



HAL
open science

Un modèle d'interaction 3D : Interaction Homme-Machine et Homme-Homme dans les interfaces 3D pour le TCAO Synchroné.

Cédric Dumas

► **To cite this version:**

Cédric Dumas. Un modèle d'interaction 3D : Interaction Homme-Machine et Homme-Homme dans les interfaces 3D pour le TCAO Synchroné.. Interface homme-machine [cs.HC]. Université des Sciences et Technologie de Lille - Lille I, 1999. Français. NNT : . tel-00402247

HAL Id: tel-00402247

<https://theses.hal.science/tel-00402247v1>

Submitted on 7 Jul 2009

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.



THESE

présentée à

l'Université des Sciences et Technologies de Lille

pour obtenir le titre de

Docteur en Informatique

par

Cédric Dumas

Un modèle d'interaction 3D : Interactions homme-machine et
homme-machine-homme dans les interfaces 3D
pour le TCAO synchrone

Thèse soutenue le 1 octobre 1999, devant la commission d'examen :

Président :	Alain Derycke	Trigone, USTL
Directeurs de Thèse :	Christophe Chaillou	LIFL, USTL
	Marie-Luce Viaud	INA, Bry sur Marne
Rapporteurs :	René Caubet	IRIT, Université Paul Sabatier
	Jocelyne Nanard	LIRMM, Université Montpellier II
Examineurs :	Dave Snowdon	Xerox, Grenoble
	Jean-Daniel Fekete	EMN, Nantes

Université des Sciences et Technologies de Lille
U.F.R. d'I.E.E.A. Bât. M3. 59655 Villeneuve d'Ascq Cedex
Tél. +33-3-20-43-65-57 Fax. +33-3-20-43-65-66

Remerciements

Tout d'abord, tout mes remerciements (et plus) pour Christelle pour son soutien et sa patience au cours de mes quatre années de travail entre Lille et Paris.

Je tiens à remercier Jocelyne Nanard et René Caubet d'avoir accepté d'être mes rapporteurs, Alain Derycke d'être président du jury et pour les commentaires pertinents qu'il m'a fait à la lecture de ma thèse. Merci à Dave Snowdon et Jean-Daniel Fekete d'avoir bien voulu examiner ma thèse.

Mon travail de recherche a été multi-disciplinaire, touchant à des domaines dans lesquels j'ai débuté avec cette thèse. Je tiens à remercier les gens qui ont produit ce travail de recherche avec moi pour les pistes et les enseignements qu'ils ont su m'apporter, en particulier Marie-Luce Viaud sans qui cette thèse n'aurait pas démarré, et Patricia Plénacoste sans qui cette thèse n'aurait pas abouti.

L'ensemble de l'équipe SpIn, qui a su mener ce projet où il en est aujourd'hui, en particulier Christophe Chaillou pour sa gestion du projet et Samuel Degrande pour son investissement.

Les personnes que j'ai côtoyé dans l'équipe Télévirtualité, au Médiaport et au sein de la direction de la recherche de l'INA qui m'ont aidé à avancer techniquement, scientifiquement et intellectuellement.

Un grand merci au travail énorme qu'a fourni Carmen Benito-Garcia pour les traductions et les corrections de nos articles.

Le laboratoire Trigone et plus particulièrement l'équipe NOCE pour nous avoir accueilli pendant plus d'un an de développement.

Yann Coello pour son ouverture d'esprit, les discussions et le début de coopération avec le laboratoire de psychologie cognitive de Lille III.

L'ensemble de l'équipe Graphix pour l'ambiance et le contexte de travail incomparable.

Merci à Michel Petitot pour ses discussions nocturnes, Ramon Pino-Perez pour son soutien en algorithmique, Alain Fargue pour son support technique, Nouredine Bensaïd pour les sorties, l'équipe GOAL pour les pizzas, Luigi pour ses développements, Luc pour les coups de main motorisés, Dual pour faire taire Grincheux, GregouInet pour son support "moral", PK pour les défoulements téléphoniques, Mike pour les figures libres, Olivier et Sabine pour les jeudis soirs, Alex pour la cuisine, Romain et Mathieu pour les discussions sur l'architecture et l'informatique et tous les parasites nocturnes du M3 (Alex, Bugz, Karim, Toyz, Azzedine, etc).

Une thèse est un travail long, qui nous rend parfois (souvent ?) peu disponible physiquement ou psychologiquement, je tiens à remercier mes amis et surtout ma famille d'avoir su me supporter et me soutenir pendant cette période.

Table des matières

INTRODUCTION	1
--------------------	---

Première Partie

1	Etat de l'art : aspects technologiques.....	7	
	1.1	Présentation de la réalité virtuelle	7
	1.2	Les domaines d'application de la réalité virtuelle.....	9
	1.2.1	La simulation.....	10
	1.2.2	La visualisation de données	10
	1.2.3	La communication.....	11
	1.2.4	Bilan	13
	1.3	Synthèse d'image temps réel.....	14
	1.3.1	Le matériel	14
	1.3.2	Les algorithmes	15
		Le rendu projectif	15
		Le calcul des parties cachées	16
		L'ombrage : Gouraud et Phong	17
		Les textures	18
		Les transparences	19
		Les ombres projetées	19
	1.3.3	Les moyens de développement informatique.....	20
	1.4	Les périphériques de sortie et d'entrée	21
	1.4.1	Les modes de visualisation	21
	1.4.2	Les périphériques gestuels	23
		Les périphériques isométriques	25
		Les périphériques isotoniques	25
		Les périphériques élastiques	25
		Les périphériques haptiques	26
	1.5	Aspect multi-utilisateur	26
	1.5.1	La vidéo.....	28
	1.5.2	Les avatars.....	28
	1.5.3	Les techniques de clonage réaliste	29
		Le corps et les bras	29
		Le visage et la bouche	30
		Découpage vidéo	30
		Analyse vidéo	31
		Analyse directe du son	32
		Analyse par Text-To-Speech	32
	1.5.4	Bilan	33

2	Etat de l'art : aspects utilisateur	35
	2.1 Problèmes de perception	36
	2.1.1 La vue	36
	Aspect biologique	36
	Aspect psychologique	37
	L'ergonomie	39
	Vision stéréoscopique	40
	L'immersion	40
	2.1.2 L'ouïe	41
	2.1.3 Le toucher	41
	2.1.4 Bilan	43
	2.2 Le Travail Coopératif Assisté par Ordinateur (TCAO).....	43
	2.2.1 L'hégémonie de la vidéo.....	44
	Les outils audiovisuels	44
	Les outils informatiques	45
	Bilan	45
	2.2.2 Les concepts essentiels du TCAO synchrone.....	45
	L'activité de groupe	45
	Les avatars	46
	L'activité centrée autour de la tâche	46
	La métaphore de la table de réunion	47
	Perception et soutien de la conscience mutuelle	47
	L'animation interactive	47
	2.3 Interfaces 3D.....	48
	2.3.1 Navigation	49
	Perception	49
	Les navigateurs	50
	Bilan	51
	2.3.2 Interaction	52
	La souris?	52
	Pointage	54
	Espace d'action	54
	Sélection	55
	Indices perceptifs	55
	Le geste	55
	Interaction à deux mains	56
	Bilan	57
3	Le projet Space	59
	3.1 Présentation du projet	59
	3.2 Space.....	60
	3.2.1 Objectifs	60
	3.2.2 Etude du domaine d'application.....	60
	3.2.3 Les contraintes	61
	3.2.4 Les actions du monde réel	61
	Les phases	61
	Les tâches à accomplir	61

3.2.5	Les fonctions à réaliser	62
3.3	Odesca / Scenic.....	62
3.3.1	Odesca	63
3.3.2	ScenIC	63
3.4	Spin	64
3.4.1	Les métaphores visuelles liées au travail coopératif : les recommandations.....	64
3.4.2	L'organisation spatiale	65
	Distorsion	66
	Le tout Visible	67
	Un point de focus	67
3.4.3	La téléprésence.....	67
3.4.4	La navigation.....	68
3.5	Mon sujet.....	68

Deuxième Partie

4	Un modèle d'interaction 3D	73
4.1	L'essentiel du geste : passage du monde réel au virtuel.....	76
4.1.1	Décomposition du geste d'action	76
4.1.2	Le geste d'action dans l'espace virtuel	78
4.1.3	Bilan	79
4.2	Désignation.....	80
4.2.1	Choix du périphérique.....	80
	Dispositif expérimental	82
	Procédure	83
	Données recueillies et traitement	84
	Résultats	84
	Bilan	88
4.2.2	Précision.....	88
	Bilan	88
4.2.3	Désignation dans l'interface.....	89
	Le pointeur	89
	Transparence	89
	Dynamique du pointeur	89
	Taille du pointeur	89
	Calibration	90
4.2.4	Bilan	90
4.3	Indices visuels	90
	Les textures	92
	L'ombre	92
	Les boîtes englobantes	96

	La mobilité relative	98
	Bilan	100
4.4	Sélection de l'objet : interaction bimanuelle	100
4.4.1	Périphérique	100
4.4.2	Visualisation de la sélection	101
4.4.3	L'instrumentation	102
4.4.4	La manipulation	102
	Manipulation directe	102
	Les documents dans SpIn	103
	Quel moyen pour manipuler les documents?	103
	Bilan	105
4.5	Bilan sur le modèle d'interaction	105
5	SpIn : interactions locales	107
5.1	retour sur les concepts de SpIn	108
5.1.1	la navigation	108
5.1.2	l'organisation spatiale de l'espace de travail	109
5.1.3	La notion de focus	110
	mode explicite	110
	mode implicite	112
	bilan	112
5.2	gestion des outils	113
5.2.1	palette d'outils	113
	palette bimanuelle	113
	bilan	114
	Toolglass	114
5.2.2	le conetree	115
5.2.3	barre d'outils	115
5.3	gestion des actions	117
5.3.1	Les menus 3D	118
5.3.2	Les menus 3D dans les jeux	119
5.3.3	Propositions	119
	Une dimension	120
	Deux dimensions	120
	Trois dimensions	121
5.3.4	Bilan	122
5.4	Accès aux documents	122
5.4.1	Visualisation	122
5.4.2	Interaction	124
5.5	Bilan	125

6	SpIn : interactions à distance	127
6.1	La téléprésence.....	128
6.2	Communication	130
6.2.1	La voix	130
6.2.2	L'animation faciale.....	130
6.2.3	Le geste	131
6.2.4	Bilan	132
6.3	Coopération.....	132
6.3.1	La présence	133
6.3.2	La position.....	133
6.3.3	Le point de vue.....	133
6.3.4	Le point d'action	134
6.3.5	Disponibilité et degré de présence	134
6.3.6	Les gestes et les expressions faciales	135
6.3.7	Liens avec les objets et informations	136
6.3.8	Historique de l'activité.....	136
6.3.9	Bilan	136
6.4	Désignation à distance.....	136
6.4.1	Analyse de l'apparence (la forme)	137
6.4.2	Le mouvement.....	138
6.4.3	Durée de vie	139
6.4.4	Analyse du facteur de groupe.....	139
	Nombre de pointeurs	139
	Identification du manipulateur	139
	Politique de partage	139
	Degré d'autonomie	139
6.4.5	Bilan	140
6.5	Manipulation à distance.....	140
6.6	Gestion du point de vue.....	141
6.7	Partage et mise à disposition de documents.....	141
6.7.1	La mise à disposition de document	142
6.7.2	Les modes de partages	143
	Mode dupliqué	143
	Mode partagé complet	143
	Modes intermédiaires	143
	Récapitulatif	146
6.8	Bilan	147

7	SpIn : un exemple d'application	149
7.1	L'interface actuelle.....	149
7.1.1	Les périphériques.....	149
7.1.2	Le point d'action.....	150
7.1.3	L'organisation des documents	152
7.1.4	Les clones	152
7.1.5	Les outils	153
7.2	L'appareil photo	154
7.2.1	Les documents	155
Un appareil photo		155
Un objectif virtuel		155
Un simulateur de prise de photo		155
Une documentation		156
7.2.2	Modes de partage.....	156
7.3	Evaluation.....	157
7.3.1	Les objectifs.....	157
7.3.2	Les conditions.....	157
Personnes concernées		158
Matériel utilisé		158
7.3.3	Résultats des questionnaires	158
Bilan		163
Les problèmes de fond		164
7.3.4	Résultats de l'analyse vidéo	165
Coopération		165
Interaction		165
7.3.5	Bilan	166
	 CONCLUSION.....	 169

Bibliographie

.....	175
-------	-----

Annexes

Annexe 1: Architecture logicielle de SpIn.....	183
1.1 Structure de données.....	184
1.1.1 Le format VRML.....	184
1.1.2 Le format VRML dans SpIn.....	185

1.1.3	Structure objet	187
1.2	Gestion de l'interaction.....	189
1.2.1	Dynamique des documents	189
1.2.2	Evènements	189
	Pile de messages	190
	Structure objet	190
1.2.3	Actions	192
	Une phase d'action dans SpIn	192
	Action persistante	193
	Description du comportement	193
	Structure objet	194
1.2.4	La barre d'outils	195
1.2.5	Bilan	196
1.3	Entrée/Sortie	197
1.3.1	Graphique.....	197
1.3.2	Son	197
1.3.3	Périphériques.....	198
1.3.4	Communication réseau.....	199
1.4	Informations interprétées	200
1.4.1	Télépointeur	200
1.4.2	Le regard des clones.....	200
Annexe 2: Eléments de l'évaluation		203



INTRODUCTION

Les progrès de la technologie font évoluer les modes de travail. En particulier en matière de communication où le développement de nouveaux modes, comme les services d'Internet, ou de nouveaux moyens, comme le téléphone portable, changent les habitudes de travail. Cela permet aux entreprises de se déployer géographiquement en profitant de plus en plus de systèmes de communication adaptés aux grandes distances; de même à l'intérieur de l'entreprise, où le personnel s'approprie de nouvelles stratégies d'organisation. Le lieu de travail peut désormais être modifié selon les nécessités de l'entreprise, chez soi, chez le client ou dans un local loué, faisant du bureau traditionnel et fixe un lieu obsolète. Au niveau des supports, la notion d'Intranet se développe, on utilise de plus en plus les systèmes informatiques comme support de communication dans l'entreprise.

Autrement dit, les distances géographiques sont de plus en plus grandes, les personnes s'éloignent et se déplacent plus, cela crée des besoins en matière de communication. Pour répondre à ces nouvelles exigences, on dispose de solutions de travail coopératif synchrone; en particulier l'audio-conférence sur des réseaux comme Internet, ou les outils de visio-conférence, basés sur la vidéo, qui ont pu profiter des avancées technologiques en matière de compression, autorisant la diffusion à travers les réseaux de communication actuels (Internet, RNIS ou des supports spécialisés).

Parallèlement, la puissance des micro-ordinateurs s'est accrue, en terme de calcul, et en terme d'affichage. La synthèse d'images temps réel est à la portée du plus grand nombre de nos jours. La question porte donc désormais plus dans beaucoup de domaines (comme la bureautique ou le dessin) sur l'utilisation de cette puissance pour un meilleur rendement, plutôt que sur son augmentation.

C'est pourquoi, il y a quelques années, l'idée a été lancée de proposer «un nouvel outil de communication autour d'un calculateur permettant de télé-travailler» :

Le terme calculateur est utilisé pour faire référence aux performances des machines actuelles, ce qui se traduit par le développement d'un environnement permettant d'établir une coopération dans un petit groupe au moyen d'une nouvelle interface. Il sous-entend également la notion de terminal de communication individuel.

Le télé-travail a ici pour objectif principal de recréer les conditions d'une réunion à distance. Les situations concernées sont les réunions d'étude, de conception, de discussion dans les entreprises, et plus généralement des situations de télé-enseignement, de support technique ou de vente à distance.

Il s'agit dans tous les cas de discussions autour d'un ou plusieurs documents, de la modification de l'un d'entre eux, ou de la production d'un nouveau. Le groupe ayant une activité de travail commune, il s'agit de lui permettre de communiquer à distance (chacun devant sa machine sur un réseau local ou étendu), puis de coopérer réellement à travers un outil informatique. Les documents partagés peuvent être de n'importe quel type, leur aspect éventuellement multi-dimensionnel étant particulièrement bien intégrable dans un espace 3D (tableau de calculs, objet modélisé en CAO, etc...). Notre travail consiste donc à définir les exigences du Travail Coopératif Assisté par Ordinateur (TCAO) synchrone et à les corrélérer avec un ensemble d'éléments d'interface homme-machine.

Il s'agit d'apporter une solution de travail coopératif synchrone. Notre démarche est une démarche de proposition, elle est basée sur trois contraintes liées à nos objectifs :

- utiliser une représentation en trois dimensions. Nous l'avons choisie pour la richesse qu'elle pourrait apporter vis-à-vis des systèmes en deux dimensions. Notre étude montrera les avantages qu'elle apporte pour les environnements de travail coopératif synchrone.
- permettre la communication indispensable à la base d'une réunion, mais surtout mettre l'accent sur la coopération.
- rester compatible avec un "environnement bureautique", notre projet se positionnant à tous les niveaux de l'entreprise. Ce qui se traduit par plusieurs principes :
 - Tout d'abord au niveau des coûts matériels, le déploiement dans une industrie du secteur tertiaire doit rester possible. On n'envisagera donc pas des solutions fonctionnant sur des machines très puissantes, qui ne risquent pas de se démocratiser dans les années à venir (SGI Onyx par exemple) ou en utilisant des périphériques au coût prohibitif (visio-casque stéréo à large champ de vision),
 - Ensuite, l'environnement doit pouvoir être l'outil de travail de tous sur le lieu de travail. Il ne faut donc pas perdre de vue l'utilisateur potentiel auquel nous nous adressons, et envisager un environnement tout public,
 - Enfin, pendant une réunion virtuelle, un utilisateur doit pouvoir rester en contact avec son environnement de travail direct (ses collègues, son téléphone, etc).

Nous nous proposons donc de créer une interface 3D pour le travail coopératif synchrone répondant à ces critères. Le contexte de ce travail est multi-disciplinaire (travail coopératif, IHM, synthèse d'images, etc.). Nous avons donc été amenés à élargir notre champ de recherche et nos collaborations.

Le projet, appelé SPACE¹, s'est déroulé en coopération entre plusieurs équipes, en particulier l'équipe Graphix du LIFL (Laboratoire d'Informatique Fondamentale de Lille) pour l'interface 3D proprement dite et l'équipe Noce du laboratoire Trigone pour l'aspect Travail coopératif et l'évaluation. D'autres équipes de recherche se sont trouvées par la suite impliquées dans le projet :

1. Ce projet a été réalisé dans le cadre du projet régional Ganymède sur les communications avancées (1994-1998), en coopération avec le CNET (CTI sur le travail coopératif, 1995-1997), toujours partie prenante dans le projet par l'intermédiaire de l'équipe PEPS du CNET de Lannion.

l'équipe Télévirtualité de l'INA (Institut National de l'Audiovisuel), pour le clonage, et l'équipe Perception-Action du laboratoire de psychologie cognitive de Lille III, pour des aspects psychologiques.

La recherche et le développement de ce projet ont été assurés par deux doctorants, Grégory Saugis entre 1994 et 1998, grâce à une bourse du Centre National d'Etudes en Télécommunications (CNET) et de la région Nord-Pas de Calais, et moi-même depuis 1995 grâce à une bourse BDI du CNRS, en coopération avec la région Nord Pas de Calais et l'Institut National de l'Audiovisuel (INA). Les premiers travaux de G. Saugis ont permis de définir un modèle d'organisation de l'espace de travail virtuel, appelé SpIn (pour Space Interface).

Il s'agissait ensuite de rendre cet environnement interactif et communicant. J'ai donc traité les problèmes concernant l'interaction homme-machine dans une interface de travail tridimensionnelle. Il a fallu pour cela repenser complètement les rapports entre l'utilisateur et l'environnement de travail, ne pas se baser sur les conventions des interfaces classiques, qui ne traitent pas les problèmes liés à la troisième dimension ou au travail coopératif.

