le déplacement de l'icône de caméra sur la vue 2D permet la translation (comme pour *TIMC*, *TISD* et *TIOD*) tandis que le joystick est dédié à la rotation. Ainsi, dans *TICVJ* et *TIMCJ* les translations sont toujours effectuées en utilisant la table tactile et les rotations toujours en utilisant le joystick.

Dans cette section, nous réalisons une expérimentation comparant *TICVJ* et *TIMCJ*. Nous décrivons d'abord l'objectif et la procédure de l'expérimentation. Nous analysons ensuite les résultats obtenus. Finalement, nous comparons les deux techniques sur le plan de l'utilisation de plusieurs caméras, sur le plan l'occlusion, sur les situations multi-utilisateurs et sur le cas de surfaces d'affichage verticales multiples qui sont des enjeux importants dans le domaine des tables tactiles.

6.5.1 Objectif

L'objectif de l'expérimentation est de comparer les performances et les préférences des utilisateurs entre *TIMCJ* et *TICVJ*. Darken et Cevik [1999] observent que les performances entre l'usage des cartes égocentriques et allocentriques dépendent de la tâche de navigation à réaliser. Considérant ces observations et les similarités entre *TIMCJ* (respectivement *TICVJ*) et l'usage d'une carte allocentrique (respectivement égocentrique), nous posons les hypothèses suivantes :

- les performances et les préférences des utilisateurs devraient être supérieures en utilisant *TICVJ* pour des tâches de navigation égocentriques (e.g. tâche de recherche où la position de l'objet recherché est connue d'avance) ;
- les performances et les préférences des utilisateurs devraient être supérieures en utilisant *TIMCJ* pour des tâches de navigation allocentriques (e.g. tâche de *recherche naïve* [Darken & Cevik, 1999]).

Le but de l'expérimentation est donc de tester et de valider ou invalider ces deux hypothèses.

6.5.2 Protocole

Dans le but d'étudier les performances et les préférences des utilisateurs en utilisant *TICVJ* et *TIMCJ* nous considérons deux tâches de navigation à accomplir. Dans la première tâche appelée tâche *arbre*, les utilisateurs sont dans une scène virtuelle composée d'une maison avec deux fenêtres (voir figure 6.18). Une voiture est positionnée

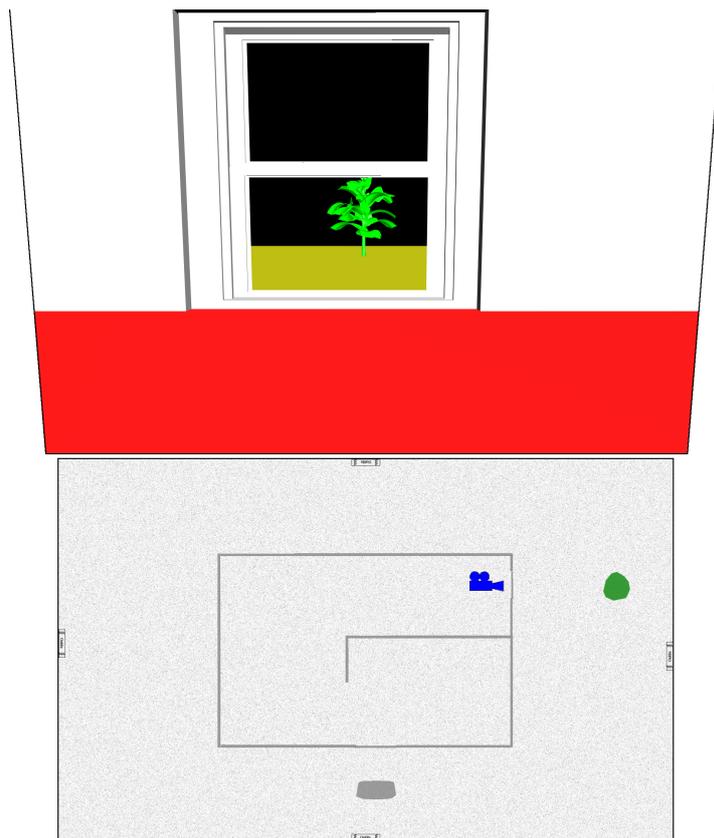


FIGURE 6.18 Vues 2D (bas) et 3D (haut) de l'environnement virtuel pour la tâche *arbre* en utilisant *TIMCJ*.

devant la première fenêtre et un arbre devant la seconde. Le but de cette tâche est de se déplacer dans la maison virtuelle pour se positionner devant la seconde fenêtre de façon à voir la totalité de l'arbre à travers la fenêtre comme le montre la figure 6.18. Dans la seconde tâche appelée tâche *boule rouge*, la scène virtuelle est composée de quinze maisons dont chacune contient deux plantes (voir figure 6.19). Une unique boule rouge, présentée exclusivement dans la vue 3D, se situe proche de l'une des trente plantes de la scène virtuelle. Pour accomplir leur tâche, les utilisateurs doivent retrouver cette boule rouge. La tâche *arbre* est une tâche égocentrique du fait que la position et l'orientation du point de vue 3D dans le référentiel du monde sont moins importants que la position et l'orientation de la vue de l'utilisateur par rapport à la fenêtre et à l'arbre. Contrairement à la tâche *arbre*, la connaissance de la position et de l'orientation du point de vue 3D dans le monde virtuel est indispensable pour la tâche *boule rouge*. Par conséquent, la tâche *boule rouge* est une tâche allocentrique.

L'architecture logicielle utilisée pour réaliser l'expérimentation est celle de MC qui est un système de type TEAMViz (voir chapitre 4). 24 utilisateurs (10 femmes et 14 hommes) ayant des formations et des niveaux d'expérience différents dans les environnements virtuels (dont les jeux 3D) ont participé à l'expérience. Les participants sont

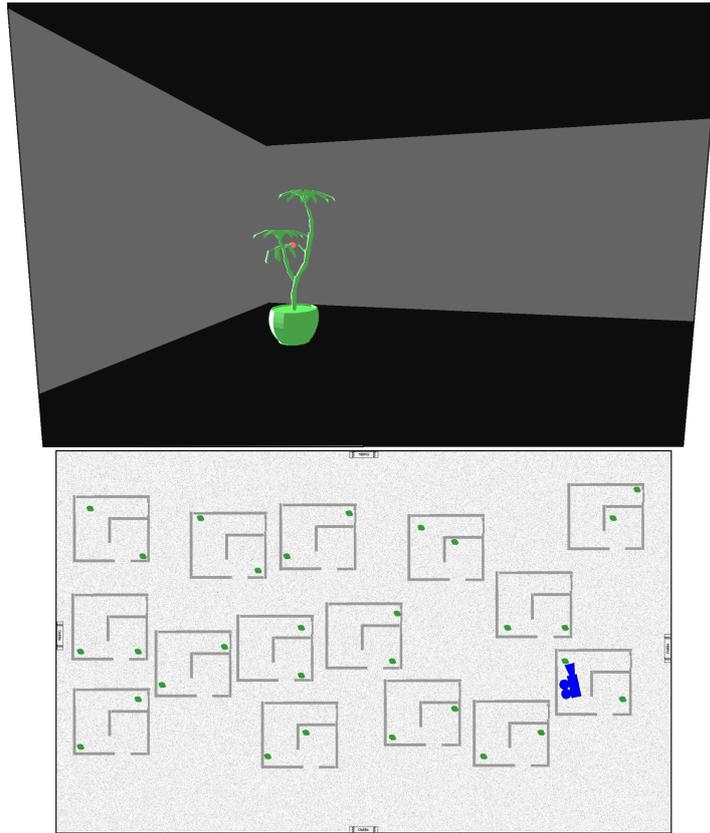


FIGURE 6.19 Vues 2D (bas) et 3D (haut) de l’environnement virtuel pour la tâche *boule rouge* en utilisant *TIMCJ*. La boule rouge est visible uniquement dans la vue 3D.

placés devant la table tactile et l’écran vertical (alignés physiquement). Le joystick se situe sur une table complémentaire du côté de la main dominante (voir figure 6.20). Au début de l’expérience, chaque utilisateur teste librement les deux techniques d’interaction durant cinq minutes. Ensuite, il doit accomplir chacune des tâches avec chacune des deux techniques d’interaction quatre fois avec des conditions finales différentes : position finale différente de l’arbre et de la fenêtre pour la tâche *arbre* et position finale de la boule rouge différente pour la tâche *boule rouge*. L’ordre des tâches et des techniques de navigation est contre-balancé. Les utilisateurs indiquent librement lorsque la tâche est accomplie. L’expérimentation dure environ trente minutes par utilisateur.

Pendant et après l’expérimentation, les utilisateurs sont priés de répondre à un questionnaire dans le but d’obtenir des résultats subjectifs pour comparer les deux techniques d’interaction. Ils doivent noter après chaque tâche et pour chaque technique la facilité, la satisfaction, l’intuitivité et l’efficacité des interactions. Les notations sont effectuées sur une échelle de Likert [1932] allant de 1 (totalement négative) à 7 (totalement positive). Après l’accomplissement d’une tâche avec les deux techniques d’interaction, les participants doivent choisir leurs technique préférée pour cette tâche. La même question

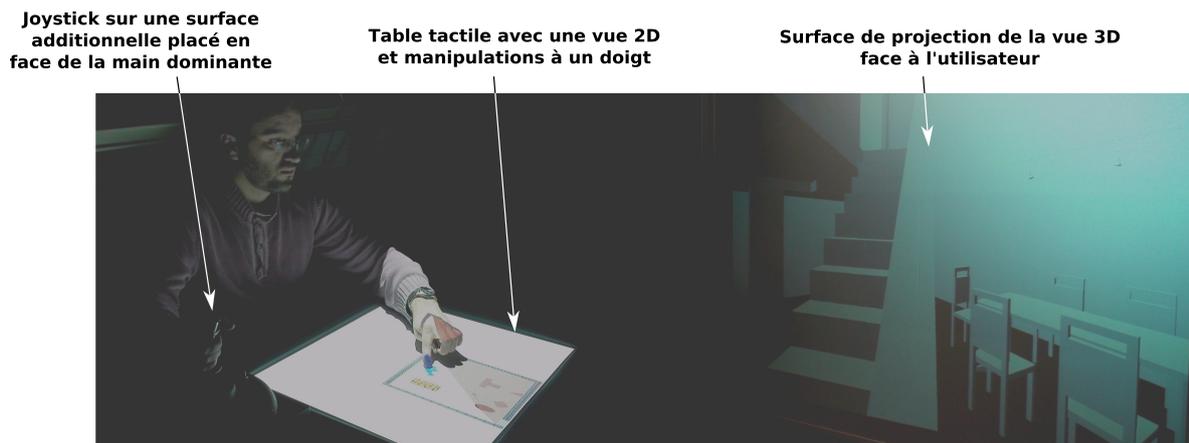


FIGURE 6.20 Utilisateur participant à l'expérimentation de comparaison de *TIMCJ* et *TICVJ*.

est posée à la fin de l'expérimentation pour la préférence générale des utilisateurs.