J'ai ensuite considéré l'interaction homme-homme médiatisée, c'est-à-dire la coopération entre les interlocuteurs, en commençant par leur donner une représentation adaptée dans l'environnement virtuel, afin d'établir des relations de présence et de coopération.

Ces différents aspects sont présentés dans ce rapport de thèse en deux parties :

- la première partie proposera un état de l'art des domaines abordés pour résoudre notre problématique, d'un point de vue technique et du point de vue de l'utilisateur.
- la seconde partie décrira et justifiera notre modèle d'interaction 3D pour une interface de travail tridimensionnelle dans le cadre du travail coopératif synchrone, d'un point de vue interaction homme-machine et homme-machine-homme. Une mise en situation de notre environnement et son évaluation y seront exposées.

L'exposé ainsi fait de notre démarche et de nos travaux débouchera alors sur les conclusions de nos recherches et les perspectives pour l'avenir.

Enfin, en annexe de cette thèse, seront présentés les aspects techniques de notre environnement qui ont permis de mettre en oeuvre nos propositions en matière d'interaction, en particulier l'architecture logicielle.

Remarque : on trouvera dans l'ensemble de ce manuscrit des textes écrits avec un style différent (police différente, texte décalé sur la droite) pour indiquer que je n'en suis pas le premier auteur.

1^{ère} Partie

Dans cette première partie, nous allons mettre en place la démarche suivie dans le cadre de cette thèse. Nous procédons à une analyse bibliographique avec une double approche - de technologie et d'usage. Ce travail s'inscrit à la croisée de nombreuses thématiques : les interfaces homme-machine, les images de synthèse, la réalité virtuelle ou encore le travail coopératif. Nous examinons chacun de ces thèmes par rapport aux problèmes de visualisation, d'interaction et de coopération que pose la définition d'un modèle d'interface 3D pour le travail coopératif synchrone.

Le premier chapitre concerne donc l'aspect technologique. Nous débutons avec une explication de la réalité virtuelle : notre projet s'inscrit dans ce domaine et partage certaines de ses problématiques, en particulier les périphériques d'entrée/sortie, la synthèse d'image temps réel et l'aspect multi-utilisateur.

Dans le second chapitre, nous abordons le point de vue de l'utilisateur, c'est-à-dire les problèmes humains liés aux interfaces 3D et à la coopération, à savoir : les capacités de perception de l'utilisateur, vis-à-vis de la vision en trois dimensions en particulier, les concepts du travail coopératif synchrone, et les IHM 3D avec les problèmes liés à la navigation et à l'interaction en trois dimensions.

Dans le dernier chapitre de cette partie, nous présentons l'état du projet au début de ma thèse, et les travaux qui ont été réalisés sur le travail coopératif et l'organisation de l'espace de travail virtuel. Puis nous abordons la description des travaux menés dans le cadre de cette thèse.



Notre projet concernant les interfaces 3D, nous nous sommes intéressés aux domaines de la synthèse d'images, des périphériques d'entrée, aux technologies de visualisation et au travail coopératif assisté par ordinateur. L'étude de ces différents aspects dans le cadre des environnements 3D nous rapproche des problématiques de la réalité virtuelle, c'est donc à partir de l'étude de ce domaine que nous allons construire notre état de l'art sur les aspects technologiques de notre recherche.

1.1 Présentation de la réalité virtuelle

Pour Claude Cadoz [Cadoz 94a], la formule «réalité virtuelle» est absurde dans la mesure où les deux termes s'opposent (c'est une oxymore). Il parle plutôt de réalité intégrale, minimisant l'impact de ce nouveau domaine en montrant que la nouveauté est dans le degré d'intégralité que l'ordinateur permet d'atteindre et dans l'usage que l'on peut en faire. Le terme réalité virtuelle est cependant le nom adopté communément. D'un point de vue fonctionnel, on peut tout de même dire sans équivoque que la réalité virtuelle se situe à l'intersection de plusieurs disciplines et techniques de l'informatique, dans le but de créer des environnements sur ordinateur qui simulent un espace particulier. Ces espaces peuvent être ceux de notre réalité, lorsqu'il s'agit par exemple de simulations (simulateur de vol, etc), ou ils peuvent être artificiels comme certains «mondes virtuels», ces espaces de communication que l'on rencontre de plus en plus sur Internet (voir un des premiers en trois dimensions World Chat de Worlds Inc.).

Pour Grigore Burdea [Burdea 96], le point clé de la réalité virtuelle est l'interaction temps réel multi-modale qui implique l'utilisateur à travers tous ses sens, et permet son immersion complète. C'est une caractéristique que l'on retrouve à travers beaucoup de définitions (toutes ?) de la réalité virtuelle. Plus formellement, [Gigante 93] la caractérise comme une application qui donne l'illusion de participer à l'activité d'un environnement virtuel plutôt que d'en être le simple observateur extérieur.

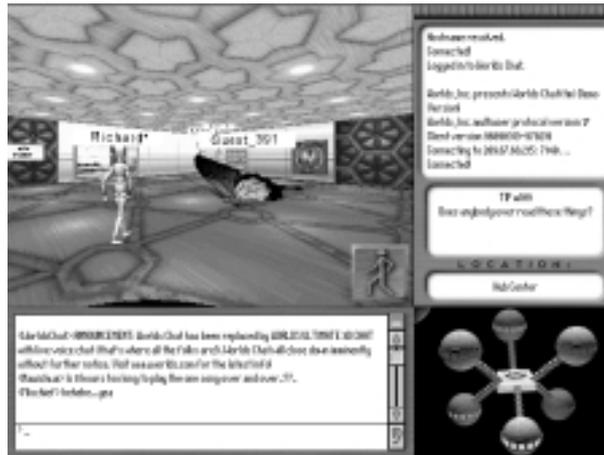


Figure 1.1 *World Chat de Worlds Inc.*

Les programmes de réalité virtuelle sont complexes à mettre en oeuvre et issus de plusieurs domaines, comme ceux de la synthèse d'images temps réel et des interfaces homme-machine (IHM). Toutefois, il est important de noter que le domaine des IHM est principalement focalisé sur les interfaces 2D. On peut l'enrichir en lui apportant des éléments concernant la troisième dimension en étudiant le domaine de la psychologie. Celle-ci nous fournit des éléments pour mettre en place des expérimentations permettant d'améliorer la position de l'utilisateur en environnement virtuel. Ainsi, certaines problématiques de la psychologie cognitive se trouvent en connection directe avec celles de la réalité virtuelle. La psychologie peut donc nous apporter, outre ses compétences expérimentales, des solutions ou des recommandations de conception (voir chapitre 2).

Un autre élément important, après la représentation de ces environnements, est l'interface entre l'utilisateur et la machine. Elle est de très haut niveau, avec des périphériques parfois complexes. Avec la troisième dimension et plus particulièrement les simulateurs, la technologie utilisée devient vite très pointue. Il s'agit en général de détecter les mouvements du corps, voire de leur répondre avec des systèmes de retour d'effort. La technologie est ciblée en fonction de l'application : faux cockpit d'avion pour les simulateurs de vol, morceau de voiture pour les simulateurs de conduite ou encore des capteurs de positions 3D pour mesurer les mouvements du corps dans un espace fini. Les aspects technologiques sont donc très présents et indissociables de l'étude d'un environnement virtuel.

Le dernier grand thème qui intéresse certains environnements de la réalité virtuelle est l'aspect multi-utilisateur : connecter plusieurs personnes ensemble dans le même lieu virtuel, qu'ils puissent se voir, communiquer, voire travailler ou jouer ensemble. Les aspects techniques, souvent assez lourds à mettre en oeuvre à grande échelle, concernent la technologie des réseaux. L'aspect communication concerne le domaine du TCAO, qui puise ses connaissances sur la coopération et les rapports inter-humains dans différents domaines des sciences humaines comme la psychologie sociale ou l'ethnographie.

1.2 Les domaines d'application de la réalité virtuelle

Les domaines d'application de la réalité virtuelle sont multiples. S'ils ont tout d'abord été principalement ludiques ou liés à la simulation, de nouvelles perspectives leurs permettent de se développer actuellement, grâce à une prise de conscience de certains acteurs de la recherche et de l'industrie : la réalité virtuelle peut simplifier le rapport de l'homme à l'ordinateur, en particulier par son approche nouvelle de l'interface homme-machine, d'un point de vue technique et d'un point de vue humain. Surtout si l'on considère que les informations que l'homme doit gérer sont de plus en plus importantes (par la taille), et de plus en plus souvent numériques, la réalité virtuelle peut-être une solution aux limites et aux conventions des systèmes à interface 2D actuels.

On peut décrire la réalité virtuelle à partir de ses champs d'application (*que contient la réalité virtuelle ?*) ou à partir des domaines auxquels elle fait appel (*qui contient de la réalité virtuelle ?*). C'est le cas de [Coomans 97], qui propose une taxonomie des interfaces utilisateur fondée sur des éléments empruntés aux interfaces homme-machine. Il délimite ensuite les champs de la réalité virtuelle pouvant s'inscrire à l'intérieur de son modèle, c'est-à-dire ceux possédant une interface utilisateur : l'interaction, l'immersion, la simulation et la visualisation. Son modèle ne dépend pas des applications, il situe les domaines d'activité de la réalité virtuelle dans un contexte plus large, celui des interfaces homme-machine.

[Fuchs 96] propose une taxonomie plus fonctionnelle, indépendante des domaines d'activité, fondée sur la définition d'une activité suivant trois points : le lieu, l'instant et l'interaction (Figure 1.2).

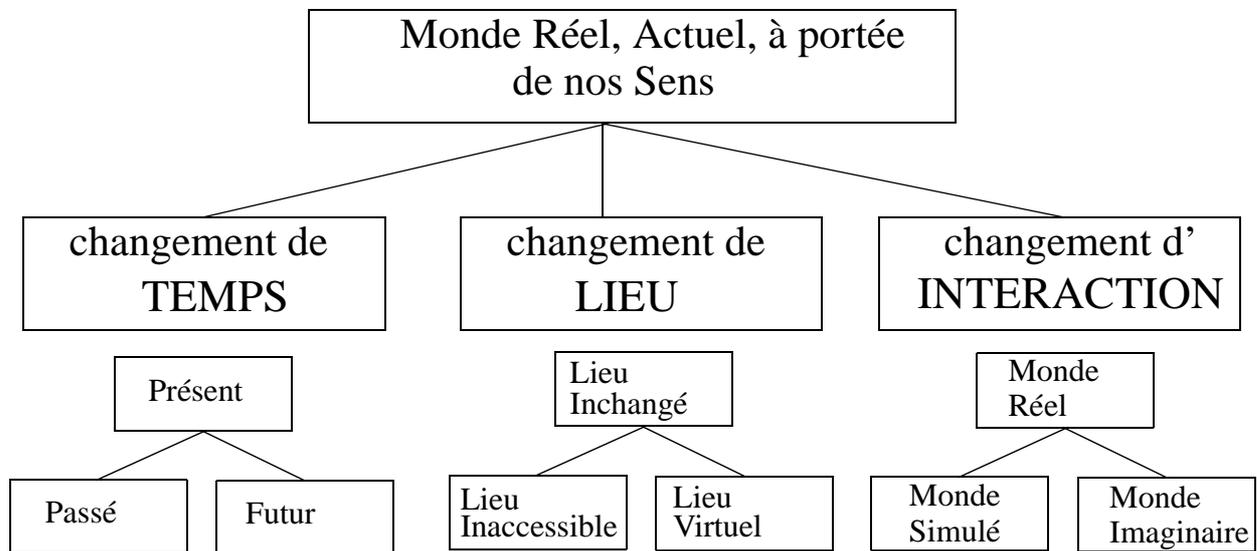


Figure 1.2 classification des applications suivant des états, d'après P. Fuchs

Par exemple, la télévirtualité (communication entre personnes à travers leurs clones dans un espace virtuel) se code sous la forme : Temps présent / Lieu virtuel / Monde Imaginaire. Cette méthode permet de classer dans une catégorie n'importe quel type d'application suivant trois critères qui sont spécifiques au monde dans lequel évolue l'utilisateur. Ces critères sont génériques, et révélateurs pour situer immédiatement un environnement donné.

Nous discutons ici de trois grands domaines de la réalité virtuelle, les plus proches de notre problématique, qui apportent des éléments d'évolution à notre modèle : la simulation, la visualisation de données et la communication.

1.2.1 La simulation

La taxonomie de ce type d'environnement est Temps présent / Lieu inchangé / Monde simulé. La simulation consiste à calculer un phénomène réel. Les simulateurs sont destinés à l'assistance en opération ou à la formation. Les problématiques tournent autour de la synthèse d'images pour représenter avec le plus de fidélité possible le phénomène simulé, et sur des problèmes mécaniques pour reproduire les dispositifs d'interfaçage homme-machine à simuler. On les trouve en particulier dans les domaines médicaux, militaires, industriels et éducatifs, mais également dans les jeux. Le simulateur n'introduit pas de problème d'IHM dans la mesure où il cherche à copier au mieux la réalité (les problèmes d'IHM étant éventuellement ceux du système réel simulé). Cependant les limitations technologiques des ordinateurs et des périphériques induisent parfois des simplifications dans certaines parties du simulateur, ce qui peut créer des problèmes d'utilisabilité ou de réalisme du simulateur, et par conséquent des problèmes de compréhension par l'utilisateur.

1.2.2 La visualisation de données

La taxonomie de ce type d'environnement est en général Temps présent / Lieu virtuel / Monde imaginaire. La visualisation de données abstraites consiste à représenter en deux ou trois dimensions, sur plusieurs axes et de façon dynamique, une quantité importante de données. En effet, si l'on maîtrise bien la représentation de tableaux à deux dimensions, par exemple sous forme de camembert dans les tableurs, il n'en va pas de même pour des tableaux possédant beaucoup de dimensions (tableau multi-dimensionnel de données abstraites). Le modèle de visualisation choisi doit être adapté au type de données visualisées : c'est lui qui doit permettre de les densifier et de mettre en valeur les particularités de l'ensemble et les variations majeures. Ceci afin de rendre possible une analyse globale et rapide, et de faire découvrir des éléments invisibles sous une forme brute. Si le modèle est bien choisi, l'effort cognitif d'interprétation sera moins important. Il s'agit donc de créer des modèles 3D dynamiques qui absorbent dans leur représentation l'ensemble des données, qui soient manipulables, et que l'on puisse visualiser afin de les annoter et de les analyser. C'est encore tout particulièrement la synthèse d'image temps réel qui permet de gérer et de manipuler facilement ces données. Xerox a mené des recherches sur différentes façons d'afficher des données en utilisant une représentation 3D, aboutissant au paradigme d'*information workspace* dont l'*Information Visualizer* [Robertson93] est une mise en oeuvre. Les deux exemples les plus connus sont le *Perspective Wall* et le *Cone Tree* (une suite de listes sur le *Perspective Wall* et un visualiseur de fichier type *Cone Tree*).

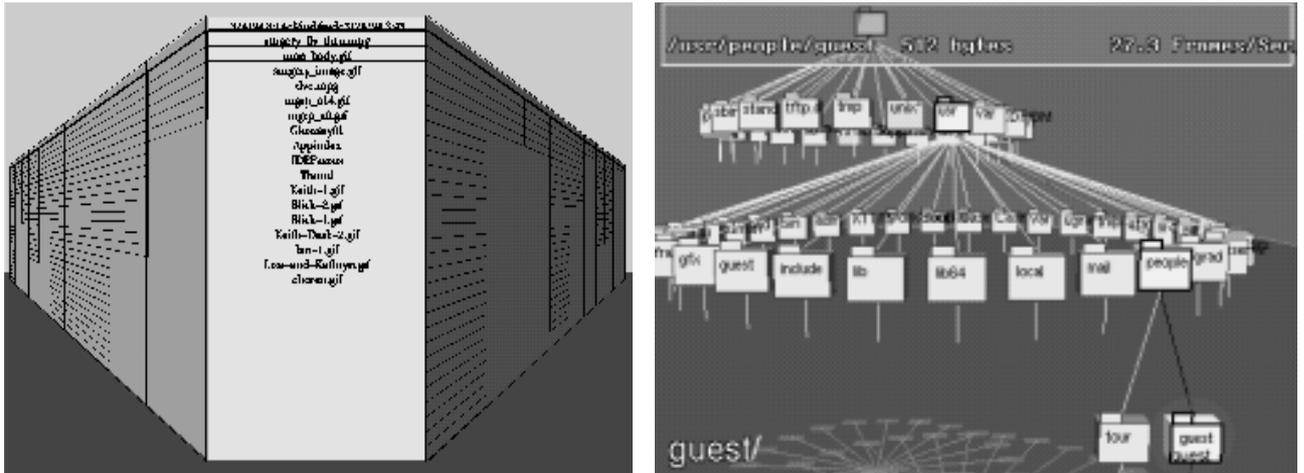


Figure 1.3 une suite de listes sur le Perspective Wall et un visualiseur de fichier type Cone Tree

1.2.3 La communication

Ce sont les loisirs qui ont permis la diffusion de logiciels de communication temps réel sur Internet (et donc à grande échelle). Tout d'abord, les MUD (Multi-User Dungeon) sont des environnements ludiques partagés qui existent depuis longtemps. Ils étaient dans un premier temps en mode texte (TinyMUD, LPMUD, DikuMUD, AberMUD), puis graphique comme BSX (une extension de LPMUD, ayant donné naissance à Regensis ou Regenbogen, BSX, un client de MUD et le 2ème monde, le monde virtuel de Cryo et Canal+). La taxonomie d'un MUD est en général Temps présent / Lieu virtuel / Monde Imaginaire.

L'émergence de la Réalité Virtuelle a coïncidé avec la généralisation de la mise en réseau des ordinateurs et la vulgarisation d'Internet. Il était inévitable que la représentation graphique 3D attrayante de l'une se joigne aux possibilités de communication rapides, presque en temps réel, de l'autre. Les logiciels de communication en mode texte avec une interface 3D (et parfois avec la voix) se sont développés rapidement avec Onlive Traveller, World Chat, le 2ème monde (premier chat 3D créé en Europe), etc.

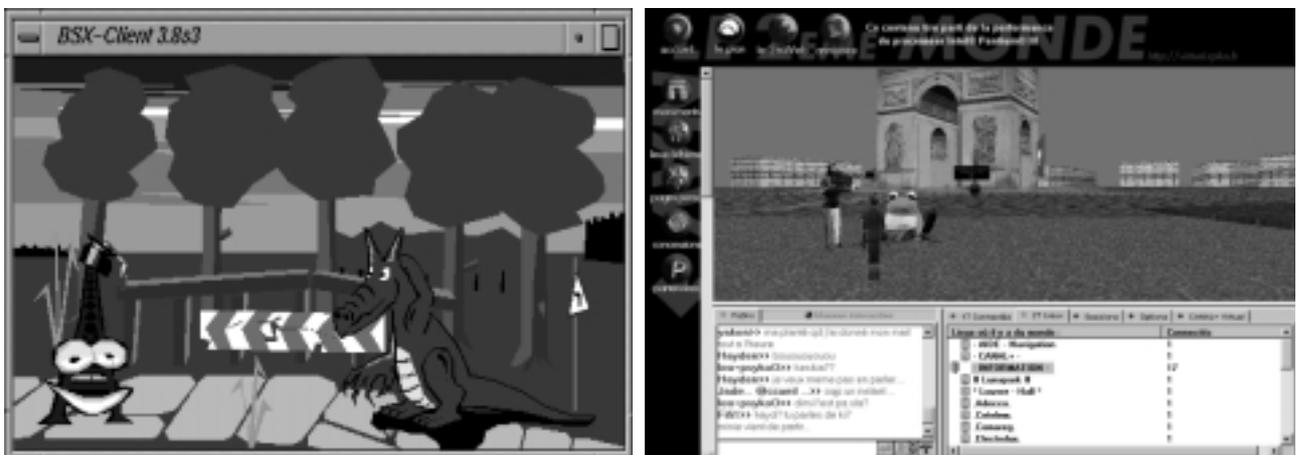


Figure 1.4 BSX, un client de MUD et le 2ème monde, le monde virtuel de Cryo et Canal+

Les mondes virtuels ont tout d'abord été des logiciels de discussion évolués sur Internet (de *chat*); ils ont servi à expérimenter des serveurs (comme chez Onlive) ou à faire des études de comportement, comme Habitat, la première communauté virtuelle reliant plusieurs centaines de personnes [Morningstar 90]. Ces logiciels de discussion, souvent en trois dimensions, ne présentent souvent que peu de fonctionnalités supplémentaires par rapport à un logiciel en mode texte. En dehors de l'interface graphique plus belle, on peut parfois ne discuter qu'avec les personnes proches de soi, créant ainsi une notion de groupe liée à la représentation graphique (cf. Onlive Traveller). Certains mondes virtuels se distinguent en incluant des notions liées au monde réel, comme l'acquisition d'un logement, sa personnalisation (cf. le 2^{ème} monde), la possibilité de se marier ou de divorcer (cf. Habitat), de publier des informations, etc.

Ces idées sont reprises maintenant au profit de jeux (Mankind de Vibes (stratégie) et Ultima Online de Origin (aventure), le premier du genre) auxquels on ne peut jouer qu'en étant connecté à Internet (la communication directe en mode texte faisant parallèlement toujours le succès d'IRC¹). Leur contenu est souvent plus riche et ne reproduit pas forcément les règles du monde réel (phénomène trop courant dans les mondes virtuels), au profit de mondes imaginaires mettant en valeur l'aspect virtuel.

Des mondes virtuels ont même donné naissance à des jeux où co-existent les personnages gérés par des joueurs (Meridian 59 de 3D0). Mais les performances des ordinateurs personnels et surtout des serveurs connectant plusieurs centaines de personnes simultanément sur Internet permettent de créer maintenant des jeux complets, de plus en plus interactifs, mêlant l'action ou la stratégie avec la communication directe vers d'autres joueurs. Ces jeux sont de plus en plus complexes et on voit les joueurs se regrouper en guildes ou en communauté afin d'évoluer ensemble à l'intérieur du jeu. Ces groupes possèdent des règles plus ou moins strictes, en particulier au niveau de la hiérarchie entre joueurs, de l'obéissance ou des temps de jeux minimums. Ces nouveaux programmes plongent l'utilisateur dans des univers virtuels de très grande dimension, dont la forme graphique présente plus d'intérêt que celle d'une simple navigation.

1. IRC : Internet Relay Chat, un service de discussion textuel organisé par forum et relayé à travers le monde par un certain nombre de serveurs dédiés. Il existe de nombreux logiciels clients.



Figure 1.5 *Mankind* de Vibes (stratégie) et *Ultima Online* de Origin (aventure), le premier du genre

La communication dans un registre moins ludique passe encore par des moyens plus traditionnels et évolue moins vite en entreprise. Le téléphone et le fax sont encore les moyens privilégiés de communication, les messageries internes et autres services d’Intranet, parfois connectés à Internet, se développent beaucoup sans pour autant proposer d’outils évolués de communication temps-réel. Il existe des systèmes de visio-conférence ou des salles de conférence équipées pour communiquer (le Reality Center de SGI France et un visiophone France Telecom de 1991), mais pas encore de produit entièrement sur ordinateur et appartenant au domaine de la réalité virtuelle. Comme notre projet, les applications actuelles font encore partie de la recherche (voir chapitre 2).

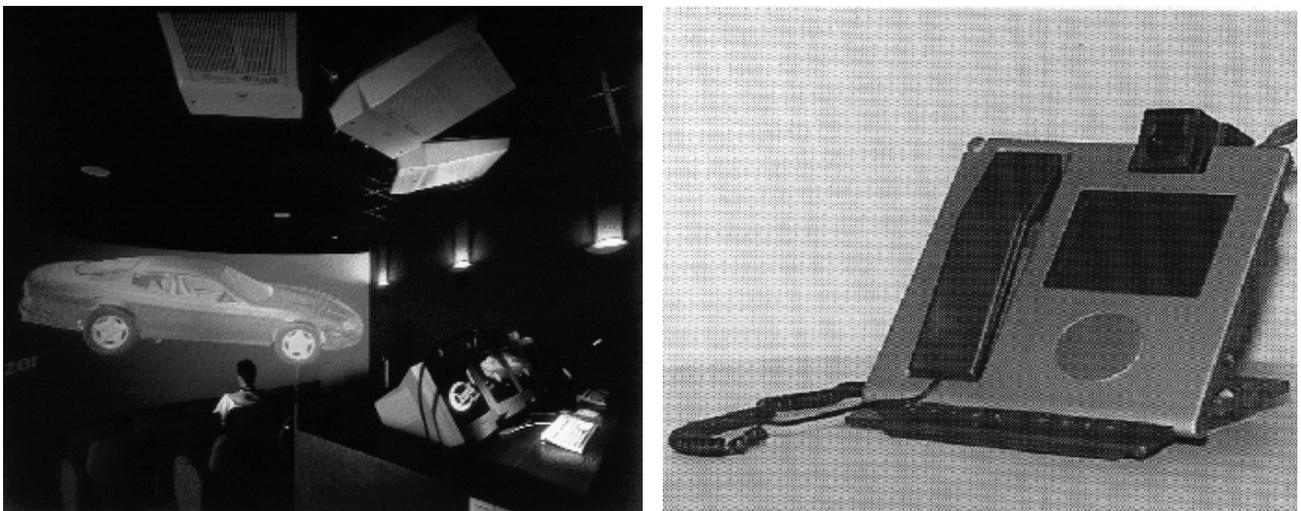


Figure 1.6 *le Reality Center* de SGI France et un visiophone France Telecom de 1991

1.2.4 Bilan

Nous venons de discuter de trois domaines de développement de la réalité virtuelle : la simulation, la visualisation et la communication. On peut constater que c’est l’évolution de plusieurs technologies, comme la synthèse d’image ou les réseaux, qui ont permis à la réalité virtuelle de se développer. Ainsi, certains domaines, comme la communication ou les jeux ne peuvent prendre leur essor dans

le domaine de l'informatique qu'en répondant aux problématiques de la réalité virtuelle. De nouveaux usages vont permettre également à la réalité virtuelle de se développer, en particulier dans des secteurs de l'industrie où des applications ont besoin d'être simplifiée ou peuvent gagner à utiliser une nouvelle approche, un nouveau mode de communication entre l'homme et l'ordinateur.

1.3 Synthèse d'image temps réel

Synthétiser des images sur ordinateur consiste à calculer la projection d'une scène 3D sur un plan, l'image projetée sur ce plan étant ensuite affichée à l'écran. Pour obtenir un rendu réaliste, c'est-à-dire possédant de multiples effets d'ombre, de reflet et de transparence, on utilise des algorithmes qui nécessitent des calculs longs, impossibles à réaliser actuellement en temps réel (sauf à des coûts prohibitifs). On entend par «temps réel», la génération (calcul) et l'affichage d'une suite d'images animées à au moins 10 images par seconde. Au delà de 25 images par seconde, l'oeil ne saisit plus les transitions d'une image à l'autre. Il est donc possible que ces séquences soient interactives puisque générées en temps réel. Le rendu «temps réel» fait appel à des algorithmes différents du rendu dit réaliste, optimisés pour le temps plus que pour la qualité d'image.

1.3.1 Le matériel

La disponibilité des cartes graphiques 3D sur station de travail (fin des années 80) puis sur des micro-ordinateurs (milieu des années 90) a permis le développement d'applications interactives en trois dimensions. On trouve sur ces cartes les algorithmes de rendu projectif, les plus rapides pour générer des images de synthèse ; il s'agit de projeter sur un plan (l'écran) l'ensemble des objets 3D de la scène, suivant le point de vue d'un observateur placé dans la scène. Ces objets sont tous représentés sous la forme d'un ensemble de facettes planes. Pour effectuer le tri en profondeur des objets, on trouve quasi-exclusivement l'algorithme du Z-buffer implémenté sur les cartes 3D actuelles (il ne nécessite aucun pré-traitement). L'éclairage des objets (détermination des couleurs) s'effectue par un calcul d'éclairage empirique, avec la méthode de Phong ou de Gouraud, aux sommets des faces des objets, puis par interpolation en ce qui concerne l'intérieur de ces faces. Les calculs d'éclairage réel avec des méthodes de type radiativité (calcul d'éclairage diffus uniquement) ne sont pas encore adaptés au temps réel.

Si cette solution s'est imposée comme référence en matière de rendu câblé, elle ne devrait pas manquer d'évoluer pour contourner les défauts du rendu projectif associé à un éclairage de type Gouraud, à savoir en particulier un manque de réalisme dans les images. Ce mode de rendu ne permet pas en effet de gérer (directement) les ombres portées et les reflets entre objets.

L'autre type de solution qui calcule l'éclairage dans son ensemble, est actuellement trop lent pour que l'on puisse envisager des calculs en temps réel : le lancer de rayon permet le calcul des éclairages entre les sources de lumière et les objets, par rapport à la position de l'observateur, et entre les objets eux-mêmes. Associé à une formule d'éclairage adaptée, le lancer de rayon permet d'obtenir un degré de réalisme important, au prix d'un temps de calcul tout aussi important.

Les solutions matérielles de rendu projectif se sont démocratisées assez rapidement dès l'apparition des premières cartes graphiques possédant quelques fonctionnalités 3D. Elles ont permis à moindre coût de fabriquer des consoles de jeux qui rivalisent de plus en plus avec leurs soeurs aînées des salles d'arcade. On leur doit ainsi l'essor de la réalité virtuelle, des jeux sur micro-ordinateur et en particulier d'applications entièrement 3D où l'utilisateur se retrouve face à une approche différente des interfaces de travail 2D de type WIMP¹. Souvent plus proche des situations de travail réelles (puisqu'en 3D), elles permettent d'introduire facilement de nouvelles formes de périphériques, plus adaptés à la troisième dimension, mais qui posent de nouveaux problèmes d'IHM.

1.3.2 Les algorithmes

Nous détaillons ici l'ensemble des algorithmes que nous avons utilisé pour créer le rendu temps réel de notre interface 3D. Nous les avons utilisé à travers la bibliothèque OpenGL (voir partie suivante) pour ceux directement câblés sur les cartes graphiques, ou nous les avons développé sur la base de cette bibliothèque.

Le rendu projectif

Techniquement :

Les coordonnées de tous les points de chaque objet sont calculées en fonction des transformations qui lui sont appliquées (translation, rotation et changement d'échelle), afin de l'amener à sa position finale pour le calcul d'une image. Effectuées en coordonnées homogènes, ces opérations donnent la position finale de l'objet dans la scène. On détermine ensuite si les objets sont visibles grâce au cône de vision de l'observateur (Figure 1.7), défini par une direction, un angle de vue vertical et horizontal. Deux plans parallèles permettent de fermer l'espace de visualisation. Si un objet est en dehors, aucun calcul n'est fait pour le visualiser (on parle de *clipping*).

Chaque objet est ensuite projeté sur le plan de l'écran (situé Figure 1.7 en Z minimum), ce qui permet de convertir les données géométriques des objets en valeur de couleur pour chaque pixel qu'ils contiennent (on parle de *rasterization*).

Les calculs géométriques sont réalisés en général par le micro-processeur central de la machine, étant donné que les cartes gérant cet aspect (transformations géométriques, *clipping*, etc) sont deux fois plus chères que les autres et sujettes à l'augmentation de puissance des ordinateurs (quand le processeur devient plus performant que la partie «géométrie» de la carte graphique).

En modifiant les valeurs des angles du cône de vision, on accentue ou on diminue le phénomène de perspective sur une distance équivalente (définie par les deux plans Z minimum et Z maximum Figure 1.7).

1. WIMP : Windows, Icons, Menus, Pointers, ou les interfaces 2D classiques type MS Windows ou X Window.

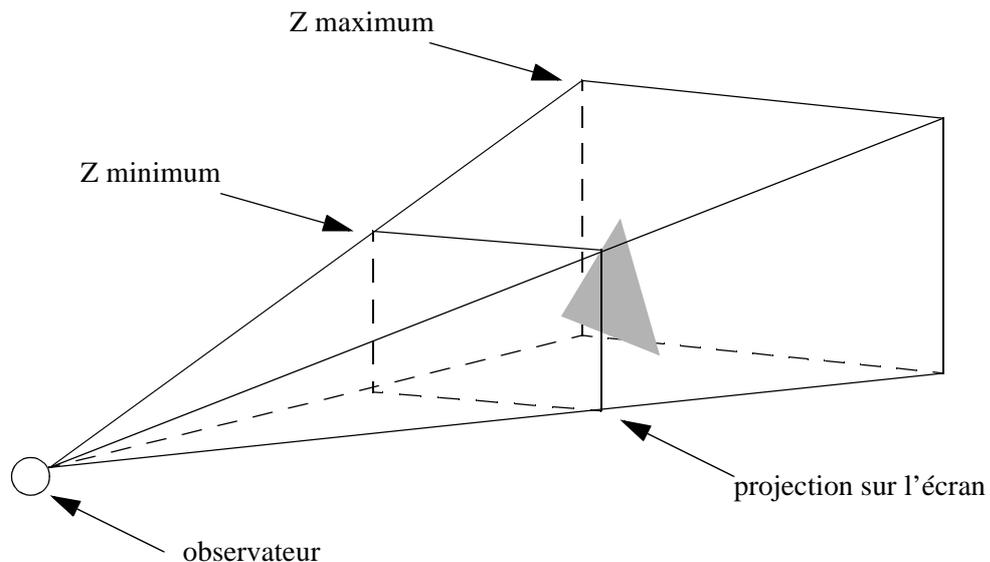


Figure 1.7 *cône de vision et projection en perspective des facettes sur le plan de l'écran*

Utilisation :

On utilise beaucoup les transformations géométriques pour déformer les objets (changement d'échelle non-uniforme suivant les trois axes) et créer des effets de *fish-eye* [Furnas 86], qui ne correspondent pas à une représentation réelle du monde mais à une vue optimisée pour afficher des volumes de données importants.

La perspective est également un paramètre sur lequel on joue beaucoup en virtuel pour augmenter l'effet de profondeur sur les objets (le rétrécissement en fonction de l'éloignement). Cela a pour effet d'introduire la perspective comme facteur d'amélioration de la perception de la profondeur, dans une scène virtuelle ayant une profondeur de champ faible.

Le calcul de perspective permet également de déterminer la taille à laquelle est vu un objet, en fonction de la taille de l'écran et de la distance à l'écran de l'utilisateur. Cette information est utilisée lors d'évaluations des indices perceptifs.

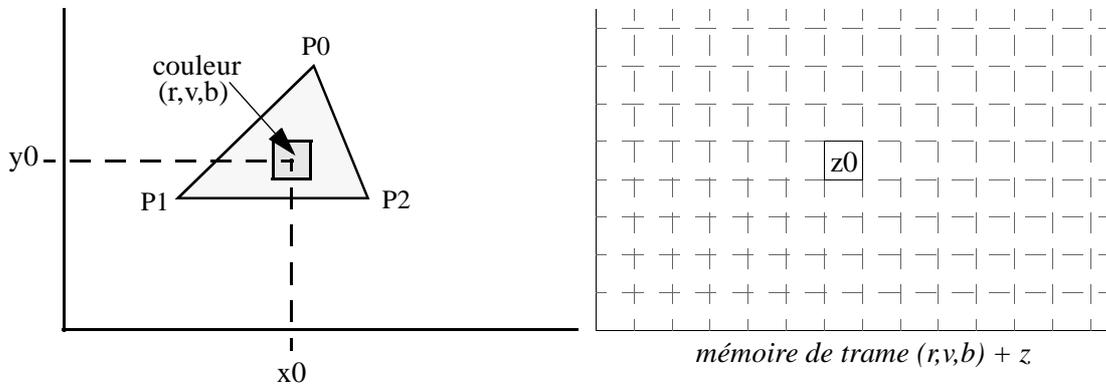
Le calcul des parties cachées

Techniquement :

L'utilisation de l'algorithme du Z-buffer s'est généralisée dans les cartes graphiques.

Pour chaque pixel de l'écran, on crée une zone mémoire où stocker une valeur de profondeur (initialisée à la valeur de profondeur maximum). L'algorithme consiste à convertir chaque polygone, ligne à ligne, en pixels. La valeur de couleur d'un pixel est stockée en mémoire de trame si sa valeur de profondeur calculée en ce point du polygone est inférieure à la valeur déjà stockée (Figure 1.8), i.e. si le point du polygone est plus proche de l'observateur que celui déjà stocké.

La facette est parcourue à partir de son sommet le plus haut (P0) vers les deux autres sommets (P1 et P2) suivant une double interpolation (parcours avec l'algorithme de Bresenham). Une façon de faciliter ce parcours est de décomposer les facettes de façon à avoir une base horizontale.



si $z(x_0, y_0) < z_0$ alors stocker $(r, v, b, z(x_0, y_0))$

Figure 1.8 le Z-buffer : pour chaque pixel contenu dans la facette, on teste la valeur du Z

La simplicité et la généralité de l'algorithme le rendent facile à implémenter sous forme matérielle.

La précision du calcul (valeur de Z sur 16 à 32 en fonction des cartes graphiques) revêt une importance pour éviter le crénelage dû à l'arrondi des valeurs de Z.

Utilisation :

Cet algorithme est indispensable dans un système de représentation comme le notre, où l'ensemble de l'espace de travail est une vraie représentation 3D.

L'ombrage : Gouraud et Phong

Techniquement :

Le calcul d'éclairage d'OpenGL utilise l'interpolation de Gouraud : pour chaque facette, la couleur du pixel de chaque sommet est calculée à partir des paramètres du matériau de la facette et des sources lumineuses. Les 3 valeurs (R,V,B) obtenues aux sommets sont interpolées pour calculer la couleur de chaque pixel de la facette. La formule du calcul d'éclairage au sommet est celle de Phong (prise en compte de la lumière spéculaire).

Utilisation :

Le placement des sources lumineuses a une grande importance. Tout d'abord un éclairage par le haut, comme dans la réalité, est souhaitable pour que l'utilisateur retrouve ses repères. Ensuite, la variation de luminosité en fonction de la profondeur est un facteur perceptif pour positionner les objets entre eux (distances relatives). On peut donc placer des sources lumineuses pour qu'il y ait des effets visibles de variations de lumière sur les objets. Il suffit de positionner les sources en hauteur au dessus de la scène et de définir un facteur d'atténuation pour chacune d'entre elles.

Les textures

Techniquement :

On utilise du placage de texture *bitmap* (image en mode point). Plaquer une texture sur une facette consiste à attribuer à chaque valeur de pixel occupé par la facette, une couleur de texel de la texture (i.e. pixel de l'image à appliquer), en fonction des coordonnées de texture (Figure 1.9). On peut faire des moyennes de couleur pour approcher la couleur au plus juste, on parle alors de filtrage bi-linéaire ou tri-linéaire.