6.5.3 Résultats et analyse

Les données objectives du temps d'accomplissement des tâches sont acquises durant l'expérimentation. Le temps moyen pour chaque tâche et pour chaque technique d'interaction est calculé à partir des trois meilleurs temps réalisés par chaque utilisateur. Les résultats du temps d'exécution de la tâche *arbre* ne montrent aucune différence significative entre les deux techniques (voir figure 6.21). Ainsi, la première hypothèse n'est pas vérifiée : la technique de navigation égocentrique *TICVJ* ne donne pas de meilleurs résultats.

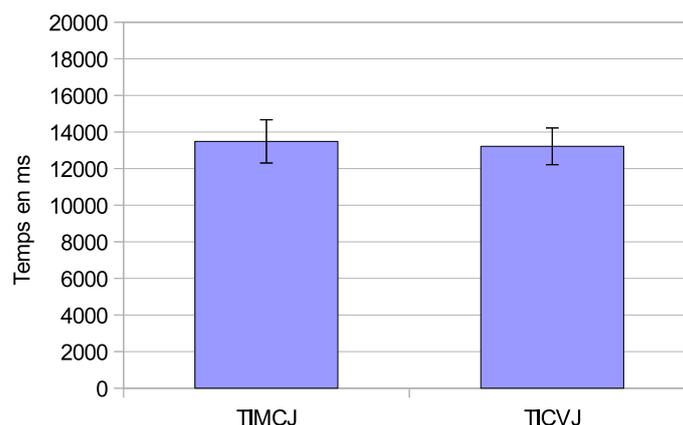


FIGURE 6.21 Temps moyen (en ms) nécessaire pour exécuter la tâche *arbre* (I : erreur type).

Le temps moyen de la tâche *boule rouge* (voir figure 6.22) est inférieur pour la technique allocentrique *TIMCJ* que pour la technique égocentrique *TICVJ* ($p=0.022$)¹.

1. p est la probabilité que l'hypothèse soit fausse

Certains utilisateurs soulignent qu'ils se sont perdus dans la scène virtuelle en utilisant *TICVJ* et qu'ils ne se souviennent plus des plantes déjà visualisées. Ces résultats soutiennent notre seconde hypothèse qui stipule que les performances des utilisateurs sont meilleures en utilisant *TIMCJ* pour des tâches de navigation allocentriques.

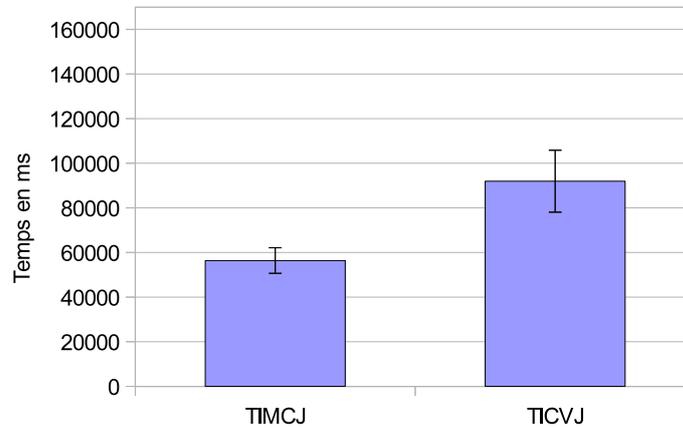


FIGURE 6.22 Temps moyen (en ms) nécessaire pour exécuter la tâche *boule rouge* (I : erreur type).

Les résultats subjectifs montrent que les utilisateurs ont mieux noté la technique égocentrique *TICVJ* que la technique allocentrique *TIMCJ* pour la facilité, la satisfaction et l'efficacité de l'interaction (voir figure 6.23). La notation de l'intuitivité est légèrement supérieure pour *TIMCJ*. Les résultats de la facilité, de la satisfaction et de l'intuitivité des interactions sont identiques pour les deux tâches. Les participants considèrent la technique allocentrique *TIMCJ* comme plus efficace pour la tâche allocentrique *boule rouge*.

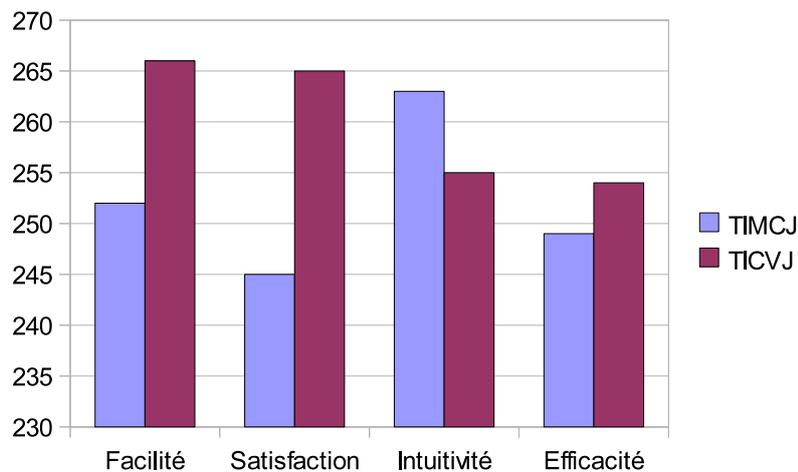


FIGURE 6.23 Somme de toutes les notations des utilisateurs pour la facilité, la satisfaction, l'intuitivité et l'efficacité des interactions.

Les résultats de la figure 6.24 montrent que 79% (respectivement 71% et 62%) des

participants préfèrent la technique égoцентриque *TICVJ* à la technique allocentrique *TIMCJ* de façon générale (respectivement pour la tâche *arbre* et la tâche *boule rouge*). Les utilisateurs justifient leurs choix par la facilité des interactions en utilisant *TICVJ*. Plus précisément, dans *TICVJ* les utilisateurs peuvent sélectionner et manipuler une zone très large (la totalité de la projection de la vue) contrairement à *TIMCJ* dans laquelle l'interaction s'effectue uniquement avec l'icône de caméra.

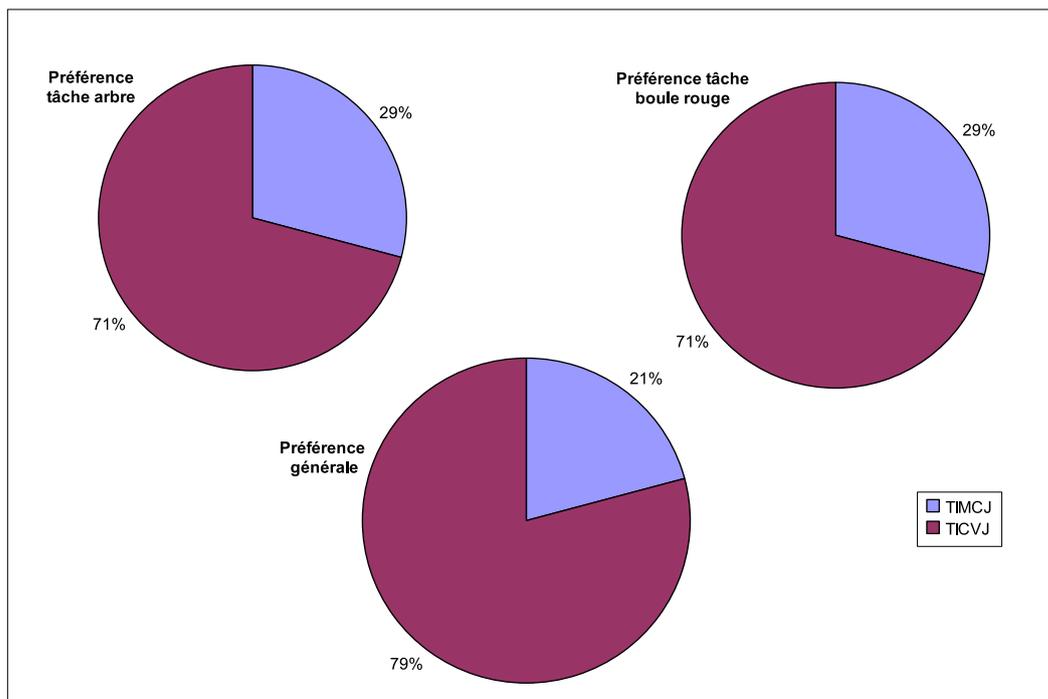


FIGURE 6.24 Préférences des utilisateurs entre *TIMCJ* et *TICVJ*. En haut à gauche : préférence pour la tâche *arbre*. En haut à droite : préférence pour la tâche *boule rouge*. En bas au centre : préférence générale.

En résumé, les utilisateurs préfèrent et notent généralement mieux *TICVJ* que *TIMCJ*. Contrairement aux préférences des utilisateurs, le temps moyen de réalisation de la tâche allocentrique *boule rouge* est inférieur en utilisant la technique allocentrique *TIMCJ*. Le fait que les participants se sont perdus dans la scène virtuelle en utilisant la technique égoцентриque *TICVJ* explique ces résultats qui soutiennent notre seconde hypothèse.

6.5.4 Discussion

Dans cette section nous avons présenté une expérimentation comparant *TIMCJ* et *TICVJ* en examinant les performances et les préférences des utilisateurs. D'autres points sont essentiels à étudier dont les caméras multiples, l'occlusion, les utilisateurs multiples et les écrans verticaux multiples. Nous comparons ici *TIMCJ* et *TICVJ* pour

les perspectives ouvertes par ces techniques sur chacun de ces points sans toutefois avoir réalisé des expérimentations qui permettraient de valider ces hypothèses.

Multi-cameras

Plusieurs icônes de caméras représentant des points de vue 3D distincts peuvent être présentées sur la vue 2D (voir figure 6.25). Une unique icône, celle du point de vue 3D affichée sur l'écran vertical, est active à un instant donné. L'activation d'un point de vue inactif peut s'effectuer par la sélection de l'icône de caméra ou par la sélection d'un bouton graphique présenté sur l'icône de caméra correspondante. L'activation par sélection de l'icône de caméra permet un passage rapide d'une caméra à l'autre mais ne permet pas de déplacer une caméra inactive sans l'activer. Avec la seconde possibilité d'activation, la manipulation des icônes de caméras inactives ne modifie pas l'affichage de la vue 3D permettant ainsi un positionnement "hors ligne" de points de vue 3D.