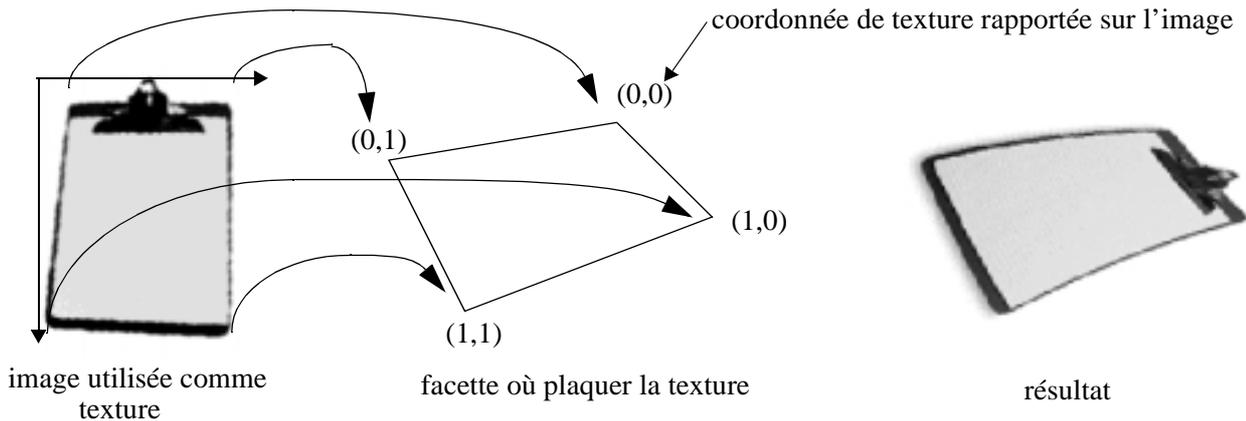


Figure 1.9 *principe du placage de texture*

Cette fonction, qui consiste à faire des correspondances entre une facette en trois dimensions et une image de points, coûte cher en terme de calculs. C'est pourtant la condition nécessaire à une certaine qualité d'images en rendu projectif.

Utilisation :

Le placage de texture est indispensable. Cela a été le facteur limitant d'une interface 3D temps réel tant que les cartes graphiques n'ont pas su en gérer une quantité importante dans les scènes 3D, sans perte de temps. C'est un calcul exigeant.

Il est utile à l'affichage de documents 2D définis sous forme *bitmap* (d'image sous forme de points), à l'amélioration des effets de rendu d'objets 3D (son utilisation d'origine). Il sert aussi à réaliser des humains virtuels réalistes (des clones), en plaquant des textures de visages sur des modèles 3D, indispensable au réalisme et donc à la reconnaissance des interlocuteurs d'une application multi-utilisateur comme la notre.

Enfin, les animations vidéo sont en fait vues comme des textures animées dans la composition de scènes 3D. C'est donc encore cette même fonction de placage de texture qui peut nous servir à insérer des documents vidéo. Cette opération est très coûteuse en temps de calculs, et seulement certaines cartes graphiques à l'architecture particulière réussissent cette opération sans ralentir dramatiquement l'ordinateur. Cela est en particulier possible lorsque la partie de la carte graphique qui réalise le placage de texture peut accéder directement et rapidement à l'image vidéo contenue en

mémoire (comme dans le cas d'une mémoire unifiée entre l'image vidéo et le *frame buffer*, à laquelle la composante placage de texture accède en général).

Les transparences

Techniquement :

L'alpha est défini comme le taux de recouvrement d'une surface dans un pixel. Il sert principalement à l'anti-aliasage et à l'affichage d'objets transparents, en indiquant le pourcentage d'opacité de l'objet (valeur comprise usuellement entre 0 et 1). On compose la couleur de l'objet transparent avec celle de l'objet derrière lui grâce à ce paramètre :

$$C_{\text{pixel}} = \text{Alpha} \times C_{\text{objet_transparent}} + (1 - \text{Alpha}) \times C_{\text{objet_derrière}}$$

Si on utilise des transparences, l'ordre de placement des objets a une importance. En effet, les objets transparents ne doivent pas influencer la valeur contenue dans le Z-buffer s'ils sont visibles, mais seulement modifier la couleur de l'objet derrière eux. Il convient donc de les tracer en dernier, ce qui impose un pré-traitement pour les trier.

Utilisation :

La transparence permet de conserver une certaine lisibilité derrière l'objet rendu transparent. Elle permet également de rendre un objet transparent au fur et à mesure qu'un autre objet le traverse, afin de conserver ce dernier visible.

Les ombres projetées

Techniquement :

Il s'agit d'un algorithme qui n'est pas câblé dans les pipelines de rendu projectif, parce qu'il n'est pas générique (algorithme avec cas particulier) et trop compliqué. Il existe plusieurs possibilités.

La première méthode, les ombres symboliques, consiste à projeter sur le sol un polygone simple de couleur noire, transparent, en forme de rond ou de carré qui fait office d'ombre pour un objet (et donc qui se déplace avec). C'est la forme d'ombre la plus simple, celle qui nécessite le moins de calculs supplémentaires.

La seconde méthode consiste à projeter au sol un objet ayant la forme de l'objet dont on veut l'ombre. En l'affichant en noir, et en transparence, on obtient une forme d'ombre réaliste. Une solution simple consiste à projeter toutes les facettes de l'objet sur le sol en les changeant en noir. Cette solution nécessite peu de pré-calcul (une matrice de projection sur un plan), mais oblige à redessiner complètement l'objet lorsqu'on le projette (donc deux fois par image). On peut sinon pré-calculer un profil de l'objet à utiliser comme ombre.

Ces deux premières méthodes ne permettent que de projeter une ombre sur un sol plat. S'il y a des reliefs, ou si un objet passe sous un autre, on ne verra pas d'ombre apparaître. Il faut dans ce cas utiliser une méthode comme les cônes d'ombres, qui permet de connaître sur quels objets l'ombre d'une facette doit être affichée. L'implémentation de cet algorithme en OpenGL nécessite de tracer deux fois la scène dans son ensemble, plus les facettes du cône d'ombre, en utilisant le *Stencil*

Buffer [Woo 98]. Elle est donc très coûteuse, et nécessite d'avoir une carte graphique possédant un *Stencil Buffer*. Il peut donc être utile de manière générale de bien mesurer le besoin en terme d'ombres afin de limiter les calculs dûs aux ombres projetées.

Utilisation :

Elle donne une information sur la position des objets en profondeur, sur leur direction lors des animations. C'est un indice perceptif indispensable. Nous avons mené une étude pour mesurer son importance et tester l'algorithme suffisant pour donner toute sa valeur à l'information sans surcharger la machine par des calculs d'ombre trop précis (voir chapitre 4).

1.3.3 Les moyens de développement informatique

Les solutions de développement sur ces machines «graphiques» se déclinent le long d'une échelle semblable à celle menant du langage naturel au langage machine :

Tout d'abord les solutions de haut-niveau où le «programmeur» compose des scènes 3D dans un éditeur graphique dédié en rendant certains éléments dynamiques (moteur d'animation, de réaction au clic, etc), comme dans *World-Up* de Sense8. Il s'agit en fait de scènes actives, visualisées avec un programme particulier. On obtient à peu près le même résultat avec un fichier VRML 97 et des scripts *JavaScript*, visualisés avec *CosmoWorlds* de Silicon Graphics. Un outil comme *Alice* permet quant à lui d'animer des objets, non plus seulement via une interface graphique ou des scripts, mais également grâce à un langage naturel de commande, du type «lapin va vers carotte» ou «chasseur suit lapin». Ces systèmes ont tous cependant le défaut de n'être peu ou pas extensibles, on ne peut y ajouter facilement des fonctionnalités. Il faut donc être sûr de ne pas déborder du cadre de leurs possibilités pour l'application voulue. Ils sont donc peu destinés à concevoir de «nouvelles» formes d'interface, mais sont en revanche très utiles pour le prototypage rapide d'un environnement 3D interactif.

On trouve ensuite les boîtes à outils comprenant un format de description de scène particulier, possédant des méthodes pour manipuler ou modifier directement les objets de la scène, et utilisant des outils complets d'interaction et de manipulation de la scène. Les plus connus sont *Open Inventor* de Silicon Graphics, *World Toolkit* de Sense8, et plus récemment *Java3D* de Sun. Ils sont composés d'une librairie très vaste de fonctionnalités complexes, qui masquent les manipulations de données bas niveau à travers le formalisme de leur structure. Ils permettent plus de choses que les outils comme *World-Up*, et sont bien sûr intégrables dans toute application. Cependant, certaines sont extensibles mais il n'est pas aisé de sortir du cadre de leur structure ou de proposer de nouvelles fonctionnalités déjà définies autrement (en matière d'interaction par exemple, il faudra ré-utiliser la structure objet des widgets déjà présents pour en créer d'autres).

Une autre approche consiste à utiliser une visionneuse de fichier VRML 97 comme *CosmoPlayer* (on n'a donc pas à s'occuper de l'affichage). On utilise alors un langage de programmation pour manipuler et modifier le graphe VRML contenu dans l'application. Ceci est possible grâce aux EAI (External Authoring Interface). Il n'existe cependant qu'une EAI, celle pour le langage Java [EAI]. Si cette méthode est très pratique pour manipuler des graphes VRML, elle limite l'interaction à celle

de VRML 97, à travers des widgets, et à la souris. Si d'autres primitives sont nécessaires, il convient de les programmer complètement à l'extérieur de la visionneuse VRML, comme par exemple la détection entre objets qui nécessite la création d'un autre graphe de position d'objets. Cette solution n'est donc pas idéale pour programmer plus que ce que VRML 97 propose, en particulier pour un projet nécessitant de faire beaucoup d'interaction 3D.

Des entreprises développent actuellement des formats de scène 3D interactives, formats propriétaires qui mettent en valeur une technologie particulière, comme *SuperScape* de SVR ou le langage *Chrome* de Microsoft.

La dernière approche est celle des bibliothèques de programmation bas-niveau. Il s'agit d'un ensemble de fonctions graphiques 3D, qui sépare l'affichage de l'interaction avec l'utilisateur et de la gestion du système d'exploitation (et des fenêtres par exemple). Elles proposent des fonctions élémentaires. Si l'approche de ces bibliothèques est plus simple en général que celle de boîtes à outils, elle demande plus de code pour arriver à des résultats équivalents. L'avantage est une plus grande souplesse et des limitations d'ordre uniquement techniques (liées à la machine), et non plus à la bibliothèque elle-même. Les plus répandus en matière de 3D sont OpenGL et Direct3D (bibliothèque propriétaire spécifique à Windows 9x), ou encore PEX, une extension du protocole X développée par le consortium X et basée sur PHIGS (un des premiers standard de librairie 3D).

Lorsqu'il s'agit d'utiliser trois dimensions, les formats de description d'objets des interfaces 2D ne sont plus adaptés. Ces formats élémentaires sont les ensembles de pixels (*bitmap*) et la description sous forme vectorielle (pour les jeux de caractères par exemple). Pour les documents 3D, il s'agit d'utiliser une description de scène complète. Chaque application utilisant ces fichiers possède son propre format, il n'y a pas de norme. Cependant, Internet a permis d'établir un langage de modélisation commun à tous : VRML. Si la première version était principalement un format de description géométrique structuré d'une scène, tiré du format d'*Open Inventor*, il en va autrement de la seconde qui propose des méthodes d'interactions avec les objets, de manipulation et d'animation par différents moyens des composants de la scène. Si ce format ne permet pas encore de décrire des scènes interactives complètes (ce sont des objets spéciaux qui sont activables qui propagent les événements dans la scène), il permet de créer des scènes animées en trois dimensions. Les prochaines versions devraient venir étoffer la description d'objets pour intégrer de plus en plus d'aspects utilisateur et multi-utilisateur.

1.4 Les périphériques de sortie et d'entrée

Notre étude porte ici sur deux points : les dispositifs de visualisation (les dispositifs de sortie), et sur les périphériques gestuels (périphériques d'entrée).

1.4.1 Les modes de visualisation

La technologie d'écran la plus répandue est celle des tubes CRT qui équipent nos moniteurs d'ordinateurs et nos télévisions : les tubes cathodiques. On voit cependant apparaître des systèmes de visualisation plus spécialisés. Les systèmes standards, les moniteurs, voient leur taille augmenter

au delà des 25 pouces (1 pouce mesure 2,54 centimètres), leur affichage passe du tube CRT à l'écran à cristaux liquides ou à plasma, leur format passe du 4/3 à un rapport du type 16/9^{ème}, mais l'image reste ici dans le cadre plat délimité par l'écran. Cela peut présenter des avantages de conserver une vue sur l'environnement immédiat de l'écran (bureau, etc) pour des besoins liés à l'application (conserver le contact avec son cadre de travail réel), ou pour des besoins liés à l'utilisateur (pour conserver la perception du monde réel). Il existe cependant des applications où ces systèmes d'affichage ne sont pas adaptés.

La vision stéréo : elle permet de visualiser un objet en trois dimensions, et non plus à plat. L'écran affiche deux images spatialement désynchronisées pour recréer une légère disparité entre les deux yeux afin que le cerveau reconstruise l'image en perspective. Les images sont affichées successivement pour chaque oeil sur un écran (CRT ou autre). Un système à base de lunettes permet de masquer l'image pour l'oeil gauche à l'oeil droit, et inversement (à base de filtre couleur ou d'obturateur à cristaux liquides). Cette technique permet d'afficher des objets «en relief» pour l'utilisateur, dont il a une perception tri-dimensionnelle complète (avec la profondeur). Ce système a un léger inconvénient technique, il faut générer deux fois plus d'images qu'en monovision.

La réalité augmentée : l'objectif est de surimposer des informations dans le champ de vision de l'utilisateur. Elle peut prendre deux formes :

Il peut s'agir pour l'utilisateur de porter un masque semi-transparent, où vont s'afficher des informations. Elles lui apparaîtront devant son champ de vision réel. Ces informations virtuelles, surimposées sur des éléments réels, apportent des renseignements sur les objets regardés et les modifient virtuellement.

L'autre solution consiste à projeter une image sur des éléments existants pour apporter un complément d'informations sur ces derniers. En projetant, par exemple, des informations supplémentaires sur une table à dessin pour compléter le plan posé dessus. Cette solution est moins nomade que la précédente.

Dans tout les cas, ce type de système est bien sûr fondé sur une analyse par l'ordinateur du monde environnant l'utilisateur au moyen d'un système de vision (de type caméra vidéo par exemple).

L'immersion est une forme particulière de vision par ordinateur, elle permet une vision complète sur les images générées, qui occupent quasiment tout le champ de vision. Elle peut se décliner sous plusieurs formes.

Les casques : ils possèdent un écran et un système optique pour calibrer la vue dessus. Certains modèles possèdent deux écrans, un pour chaque oeil (la vision stéréo est donc possible facilement). Un capteur mesurant les mouvements de la tête peut permettre de changer le point de vue sur l'environnement virtuel.

Les Caves : il s'agit d'une pièce équipée de plusieurs écrans afin d'offrir une vision panoramique complète. Un ou plusieurs utilisateurs se placent au centre de la pièce. C'est un système de projection à grande échelle, qui peut faire la taille d'une petite salle de cinéma (voir les «visionariums» de Silicon Graphics). Ce type de système permet d'immerger un utilisateur dans un

environnement à son échelle, où il peut interagir. Dans le cas d'un groupe, ce type de système permet en particulier de présenter des objets en virtuel à l'échelle un.

Les hologrammes virtuels : il faut remarquer les résultats de systèmes holographiques comme le *MIT holovideo display* [Watlington 95] qui, s'il affiche encore une image holographique de petite taille avec des moyens matériels considérables, n'en reste pas moins le seul dispositif d'affichage sans support de projection. Ces travaux présagent des futurs systèmes d'affichage 3D.

Les Workbenchs : ce sont des tables où une image est projetée par un système vidéo [Obeysekare 96]. L'utilisateur est devant cette «table» et agit directement dessus. Le système peut exister en deux dimensions, l'utilisateur interagit en général alors avec un stylet sur une tablette graphique transparente posée sur l'écran. Si le système est en 3D, il est accompagné de lunette de vision stéréo pour l'utilisateur, dont les mouvements sont mesurés pour déplacer d'autant l'image visualisée. Ce système permet une interaction sur des objets virtuels très proches et sur une grande surface de travail, au moyen de capteur 3D placés sur les mains de l'utilisateur. Il existe des systèmes à deux utilisateurs.

Bilan : tous ces systèmes permettent un travail seul ou en coopération avec d'autres personnes. Il s'agit souvent de matériel occupant beaucoup de place et/ou difficile à mettre en oeuvre (pour les Caves ou les Workbenchs en particulier). On les utilise pour des applications particulières, adaptées à leur mode de visualisation et aux conditions que ces environnements de travail imposent.

1.4.2 Les périphériques gestuels

Un des principaux problèmes de l'interaction homme-machine vient des moyens de communication mis à disposition de l'utilisateur pour interagir avec l'ordinateur. Pour ce qui est de l'interaction directe, la souris s'est imposée comme le périphérique d'entrée indispensable. Elle permet, comme les autres périphériques à deux degrés de liberté (trackball, glidepoint) de déplacer un pointeur sur un bureau virtuel dans deux dimensions. Les utilisateurs d'ordinateur devenant vite des experts du maniement de la souris, il pourrait être souhaitable de profiter de cette maîtrise pour les faire agir en trois dimensions avec ce dispositif. Les visualiseurs de fichier VRML, comme *CosmoWorld* qui propose une interface de pilotage pour naviguer en trois dimensions, s'utilisent à la souris. D'ailleurs, à l'heure actuelle, la souris est la seule solution dont disposent les développeurs de programmes 3D destinés au public pour interagir. L'utilisation des joysticks semble être réservée aux jeux. Certains périphériques plus originaux voient le jour pour le grand public comme la *SpaceOrb 360* de Spacetec (et chez ASCII Entertainment), version allégée et économique de leur série de trackball 3D Spaceball, ou le capteur à ultrason *FreeD* (anciennement *Owl*) de Pegasus. Mais leur diffusion est encore limitée.



Figure 1.10 en haut : **capteur 3D** (FreeD de Pegasus), **trackball 3D** (Spaceball 3003)
en bas : **manette de jeux à 6 degrés de liberté** (Spaceorb), **trackball 2D classique**

La manipulation d'objets 3D à la souris a donc naturellement été étudiée (voir [Chen 88] et [Emmerik 90]), et beaucoup de solutions ont été développées dans des applications manipulant des objets 3D (Figure 1.11). La manipulation se fait en général à travers des widgets apparaissant sur l'objet, comme les trackball virtuels. Certains produits se démarquent, comme *Bryce* de MetaTools, dont le design ergonomique permet une manipulation rapide des scènes 3D.

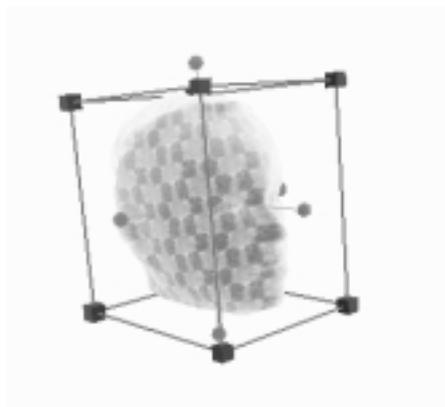


Figure 1.11 un **widget de translation** du visualiseur d'Open Inventor utilisant la souris

Chaque problème possède son périphérique adapté, et on peut voir des solutions très spécialisées apparaître : par exemple, les simulateurs de vol, où l'interaction se fait par le moyen d'une

reproduction des commandes d'un avion (manche à balai et palonnier). L'étude d'un système d'interaction adapté à une application informatique doit porter en parallèle sur le mode et sur le moyen d'interaction.

Nous allons étudier en particulier les périphériques d'entrée 3D [Fuchs 96], qui permettent l'acquisition de données en trois dimensions (les classiques souris ne fonctionnant que suivant deux axes). Ils se présentent sous la forme de trackballs 3D ou de capteurs que l'on fixe au doigt et que l'on déplace en trois dimensions. Ils renvoient à l'ordinateur des indications quant à la position du périphérique : on parle de degrés de liberté. Si le périphérique gère les déplacements suivant 3 axes (les translations suivant X, Y et Z) alors il a trois degrés de liberté; si en plus, il mesure les rotations autour de ces axes, il possède six degrés de liberté (X, Y, Z et l'angle de rotation du périphérique autour de chacun de ces axes). Il peut en posséder encore plus en fonction du type d'information recherché : de la simple position de la main avec un capteur 3D, à son mouvement complet, doigts compris, au moyen d'un gant de données.