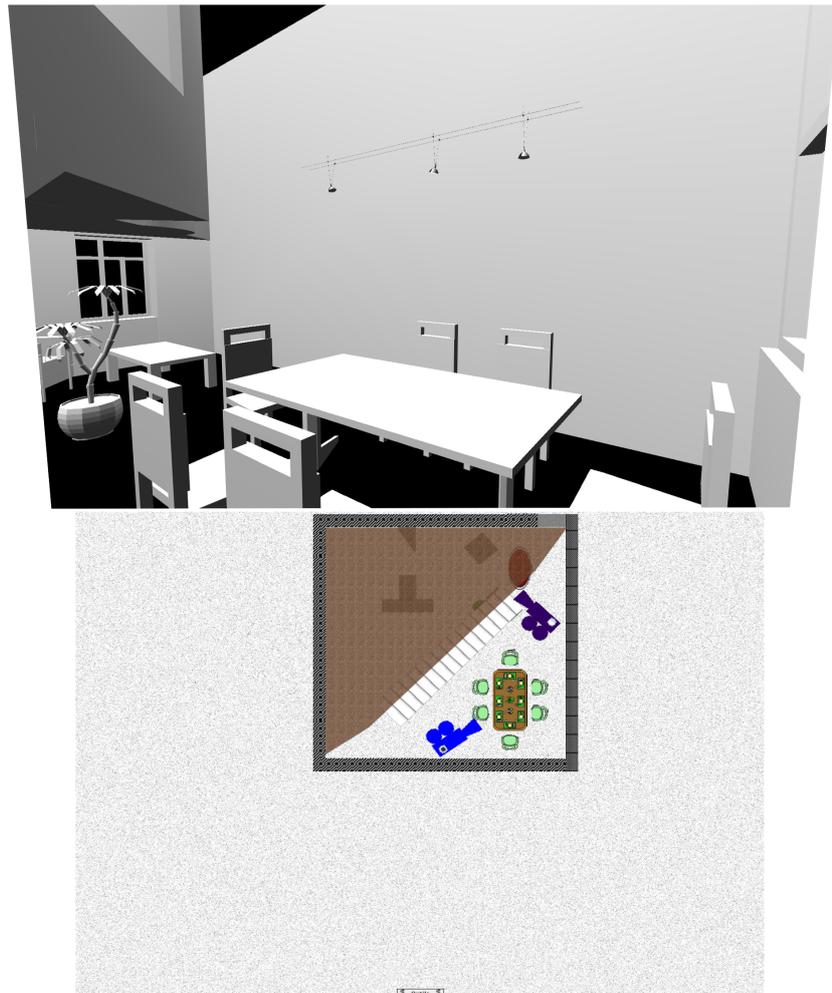


FIGURE 6.25 Caméras multiples dans la vue 2D (bas) dont une active présentée sur l'écran vertical (haut).

La manipulation d'une icône de caméra active en utilisant *TIMCJ* n'affecte pas la manipulation des autres icônes de caméra. La téléportation du point de vue (déplacement instantané du point de vue d'une position à une autre position) dans les environnements virtuels est déconseillée. Cependant, la vue 2D fixe sur la table tactile permet aux utilisateurs de conserver constamment une vue d'ensemble de la scène virtuelle. Pour cette raison, le changement du point de vue d'une caméra active à une caméra inactive peut se réaliser sans animation d'interpolation supplémentaire. L'utilisation de cette technique serait bien adaptée par exemple pour la gestion des différentes caméras d'un plateau de tournage et dont une seule serait active à la fois.

En utilisant *TICVJ*, la totalité de la vue 2D est déplacée et/ou tournée pour naviguer dans l'environnement virtuel. Les caméras inactifs sont déplacées en manipulant l'icône de caméra correspondante en utilisant par exemple l'une des techniques suivantes : *TIMC*, *TISD*, *TIOD*, *TIMDD* ou *TIMCJ*. Ainsi, les caméras ne réagissent donc pas de la même façon selon si elle est active ou non. La navigation et le positionnement d'une icône de caméra inactive dans le repère du monde virtuel de façon simultanée est difficile à réaliser en raison de la manipulation de la totalité de la projection de la vue 2D durant la navigation. Néanmoins, la manipulation simultanée de deux icônes de caméras inactives reste facile. Le passage d'une caméra active à une caméra inactive en utilisant *TICVJ* présente certains problèmes. En effet, lors de cette permutation les deux vues sont modifiées pouvant ainsi désorienter les utilisateurs. Nous conseillons d'ajouter une animation interpolant la vue 2D et la vue 3D du point de vue actif au point de vue souhaité pour aider les utilisateurs à surmonter ce problème de désorientation.

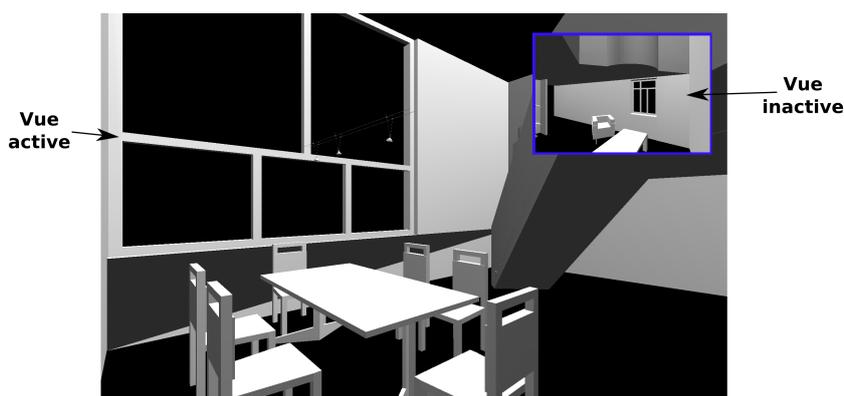


FIGURE 6.26 Présentation d'une vue inactive réduite incrustée dans la vue active 3D afin de fournir un retour visuel 3D à l'utilisateur s'il manipule une caméra inactive.

La visualisation du point de vue 3D d'une caméra inactive est parfois souhaitable durant sa manipulation. Cette visualisation est réalisable en incrustant une mini-vue

3D d'une caméra inactive dans la vue 3D d'une caméra active (voir figure 6.26). L'inconvénient de cette solution est l'occlusion de la vue 3D active produite par l'espace consacré à l'autre vue. La position et l'orientation horizontales d'un point de vue inactif étant toujours visibles dans la vue 2D, nous déconseillons l'utilisation de cette solution pour préserver la totalité de la vue 3D.

Occlusion

Dans notre système MC, une vue 2D et une vue 3D sont présentées à l'utilisateur comportant chacune des occlusions :

- les occlusions dans la vue 3D sont produites par certains objets situés devant d'autres par rapport au point de vue. De plus, selon la position du point de vue dans la scène virtuelle, certains objets ne sont pas visibles à cause du champ de vue 3D restreint. Les occlusions produites dans la vue 3D sont identiques pour *TIMCJ* et *TICVJ* ;
- dans *TIMCJ*, la vue 2D sur l'environnement virtuel est fixe. Les utilisateurs possèdent une vue d'ensemble de la scène virtuelle sans parties cachées lorsqu'une initialisation appropriée du centre et des dimensions de la vue est réalisée. Ainsi, une fonctionnalité de zoom de la vue 2D n'est pas nécessaire pour des scènes de complexité et de taille réduites. Cependant, la sélection de l'icône de caméra par manipulation directe (avec les doigts) augmente l'occlusion produite par les mains des utilisateurs [Shen *et al.*, 2006] ;
- la position et l'orientation de la vue 2D sur la table tactile sont centrées sur l'icône de caméra fixe et sont modifiées durant la navigation avec *TICVJ*. Ainsi, des parties de l'environnement virtuel peuvent être cachées dans la vue 2D nécessitant une fonctionnalité de zoom même pour des scènes simples. De plus, la position de l'icône de caméra sur la vue 2D influence l'occlusion. En effet, le positionnement de cette icône au centre de la vue 2D permet de visualiser les parties non visibles dans la vue 3D (e.g. derrière le point de vue 3D) mais réduit l'espace de visualisation devant l'icône de caméra (voir figure 6.27). Le positionnement au bord opposé à l'écran vertical de l'icône (voir figure 6.27) produit l'effet inverse. Bien que l'occlusion de parties de l'environnement virtuel soit possible en utilisant *TICVJ*, l'occlusion physique produite par les mains des utilisateurs est réduite en raison d'une zone de sélection et de manipulation très large (les utilisateurs déplacent la projection de la vue 2D en sélectionnant n'importe quel point ne correspondant ni à un objet virtuel ni à l'icône de caméra). De grandes chances qu'une telle zone existe près du bord où se situe l'utilisateur lui permettent d'interagir confortablement

sur une zone proche de lui/elle.

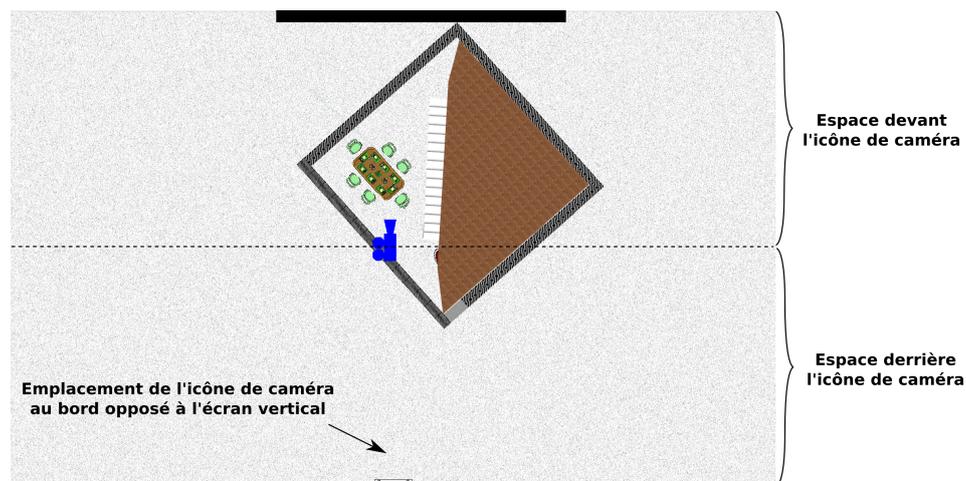


FIGURE 6.27 Position de l'icône de caméra sur la vue 2D.

Multi-utilisateurs

Un bénéfice important des dispositifs de type tabletop est la possibilité que plusieurs utilisateurs s'assoient autour et partagent des interactions. Chacun des utilisateurs possède alors une vue différente sur les vues 2D et 3D (voir figure 6.28). L'utilisateur situé face à l'écran 3D (utilisateur 1 de la figure 6.28) est le plus avantageux du fait de l'alignement des vues. Les personnes installées perpendiculairement à l'écran affichant la vue 3D (utilisateurs 2 et 3 de la figure 6.28) sont situées de façon équivalente par rapport à l'écran. Elles sont désavantagées en comparaison avec la personne située en face de l'écran en raison d'une rotation horizontale de la tête de 90 degrés nécessaire pour passer de la visualisation d'une vue à l'autre. Finalement, l'utilisateur positionné dos à l'écran (utilisateur 4 de la figure 6.28) est le plus désavantagé du fait qu'il doit réaliser une rotation de 180 degrés pour visualiser la vue 3D. Pour cette personne, l'écran vertical est quasi-inutilisable. De plus il gêne la visualisation de l'écran vertical à l'utilisateur positionné face à l'écran. Nous conseillons d'éviter d'utiliser les quatre côtés, quitte à placer plusieurs utilisateurs à côté de celui situé face à l'écran (utilisateur 1 de la figure 6.28).

Les personnes situées perpendiculairement (utilisateurs 2 et 3 de la figure 6.28) ou dos à l'écran vertical (utilisateur 4 de la figure 6.28) sont encore plus désavantagées par rapport à celle installée en face de l'écran en utilisant *TICVJ*. En effet, l'alignement de l'utilisateur, de la vue 2D et de la vue 3D favorise la personne située en face de l'écran vertical. Cet alignement supprime la nécessité de réaliser des rotations mentales pour commuter entre la vue 2D et la vue 3D.

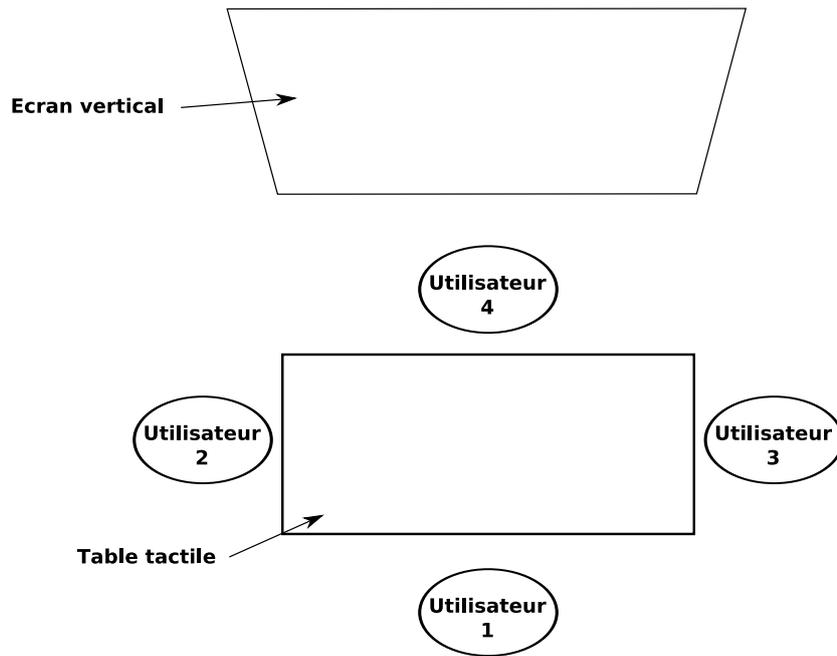


FIGURE 6.28 Disposition de quatre utilisateurs autour d'une table tactile.

Dans *TIMCJ*, la vue 2D est fixe permettant à chaque utilisateur de construire facilement sa propre représentation mentale de l'environnement virtuel selon sa position autour de la table tactile. Ainsi, *TIMCJ* réduit l'inégalité entre les utilisateurs concernant l'ergonomie de leur position par rapport à la projection de la vue 2D.

Ecrans verticaux multiples

L'augmentation de la table tactile par plusieurs écrans verticaux est intéressante pour étendre l'espace de visualisation. Nous nous intéressons à l'étude de cette augmentation en présentant des points de vue 3D différents de la scène virtuelle sur chacun de ces écrans verticaux. Ainsi, les utilisateurs peuvent visualiser plusieurs points de vue 3D de la scène virtuelle simultanément.

Pour *TIMCJ* et les autres techniques n'intégrant pas la disposition des surfaces de projection des vues dans le monde physique, l'utilisation d'écrans verticaux multiples ne nécessite pas de restrictions majeures. Une icône de caméra active doit être définie par écran à tout moment. Le point de vue de la même icône de caméra peut être présenté sur plusieurs écrans verticaux. La manipulation simultanée de plusieurs caméras actives avec *TIMCJ* est possible sans aucune gêne supplémentaire pour les utilisateurs. Le passage d'une caméra active à une caméra inactive reste facile mais nécessite le choix de l'écran sur lequel le point de vue 3D correspondant sera présenté (e.g. par choix dans un menu).

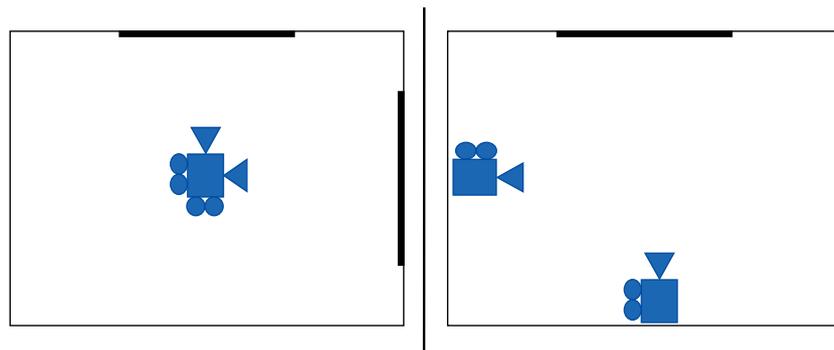


FIGURE 6.29 Deux possibilités de disposition des caméras sur la vue 2D en utilisant *TICVJ* dans la situation d'écrans verticaux multiples.

Dans le but de préserver le couplage entre les deux types de disposition des vues dans *TICVJ*, les navigations dans deux vues différentes de la même scène virtuelle affichée sur des écrans multiples sont associées. Ainsi, le déplacement dans une vue 3D implique le déplacement dans les autres vues 3D. Nous identifions deux possibilités de gestion des caméras : icônes de caméra centrées dans la vue 2D (voir la partie gauche de la figure 6.29) et icônes de caméras affichées sur les bords opposés aux bords les plus proches des écrans correspondants (voir la partie droite de la figure 6.29). Dans ces deux cas, les icônes de caméras actives pointent vers la position physique de l'écran correspondant. La première possibilité permet d'avoir un plus grand champ de vision d'un même point de vue sur la scène virtuelle (en comparaison avec une seule caméra active et un seul écran). Les vues 3D sont donc complémentaires : le bord gauche d'une vue est en raccord exact avec le bord droit de la vue 3D de gauche. Dans la seconde possibilité, les utilisateurs visualisent sur plusieurs écrans verticaux une même partie de la scène virtuelle mais à partir de points de vue différents. Certaines informations dans les vues 3D sont redondantes tandis que d'autres sont complémentaires. Le choix entre les deux cas doit être réalisé selon la scène virtuelle à visualiser et la tâche à réaliser. Par exemple, la première possibilité est plus adaptée à la navigation dans une maison virtuelle tandis que la seconde est plus appropriée pour la visualisation d'un objet virtuel 3D. Dans le cadre d'un nombre de caméras supérieur au nombre d'écrans, l'activation de caméras inactives doit être réalisée pour un groupe de caméras actives couplées vers un autre groupe de caméras inactives couplées. L'activation d'une seule caméra inactive est impossible pour préserver l'association entre les deux types de dispositions des vues. Cela nécessite donc de former des groupes de caméras de même taille que le nombre d'écrans 3D.

La combinaison de plusieurs écrans verticaux (alignés parallèlement à plusieurs bords de la table tactile) dans le cadre multi-utilisateurs présente un inconvénient majeur qui est l'occlusion. Les utilisateurs sont positionnés autour de la table ainsi que les écrans

verticaux. Cela implique que chaque personne située entre la table tactile et un écran vertical masque cet écran partiellement. Néanmoins, l'utilisation d'écrans verticaux multiples, situées chacune devant un utilisateur, réduit les inégalités entre utilisateurs pour *TICVJ*.

En résumé, l'utilisation de *TIMCJ* est plus adaptée que l'utilisation de *TICVJ* dans les situations multi-utilisateurs, les situations de caméras multiples et pour des tâches où une représentation mentale de l'environnement virtuel est nécessaire en raison de la vue 2D fixe. Cependant, *TICVJ* réduit les rotations mentales et l'occlusion physique produite par les mains des utilisateurs qui préfèrent cette technique à *TIMCJ*. De plus, dans le cas multi-écrans 3D, l'inégalité entre utilisateurs (dans les situations multi-utilisateurs) est réduite pour *TICVJ* lorsqu'un écran vertical est positionné devant chaque utilisateur autour de la table tactile.

6.6 Conclusion

Plusieurs techniques d'interaction dédiées aux TEAMViz pour la navigation dans un environnement virtuel présenté en 2D sur une table tactile et en 3D sur un écran vertical ont été décrites dans ce chapitre. Dans toutes ces techniques une représentation du point de vue 3D est affichée sur la vue 2D présentant ainsi la disposition des points de vues dans le monde virtuel. Deux techniques (*TICV* et *TICVJ*) intègrent la disposition des surfaces de projection des vues dans le monde physique et la couplent avec la disposition des points de vues dans le monde virtuel. L'expérimentation montre que *TICVJ* est préférée par les utilisateurs par rapport à une autre technique n'intégrant pas la disposition des surfaces de projection dans le monde physique (*TIMCJ*). Cependant les résultats expérimentaux montrent que les utilisateurs sont moins performants avec *TICVJ* qu'avec *TIMCJ* pour des tâches nécessitant une représentation mentale de l'environnement virtuel. Ainsi, le couplage des deux types de disposition des vues doit être choisi en fonction du type de tâches de navigation à accomplir.

L'étude de la navigation dans un environnement virtuel en utilisant un système de type TEAMViz nécessite de divers approfondissements. Une comparaison entre les techniques d'interaction présentées dans ce chapitre est indispensable pour examiner l'influence des différences de techniques de rotation sur les utilisateurs. De plus, les techniques que nous avons présentées n'intègrent pas tous les degrés de liberté souhaités pour la navigation dans un environnement virtuel (e.g. translation en Z). L'intégration de ces degrés de liberté ainsi que le zoom étant important, il serait nécessaire de leur dédier une étude formelle et expérimentale. Finalement, d'autres techniques d'interac-

tion pour la navigation dans un environnement virtuel peuvent être développées pour les systèmes TEAMViz nécessitant leurs comparaisons avec les techniques décrites dans ce chapitre.

Conclusion

Les dispositifs d'entrée dédiés aux interactions avec des environnements 3D présentent chacun des avantages et des inconvénients et aucun ne satisfait tous les besoins des tâches possibles dans ces environnements. De plus, il n'existe pas une seule meilleure technique d'interaction pour les interactions dans les environnements 3D. Pour ces raisons, nous proposons de développer des systèmes appelés TEAMViz intégrant des interactions au moyen d'une vue 2D d'une scène virtuelle présentée sur une table tactile et des interactions au moyen d'une vue 3D de cette même scène affichée sur un écran vertical. Les TEAMViz combinent plusieurs dispositifs en entrée (dont une table tactile) et plusieurs vues en sortie présentées sur des surfaces d'affichage distinctes.