On peut les classer suivant trois types :

Les périphériques isométriques

Ils possèdent une résistance infinie, ils sont donc immobiles. Ils mesurent le mouvement à partir de la force et du couple qu'on leur applique. Ils ont pour défaut le manque de rapport direct avec le mouvement; en effet, c'est la force appliquée qui détermine le mouvement et non pas le déplacement, quasi nul, de la main. Le manque de retour (donc de proprioception) est un autre défaut, d'où un certain temps d'adaptation pour les tâches complexes. Les trackballs 3D en sont un exemple (Figure 1.10).

Les périphériques isotoniques

Ils bougent librement pendant le mouvement de l'utilisateur, ils possèdent donc une résistance nulle. C'est le cas des souris en 2D, ou des gants de données. Ils ont pour défaut la fatigue éventuelle due à leur utilisation prolongée, et pour certains types d'applications les limites de leur champ d'action. Cependant ils permettent souvent d'avoir six degrés de liberté. Les capteurs à ultrasons en sont un exemple (Figure 1.10).

Les périphériques élastiques

Ils sont à mi-chemin entre les deux derniers types. On doit appliquer une force qui varie en fonction du déplacement, et ils sont autocentrés dans leur position stable. Ils sont réputés plus proprioceptifs pour l'utilisateur et donc plus faciles à utiliser.

Il existe des controverses sur le périphérique qui permet d'obtenir les meilleures performances [Zhai 94]. Les isométriques fonctionnent mieux dans les modes de contrôle de taux (robotique), alors que les isotoniques sont plus adaptés aux modes de contrôle de position, lorsqu'il y a un rapport direct entre le mouvement de la main et le mouvement d'un pointeur.

Les périphériques haptiques

Certains périphériques, quelque soit leur type (isométrique, etc), permettent de réaliser un retour de toucher, en particulier une sensation de contact sur les doigts. On dit alors qu'ils sont à retour de toucher, ou à retour d'effort.

Le retour d'effort consiste à produire une résistance au mouvement qu'un utilisateur imprègne à un périphérique d'entrée. Cette résistance est pilotée par l'application, dont elle dépend, ainsi que des moyens utilisés. Le retour d'effort étant produit par des moteurs, les dispositifs sont souvent lourds et encombrants, a fortiori si l'effort à retourner est important.



Figure 1.12 le gant à retour tactile *Cybertouch* de VTI

Pour améliorer la perception de l'utilisateur, une simple information de toucher peut donc être très utile au moment du contact avec un objet virtuel; voire une sensation sur les doigts correspondant à la texture du matériau touché. Cette information peut être simulée en utilisant un système à retour d'effort (où les secousses et les vibrations seront autant d'informations à interpréter), mais l'idéal est encore un dispositif donnant directement la sensation de contact et de toucher à l'extrémité des doigts, comme le *Cybertouch* (Figure 1.12) de VTI (Virtual Technologies Inc) qui dispose de stimulateurs vibrotactiles sous chaque doigt du gant.

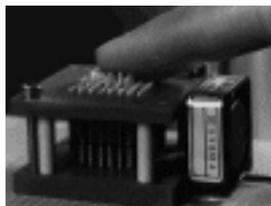


Figure 1.13 le «*haptical Actuator Array*» du Dr. Harald Fischer, du HIT de Karlsruhe (All.)

Ces stimulateurs sont programmables de manière à simuler de simples contacts (continus ou par impulsion) ou des surfaces complexes (Figure 1.13).

1.5 Aspect multi-utilisateur

L'introduction des réseaux dans les entreprises, les besoins grandissant en matière de communication ont permis à des outils comme la messagerie électronique de se développer largement. Il s'agit d'un outil de communication asynchrone, qui permet une forme de collaboration.

D'après le Petit Larousse, collaborer, c'est travailler avec d'autres, et coopérer, c'est agir conjointement avec quelqu'un. La coopération implique une proximité plus grande dans la tâche à accomplir, on agit en même temps, sur les mêmes objets. La collaboration a un sens plus large. Les deux termes sont confondus en général lorsque l'on parle de TCAO. Cependant, nous nous intéressons bien ici à des problèmes de coopération.

L'introduction du courrier électronique ne modifie pas en général les méthodes de travail. Elle vient s'ajouter aux outils de communication, et ajoute des tâches supplémentaires. Toutefois on devrait voir s'intégrer à l'avenir des outils de collaboration synchrone dans les entreprises qui créeront de nouvelles habitudes, en réponse à de nouveaux besoins : le rapprochement de bureaux d'études distants, les réunions à distance, etc. Ainsi les applications pouvant permettre un gain de temps, d'espace ou encore la diffusion d'information à large échelle, plus facilement que par des moyens traditionnels (pour le télé-enseignement, par exemple) vont prendre place dans les prochaines années. Le développement des technologies de TCAO synchrone devrait également faciliter la diffusion de nouvelles habitudes de travail comme la vente, le service après-vente ou le travail à domicile, même si pour ce dernier, cela implique surtout des changements dans l'organisation et la structure de l'entreprise.

Un système de TCAO synchrone suppose la connection en temps réel des différents utilisateurs, afin qu'ils puissent travailler ensemble sur des documents communs, voire simultanément grâce à des outils de coopération. Il existe différents niveaux de coopération. Actuellement, on trouve surtout des outils de communication (homme-machine-homme), possédant parfois des outils de travail comme des tableaux blanc ou des éditeurs de texte. Ces derniers sont en général à usage exclusif (système maître/esclave où un seul utilisateur peut travailler à la fois, avec un mécanisme de prise de main).

Il existe plusieurs catégories d'environnements de travail virtuels :



Les médiaspace : il s'agit d'environnements de travail médiatisés. Les interlocuteurs réunis ensemble ne sont plus confrontés l'un à l'autre directement, mais communiquent via un ou plusieurs médias (liaison vidéo, téléphone, réseau informatique, etc). Si on élargit la définition, les personnes ne sont pas forcément représentées dans ce médiaspace, et elles ne partagent pas forcément des documents communs. Il peut s'agir de systèmes très peu interactifs.

Les environnements virtuels : il s'agit de créer un environnement de travail nouveau, où les utilisateurs pourront se projeter pour avoir une activité commune. C'est l'informatique et les réseaux qui permettent de donner tout son intérêt à ce type d'environnement.

Les environnements virtuels coopératifs : la notion de CVE s'est imposée ces derniers temps, marquant une nuance de taille dans l'ensemble des environnements virtuels. Ils introduisent une dimension coopérative entre les utilisateurs. Ces derniers doivent avoir la sensation de présence des

autres utilisateurs. C'est l'élément de base. Il s'agit ensuite de leur permettre de communiquer puis de coopérer facilement en les renseignant sur l'activité des autres.

Les applications d'environnements médiatisés sont multiples, comme la conception collaborative dans l'industrie, l'architecture, les arts, les évolutions des MUDs et des MOOs vers les jeux et les environnements de chat 3D, les aspects éducatifs.

1.5.1 La vidéo

On peut trouver la vidéo sous plusieurs formes dans un environnement de travail collaboratif. Tout d'abord sous forme de document, traitant d'un problème donné. Elle peut alors prendre la forme d'un fichier, d'une liaison vidéo temps réel où un des intervenants filme une scène particulière. Cette option est très intéressante et permet de disposer d'un support d'information assez souple. Nous allons cependant discuter ici plutôt de l'aspect téléprésence. C'est en effet la deuxième utilisation que l'on peut faire de la vidéo, au moyen d'une caméra. Cette dernière permet de transmettre l'image des interlocuteurs en temps réel via un réseau.

Si les algorithmes de compressions ont énormément évolué dans le sens imposé par la visio-conférence (celui des débits très faibles), la vidéo souffre de défauts inhérents à sa condition. Il s'agit en effet de tenir en permanence un flux de données. Tout ralentissement crée une animation morcelée et des arrêts sur image qui tombent rarement à propos. Si ces deux premiers défauts tentent d'être compensés au maximum par des algorithmes extrêmement efficaces et adaptés au bas débit, la qualité de l'image des interlocuteurs, au sens propre et au sens figuré, s'en trouve affectée.

La solution de la vidéo, si elle est idéale (voire indispensable dans certaines situations) comme support d'information, n'est pas le meilleur moyen pour la téléprésence. Le fait que des utilisateurs d'outils comme *PictureTel* finissent par se passer de la vidéo faute d'intérêt, montre ses limites. Ce manque d'intérêt peut venir des inconvénients techniques cités ci-dessus, ou d'un manque d'intégration dans l'environnement de communication. Les flux vidéo sont en effet difficiles à manipuler, donc à modifier, et on se contente en général de les afficher dans une fenêtre isolée. Il n'y a pas alors de lien évident avec leur activité, obligeant l'utilisateur à inférer ce que son interlocuteur fait.

Certains collecticiels, construits autour de la vidéo, ne souffrent cependant pas de tous ces défauts [Kuzuoka 94]. Créés avec des technologies adaptées, ils permettent d'obtenir des outils de collaboration idéaux. Leur coût en terme de mise en oeuvre, de matériel et certaines limitations (deux utilisateurs en général, etc) empêchent de sortir ces développements du cadre de l'expérimentation. Ils ont permis cependant de mettre en évidence beaucoup de paramètres essentiels des applications de TCAO synchrone (voir chapitre suivant).

1.5.2 Les avatars

En anglais, un avatar désigne une manifestation visible d'un concept abstrait. Dans les environnements virtuels partagés, il s'agit donc d'un corps virtuel. On les retrouve en particulier dans les CVE (Collaborative Virtual Environments) dont l'objectif est de créer des environnements

multi-utilisateurs où s'établit une coopération, comme dans certains jeux (voir [Ultima On Line] Figure 1.5), des outils de communication 3D (voir [Worlds Chat] Figure 1.1), ou encore des outils de TCAO synchrone comme certains environnements virtuels partagés (voir [DIVE] Figure 1.17).

En Français, nous devrions donc plutôt parler de représentation ou de clone d'un utilisateur. Le terme «avatar» n'ayant pas le même sens, si ce n'est par rapport à la religion hindoue, où il désigne les différentes incarnations du dieu Visnu (le mot «avatar» vient du sanskrit).

Un clone est donc également une représentation d'utilisateur. Dans le cadre des mondes virtuels, on étend sa définition à n'importe laquelle des représentations que peut avoir un utilisateur. Et on parle de clone réaliste ou clone anthropomorphique, lorsque la représentation d'un utilisateur renvoie son image réelle.

1.5.3 Les techniques de clonage réaliste

Un clone possède un corps, des bras et une tête. Le processus de création d'un clone se fait à partir de prises de vue et de mesures de l'utilisateur afin de créer un modèle 3D du visage, une texture du visage à appliquer dessus, et un modèle 3D de corps aux mensurations du clone. Ce processus peut prendre différentes formes, il est plus ou moins automatique et plus ou moins long en fonction des méthodes employées, mais aboutit toujours au même résultat, commun à quasiment tous les systèmes de détection/animation d'un clone : un maillage 3D du visage, du corps et une texture bitmap du visage.

Le corps et les bras

On peut détecter simplement le mouvement du corps et des bras à partir de capteurs de position 3D (comme les Polhemus, voir Figure 1.14) qui donnent la position des extrémités du corps. A partir d'algorithmes comme ceux de cinématique inverse, on peut recalculer les positions de chaque partie du buste et des bras avec un nombre limité de capteurs (voir Figure 1.15).



Figure 1.14 *Système de capteurs de position 3D Polhemus*

On peut également animer les mains d'un clone pour qu'elles puissent saisir les objets vers lesquels elles se dirigent. L'opération étant coûteuse [Noser 96], on fait plutôt une approximation en n'animant pas ou peu les doigts de la main.

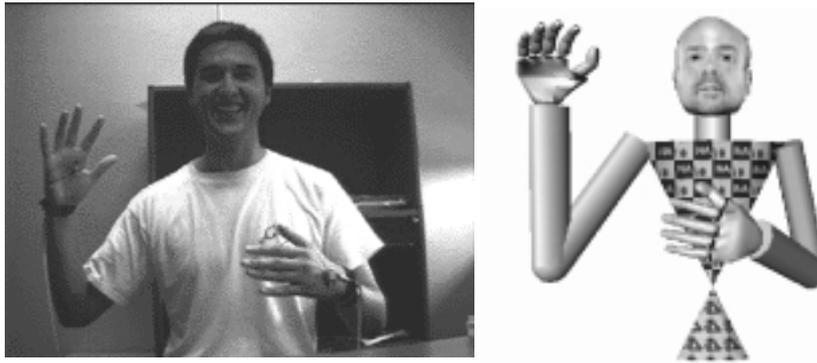


Figure 1.15 *Un utilisateur anime un clone au moyen de deux capteurs et d'un algorithme de cinématique inverse (système développé à l'Institut National de l'Audiovisuel)*

Le visage et la bouche

Le visage donne deux types d'informations essentielles. Tout d'abord les expressions faciales qui expriment un sentiment puis les mouvements des lèvres qui doivent suivre les paroles de la personne.

Nous nous intéressons au système d'animation de clones à distance en temps réel. Il existe un certain nombre de systèmes permettant d'animer un visage de façon très satisfaisante. Par exemple pour reproduire fidèlement et synchroniser les mouvements des lèvres d'une personne filmée sur de nouvelles paroles (donc de nouveaux mouvements de lèvres), il existe des algorithmes comme ceux d'Interval Research [Bregler 97] qui animent les lèvres d'un personnage à partir d'une série de phonèmes, une base de données ayant été préalablement générée pour tous les phonèmes possibles, le système suivant les mouvements de tête pour placer au mieux les lèvres sur le visage. Si ce système d'animation faciale est particulièrement efficace, comme d'autres, il n'est absolument pas temps réel, et ne peut pas intervenir dans un processus temps réel d'analyse des mouvements, puis de synthèse de clones.

Les données de base pour l'animation d'un clone sont soit le texte, soit le son, soit le son et la vidéo (une caméra de visioconférence filmant le visage).

Découpage vidéo

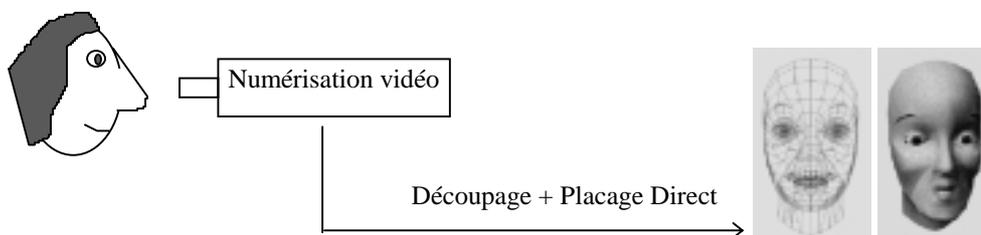


Figure 1.16 *animation d'un clone par placage directe*

Placage direct de la vidéo sur un polygone ou le maillage 3D du visage (Figure 1.16). C'est la solution la plus simple, mais elle nécessite le transport de la vidéo sur un réseau, ce qui est très

coûteux. L'effet obtenu peut manquer de réalisme. C'est un des systèmes utilisés pour les clones de *DIVE* (Figure 1.17).

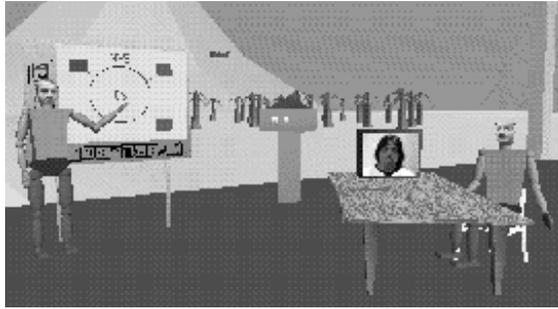


Figure 1.17 *une réunion dans DIVE*

Une solution alternative consiste à ne transmettre qu'une découpe du visage, celle des yeux et de la bouche pour animer le visage distant (Figure 1.18).



Figure 1.18 *le système de clonage de l'équipe VIS de l'IMAG, par découpage vidéo*

Analyse vidéo

Ce système permet, à partir d'une image vidéo, d'analyser les points clés du visage, et de calculer leurs déplacements au cours du temps tout en tenant compte des mouvements de la tête [Télévirtualité]. On transmet ces paramètres d'animation à distance (débit réseau très faible), on déforme le modèle 3D du visage avec ces points, et on obtient, après placage d'une texture représentant le visage, une animation du clone en temps réel, avec expressions faciales et mouvements de la bouche (voir Figure 1.19).

La limite de ce système est la nécessité de synchroniser le son avec l'image au moment de générer le clone 3D. Si le problème est connu en dessin animé par exemple, la synchronisation des lèvres et de la voix (*lips sync*) doit être réalisée ici automatiquement, et après passage sur un réseau qui introduit des temps de latence importants. A l'arrivée, les positions clés du visage doivent être interpolées et synchronisées avec le son. Si l'utilisateur parle vite, les mouvements de la bouche sont trop rapides pour permettre une analyse fine.

Ce système est cependant le seul à permettre de reproduire toutes les expressions faciales d'une personne, ainsi que les mouvements de tête (rotation et translation dans l'espace).

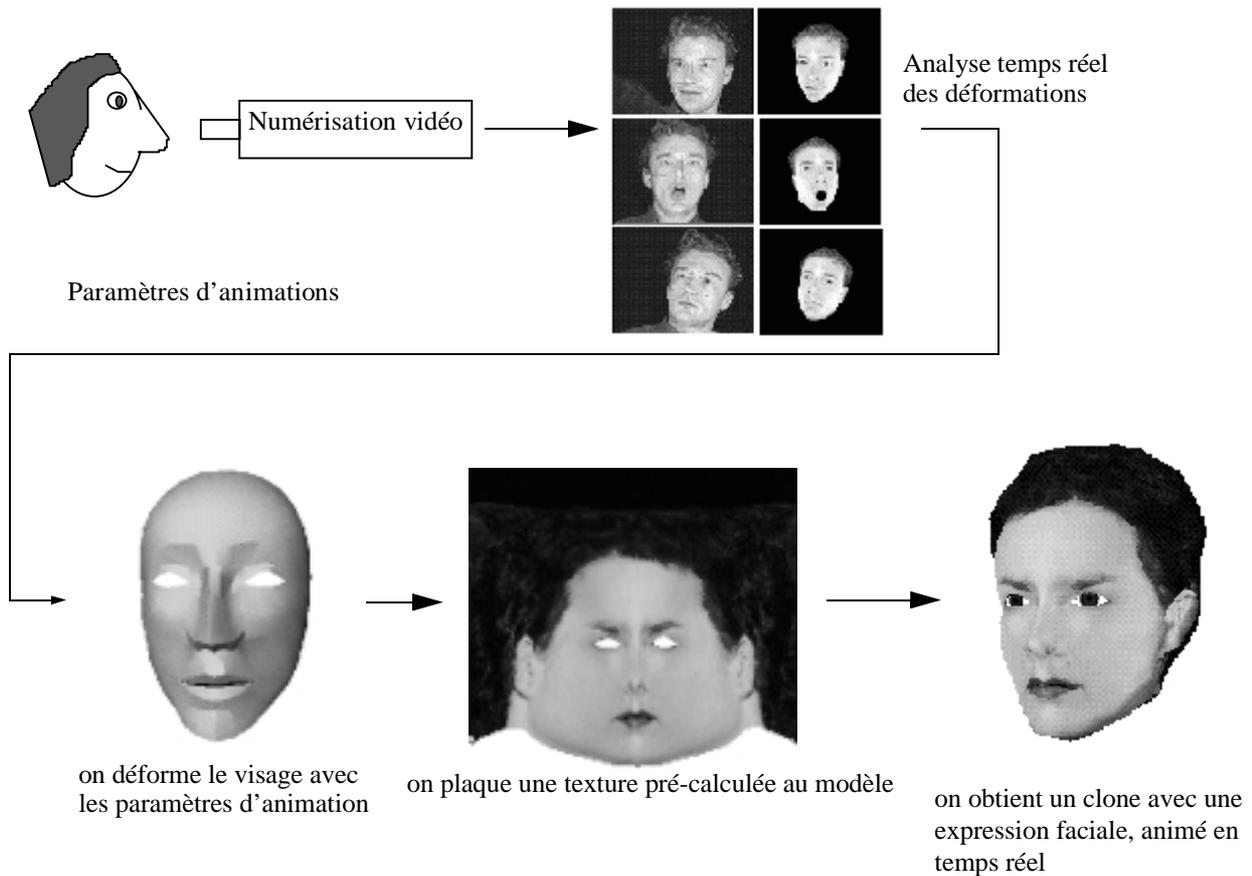


Figure 1.19 animation d'un clone par analyse synthèse

Il est également possible de déformer la texture plutôt que le modèle 3D. Ce système est moins souple et sert en général à modifier la position des yeux ou des lèvres.