Bilan

Le développement des TEAMViz et des leurs interactions nécessite la prise en considération de plusieurs éléments. L'élément essentiel est l'étude des interactions en entrée et en sortie de façon couplée et non de façon disjointe en raison des couplages forts nécessaires entre les entrées et les sorties des surfaces tactiles. Dans la littérature, l'étude des vues multiples, des surfaces multiples et des dispositifs en entrées multiples est souvent réalisée de façon disjointe. Certaines études nous ont fortement inspiré mais aucun espace de conception existant n'est réellement adapté à la conception des interactions pour les TEAMViz en raison de cette séparation des entrées et des sorties. Le couplage fort des entrées et des sorties des surfaces tactiles soutient notre choix de se focaliser sur des interactions par pointage et par manipulation directe et de se dispenser de l'étude des modalités de parole ou l'utilisation d'objets tangibles.

Dans les TEAMViz, la présentation de deux vues d'une même scène virtuelle sur des surfaces d'affichage différentes nécessite la mise en relation des interactions dans le monde physique (réel) avec leurs effets dans le monde numérique. De plus, la relation entre la disposition des surfaces de projection des vues dans le monde physique et la disposition des points de vue dans le monde virtuel constitue un élément important

lors de la conception des interactions. Notre expérience nous a montré qu'un espace de conception adapté à la conception des TEAMViz et de leurs interactions doit prendre en compte la relation entre le monde physique et le monde numérique ainsi que la conception des interactions en entrée et en sortie de façon couplée.

La contribution majeure de cette thèse est l'espace de conception pour la modélisation des interactions dédiés aux systèmes de type TEAMViz. Cet espace de conception, composé de trois étapes, couvre plusieurs aspects allant de la conception des interactions à haut niveau jusqu'à la gestion de la concurrence des interactions en entrée. L'intérêt principal de cet espace de conception est la mise en relation du monde physique avec le monde virtuel 3D en couplant les entrées et les sorties.

La première étape de conception (conception par *mapping*) se situe à un haut niveau et se base sur le modèle *mapping*. Ce modèle connecte les dispositifs d'entrée (dans le monde physique) aux tâches élémentaires spatiales (dans le monde numérique) en passant par une table de techniques d'interaction (définie selon les interactions dans le monde physique). Il permet donc de relier le monde physique au monde numérique à haut niveau. De plus, la relation entre les vues dans le monde physique et le monde numérique est étudiée durant la phase de définition des vues de la conception par *mapping*. Le couplage des entrées et des sorties dans la conception par *mapping* est réalisé durant la phase de choix des dispositifs d'entrée et de définition des vues. La conception par *mapping* permet d'explorer plusieurs possibilités de combinaisons d'interactions à haut niveau tout en éliminant celles qui ne respectent pas certaines contraintes exprimées dans les règles logiques. L'expérimentation comparant plusieurs combinaisons possibles montre que la souris reste le dispositif préféré des utilisateurs et le plus performant. Néanmoins, la combinaison des interactions au moyen d'une table tactile et d'un dispositif à plusieurs degrés de liberté est prometteuse avec plus d'apprentissage.

La conception par composantes logiques (seconde étape de notre espace de conception) se situe à un niveau intermédiaire entre la conception par *mapping* et la conception logicielle. Cette phase de conception se base sur la notion de composantes logiques qui gère, selon son type, la présentation de l'information d'une entité (objet virtuel, représentation d'un point de vue ou un point de vue) sur une vue et/ou les interactions en entrée. Les composantes logiques sont connectées aux dispositifs physiques et aux éléments du contrôleur de dialogue (éléments contenant une version cohérente des données d'un objet virtuel ou d'une vue). Certains types de composantes logiques relient les entrées et les sorties permettant ainsi de préserver le couplage fort entrée/sortie nécessaire dans certains types d'interfaces comme les surfaces tactiles.

La gestion de la concurrence des interactions en entrée constitue la dernière étape de

notre espace de conception. Une architecture logicielle centralisée reliant des applications d'IHM (composante IHM 2D et composante IHM 3D de notre système MC) à une application centrale facilite la gestion de la concurrence. Nous appliquons un système de permission pour gérer la concurrence des interactions. Nous modélisons également des machines à états, d'une part, au niveau de l'application centrale pour la gestion des permissions et, d'autre part, au niveau des applications IHM pour gérer le blocage des interactions. Plusieurs niveaux de granularité de gestion de la concurrence peuvent exister allant du blocage totale d'une application IHM lorsqu'un utilisateur interagit avec une autre application jusqu'au blocage de la manipulation de certains degrés de liberté uniquement. L'expérimentation montre que la granularité fine de la gestion de la concurrence est appréciée des utilisateurs. Dans cette étape de la conception, la relation entre le monde physique et le monde numérique n'est pas intégrée. Cependant, l'étude de la concurrence des degrés de liberté en entrée pour une granularité fine nécessite la prise en compte de la présentation de l'information en sortie.

Notre espace de conception est spécifique aux tâches spatiales et aux interactions réalisables au moyen de vues multiples d'un même environnement virtuel présentées sur des surfaces d'affichage distinctes. Cependant, plusieurs domaines d'application existent pour ce type d'interaction et d'interface dont l'architecture, l'urbanisme, la visualisation interactive de l'acoustique de salles virtuelles, etc. L'extension d'un système conçu avec notre espace de conception à d'autres systèmes a été étudiée dans ce manuscrit et montre qu'elle est simple lorsque les deux systèmes présentent des caractéristiques identiques (e.g. mêmes dimensions d'affichage). Lorsque les systèmes présentent des caractéristiques différentes, des modifications importantes au niveau des trois étapes de notre espace de conception (ou même une re-modélisation totale) sont nécessaires. Néanmoins, certaines de ces modifications pourraient être automatisées.

Nous avons appliqué notre espace de conception à l'étude de la navigation dans une scène virtuelle 3D en utilisant un système de type TEAMViz. Dans cette étude nous différencions deux types de disposition des vues : la disposition des points de vue dans le monde virtuel et la disposition des surfaces de projection dans le monde physique. Nous avons développé plusieurs techniques de navigation avec notre espace de conception dont une couplant les deux types de dispositions. L'expérimentation montre que les utilisateurs préfèrent la technique couplant les deux types de dispositions à une autre technique intégrant uniquement la disposition des points de vue dans le monde virtuel. Cette préférence des utilisateurs justifie l'importance de la relation entre le monde physique et le monde numérique dans la conception des interactions pour les systèmes de type TEAMViz. Cependant, les différences de performances des utilisateurs entre les deux techniques sont fortement liées au type de tâche de navigation à réaliser.

Ces observations montrent la complexité de conception des interactions dans les systèmes TEAMViz.

Perspectives

Les travaux entrepris dans cette thèse présentent plusieurs ouvertures possibles. Certaines de ces ouvertures sont des extensions et améliorations de notre espace de conception tandis que d'autres sont des élargissements du champ d'étude engagé.

L'extension principale de notre espace de conception est le développement d'une plate-forme permettant une conception rapide des interactions et des interfaces en se basant sur la conception par composantes logiques. Cette plate-forme pourrait s'inspirer de l'ICON [Dragicevic & Fekete, 2004], de la plate-forme OpenInterface [OpenInterface][OpenInterface Project] ou des travaux de Kobayashi *et al.* [2005] mais elle devrait intégrer les sorties et des composantes d'entrée/sortie. La modélisation des machines à états pour la gestion de la concurrence devrait aussi être facilitée par cette plate-forme. Elle pourrait s'inspirer de HsmTk [Blanch & Beaudouin-Lafon, 2006] mais serait appliquée à la gestion des permissions et de blocage des interactions.

L'intégration et l'étude du type d'interaction selon qu'elle est isotonique ou isométrique au niveau de la conception par *mapping* constitue la seconde extension possible de notre espace de conception. La différenciation entre ces deux types d'interactions s'effectuerait au niveau du modèle *mapping* et plus précisément au niveau des connexions reliant la *table des techniques d'interaction* à la composante *tâche*. La comparaison des performances et des préférences des utilisateurs entre ces deux types d'interactions permettrait de définir de nouvelles règles d'élimination de combinaisons indésirables. Ainsi la conception des interactions en serait facilitée.

Nous identifions plusieurs thèmes possibles d'élargissement du champ d'étude entrepris dans cette thèse. L'étude dans les systèmes TEAMViz des interactions pour les tâches de type "*cross-task*" (soit en français inter-tâches) nous semble très intéressante. Un exemple d'un tel type de tâches est le déplacement du point de vue 3D tout en positionnant un objet virtuel. Une technique d'interaction possible pour cette tâche est le déplacement du point de vue 3D en manipulant une représentation de ce point de vue sur la table tactile et la manipulation simultanée de l'objet virtuel (relativement au point de vue 3D) en utilisant un dispositif à six degrés de liberté. L'intérêt de des interactions dédiées à ce type de tâches est de faciliter la réalisation de l'une des deux tâches en permettant de réaliser une autre tâche simultanément. Par exemple, ce type d'interaction faciliterait le déplacement simultané d'un objet virtuel et du point de vue

3D pour éviter des problèmes d'occlusion. Une expérimentation avec des utilisateurs serait nécessaire pour étudier l'utilisabilité de ces techniques d'interaction. Les résultats de cette expérimentation permettraient de définir de nouvelles règles de conception par *mapping*.

L'extension des TEAMViz à plusieurs surfaces d'affichage verticaux nous semble également intéressante à étudier. Cette extension offrirait aux utilisateurs des vues supplémentaires et ainsi un champ de vue plus étendue sur la scène virtuelle. L'ajout de surfaces d'affichage verticales supplémentaires et ainsi des vues supplémentaires aux TEAMViz augmente la complexité d'utilisation du système pour l'utilisateur. Ainsi, il est indispensable d'étudier les possibilités de faciliter la compréhension du fonctionnement du système (e.g. la relation entre les vues, la relation entre dispositifs d'entrée et vues, etc.). Ces améliorations se situent au niveau de la présentation de l'information sur la relation entre les vues ainsi que la liaison entre un dispositif d'entrée et une (des) vue(s).

Dans cette thèse, nous avons appliqué notre espace de conception aux TEAMViz. Cependant, il n'est pas spécifique uniquement aux TEAMViz et nous semble applicable à d'autres types d'interfaces combinant des surfaces d'affichages multiples présentant différentes vues d'une même scène virtuelle. Les interfaces présentant des vues 3D d'une même scène virtuelle mais affichées sur des surfaces d'affichage verticales distinctes sont des exemples de telles interfaces. L'application de notre espace de conception à d'autres types de systèmes que les TEAMViz constitue donc une autre perspective qui permettrait de vérifier et valider la généralité de notre approche.

Finalement, la dernière perspective de notre travail que nous présentons consiste à étudier l'utilisation des TEAMViz dans plusieurs domaines d'application. Dans notre étude nous avons développé des systèmes de type TEAMViz en nous focalisant uniquement sur des scènes virtuelles architecturales. D'autres domaines d'application sont possibles dont la biologie pour le "docking" moléculaire ou les domaines artistiques (e.g. cinéma, muséographie, installation numérique interactive, etc.).

Bibliographie

3Dconnexion. www.3dconnexion.fr.

ArchiCAD : 3D Architectural CAD Software for Architects & Designers. www.graphisoft.com/products/archicad.

AutoCAD. <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2704278>.