Analyse directe du son

Le projet [Teleface] tente d'animer la bouche d'un clone uniquement à partir du flux sonore de la voix. Cela permettrait de ne transmettre que le son à distance. Si le problème de synchronisation texte/image se résout plus aisément, on perd toutes les informations d'expression faciale.

Le couplage d'un tel système avec le précédent devrait permettre de résoudre les problèmes d'animation de la bouche, tout en conservant les expressions faciales.

Analyse par Text-To-Speech

On génère automatiquement les positions de bouche à partir d'un découpage du son en phonèmes, puis on interpole les positions entre elles (voir le projet Multimodal Speech Synthesis [MSS]). On se sert de la hauteur et de la durée du son pour paramétrer l'animation. On obtient des résultats moins hachés en utilisant des diphtongues, unités composées d'une série de phonèmes auxquels on associe

directement une animation. Il en existe plus que de phonèmes mais ils permettent une animation plus précise.

La synchronisation son / image est automatique. Mais le problème des expressions faciales se pose encore. Comme le précédent, ce système permettrait s'il était couplé à l'analyse vidéo, d'avoir une animation fluide et complète du visage.

Ce système repose sur une connaissance préalable de ce que dit un interlocuteur sous la forme de son mais également sous forme de phonèmes. Si la traduction de phonème en son est aisée, elle l'est beaucoup moins dans l'autre sens. Ce qui limite actuellement l'utilisation d'un tel système pour de la communication à distance avec voix. On la réserve plutôt pour des assistants ou à des *chats* 3D [Virtual Friends].

1.5.4 Bilan

Les systèmes permettant de créer un avatar, à son image ou non, sont en pleine évolution. Drainés par le secteur des jeux, on peut voir apparaître certains systèmes pour les composer soi même. Les scanners 3D et les solutions logicielles se développent, les standards de définition d'humain virtuel également (dans MPEG 7 et un groupe de travail MPEG 4). En adaptant la qualité du rendu temps réel des cartes 3D actuelles aux contraintes de rendu des visages humains, avec les modèles adaptés que l'on possède déjà, on voit migrer les solutions de téléprésence de la vidéo vers des modes virtuels dont la qualité devra être bientôt équivalente, les problèmes techniques liés à la diffusion en moins.

Concevoir un modèle d'interface tridimensionnelle met en jeu plusieurs domaines. Nous venons de citer la synthèse d'images temps réel et le travail coopératif assisté par ordinateur, propres à la définition même du cadre des travaux. Mais certaines disciplines des sciences humaines interviennent également dans l'élaboration, comme l'ergonomie ou la sociologie, en ce qui concerne le comportement de l'utilisateur dans l'interface ou parmi un groupe d'utilisateurs. La synthèse de ces domaines permet d'avancer des concepts généraux sur la réalisation d'une interface 3D pour les activités coopératives.

L'évolution de la technologie, de la communication et de l'information est très importante, elle se matérialise à travers l'amélioration des moyens informatiques et le développement des moyens de communications. Face à cette rapide évolution technologique, les sciences humaines se sont intéressées tout d'abord au phénomène d'évolution de la technologie, puis à la technologie elle-même. Elles étudient son implication sur les activités de l'utilisateur et son impact sur leur organisation.

Ainsi, dans la conception des systèmes informatiques, outre les aspects techniques, on prend en compte des paramètres venant de la sociologie, de la psychologie, des sciences de l'éducation, de l'ergonomie, ou encore des sciences du langage, bien que ces dernières accompagnent l'informatique depuis ses débuts. Les psychologues étudiant le point de vue de l'utilisateur dans les interfaces homme-machine, ont évalué l'existant pour élaborer des règles de conception des interfaces. Issues d'une certaine expérience, ces règles d'ergonomie [Scapin 86] sont principalement dédiées aux interfaces 2D de type WIMP, les plus répandues.

Toutefois l'informaticien de nos jours ne tient que rarement compte de ces règles d'ergonomie dans la conception de ses programmes, en se basant sur son expérience d'expert et sur les bibliothèques de développement actuelles. L'ergonomie ne fait d'ailleurs que très rarement partie de la formation d'un programmeur, et on peut juste espérer que le cahier des charges de toute application soit rédigée par des gens sachant prendre en compte le point de vue de l'utilisateur. La réalisation de programme interactif en trois dimensions confronte plus que tout autre le concepteur à la perception-action.

Les bibliothèques de programmation proposant des systèmes interactifs 3D valables n'existent pas ou très peu, et on doit bien se rendre à l'évidence : les difficultés à percevoir la troisième dimension (simulée en images de synthèse sur un écran d'ordinateur) et les problèmes de localisation dus à la navigation se révèlent fortement dans les espaces infinis des mondes virtuels. Après la présentation des aspects techniques, nous allons donc discuter des problèmes d'utilisation posés par les interfaces 3D.

Nous allons tout d'abord discuter des problèmes de perception d'un utilisateur de système informatique, en particulier ceux liés à la vue. Puis nous présenterons les éléments essentiels du TCAO synchrone, afin de terminer sur les aspects principaux qui nous intéressent dans ce mémoire : ceux liés à l'interaction 3D.

2.1 Problèmes de perception

Un système informatique, aussi développé soit-il, n'utilise encore que des moyens limités pour échanger des informations avec l'homme en profitant de tous ses sens. Si les dispositifs d'entrée ont la possibilité d'utiliser la motricité de tous les membres de l'utilisateur, le cas de cet usage reste exceptionnel (en simulation par exemple), c'est en général le seul poignet qui est mis à contribution avec l'utilisation de la souris ou les deux mains pour le clavier. Pour le retour d'information de l'ordinateur, c'est la vue qui est mise à contribution avec les différentes sortes d'écran qu'un ordinateur peut gérer, et l'organisation de l'espace affiché. L'ouïe ensuite est le second sens le plus utilisé. Le retour d'efforts commence à se développer doucement en bureautique, dans les jeux grands publics et plus certainement et sérieusement dans les simulateurs. Toutefois, ce sens est peu utilisé de manière générale comme retour d'information; il reste uniquement un retour d'action pour beaucoup d'usages. L'odorat et le goût n'ont amené à l'heure actuelle que très peu d'expérimentations, probablement par manque de marché et également de dispositif permettant de reproduire le phénomène.

2.1.1 La vue

De l'écran cathodique aux vidéo-projecteurs équipant les *caves*, la vision est mise à contribution, permettant ainsi à l'utilisateur de déchiffrer, lire, comprendre ce qu'il est en train de faire. La vision en trois dimensions met en jeu des mécanismes d'interprétations particuliers, qu'il peut être intéressant de connaître en ce qui concerne la vision d'une scène réelle, afin de diminuer au maximum l'effort cognitif d'un utilisateur pour comprendre une scène virtuelle, en reproduisant ou en créant les bons indices perceptifs.

Aspect biologique

Le phénomène de vision peut être tout d'abord décrit d'un point de vue physiologique par l'étude du système sensoriel. Pour une étude complète, on verra [Vander 89]. Nous retiendrons particulièrement les éléments suivants. Tout d'abord, le champ récepteur d'un neurone visuel est la surface de la rétine dont la stimulation lumineuse modifie la réponse du neurone. La taille de ces champs récepteurs augmente régulièrement avec leur excentricité rétinienne. Il en résulte une diminution de la sensibilité oculaire du centre vers l'extérieur de la rétine. Les humains ont une vision frontale, qui les oblige souvent à parcourir une scène pour la comprendre dans son ensemble par saccade (le regard se déplace par sauts successifs). La fréquence et la taille de ces sauts est très difficile à mesurer. Il faut cependant prendre en compte ces paramètres lors de l'animation d'une scène visuelle. Il faut présenter chaque parcelle d'information pendant un temps suffisant pour permettre au regard de parcourir la scène, sachant que le temps de réaction oculomotrice est entre 150 et 200ms (on peut considérer que ce temps est suffisant pour extraire l'essentiel de l'information d'une image). Ce qui régit l'ordre et le déclenchement de ces saccades n'est pas établi mais il s'agit de paramètres tels que des régions de fort contraste, en luminance ou en chrominance, ou la recherche d'un élément particulier, l'exploration de détails, etc.

Les champs récepteurs sont en fait constitués de cellules réceptrices absorbant les photons, les cônes et les bâtonnets. Les cônes sont concentrés dans la région centrale de la rétine (jusqu'à un cône par neurone visuel), leur densité diminue rapidement lorsque l'on s'éloigne du centre de la rétine (la fovéa). Au contraire la densité maximale de bâtonnets est située à l'extérieur de la fovéa (il n'y en a pas dedans), et diminue avec l'excentricité (et on en dénombre plusieurs connectés à un neurone visuel). Les cônes sont sensibles à la couleur et les bâtonnets à la lumière. En fonction de leur type de réaction, on peut en déduire des couleurs et certains contrastes plus ou moins bien détectés. Par exemple, les couleurs cyan à bleu profond sont les plus difficiles à nuancer pour l'oeil, à l'opposé des couleurs orange et jaune. Il faut donc éviter le bleu pour faire ressortir un objet, sauf si l'effet est recherché, comme pour marquer un contraste de profondeur avec un objet rouge. Ce type de propriétés permet d'éditer des règles qui peuvent servir pour les applications interactives sur ordinateur. On les retrouve en effet sous formes de directives de création d'interface homme-machine dans de nombreux guides [IBM a] [Apple].

Le système nerveux rétinien sait détecter et traiter trois types d'information en provenance de la rétine : le mouvement, la forme et la couleur. Pour ce qui est de l'interprétation des formes et des mouvements, et plus encore la perception des scènes tridimensionnelles, il faut étudier des mécanismes au niveau neuronal en plus des aspects physiologiques.

Aspect psychologique

La description biologique de l'appareil optique ne nous apporte que peu d'informations sur la manière dont un utilisateur réagit en présence d'environnements 3D. Elle ne nous apporte des informations que sur le premier maillon de la chaîne, l'information qui en sort est ensuite traitée, décodée et interprétée dans un certain contexte (dépendant de l'environnement, de la tâche et des acquis de l'utilisateur). Il serait aussi intéressant d'avoir une vue globale du processus ce qui permettrait de savoir comment un utilisateur reconstruit une scène tri-dimensionnelle (réelle ou virtuelle), afin d'affiner des paramètres pour diminuer l'effort cognitif de l'utilisateur. Cependant les études psychologiques du phénomène sont assez complexes, et il en existe plusieurs. Nous retiendrons celle de [Bonnet 82]. Celui-ci explique le système visuel comme une entité isolable de l'organisme et composée de sous-systèmes, tout du moins d'un point de vue conceptuel. L'information visuelle y est traitée en trois étapes principales :

- le niveau neuro-sensoriel, qui concerne les mécanismes de codage des dimensions élémentaires de la stimulation visuelle. Une dimension élémentaire est une distribution spatio-temporelle de luminance qui active le champ récepteur d'un certain nombre de neurones visuels et dont il ressort une information, comme par exemple une orientation de contour .

- le niveau perceptif, où les dimensions élémentaires sont mises en relation les unes aux autres, créant des représentations structurales, des formes, égocentrées (elles correspondent encore à la structure de la stimulation rétinienne). A ce niveau, les représentations sont 2D 1/2 [Edgar 95]. En effet, l'information sur la 3^{ème} dimension est présente, mais sous forme de distances relatives et ne concerne que les parties visibles de l'objet (pas d'organisation hiérarchique ou de représentation globale à ce niveau).

- le niveau cognitif, où les objets sont identifiés, il s'agit d'une représentation cognitive des objets. C'est à ce niveau qu'il y a une prise de conscience de l'information, à partir des niveaux précédents, mais aussi du contexte visuel global et celui de l'activité (d'autres informations viennent compléter l'information «rétinienne»). C'est à ce niveau que l'observateur construit une représentation structurale en trois dimensions de la scène visuelle où chaque objet identifié y est intégré en terme de position spatiale.

L'information visuelle est traitée dans des circuits courts, auto-régulés, ou des circuits plus longs qui permettent une représentation consciente de ce qui est vu.

Les expériences menées au niveau cognitif font ressortir quelques règles qui pourraient se résumer dans le mot «clarté». En effet, il faut éviter les regroupements de petits objets aux formes similaires si l'on veut que l'utilisateur les distingue. Le respect de cette règle ne peut que réduire les temps d'interprétation. Sachant que lorsque l'on parle de forme similaire, cela signifie que les objets ont des caractéristiques élémentaires identiques en forme et en orientation.

La similarité des objets pourra, par contre, être exploitée dans le cas d'un gradient de texture où l'objectif est de répéter le même motif pour visualiser une perspective. On a en effet une augmentation de densité du motif de texture due à la perspective, ce qui est un facteur de différenciation rapidement évalué par le système visuel.

Mais l'étude du niveau cognitif de la perception nous permet surtout de comprendre la façon dont un observateur perçoit la troisième dimension. Pour cela, il doit interpréter l'information sur la profondeur, c'est-à-dire sa distance aux objets et la distance relative des objets entre eux. A cette fin, il utilise :

- la disparité rétinienne (voir plus loin : Vision stéréoscopique) est une information pour la profondeur relative (phénomène de stéréoscopie statique), tout comme les déplacements de l'observateur (indice de parallaxe relative).
- les vitesses des images rétiniennes lors du mouvement d'un objet qui s'éloigne sont légèrement différentes. C'est un indice du mouvement en profondeur (phénomène de stéréoscopie dynamique), tout comme le changement de couleur des objets lié à leur déplacement en profondeur.
- les indices picturaux : la perspective linéaire, les interpositions entre objets, les variations du flux lumineux sur le paysage observé et les ombres portées à condition de localiser la source de lumière.
- les invariances de représentations perceptives, comme lors des saccades ou de mouvements limités de la tête, la forme perçue qui reste constante ou la taille perçue des objets qui reste indépendante de la distance de l'observateur à l'objet et l'invariant taille-distance. Mais ces constantes sont tout de même sujettes au contexte de la scène observée.

Cependant, on ne peut que difficilement s'en tenir à une approche qui étudie uniquement les processus cognitifs de perception de scènes tridimensionnelles pour créer une interface idéale. Tout d'abord parce que le phénomène est trop compliqué à traduire dans son intégralité : il intègre beaucoup trop de paramètres. Ensuite, le système perceptif construit une représentation du monde à

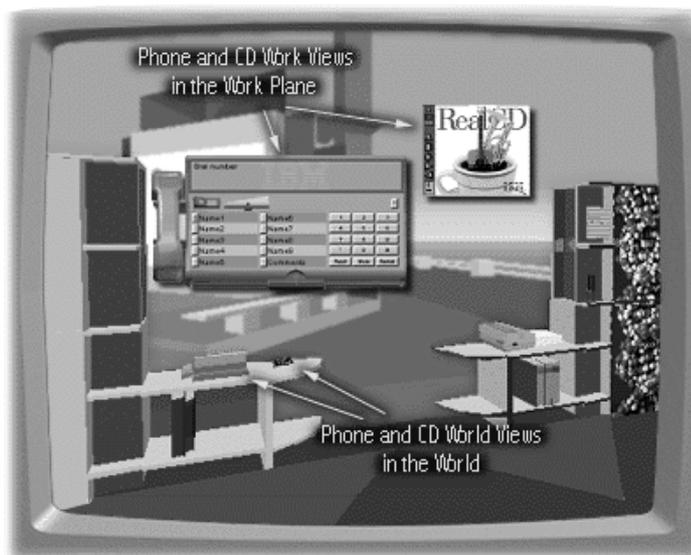
partir des informations du monde physique, mais aussi des caractéristiques de son organisme. Et assez tôt dans le processus d'interprétation, les connaissances de chaque utilisateur interviennent.

La perception ne s'intéressant qu'à la structure de la scène visuelle, il faut s'intéresser à des processus de plus haut niveau. A commencer par l'étude de la perception-action, qui crée une relation entre cette structure et la motricité. Et finalement au dernier niveau, on s'intéresse à l'adéquation entre la représentation de l'utilisateur et celle de l'interface (paragraphe 2.3).

L'ergonomie

Comme nous l'avons dit précédemment, certaines règles d'ergonomie sont directement tirées de l'étude des capacités humaines. Mais elles sont principalement constituées à partir d'expérience sur les interfaces 2D [Scapin 86]. Si certaines de ces règles sont ré-adaptables directement (sur les couleurs par exemple), on ne peut pas tenir compte de règles dictées pour des systèmes de type WIMP.

En trois dimensions, il existe peu de guides de conception d'interface. Principalement parce qu'il n'existe pas de consensus sur un modèle d'interface 3D générique, d'ailleurs difficilement imaginable. Les travaux que l'on trouve sur les interfaces 3D sont en effet spécialisés sur des types d'applications (voir partie 2.3). En fait, nous ne connaissons qu'un seul guide de directives d'interface 3D [IBM b], rédigé chez IBM à la suite du projet RealPlaces [Roberts 98]. Ce dernier a été initié en 1995 comme ce qui pourrait devenir la nouvelle interface graphique de OS/2.



Ce projet a connu une ampleur difficile à évaluer dans la mesure où il a avancé sur plusieurs fronts :

- le modèle objet de l'interface,
- les composants bas niveau,
- le style visuel global,
- l'interaction,
- la configuration matérielle,
- l'assistance à l'utilisateur,
- l'interopérabilité (en particulier avec les applications existantes),
- la technologie informatique utilisée.

L'ensemble des travaux donne une description complète du système, et fournit une suite de *guidelines* pour les interfaces 3D. Une des conclusions de l'évaluation de l'environnement de travail insiste sur le fait que même si le public et la technologie ne sont pas encore prêts, la migration de certaines applications vers la 3D sera inévitable.

D'autres interfaces se sont intéressées à la réorganisation de l'espace de travail virtuel en trois dimensions. Il existe des *window manager* en trois dimensions, qui se substituent aux gestionnaires classiques sous X11 ou BeOs. Ils ré-organisent l'agencement des fenêtres et des icônes en trois

dimensions, ou plutôt en 2D 1/2 puisqu'ils créent un axe de profondeur uniquement pour placer sur plusieurs plans les composantes de l'interface. Ils conservent cependant tous les autres métaphores (icônes, boutons, fenêtres rectangulaires). Si l'aspect esthétique peut avoir un intérêt, la troisième dimension n'y est pas exploitée.

Vision stéréoscopique

La disparité rétinienne est l'écart angulaire entre les positions rétiniennes des images du même objet sur les deux hémirétines par rapport aux fovéas. C'est ce qui caractérise la vision stéréoscopique humaine. Cette disparité est interprétée par le cerveau comme une information sur la profondeur de l'objet. L'acuité stéréoscopique est l'écart angulaire des deux images rétiniennes qui permet de percevoir une différence de profondeur, soit 12 secondes d'angle visuel en moyenne. Cela signifie par exemple qu'à partir de 10 mètres, il faudra au moins 90 mm entre deux objets pour que l'on perçoive une différence de profondeur. En synthèse d'image, on peut augmenter artificiellement l'effet de perspective pour renforcer l'effet de profondeur, ce qui revient à «tricher» sur la taille des objets.