Autodesk 3ds Max. <http://usa.autodesk.com/adsk/servlet/pc/index?id=13567410&siteID=123112>.

Blender. www.blender.org.

Google Docs. docs.google.com.

Kensington France. www.kensington.fr.

Microsoft Surface. www.microsoft.com/surface.

MultiTouch - Modular MultiTouch LCD. www.multitouch.fi.

NUI Group - Natural User Interface Group. www.nuigroup.com.

OpenInterface. <http://www.openinterface.org>.

OpenInterface Project. <http://www.oi-project.org>.

Verse. verse.blender.org.

VirChor. http://sourceforge.net/apps/mediawiki/virchor/index.php?title=Main_Page.

Wacom. <http://www.wacom-europe.com>.

Ajaj, Rami, Savioja, Lauri, & Jacquemin, Christian. 2008. Software platform for real-time room acoustic visualization. *Pages 247–248 of : Proceedings of the ACM symposium on Virtual reality software and technology*. Bordeaux, France : ACM.

Aliakseyeu, Dzmitry, Subramanian, Sriram, Martens, Jean-Bernard, & Rauterberg, Matthias. 2002. Interaction techniques for navigation through and manipulation of 2D and 3D data. *Pages 179–188 of : Proceedings of the workshop on Virtual environments 2002*. Barcelona, Spain : Eurographics Association.

- Allen, James F. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, **26**(11), 832–843.
- Allen, Jont B., & Berkley, David A. 1979. Image Method for Efficiently Simulating Small-Room Acoustics. *The Journal of the Acoustical Society of America*, **65**(4), 943–950.
- Appert, Caroline, & Beaudouin-Lafon, Michel. 2006. SwingStates : adding state machines to the swing toolkit. *Pages 319–322 of : Proceedings of the 19th annual ACM symposium on User interface software and technology*. Montreux, Switzerland : ACM.
- Baer, Ralph H. 2005. *Videogames : In the Beginning*. Rolenta Press.
- Bailly, Gilles, Nigay, Laurence, & Auber, David. 2006. NAVRNA : visualization - exploration - editing of RNA. *Pages 504–507 of : Proceedings of the working conference on Advanced visual interfaces*. Venezia, Italy : ACM.
- Baldonado, Michelle Q. Wang, Woodruff, Allison, & Kuchinsky, Allan. 2000. Guidelines for using multiple views in information visualization. *Pages 110–119 of : Proceedings of the working conference on Advanced visual interfaces*. Palermo, Italy : ACM.
- Ballagas, Rafael, Ringel, Meredith, Stone, Maureen, & Borchers, Jan. 2003. iStuff : a physical user interface toolkit for ubiquitous computing environments. *Pages 537–544 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. Ft. Lauderdale, Florida, USA : ACM.
- Bastide, Rémi, Palanque, Philippe A., Le, Duc-Hoa, & Munoz, Jaime. 1998. Integrating Rendering Specifications into a Formalism for the Design of Interactive Systems. *Pages 171–190 of : Proceedings of Design Specification and Verification of Interactive Systems*. Springer Verlag.
- Beaudouin-Lafon, Michel. 2000. Instrumental interaction : an interaction model for designing post-WIMP user interfaces. *Pages 446–453 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. The Hague, The Netherlands : ACM.
- Beaudouin-Lafon, Michel. 2001. Novel interaction techniques for overlapping windows. *Pages 153–154 of : Proceedings of the 14th annual ACM symposium on User interface software and technology*. Orlando, Florida : ACM.
- Bérard, François, Ip, Jessica, Benovoy, Mitchel, El-Shimy, Dalia, Blum, Jeffrey R., & Cooperstock, Jeremy R. 2009. Did "Minority Report" Get It Wrong? Superiority of the Mouse over 3D Input Devices in a 3D Placement Task. *Pages 400–414 of : Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction : Part II*. Uppsala, Sweden : Springer-Verlag.
- Blanch, Renaud, & Beaudouin-Lafon, Michel. 2006. Programming rich interactions using the hierarchical state machine toolkit. *Pages 51–58 of : Proceedings of the working conference on Advanced visual interfaces*. Venezia, Italy : ACM.
- Bouchet, Jullien, & Nigay, Laurence. 2004. ICARE : a component-based approach for the design and development of multimodal interfaces. *Pages 1325–1328 of : CHI '04 extended abstracts on Human factors in computing systems*. Vienna, Austria : ACM.

- Bouchet, Jullien, Nigay, Laurence, & Balzagette, Didier. 2004a. ICARE : a component-based approach for multimodal interaction. *Pages 36–43 of : Proceedings of the 1st French-speaking conference on Mobility and ubiquity computing*. Nice, France : ACM.
- Bouchet, Jullien, Nigay, Laurence, & Ganille, Thierry. 2004b. ICARE software components for rapidly developing multimodal interfaces. *Pages 251–258 of : Proceedings of the 6th international conference on Multimodal interfaces*. State College, PA, USA : ACM.
- Bowman, Doug A., Johnson, Donald B., & Hodges, Larry F. 1999. Testbed evaluation of virtual environment interaction techniques. *Pages 26–33 of : Proceedings of the ACM symposium on Virtual reality software and technology*. London, United Kingdom : ACM.
- Bowman, Doug A., Kruijff, Ernst, LaViola, Joseph J., & Poupyrev, Ivan. 2004. *3D User Interfaces : Theory and Practice*. Addison Wesley Longman Publishing Co., Inc.
- Buxton, William. 1983. Lexical and pragmatic considerations of input structures. *SIG-GRAPH Computer Graphics*, **17**(1), 31–37.
- Chalon, René, & David, Bertrand. 2004. IRVO : an Architectural Model for Collaborative Interaction in Mixed Reality Environments. *In : Proceedings of the Workshop MIXER'04*.
- Coutaz, Joëlle. 1987. PAC, on Object Oriented Model for Dialog Design. *Pages 431–436 of : Proceedings of Interact'87*.
- Coutaz, Joëlle, Nigay, Laurence, Salber, Daniel, Blandford, Ann, May, Jon, & Young, Richard M. 1995. Four easy pieces for assessing the usability of multimodal interaction : the CARE properties. *Pages 115–120 of : Proceedings of INTERACT'95*.
- Coutrix, Céline, & Nigay, Laurence. 2006. Mixed reality : a model of mixed interaction. *Pages 43–50 of : Proceedings of the working conference on Advanced visual interfaces*. Venezia, Italy : ACM.
- Coutrix, Céline, & Nigay, Laurence. 2008. Balancing physical and digital properties in mixed objects. *Pages 305–308 of : Proceedings of the working conference on Advanced visual interfaces*. Napoli, Italy : ACM.
- Cruz-Neira, Carolina, Sandin, Daniel J., DeFanti, Thomas A., Kenyon, Robert V., & Hart, John C. 1992. The CAVE : audio visual experience automatic virtual environment. *Commun. ACM*, **35**(6), 64–72.
- Darken, Rudolph P., & Cevik, Helsin. 1999. Map usage in virtual environments : orientation issues. *Pages 133–140 of : Proceedings of IEEE Virtual Reality 99*.
- De La Rivière, Jean-Baptiste, Kervégant, Cédric, Orvain, Emmanuel, & Dittlo, Nicolas. 2008. CubTile : a multi-touch cubic interface. *Pages 69–72 of : Proceedings of the ACM symposium on Virtual reality software and technology*. Bordeaux, France : ACM.
- Dietz, Paul, & Leigh, Darren. 2001. DiamondTouch : a multi-user touch technology. *Pages 219–226 of : Proceedings of the 14th annual ACM symposium on User interface software and technology*. Orlando, Florida : ACM.

- Dragicevic, Pierre. 2001. *Un modèle d'interaction en entrée pour des systèmes interactifs multi-dispositifs hautement configurables*. Ph.D. thesis.
- Dragicevic, Pierre, & Fekete, Jean-Daniel. 2004. The Input Configurator toolkit : towards high input adaptability in interactive applications. *Pages 244–247 of : Proceedings of the working conference on Advanced visual interfaces*. Gallipoli, Italy : ACM.
- Dubois, Emmanuel, Nigay, Laurence, & Troccaz, Jocelyne. 2001. Consistency in Augmented Reality Systems. *Pages 111–122 of : Engineering for Human-Computer Interaction*.
- Dubois, Emmanuel, Nigay, L., & Troccaz, J. 2002. Assessing Continuity and Compatibility in Augmented Reality Systems. *International Journal on Universal Access in the Information Society*, **1**(4), 263–273.
- Fjeld, Morten, Lauche, Kristina, Dierssen, S., Bichsel, Martin, & Rauterberg, Matthias. 1998. BUILD-IT : A Brick-based integral Solution Supporting Multidisciplinary Design Tasks. *Pages 122–133 of : Proceedings of the IFIP Working Group 13.2 Conference on Designing Effective and Usable Multimedia Systems*. Kluwer, B.V.
- Fjeld, Morten, Ironmonger, Nicholas, Voorhorst, Fred, Bichsel, Martin, & Rauterberg, Matthias. 1999a. Camera Control in a Planar, Graspable Interface. *Pages 242–245 of : Proceedings of 17th international conference on Applied Informatics*.
- Fjeld, Morten, Voorhorst, Fred, Bichsel, Martin, & Krueger, Helmut. 1999b. Exploring brick-based camera control. *Pages 1060–1064 of : Proceedings of the HCI International '99 (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction : Communication, Cooperation, and Application Design-Volume 2 - Volume 2*. L. Erlbaum Associates Inc.
- Fjeld, Morten, Voorhorst, Fred, Bichsel, Martin, Lauche, Kristina, Rauterberg, Matthias, & Krueger, Helmut. 1999c. Exploring Brick-Based Navigation and Composition in an Augmented Reality. *Pages 102–116 of : Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. Springer-Verlag.
- Forlines, Clifton, Esenther, Alan, Shen, Chia, Wigdor, Daniel, & Ryall, Kathy. 2006. Multi-user, multi-display interaction with a single-user, single-display geospatial application. *Pages 273–276 of : Proceedings of the 19th annual ACM symposium on User interface software and technology*. Montreux, Switzerland : ACM.
- Forlines, Clifton, Wigdor, Daniel, Shen, Chia, & Balakrishnan, Ravin. 2007. Direct-touch vs. mouse input for tabletop displays. *Pages 647–656 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. San Jose, California, USA : ACM.
- Frohlich, Bernd, Plate, John, Wind, Jürgen, Wesche, Gerold, & Gobel, Martin. 2000. Cubic-Mouse-based interaction in virtual environments. *IEEE Computer Graphics and Applications*, **20**(4), 12–15.