L'écran d'un ordinateur est toujours plat, physiquement sans profondeur. On peut tout de même recréer les conditions de vue stéréo en générant une image pour chaque œil, on utilise alors un dispositif approprié (voir chapitre précédent) pour que chaque œil voit la bonne image. C'est le cerveau qui interprète la différence entre les deux images, et crée une représentation mentale de la scène en trois dimensions. Cependant, le travail imposé au cerveau pour recalibrer les deux images peut vite générer des fatigues importantes (surtout visuelle). En effet, dans les systèmes de stéréovision actuels, la convergence et l'accommodation ne peuvent pas se comporter l'une par rapport à l'autre comme dans la réalité [Edgar 95]. Cette fatigue peut être également due à un mauvais calibrage. Mais un calibrage précis est assez difficile à réaliser.

De plus, la stéréovision n'améliore par forcément l'efficacité de l'utilisateur, en particulier dans un espace 3D qui n'est pas surchargé, l'important étant alors surtout la fréquence des images [Burdea 96]. Il faut donc être prudent quant à son intégration dans une application 3D.

On peut placer un système de vision stéréo sur n'importe quel dispositif d'affichage, de la salle de cinéma à l'écran d'ordinateur de bureau en passant par les casques d'immersion. C'est au concepteur de l'application de déterminer l'intérêt de la vision stéréoscopique.

L'immersion

La vision plane des moniteurs classiques peut être améliorée par un dispositif d'immersion qui plonge l'utilisateur dans un monde virtuel. Pour cela, on utilise des casques de réalité virtuelle ou des *caves* pour couvrir tout le champ de vision de l'utilisateur. L'immersion impose de suivre les mouvements de la tête de l'utilisateur. Sans ce capteur, ce dernier serait complètement désorienté avec probablement des effets de mal de mer, etc. Au bout du compte, ces dispositifs sont assez encombrants et chers, même si les choses vont en s'améliorant (voir le *Glasstron* de Sony).

Qui plus est, des défauts techniques comme le temps de retard fréquent entre les images générées et les mouvements de l'utilisateur, ou des problèmes intrinsèques à l'immersion en monde virtuel, créent souvent des malaises chez l'utilisateur [Regan 95].

L'immersion est réservée à des applications particulières, et ne peut concerner une application comme celle envisagée dans notre projet. Le cadre bureautique d'un environnement de travail virtuel (utilisation dans un milieu de bureau, dispositif léger, qui permette de garder le contact avec son entourage, etc.) n'autorise pas l'immersion de l'utilisateur.

2.1.2 L'ouïe

Les informations sonores sont essentielles dans la plupart des applications, elles ne sont pourtant guère utilisées en dehors du retour d'actions. En simulation, elles participent au réalisme de la situation en reproduisant le contexte sonore de l'activité simulée. Elles permettent de donner une ambiance à un moment donné (musique ou bruitages appropriés). Dans le cas des IHM, le son permet de renforcer les retours visuels dans l'interface, voire d'aider à positionner certaines informations (avec un son spatialisé). Les sons sont souvent dispensés de façon ponctuelle, en fin ou début d'événements, pour renforcer un traitement. Le son est cependant peu présent dans les applications, des structures d'interface pensées pour le son permettraient d'intégrer des paramètres sonores plus adaptés et plus fréquemment dans les interfaces [Beaudouin 94].

La spatialisation du son est particulièrement indispensable dans un espace où des éléments ne sont pas dans le champ de vision de l'utilisateur. Il est alors seulement relié à ces objets par le son, seule source d'information sur leur position avec la spatialisation. Le son positionné en trois dimensions peut être également une source d'information appréciable pour aider l'utilisateur à retrouver plus facilement le contexte du message sonore (s'il est dit par un utilisateur représenté dans la scène par exemple). L'utilisation du son 3D devrait donc se généraliser [Demarey 98]. Elle nécessite cependant de connaître en permanence la position et l'orientation de l'utilisateur dans l'espace virtuel où il évolue. De plus, si les techniques de spatialisation du son sont bien maîtrisées, elles nécessitent une puissance de calcul importante. [Barme 96] a étudié les critères importants dans la localisation d'un son afin de simplifier les calculs de spatialisation. L'apparition de carte son bon marché, sachant spatialiser le son, devrait cependant résoudre rapidement la partie technologique de la spatialisation.

C'est cependant l'aspect visuel, le plus difficile à maîtriser, qui a été privilégié dans la recherche de notre modèle d'interface 3D. Le son n'intervenant que comme un apport local d'informations et ce, sans conception de notre part d'un environnement sonore spécifique [Gaver 97].

2.1.3 Le toucher

On trouve surtout des systèmes à retour d'effort dans les simulateurs : sur le volant d'un simulateur de conduite, sur le manche à balai d'un simulateur de vol, etc, jusque dans les jeux sur micro-ordinateur qui commencent à utiliser des joysticks à retour d'effort. Mais le phénomène le plus marquant sur le retour d'effort, est le nombre croissant de jeux possédant un tel dispositif dans les salles d'arcade (simulateur de ski, de surf, de voiture, etc).

En ce qui concerne le travail coopératif, il ne semble pas qu'il y ait de besoin correspondant à l'utilisation de système à retour d'effort, faute de situation de travail nécessitant ce type d'outil. Il présente peu d'intérêt avec nos contraintes, l'objectif étant de donner un maximum d'autonomie et d'interactivité à l'utilisateur, on voit difficilement son intérêt en dehors des simulateurs. Cependant, pour ce qui est de l'utilisation globale d'une interface de type WIMP, [Rosenberg 96] a montré que l'utilisation d'un joystick à retour d'effort peut améliorer l'efficacité des utilisateurs dans des tâches de pointage de bouton, de sélection d'item dans un menu ou encore de positionnement du pointeur sur des ascenseurs.

Si on n'a pas besoin d'un retour de force, on peut donc avoir besoin d'un retour de contact. En effet, la sensation que l'utilisateur perd en s'immergeant dans des mondes virtuels est le toucher. S'il conserve la vision des objets, il n'a plus la possibilité de les sentir sous ses mains lorsqu'il les manipule. Le retour d'effort peut donc par contre être utilisé à dose limitée pour donner des informations similaires au retour de toucher.



Figure 2.1 le *Phantom* de Sensable Technologies, qui existe en version bureautique

On peut, par exemple, imaginer qu'avec un périphérique de type *Phantom* (Figure 2.1) comme capteur de position 3D pour le pointeur, l'utilisateur ressent un léger choc au moment de l'entrée de l'objet, à la manière des premières souris à retour «de toucher» (Figure 2.2) lorsque l'on atteint le bord de l'écran ou d'une fenêtre.



Figure 2.2 *souris à retour de force* : la *Feelit* de Immerse Corporation et la *Virtual Reality Mouse* de Control Advancements

On peut également faire en sorte que le mouvement de la main soit ralenti quand il traverse un objet en trois dimensions, voire en fonction du type d'objet traversé et selon certaines propriétés (par exemple en CAO). Cela revient tout de même à entraver un peu le mouvement de l'utilisateur. Il pourrait être intéressant de mesurer pour certaines tâches le rapport entre la gêne suscitée et la diminution de l'effort cognitif de l'utilisateur pour comprendre la composition de l'espace 3D. Mais vu la diffusion restreinte, le coût prohibitif de ces périphériques, et le manque d'adéquation avec l'environnement bureautique d'un projet comme le notre, nous n'avons pas encore envisagé la piste des périphériques à retour de toucher ou d'effort.

A l'avenir, plus loin dans le retour d'effort, l'utilisation du «sens musculaire» [Roll 98] devrait permettre de donner des informations plus importantes, sous forme de sensations de mouvement données par des vibrateurs placés sur les tendons musculaires.

[Hinckley 99] propose d'utiliser une autre forme d'information de toucher en détectant le contact de la main sur un périphérique. Comme avec le *TouchTrackball*, qui informe l'ordinateur si la main est en contact avec lui. On dispose d'une information supplémentaire pour créer de nouvelles formes d'interaction, comme afficher un signal donné uniquement quand la main est en contact, ou descendre un document d'une page quand on touche simplement le périphérique. En travail coopératif, on pourrait l'utiliser pour détecter dans quelle phase d'activité est une personne.

On peut remarquer que certains périphériques isométriques extrêmement sensibles, comme la *Spacemouse*, permettent directement de savoir si la main touche la partie sensible du périphérique.

2.1.4 Bilan

Nous avons donc essayé de tenir compte au maximum des apports de la psychologie, des expériences sur la perception, sur la motricité, sur les processus cognitifs qui pouvaient nous aider à comprendre les facteurs humains qui entrent en jeu dans les tâches que nous explorons. Au niveau des expérimentations bien sûr, mais aussi en travaillant avec des psychologues dès les premières étapes de conception. Notre approche a cependant été assez pragmatique, dans le sens où, à partir des connaissances des capacités de perception d'un utilisateur, de ses attentes et de notre objectif, nous avons proposé des primitives d'interaction que nous avons développées et ensuite vérifiées expérimentalement.

2.2 Le Travail Coopératif Assisté par Ordinateur (TCAO)

Nous allons exposer maintenant les différents concepts inhérents au travail simultané de groupe (TCAO synchrone). Les outils de TCAO (les collecticiels) existants sont nombreux, mais les derniers progrès technologiques permettent plus que jamais d'espérer des applications à la hauteur des exigences des utilisateurs, tenant compte de leurs capacités et respectant les conventions sociales. Les collecticiels disponibles actuellement utilisent des moyens classiques, basés notamment sur la vidéo. Nous allons d'abord décrire brièvement ces derniers avant de nous intéresser aux concepts fondamentaux du TCAO synchrone (on trouvera une étude complète dans [Saugis 98]).

2.2.1 L'hégémonie de la vidéo

Les systèmes actuels sont donc basés sur la vidéo. Ils sont équipés de caméra pour filmer les différents interlocuteurs et de micros pour capter le son. Ils sont adaptés à des réunions de deux ou plusieurs personnes seules, ou de groupes entiers. On parle de visioconférence. Les dispositifs pour représenter les interlocuteurs vont de l'écran vidéo le plus simple au grand écran en passant par les *caves*.

La caméra pose le problème du point de vue, on est dépendant de la position de l'utilisateur par rapport à la caméra, ce qui modifie la perception que l'on a de lui. Il est de plus difficile d'avoir le même point de vue sur tout le monde, ce qui serait idéal pour ne pas créer d'a priori lié à la dimension ou à l'orientation de l'image des interlocuteurs entre eux.

Un autre inconvénient de la vidéo est l'impossibilité d'utiliser le regard, sauf dispositif vraiment particulier [Buxton 92]. En effet, si un utilisateur regarde un écran de projection, il est filmé à partir de la caméra fixée à côté de cet écran, on perd donc l'information liée au regard qui est décalé de la même manière que l'écran et la caméra. Typiquement, il n'est pas possible de regarder quelqu'un directement dans les yeux.

La vidéo n'est pas un flux manipulable facilement, et dont il n'est pas évident de faire ressortir des informations (par analyse sur ordinateur). Sa représentation est généralement dissociée du reste des éléments de la réunion (documents, etc), par un affichage sur un écran ou dans une fenêtre séparée. Il devient alors difficile, quand il y en a besoin, de créer des relations entre la représentation vidéo des utilisateurs et les différentes représentations des documents présents dans la réunion.

Enfin, il n'est pas démontré que la vidéo soit utile dans l'utilisation actuelle que l'on fait des collecticiels [Williams 77], son apport peut même être jugé négligeable pour faire de la téléprésence. Il s'agit peut-être d'étendre le champ d'utilisation des collecticiels, afin d'aborder des domaines où la téléprésence serait indispensable : en matière de réunions par exemple, en rendant possible leur utilisation dans des rencontres critiques inter-entreprises plutôt qu'en réunion intra-entreprise sans négociation. Il faudra pour cela réduire les manques des réunions médiatisées par rapport aux réunions réelles, et tirer parti de l'aspect virtuel. La vidéo, principalement à cause des limites citées ci-dessus, risque d'être écartée pour représenter les interlocuteurs au profit de solutions plus innovantes, moins contraignantes.

Les outils audiovisuels

Les projets les plus aboutis en matière de travail coopératif concernent surtout des solutions dédiées. Ces projets de recherche sont les *Hydra Units* [Buxton 92], le *ClearBoard* [Ishii 92] ou *MAJIC* [Okada 94], qui concernent des outils de réunions de petits groupes.

On peut trouver également des solutions d'enseignement à distance basées sur des outils audiovisuels [Peters 94] ou des formes plus originales, comme les médiaspace [Beaudouin 96] qui consistent à équiper chaque bureau des collaborateurs d'une caméra, d'un moniteur et d'un système de commande permettant d'entrer immédiatement en communication avec qui l'on veut, de savoir si quelqu'un est présent, etc.

Les outils informatiques

Il s'agit des applications informatiques de travail collaboratif utilisant une caméra vidéo, principalement pour faire de la téléprésence, on parle alors de *desktop conferencing* (réunion à plusieurs, mais chacun dans son bureau devant sa machine). Ces outils proposent en plus de partager des applications de bureautique classique pour travailler à plusieurs sur un même document, mais souvent en mode maître-esclave. Les produits commerciaux sont nombreux, les plus connus sont *PictureTel* (ZDS), *ProShare* (Intel) ou *CUSEeMe*.

Bilan

Même si la représentation des autres utilisateurs passe après la nécessité d'un son de bonne qualité (i.e. sans interruption) ou d'une gestion adaptée des documents entre les utilisateurs, la téléprésence n'en est pas moins indispensable pour créer un véritable CVE (Collaborative Virtual Environment), où les intervenants ont naturellement conscience de la présence des autres. La vidéo sur ce point pose le problème de l'intégration à un environnement de travail virtuel, et rappelons-le, nécessite des moyens techniques importants (en terme de réseau en particulier). Si on imagine difficilement d'autres alternatives dans le cas de visioconférence entre des salles contenant plusieurs personnes, d'autres solutions sont envisageables pour le desktop-conferencing. Notre projet s'est employé à les explorer.

2.2.2 Les concepts essentiels du TCAO synchrone

L'interface doit supporter le travail coopératif synchrone, donc des applications où les participants doivent communiquer pour prendre des décisions, comparer leurs opinions ou trouver des solutions (téléconférence, enseignement). Le réalisme des situations est crucial. L'utilisateur doit sentir immédiatement qu'il est avec d'autres personnes. Des expériences comme celle des *Hydra Units* [Buxton 92] ou de *Majic* [Okada 94] nous permettent de tirer quelques points fondamentaux des réunions médiatisées :

- le contact visuel : un interlocuteur perçoit qu'on le regarde, il peut échanger des regards.
- la conscience du regard : on trouve le centre d'intérêt d'une personne en suivant son regard.
- les expressions faciales : elles donnent des informations sur les impressions des interlocuteurs, leur acquiescement ou leur mécontentement, etc.
- les gestes : ils sont importants pour la désignation (manipulation, interface gestuelle, etc.), mais aussi comme moyen d'expression dans le dialogue.

L'activité de groupe

La parole est loin d'être le seul vecteur de communication pendant un dialogue [Cassel 94]. Les gestes (volontaires ou réflexes), le regard et l'expression du visage apportent tout autant d'informations et aident à communiquer. De plus, le travail à plusieurs exige de connaître le point de vue et les actions de ses interlocuteurs [Shu 94] [Kuzuoka 94]. Il faut donc définir des métaphores

permettant aux différents participants d'une activité coopérative de comprendre les actions des autres et de pouvoir interagir.

Les avatars

Tout d'abord, les différents interlocuteurs peuvent être vus dans l'interface sous forme de vignette fixe, animée (vidéo), ou sous forme 3D. Chacune de ces représentations ayant un contenu sémantique croissant (le plus pauvre étant l'image fixe qui ne fait qu'identifier visuellement une personne). A ce propos, [Benford 95] définit, quand il s'agit de communication, les différents critères à envisager pour représenter une personne dans un environnement virtuel. Il établit des règles pour chaque caractéristique, et les applique pour leur propre système *DIVE* (Distributed Interactive Virtual Environment) [Benford 93]. Ses travaux mettent en avant les qualités d'une représentation sous forme de clone 3D, qui permet donc de garder un contact visuel (on maîtrise le mouvement des yeux d'un clone), les gestes et les expressions faciales. Le clone est piloté par les actions et/ou les mouvements naturels de l'utilisateur.

Si on choisit une représentation 3D des utilisateurs que l'on pilote à partir des mouvements de l'utilisateur, les mouvements du clone seront naturels et continus. Cependant, plus le clone sera complet (des pieds à la tête), plus les mouvements de l'utilisateur seront durs à capter pour être fidèles, tellement il existe de possibilités de mouvements [Cadoz 94b]. Une autre solution consiste à capturer le mouvement de l'utilisateur en l'équipant avec un minimum de capteurs pour ne pas (trop) le gêner. On peut par exemple facilement calculer la position des différentes articulations du bras à partir de la seule position de la main grâce aux algorithmes de cinématique inverse (avec une incertitude sur la position du coude).

Une autre alternative consiste à animer le clone à partir des actions de l'utilisateur. Celles-ci sont analysées pour en déduire un comportement et animer le clone en conséquence. Cette approche est préférable dans les environnements qui ne sont pas complètement immersifs, et où le mouvement d'un utilisateur n'a pas forcément de sens en lui-même (s'il va cliquer sur un icône pour ouvrir un fichier, il faut mieux voir son avatar ouvrir un fichier que bouger le bras comme l'a fait son utilisateur).

Avec sa représentation, chaque intervenant doit posséder un moyen de désignation à distance, visible dans l'interface de ses interlocuteurs, un télépointeur [Primet 94]. Le télépointeur est un élément qui permet de faire de la téléprésence. Il peut être amélioré par des éléments sonores ou visuels comme les marqueurs.

L'activité centrée autour de la tâche

Les participants d'une réunion doivent se concentrer autour d'un ou de plusieurs documents partagés (on parle de «document» au sens large, quel que soit son contenu, statique ou dynamique, sa dimension ou la dimension de sa représentation). Il est donc préférable de renforcer l'impression de collaboration autour de ces derniers, en les plaçant de manière centrale. Ce concept consistant à regrouper spatialement les documents pour recentrer l'attention est née du projet *Rooms* [Henderson 86] de Xerox. L'idée originale était de diminuer la concentration et les recouvrements de

fenêtres en les regroupant en fonction d'une tâche particulière dans des bureaux virtuels, un seul étant affiché à la fois.

La métaphore de la table de réunion

Il semble logique de vouloir matérialiser cette séparation des tâches. Un espace ouvert et homogène ne conviendrait pas. On rencontre souvent dans la littérature la notion de pièce (au sens visuel et sémantique). Elle est caractérisée par un espace clos et associée à une seule tâche. Cela ne permet cependant pas de garder une vue, même subjective, des autres activités se déroulant en parallèle. Nous allons étendre ce concept à la métaphore de la table de réunion, où un utilisateur peut rejoindre un groupe de personnes en gardant un contact (visuel ou sonore) sur les autres tables.

Ce concept est proche de la réalité. En effet pour que l'utilisateur se retrouve dans une situation familière, un environnement tridimensionnel est une solution idéale - ceci pour obtenir une interface naturellement conviviale et intuitive.

Perception et soutien de la conscience mutuelle

Certaines tâches requièrent de se concentrer sur un point particulier, tandis que d'autres nécessitent une vue globale du milieu. En général, l'attention est temporellement partagée entre ces deux cas. Cependant, le TCAO exige de chaque participant qu'il sache ce qui se produit, ce qui change, où et par qui : l'interface doit alors pouvoir soutenir cette conscience mutuelle. L'idéal serait de voir tout ce qui se passe à tout moment dans la pièce : le «tout visible». Il y a cependant une limite à la quantité d'information qu'il est possible d'afficher sur un écran. On peut néanmoins tirer parti (tout en s'adaptant) des spécificités de la perception humaine. En particulier, il existe une zone centrale dans le champ visuel où l'image est très précise, alors que la vision périphérique, manquant de netteté, réagit surtout aux mouvements et aux changements (ce qui correspond aux cônes et aux bâtonnets qui composent les champs récepteurs, voir la partie précédente). Un système de visualisation utilisant des déformations (style *fish eyes*) pourrait mettre à profit cette caractéristique.