- Funkhouser, Thomas, Carlbom, Ingrid, Elko, Gary, Pingali, Gopal, Sondhi, Mohan, & West, Jim. 1998. A Beam Tracing Approach to Acoustic Modeling for Interactive Virtual Environments. *Pages 21–32 of : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, vol. 32.
- Greenhalgh, Chris, Purbrick, Jim, & Snowdon, Dave. 2000. Inside MASSIVE-3 : flexible support for data consistency and world structuring. *Pages 119–127 of : Proceedings of the third international conference on Collaborative virtual environments*. San Francisco, California, United States : ACM.
- Grossman, Tovi, & Wigdor, Daniel. 2007. Going Deeper : a Taxonomy of 3D on the Tabletop. *Pages 137–144 of : Proceedings of the Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07*.
- Hachet, Martin, Guitton, Pascal, & Reuter, Patrick. 2003. The CAT for efficient 2D and 3D interaction as an alternative to mouse adaptations. *Pages 225–112 of : Proceedings of the ACM symposium on Virtual reality software and technology*. Osaka, Japan : ACM.
- Hachet, Martin, Declé, Fabrice, Knodel, Sebastian, & Guitton, Pascal. 2008. Navidget for Easy 3D Camera Positioning from 2D Inputs. *Pages 83–89 of : Proceedings of the 2008 IEEE Symposium on 3D User Interfaces*.
- Han, Jefferson Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. *Pages 115–118 of : Proceedings of the 18th annual ACM symposium on User interface software and technology*. Seattle, WA, USA : ACM.
- Hancock, Mark, Carpendale, Sheelagh, & Cockburn, Andy. 2007. Shallow-depth 3d interaction : design and evaluation of one-, two- and three-touch techniques. *Pages 1147–1156 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. San Jose, California, USA : ACM.
- Hancock, Mark S., Carpendale, Sheelagh, Vernier, Frederic D., Wigdor, Daniel, & Shen, Chia. 2006. Rotation and Translation Mechanisms for Tabletop Interaction. *Pages 79–88 of : Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE Computer Society.
- Hinckley, Ken, Tullio, Joe, Pausch, Randy, Proffitt, Dennis, & Kassell, Neal. 1997. Usability analysis of 3D rotation techniques. *Pages 1–10 of : Proceedings of the 10th annual ACM symposium on User interface software and technology*. Banff, Alberta, Canada : ACM.
- Hodges, Steve, Izadi, Shahram, Butler, Alex, Rustemi, Alban, & Buxton, Bill. 2007. ThinSight : versatile multi-touch sensing for thin form-factor displays. *Pages 259–268 of : Proceedings of the 20th annual ACM symposium on User interface software and technology*. Newport, Rhode Island, USA : ACM.
- Iwata, Hiroo, Yano, Hiroaki, Fukushima, Hiroyuki, & Noma, Haruo. 2005. CirculaFloor. *IEEE Computer Graphics and Applications*, **25**(1), 64–67.

- Iwata, Hiroo, Yano, Hiroaki, & Tomioka, Hiroshi. 2006. Powered shoes. *Page 28 of : ACM SIGGRAPH 2006 Emerging technologies*. Boston, Massachusetts : ACM.
- Johanson, Brad, Hutchins, Greg, Winograd, Terry, & Stone, Maureen. 2002. PointRight : experience with flexible input redirection in interactive workspaces. *Pages 227–234 of : Proceedings of the 15th annual ACM symposium on User interface software and technology*. Paris, France : ACM.
- Jordà, Sergi, Geiger, Günter, Alonso, Marcos, & Kaltenbrunner, Martin. 2007. The reacTable : exploring the synergy between live music performance and tabletop tangible interfaces. *Pages 139–146 of : Proceedings of the 1st international conference on Tangible and embedded interaction*. Baton Rouge, Louisiana : ACM.
- Kara, Levent Burak, Shimada, Kenji, & Marmalefsky, Sarah D. 2007. Calligraphic Interfaces : An evaluation of user experience with a sketch-based 3D modeling system. *Computers and Graphics*, **31**(4), 580–597.
- Kato, Hirokazu, Billingham, Mark, Poupyrev, Ivan, Imamoto, Kenji, & Tachibana, Keihachiro. 2000. Virtual object manipulation on a table-top AR environment. *Pages 111–119 of : Proceedings of IEEE and ACM International Symposium on Augmented Reality, 2000. (ISAR 2000)*.
- Knödel, Sebastian, Hachet, Martin, & Guitton, Pascal. 2008. Navidget for immersive virtual environments. *Pages 47–50 of : Proceedings of the ACM symposium on Virtual reality software and technology*. Bordeaux, France : ACM.
- Kobayashi, Nobuyuki, Tokunaga, Eiji, Kimura, Hiroaki, Hirakawa, Yasufumi, Ayabe, Masaki, & Nakajima, Tatsuo. 2005. An input widget framework for multi-modal and multi-device environments. *Pages 63–70 of : Proceedings of the Third IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, 2005. SEUS 2005*.
- Krasner, Glenn E., & Pope, Stephen T. 1988. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, **1**(3), 26–49.
- Kruger, Russell, Carpendale, Sheelagh, Scott, Stacey D., & Tang, Anthony. 2005. Fluid integration of rotation and translation. *Pages 601–610 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA : ACM.
- Kuchar, Robert, Schairer, Timo, & Straßer, Wolfgang. 2007. Photorealistic Real-Time Visualization of Cultural Heritage : A Case Study of Friedrichsburg Castle in Germany. *Pages 9–16 of : EG Cultural Heritage Papers*.
- Lachenal, Christophe. 2004. *Modèle et infrastructure logicielle pour l'interaction multi-instrument multisurface*. Ph.D. thesis.
- Lee, Johnny C. 2008. Hacking the Nintendo Wii Remote. *Pervasive Computing, IEEE*, **7**(3), 39–45.
- Likert, Rensis. 1932. A technique for the measurement of attitudes. *Archives of Psychology*, **22**(140), 1–55.

- Liu, Jun, Pinelle, David, Sallam, Samer, Subramanian, Sriram, & Gutwin, Carl. 2006. TNT : improved rotation and translation on digital tables. *Pages 25–32 of : Proceedings of Graphics Interface 2006*. Quebec, Canada : Canadian Information Processing Society.
- Mackinlay, Jock, Card, Stuart K., & Robertson, George G. 1990. A semantic analysis of the design space of input devices. *Human-Computer Interaction*, **5**(2), 145–190.
- Maher, Mary Lou, & Kim, Mi Jeong. 2006. Studying Designers using a Tabletop System for 3D Design with a Focus on the Impact on Spatial Cognition. *Pages 105–112 of : Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE Computer Society.
- Martin, Jean-Claude. 1998. TYCOON : Theoretical framework and software tools for multimodal interfaces. *In : Intelligence and Multimodality in Multimedia Interfaces*. AAAI Press.
- Masry, Mark, Kang, Dong, & Lipson, Hod. 2005. A freehand sketching interface for progressive construction of 3D objects. *Computers and Graphics*, **29**(4), 563–575.
- Matsushita, Nobuyuki, & Rekimoto, Jun. 1997. HoloWall : designing a finger, hand, body, and object sensitive wall. *Pages 209–210 of : Proceedings of the 10th annual ACM symposium on User interface software and technology*. Banff, Alberta, Canada : ACM.
- Milgram, Paul, & Kishino, Fumio. 1994. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, **E77-D**(12).
- Nacenta, Miguel A., Aliakseyeu, Dzmitry, Subramanian, Sriram, & Gutwin, Carl. 2005. A comparison of techniques for multi-display reaching. *Pages 371–380 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA : ACM.
- Navarre, David, Palanque, Philippe, Bastide, Rémi, Schyn, Amélie, Winckler, Marco, Nedel, Luciana P., & Freitas, Carla M.D.S. 2005. A Formal Description of Multimodal Interaction Techniques for Immersive Virtual Reality Applications. *Pages 170–183 of : Human-Computer Interaction - INTERACT 2005*. Rome, Italy : Springer Berlin / Heidelberg.
- Nigay, Laurence, & Coutaz, Joëlle. 1991. Building user interfaces : Organizing software agents. *Pages 707–719 of : Proceedings of ESPRIT'91 conference*.
- Nigay, Laurence, & Coutaz, Joëlle. 1995. A generic platform for addressing the multimodal challenge. *Pages 98–105 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. Denver, Colorado, United States : ACM Press/Addison-Wesley Publishing Co.
- Noisternig, Markus, Katz, Brian F.G., Siltanen, Samuel, & Savioja, Lauri. 2008. Framework for real-time auralization in architectural acoustics. *Acta Acustica United with Acustica*, **94**(6), 1000–1015.

- Norman, Donald A. 1981. Categorization of action slips. *Psychological Review*, **88**, 1–15.
- Ørbæk, Peter. 2004 (Apr.). *A Multi-sided Tabletop Interface*. Unpublished report.
- Palanque, Philippe A., & Schyn, Amélie. 2003. A Model-Based Approach for Engineering Multimodal Interactive Systems. *Pages 543–550 of : Human-Computer Interaction - INTERACT 2003*.
- Peltonen, Peter, Kurvinen, Esko, Salovaara, Antti, Jacucci, Giulio, Ilmonen, Tommi, Evans, John, Oulasvirta, Antti, & Saarikko, Petri. 2008. It's Mine, Don't Touch! : interactions at a large multi-touch display in a city centre. *Pages 1285–1294 of : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. Florence, Italy : ACM.
- Penn, Alan, Mottram, Chiron, gen. Schieck, Ava Fatah, Wittkämper, Michael, Störing, Moritz, Romell, Odd, Strothmann, Andreas, & Aish, Francis. 2005. Augmented Reality Meeting Table : a Novel Multi-User Interface for Architectural Design. *Pages 213–231 of : Recent Advances in Design and Decision Support Systems in Architecture and Urban Planning*.
- Petri, Carl A. 1962. Fundamentals of a Theory of Asynchronous Information Flow. *Pages 386–390 of : Proceedings of IFIP Congress*.
- Pfaff, Günther E. 1985. User Interface Management Systems. *In : Proceedings of the Seeheim Workshop*. Springer Verlag.
- Pinho, Márcio S., Bowman, Doug A., & Freitas, Carla M.D.S. 2002. Cooperative object manipulation in immersive virtual environments : framework and techniques. *Pages 171–178 of : Proceedings of the ACM symposium on Virtual reality software and technology*. Hong Kong, China : ACM.
- Poupyrev, Ivan, Billinghamurst, Mark, Weghorst, Suzanne, & Ichikawa, Tadao. 1996. The go-go interaction technique : non-linear mapping for direct manipulation in VR. *Pages 79–80 of : Proceedings of the 9th annual ACM symposium on User interface software and technology*. Seattle, Washington, United States : ACM.
- Poupyrev, Ivan, Tomokazu, Numada, & Weghorst, Suzanne. 1998. Virtual Notepad : Handwriting in Immersive VR. *Pages 126–132 of : Proceedings of the Virtual Reality Annual International Symposium*. IEEE Computer Society.
- Rekimoto, Jun. 2002. SmartSkin : an infrastructure for freehand manipulation on interactive surfaces. *Pages 113–120 of : Proceedings of the SIGCHI conference on Human factors in computing systems : Changing our world, changing ourselves*. Minneapolis, Minnesota, USA : ACM.
- Roberts, Jonathan C. 1998. On encouraging multiple views for visualization. *Pages 8–14 of : Proceedings of the 1998 IEEE Conference on Information Visualization*.
- Rogers, Yvonne, & Lindley, Siân E. 2004. Collaborating around vertical and horizontal large interactive displays : which way is best? *Interacting with Computers*, **16**(6), 1133–1152.

- Roman, Manuel, Hess, Christopher, Cerqueira, Renato, Ranganathan, Anand, Campbell, Roy H., & Nahrstedt, Klara. 2002. A middleware infrastructure for active spaces. *Pervasive Computing, IEEE*, **1**(4), 74–83.
- Rousseau, Cyril, Bellik, Yacine, & Vernier, Frédéric. 2005. WWHT : un modèle conceptuel pour la présentation multimodale d'information. *Pages 59–66 of : Proceedings of the 17th French-speaking conference on human-computer interaction IHM'05*. Toulouse, France : ACM.
- Schmalstieg, Dieter, ao, L. Miguel Encarnaç & Szalavári, Zsolt. 1999. Using transparent props for interaction with the virtual table. *Pages 147–153 of : Proceedings of the 1999 symposium on Interactive 3D graphics*. Atlanta, Georgia, United States : ACM.
- Schuster, Grit. 2008 (Sept.). *Multitouch Input*. Master Thesis. Hochschule Magdeburg-Stendal (FH) Fachbereich Ingenieurwesen und Industriedesign Institut für Industrial Design, Magdebourg.
- Schwaiger, Martin, Thümmel, Thomas, & Ulbrich, Heinz. 2007. Cyberwalk : Implementation of a Ball Bearing Platform for Humans. *Pages 926–935 of : Human-Computer Interaction. Interaction Platforms and Techniques*. Springer Berlin / Heidelberg.
- Schyn, Amélie, Navarre, David, Palanque, Philippe, & Nedel, Luciana Porcher. 2003. Formal description of a multimodal interaction technique in an immersive virtual reality application. *Pages 150–157 of : Proceedings of the 15th French-speaking conference on human-computer interaction IHM'03*. Caen, France : ACM.
- Shen, Chia, Vernier, Frédéric D., Forlines, Clifton, & Ringel, Meredith. 2004. DiamondSpin : an extensible toolkit for around-the-table interaction. *Pages 167–174 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. Vienna, Austria : ACM.
- Shen, Chia, Ryall, Kathy, Forlines, Clifton, Esenther, Alan, Vernier, Frederic D., Everitt, Katherine, Wu, Mike, Wigdor, Daniel, Morris, Meredith Ringel, Hancock, Mark, & Tse, Edward. 2006. Informing the Design of Direct-Touch Tabletops. *IEEE Computer Graphics and Applications*, **26**(5), 36–46.
- Steinicke, Frank, Bruder, Gerd, Jerald, Jason, Frenz, Harald, & Lappe, Markus. 2008. Analyses of human sensitivity to redirected walking. *Pages 149–156 of : Proceedings of the ACM symposium on Virtual reality software and technology*. Bordeaux, France : ACM.
- Stoakley, Richard, Conway, Matthew J., & Pausch, Randy. 1995. Virtual reality on a WIM : interactive worlds in miniature. *Pages 265–272 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. Denver, Colorado, United States : ACM Press/Addison-Wesley Publishing Co.
- Tandler, Peter. 2004. The BEACH application model and software framework for synchronous collaboration in ubiquitous computing environments. *Journal of Systems and Software*, **69**(3), 267–296.

- Thompson, Mark K., Weal, Mark J., Michaelides, Danius T., Cruickshank, Don G., & Roure, David C. De. 2003. *MUD Slinging : Virtual Orchestration of Physical Interactions*. Tech. rept. ECSTR-IAM03-001, ECS, University of Southampton.
- UIMS. 1992. A metamodel for the runtime architecture of an interactive system : the UIMS tool developers workshop. *ACM SIGCHI Bulletin*, **24**(1), 32–37.
- Vernier, Frédéric. 2001. *La multimodalité en sortie et son application à la visualisation de grandes quantités d'information*. Ph.D. thesis.
- Ware, Colin, & Jessome, Danny R. 1988. Using the bat : a six dimensional mouse for object placement. *Pages 119–124 of : Proceedings on Graphics interface '88*. Edmonton, Alberta, Canada : Canadian Information Processing Society.
- Ware, Colin, & Osborne, Steven. 1990. Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH Computer Graphics*, **24**(2), 175–183.
- Weiser, Mark. 1999. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review*, **3**(3), 3–11.
- Welch, Greg, & Foxlin, Eric. 2002. Motion Tracking : No Silver Bullet, but a Respectable Arsenal. *IEEE Computer Graphics and Applications*, **22**(6), 24–38.
- Wigdor, Daniel, Shen, Chia, Forlines, Clifton, & Balakrishnan, Ravin. 2006. Table-centric interactive spaces for real-time collaboration. *Pages 103–107 of : Proceedings of the working conference on Advanced visual interfaces*. Venezia, Italy : ACM.
- Wilson, Andrew D. 2005. PlayAnywhere : a compact interactive tabletop projection-vision system. *Pages 83–92 of : Proceedings of the 18th annual ACM symposium on User interface software and technology*. Seattle, WA, USA : ACM.
- Wu, Mike, & Balakrishnan, Ravin. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Pages 193–202 of : Proceedings of the 16th annual ACM symposium on User interface software and technology*. Vancouver, Canada : ACM.
- Zhai, Shumin. 1998. User performance in relation to 3D input device design. *SIGGRAPH Computer Graphics*, **32**(4), 50–54.

Glossaire

CDI

Type de Contrôleur de Dialogue d'Initialisation. Un élément du contrôleur de dialogue de ce type n'est pas modifiable dynamiquement.

CDIT

Type de Contrôleur de Dialogue d'Initialisation/Tâche. Un élément du contrôleur de dialogue de ce type est modifiable dynamiquement.

CL

Composante Logique. Elle appartient à la couche interaction logique du modèle physique/numérique. Elle gère la présentation de l'information en sortie d'une entité (objet virtuel, représentation d'un point de vue, point de vue, etc.) sur une vue et/ou les interactions en entrée d'un dispositif en entrée (par exemple sur cette représentation de l'entité).

CLS

Composante Logique Spatiale. Un type de composante logique intégrant les degrés de liberté en entrée et/ou les dimensions d'affichage en sortie.

Combinatoire

Ensemble des combinaisons de composantes (de la conception par *mapping* ou de la conception par composantes logiques) et des connexions possibles entre elles.

EDE

Ensemble des Dispositifs d'Entrée. Une EDE est une combinaison d'un ou plusieurs dispositifs en entrée.

Interaction directe

Une technique d'interaction nécessitant une action de l'utilisateur selon un degré de liberté dans le monde physique et transposable sur un ou des degré(s) de liberté de manipulation ou de navigation dans le monde virtuel.

Interaction indirecte

Une technique d'interaction ne nécessitant pas une action de l'utilisateur selon un

degré de liberté dans le monde physique pour la manipulation d'un objet virtuel ou la navigation.

Interaction partiellement directe

Une technique d'interaction discrète nécessitant une action de l'utilisateur selon un degré de liberté dans le monde physique affectant un degré de liberté d'un interacteur numérique. Les effets de la manipulation du degré de liberté de l'interacteur sont transposables sur un ou des degré(s) de liberté de manipulation de l'objet virtuel ou de la navigation auquel il est associé.

MC

Management Cabin (en français cabine de commande). Système développé combinant des interactions au moyen d'une vue 2D d'une scène virtuelle présentée sur une table tactile et au moyen d'une vue 3D verticale de cette même scène en utilisant différents dispositifs d'entrée.

SEDE

Solution d'un Ensemble de Dispositifs d'Entrée. C'est une solution possible d'une combinaison d'un ou plusieurs dispositifs en entrée.

Solution

Combinaison possible qui satisfait les règles de conception.

Taux d'équivalence

Taux de possibilités de réaliser une tâche de façons différentes.

TEAMViz

Tabletop Environment Augmented with Mural Visualization (en français environnement tabletop augmenté par une visualisation murale). Type de systèmes combinant des interactions au moyen d'une vue 2D d'une scène virtuelle présentée sur une table tactile et au moyen d'une vue 3D de cette même scène virtuelle présentée verticalement.

TICV

Technique d'Interaction de Couplage des Vues. Technique d'interaction permettant de déplacer et d'orienter (de façon disjointe) horizontalement le point de vue 3D par manipulation de la totalité de la projection de la vue 2D présentée sur la table tactile. L'icône de caméra reste fixe sur la table tactile et pointe constamment vers la position de l'écran vertical.

TICVJ

Technique d'Interaction de Couplage des Vues avec Joystick. Technique d'interaction permettant de déplacer horizontalement le point de vue 3D en déplaçant la

totalité de la projection de la vue 2D sur une table tactile. La rotation horizontale du point de vue est réalisée par l'utilisation d'un dispositif de type joystick. L'icône de caméra reste fixe sur la table tactile et pointe constamment vers l'écran vertical.

TIMC

Technique d'Interaction par Manipulation de Caméra. Technique d'interaction permettant de déplacer horizontalement le point de vue 3D en déplaçant une icône de caméra présentée sur une table tactile. La rotation horizontale s'effectue par manipulation d'un interacteur collé graphiquement à l'icône de caméra.

TIMCJ

Technique d'Interaction par Manipulation de Caméra avec Joystick. Technique d'interaction permettant de déplacer horizontalement le point de vue 3D en déplaçant une icône de caméra présentée sur une table tactile. La rotation horizontale du point de vue est réalisée par l'utilisation d'un dispositif de type joystick.

TIMDD

Technique d'Interaction de Manipulation à Deux Doigts. Technique d'interaction permettant de déplacer et d'orienter horizontalement le point de vue 3D en manipulant avec deux doigts une icône de caméra présentée sur une table tactile.

TIOD

Technique d'Interaction d'Orientement du Doigt. Technique d'interaction permettant de déplacer horizontalement le point de vue 3D en déplaçant une icône de caméra présentée sur une table tactile. La rotation horizontale du point de vue est définie selon l'orientation du doigt de l'utilisateur sur l'icône de caméra.

TISD

Technique d'Interaction de Suivi du Doigt. Technique d'interaction permettant de déplacer horizontalement le point de vue 3D en déplaçant une icône de caméra présentée sur une table tactile. L'angle de rotation horizontal est calculé à partir de la tangente de la trajectoire de déplacement de l'icône de caméra.

TISDA

Technique d'Interaction de Suivi de Doigt Augmentée. Technique d'interaction permettant de déplacer horizontalement le point de vue 3D en déplaçant une icône de caméra présentée sur une table tactile. La rotation horizontale est calculée selon la tangente de la trajectoire de déplacement de l'icône de caméra sauf lorsque le déplacement est effectué dans le sens opposé vers lequel l'icône pointe. L'angle de rotation verticale (pitch) est calculé selon la direction (vers l'avant ou vers l'arrière) de déplacement de l'icône de caméra.