L'animation interactive

Elle permet deux choses. Tout d'abord, on peut densifier la quantité d'informations affichées sur l'écran, et rendre parfaitement lisible seulement une faible partie de ces dernières [Robertson 91] [Mackinlay 91], le reste donnant, sous sa forme dégradée, une idée globale du contenu (Figure 1.3). Une animation interactive de ces données permet à l'utilisateur de visualiser correctement et rapidement chaque partie du document [Robertson 93].

Une autre utilisation de l'animation interactive est celle de l'interface elle-même, qui devra être fluide. Nous voulons éviter les ruptures abstraites dans la continuité de la scène, qui augmentent la charge cognitive de l'utilisateur. Une simple observation doit permettre d'éviter un effort cognitif inutile. Par exemple, certains systèmes graphiques décomposent la réduction d'une fenêtre à l'état d'icone en une animation linéaire, au lieu de fermer brutalement la fenêtre et de créer un nouvel icône : plus besoin de chercher à reconstituer mentalement le lien entre les deux formes du même objet. C'est la représentation graphique qui diminue le coût cognitif.

2.3 Interfaces 3D

Les interfaces de bureau actuelles, de type WIMP montrent vite leurs limites d'un point de vue ergonomique dans le cas du TCAO. L'espace 2D est très contraignant pour l'affichage de quantités d'information importantes. Il est inadapté au travail coopératif où l'on n'est plus seul à agir dans son interface. Le recouvrement des fenêtres, la nécessité de réduire l'information à l'état d'icone empêchent de voir ce que d'autres utilisateurs font sur les documents ainsi masqués. Quant au système d'affichage des fenêtres, que ce soit X Window ou MS Windows, il banalise les applications dans des fenêtres toutes identiques, sans rapport avec la tâche de l'utilisateur. Les fenêtres étant dissociées l'une de l'autre, lorsqu'elles contiennent des documents ou des acteurs (les interlocuteurs), il est difficile de créer des liens visuels entre elles de façon naturelle, sans surcharger l'interface.

Les représentations 2D+3D lors de présence de document 3D montrent vite leurs limites. En effet, les contrôles restent en deux dimensions, avec les classiques menus et boutons, désolidarisant complètement la partie commande du document lui-même. Ces applications exigent en général un apprentissage important de la part de l'utilisateur (comme les outils de modélisation 3D).

Enfin, les interfaces 2D classiques posent des problèmes du rapport de l'utilisateur dans l'espace de travail (système peu proprioceptif) lors de sessions contenant plusieurs utilisateurs et documents, le système ne permettant pas facilement de construire une représentation mentale globale de l'activité. D'où l'intérêt des interfaces 3D pour ré-organiser l'espace de travail en se dégageant des contraintes des systèmes classiques.

Si l'organisation de l'espace de travail est plus aisée, la navigation dans cet espace et l'interaction se passent différemment. Nous allons maintenant regarder de plus près ces deux aspects.

2.3.1 Navigation

Nous définissons la navigation comme les déplacements de l'utilisateur à l'intérieur de l'espace 3D, ce qui se traduit par des changements de point de vue de la scène. L'interaction, quant à elle, correspond aux actions de l'utilisateur dans la scène, qui manipule des objets sans changer de point de vue global sur cette dernière.

Perception

La navigation pose des problèmes tout à fait différents suivant le mode de visualisation dans lequel on se trouve, immersif ou non, et le nombre de degrés de liberté suivant lesquels l'utilisateur peut se déplacer. Les problèmes de positionnement sont plus importants en immersif, l'utilisateur a du mal à se construire une représentation mentale de l'environnement, les distances sont sous-estimées [Henry 93]. Il s'agit donc de donner des informations supplémentaires pour aider l'utilisateur.

Pour ce qui est de l'orientation de l'utilisateur par rapport au monde, dans le cas d'un système immersif, il est conseillé de ne pas autoriser les rotations de l'utilisateur sur lui-même (azimut fixe), et de le laisser au sol pour limiter les problèmes d'orientation spatiale. Tout dépend cependant de l'application. Dans le cas d'un système non-immersif, l'environnement de l'image virtuelle (le

monde réel) sert de référent spatial. L'utilisateur a donc moins de chance de se retrouver lui-même désorienté.

Pour ce qui est de l'orientation globale dans le monde, le principal indice visuel supplémentaire est une carte, où l'utilisateur et sa direction sont représentés, si possible auto-orientée et auto-centrée sur lui. [Stoakley 95] propose un outil de navigation avancée (Figure 2.3), le WIM (Worlds In Miniature), pour aider l'utilisateur à s'orienter. Ce dernier voit une carte 3D représentant les environs de l'endroit où il se trouve. Il peut se déplacer ou explorer à l'aide du WIM.

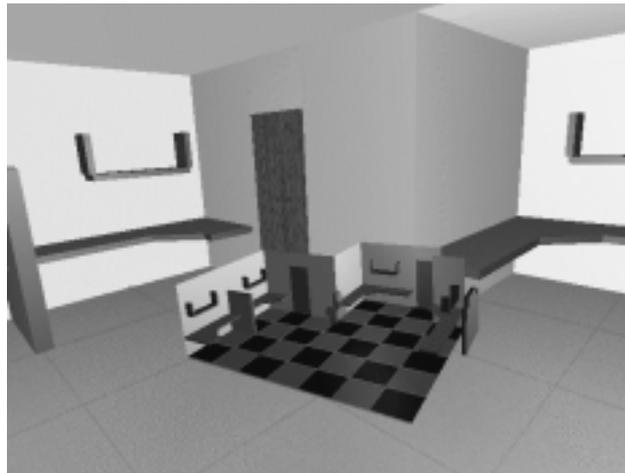


Figure 2.3 *Worlds In Miniature*

Le WIM est une carte très évoluée, proposant une deuxième vue sur le monde virtuel. Il est représentatif du problème de la navigation où le repérage de l'utilisateur, au moins par un simple système de carte, est indispensable.

Les navigateurs

Les navigateurs pour les mondes 3D les plus courants sont ceux dédiés aux fichiers VRML (voir Figure 2.4). Par leur vocation à être utilisés sur internet, ils s'adressent au plus grand nombre. L'interaction s'y fait au moyen de la souris, en interaction directe pour le déplacement plan (en général) et grâce à des widgets (Figure 2.5) pour ce qui est des autres (rotation autour des axes, élévation, etc). Ils sont assez peu pratiques à utiliser, principalement à cause de widgets de petites tailles, pas toujours faciles à sélectionner, et d'un mode de déplacement en translation dans le plan du sol (le plus souvent utilisé) assez peu pratique. *CosmoWorlds* utilise par exemple deux vitesses de déplacement, on accède à la seconde en appuyant sur la touche SHIFT de son clavier.



Figure 2.4 *WorldView Platinum*, visionneuse VRML (navigation possible au joystick). Le pointeur de la souris change de rôle en fonction des outils que l'on sélectionne sur la palette à gauche.



Figure 2.5 *CosmoWorlds*, la visionneuse VRML de référence d'un point de vue technique. Une petite manette permet de passer du mode manipulation au mode navigation.

L'utilisateur est en général sauvé par les points de vue prédéfinis (par le concepteur du monde 3D) qui lui permettent de se déplacer aisément à travers le monde en lui donnant une vision globale de celui-ci (si les *viewpoints* ont été bien placés).

Grâce à des environnements construits avec les outils de *Sense8* par exemple (voir le chapitre 1), la navigation au moyen de périphériques comme les joysticks ou les trackballs 3D peut facilement être mise en oeuvre avec des mondes VRML. Grâce à ces périphériques, elle est plus aisée et plus intuitive qu'à la souris : le périphérique est utilisé en mode relatif, et les déplacements de la main correspondent au sens du déplacement (avant/arrière et droite/gauche).

Lorsque l'on est placé en immersif, un capteur d'orientation placé sur la tête permet de faire suivre au monde virtuel les mouvements de tête. Le regard est ainsi facilement intégré, à la condition que le temps de réponse du système soit très faible, autrement, un «mal du virtuel» est à prévoir.

Il est intéressant de remarquer à quel point des utilisateurs expérimentés de jeux comme *Quake* arrivent à se déplacer facilement dans les mondes 3D de la même manière : le joueur oriente son regard de haut en bas et le personnage est orienté de la gauche vers la droite grâce au même mouvement à la souris, et l'ordre d'avancer ou de reculer est donné grâce à deux touches au clavier (utilisation de l'autre main, la main dominante tenant la souris). Cette configuration a fini par faire l'unanimité dans ce monde de joueurs, au profit d'un déplacement rapide et sûr, et au détriment d'un temps d'apprentissage excessivement long. Cette configuration est peut-être donc la solution la plus efficace pour ce qui est de la navigation dans des mondes 3D au moyen des périphériques standards, le clavier et la souris.

Bilan

Les deux éléments, navigation et interaction, sont liés car on doit pouvoir se déplacer pour agir dans l'interface. Les modes varieront suivant l'application. Mais la troisième dimension posant de nouveaux problèmes comme celui du positionnement ou de la direction du point de vue de l'utilisateur, il faut gérer les limites de sa perception pour ne pas le désorienter [Gomez 94] - a fortiori dans notre application où le point central n'est pas la navigation à travers l'interface, mais le travail qu'on y effectue. En effet, dans le cadre du travail coopératif, l'utilisateur n'a pas seulement une position physique dans l'interface, mais aussi une position liée au groupe. Au delà de la conscience de groupe, chacun possède un rôle (par exemple, le chef de réunion et un intervenant ne feront pas la même chose). Ce rôle peut induire des limites et l'utilisateur devra disposer d'une instance de l'interface adaptée à son propre rôle, avec sa propre vue des choses. Il devra se trouver d'autant plus impliqué dans son espace de travail, et ne pas rencontrer des difficultés de déplacements. Ces problèmes nous contraignent à définir un cadre géométrique de navigation limitée, suffisant et facile à utiliser.

2.3.2 Interaction

L'interaction est le problème majeur d'une nouvelle forme d'interface homme-machine. S'il existe peu d'environnement d'un type donné, on a peu de chance de trouver beaucoup d'expérimentations ou de règles d'ergonomie qui permettent de le faire évoluer. Il existe cependant un certain nombre d'études, certaines empiriques, sur des méthodes d'interaction 3D [Hinckley 94] [Hand 97] [Manitiu 98]. Certaines explorent les possibilités basées sur de nouveaux périphériques, alors que d'autres développent des environnements 3D en continuant à interagir avec un périphérique 2D. En effet, les utilisateurs d'ordinateur devenant vite des experts (au sens de la compétence) du maniement de la souris, il est peut-être souhaitable de profiter de cette maîtrise pour les faire agir en trois dimensions.

La souris?

La manipulation d'objets 3D à la souris a été étudiée par [Chen 88] [Emmerik 90]. Ces travaux ont permis de développer des solutions pour des applications contenant des objets 3D. La manipulation se fait en général à travers un widget 3D (nous nous référons à la définition de [Brookshire 92]). C'est un périphérique virtuel qui apparaît en général sur l'objet (Figure 1.11). Certains produits se démarquent, comme *Bryce 4D* de MetaTools, dont le design ergonomique permet une manipulation rapide de scènes de paysage 3D. On a vu également des widgets dans la première version de *Cosmoplayer*, comme un widget de navigation sous la forme d'un manche à balai (Figure 2.6). Ce «joystick virtuel» a été abandonné par la suite au profit d'une navigation directe à la souris, le curseur dynamique de la souris remplaçant l'animation de la manette.



Figure 2.6 le navigateur de *CosmoPlayer 1.0* sur Silicon Graphics

Mais il ne s'agit pas d'interaction directe sur le document lui-même, elle est détournée à travers un autre objet ce qui rend l'opération moins intuitive et moins pratique.

[Hinckley 97a] a réalisé une étude sur la rotation d'objets 3D qui montre la meilleure efficacité des périphériques 3D (des capteurs électromagnétiques type *Polhemus* en l'occurrence) par rapport à des périphériques virtuels manipulés à la souris (l'*Arbball* de [Shoemake 92] et la sphère virtuelle de [Chen 88]).

Ces solutions pour la manipulation et la navigation en trois dimensions permettent d'utiliser un périphérique (la souris) qui n'est pas adapté à la tâche originale. Elles ont pour avantage de pouvoir conserver les périphériques auxquels les utilisateurs sont éventuellement habitués, et dont les ordinateurs sont équipés (en particulier quand aucune évolution matérielle n'est possible). Du point de vue de l'utilisation, cela limite beaucoup le développement, qui retombe sur des solutions peu pratiques et/ou peu intuitives. Par exemple, le widget de manipulation d'objet de *CosmoCreate* s'utilise avec la souris, mais en conjonction avec plusieurs touches du clavier pour accéder à tous les modes de déplacement (rotation, translation suivant les axes, etc.), ce qui le rend difficile d'accès.

Lorsque l'application propose une visualisation tri-dimensionnelle, l'utilisation d'un périphérique à plus de deux degrés de liberté peut devenir vite très utile. Tout dépendra bien sûr du type d'interaction désiré.

La manipulation directe consiste à agir sans intermédiaire sur un objet pour changer son état. Une souris permet par exemple de faire de la manipulation directe sur les fenêtres en les déplaçant par un cliqué/glissé sur la barre de titre. Elle a été formalisée par Schneidermann, qui la caractérise par trois propriétés (voir dans [Morineau 96]) :

- une représentation continue de l'objet sur lequel porte l'attention de l'utilisateur,
- une action physique pour remplacer des syntaxes complètes,
- des actions rapidement réversibles dont l'impact est visible.

La manipulation directe permet de gagner en efficacité grâce à la manipulation de représentations graphiques simples, elle est plus intuitive et permet de travailler sur des objets visuels. Cependant, la manipulation d'un menu reste simplement de la manipulation de menus, i.e. de la manipulation indirecte vis-à-vis du document concerné [Beaudouin 97]. Le fait d'aller sélectionner l'objet COUPER dans le menu EDITION d'un éditeur de texte ne constitue pas une action directe de manipulation sur le texte concerné. Il faut bien retrouver, lire et interpréter les options dans les menus pour effectuer l'action. On pourrait parler de manipulation directe sur le texte s'il s'agissait de prendre un outil COUPER (matérialisé par exemple par un pointeur souris en forme de ciseau) et d'aller cliquer sur le texte. Il faut donc cerner la différence entre les objets que l'on manipule et les objets que l'on modifie. Plus ils seront proches, moins l'effort pour retrouver et effectuer l'action sera importante, et plus la manipulation sera directe. Par exemple, un menu contextuel sur le texte sélectionné où apparaît l'option COUPER sera déjà plus facile à interpréter, même s'il ne s'agit pas encore de manipulation directe sur le texte.

Les périphériques que l'on utilise doivent être adaptés à la tâche. On ne trouve pas de périphérique idéal pour toutes les tâches. Ainsi, si on utilise beaucoup la souris dans les environnements de travail 2D classiques, les circonstances d'utilisation (l'environnement de la machine) ou les particularités du logiciel utilisé, rendront indispensable l'utilisation d'autres périphériques plus adaptés, là où la souris n'aurait pu suffire a priori : les dessinateurs utilisent des tablettes graphiques, les ordinateurs portables substituent maintenant les *glidepoints* à la souris ou au trackball, etc. On peut dire que les périphériques sont plus ou moins proprioceptifs, et ce en fonction de la tâche à accomplir.

Une alternative pour l'utilisateur entre la nouveauté des périphériques 3D et la souris pourrait être à l'avenir des souris évoluées comme la *Roller Mouse* [Venolia 93] ou la *Rockin'Mouse* [Balakrishnan 97]. Cette dernière propose une évolution de la souris vers un périphérique 3D. Elle permet, avec ses quatre degrés de liberté, d'étendre les possibilités d'une souris classique, tout en conservant son mode de fonctionnement. Elle permet un gain notable d'efficacité sur des tâches de placement d'objets 3D par rapport à la souris.

Pointage

[Ware 88] a montré que le placement approximatif d'objets en trois dimensions devenait trivial avec leur capteur de position 3D *The Bat*, comparé à la souris. Une étude de [Driver 90], qui compare un trackball 3D, une souris et un capteur de position 3D dans une tâche de pointage d'objet, révèle la supériorité du capteur 3D. Les utilisateurs font moins d'erreurs liés à la perception en trois dimensions, ce que l'auteur explique par le rapport direct entre le mouvement de la main et le mouvement du pointeur (le retour d'information kinesthésique), qui donne une information supplémentaire pour percevoir la profondeur. Et [Zhai 98] montre que les périphériques isotoniques sont les plus intuitifs et les plus rapides à maîtriser.

Aussi, les capteurs de position 3D semblent intéressants pour le pointage d'objet, mais cela pose des problèmes de positionnement des mains, de perception des mouvements entre l'espace réel et l'espace virtuel [Venolia 93]. Le modèle d'interaction doit être en étroite relation avec le moyen (matériel) de navigation que l'on propose : l'étude doit comprendre en parallèle le mode et le moyen d'interaction.

Espace d'action

On peut utiliser chaque périphérique en mode relatif ou en mode absolu. Les périphériques isométriques sont généralement plus adaptés au mode relatif et les périphériques isotoniques au mode absolu.

[Hinckley 94] suggère à partir de l'analyse des travaux d'Yves Guiard que les déplacements d'un périphérique dans un espace fixé (sans moyen de configuration) diminuent l'efficacité de l'utilisateur.

L'espace d'action de l'utilisateur peut exiger de lui qu'il navigue pour atteindre les objets, il doit donc d'abord se déplacer pour aller saisir un objet (typiquement dans les systèmes de réalité virtuelle réaliste). A l'inverse, l'utilisateur peut n'avoir qu'un périphérique pour pointer dans tout l'espace, sans besoin de naviguer (comme la souris dans une interface de travail 2D). [Poupyrev 96] propose une approche intermédiaire, sa méthode d'interaction permet en environnement immersif à un utilisateur équipé d'un capteur de position 3D d'atteindre des objets distants en déplaçant une main virtuelle rapidement vers l'objet le plus près (plus l'objet est loin, plus le déplacement de la main est sensible). Le système agissant normalement (correspondance exacte entre le mouvement de la main réelle et virtuelle) quand l'utilisateur est à portée de bras de l'objet.

Sélection

La méthode classique consiste à sélectionner l'objet une fois que le pointeur, i.e. le capteur de position 3D, est directement positionné sur l'objet. [Yoshimura 94] présente un système d'interaction 3D dans le cadre de la CAO. Il s'agit de placer des objets dans une pièce, en modifiant son point de vue et les sources d'éclairage. Il justifie en particulier l'utilisation de la manipulation directe en 3D, en reprenant la définition de [Schneiderman 83]. Leur modèle d'interaction n'est pas généralisable, mais il utilise un pointeur 3D se terminant par un rayon qui traverse les objets dans l'axe du pointeur (technique du *ray casting*). On utilise dans ce cas l'orientation du capteur 3D, plutôt que sa position.

Indices perceptifs

Il s'agit d'un retour d'informations visuelles. La synthèse d'image temps réel nous permet de créer facilement et à des fréquences suffisantes (au moins 20 images/seconde), des images d'objets 3D éclairés. Les effets d'ombrage sont une première information essentielle à la perception de la profondeur. L'autre indice indiscutable est la projection en perspective des objets 3D (à comparer à une projection orthogonale). On peut ensuite utiliser un certain nombre d'indices visuels statiques ou dynamiques, comme l'ombre du pointeur 3D [Hudson 92]. Mais [Wanger 92] a montré que les indices perceptifs d'une interface 3D dépendent de la tâche effectuée. L'objectif est de donner les

bonnes informations sans surcharger l'interface. Il s'agit de définir le type de rendu que l'on veut en fonction des objectifs de l'application visée mais aussi des tâches élémentaires à effectuer, c'est-à-dire en fonction de la représentation qu'a l'utilisateur de ce qu'il fait (sa compréhension).

Une autre forme de retour d'information utile est le retour d'information tactile. D'après [Burdea 96], il permet une sensation d'immersion impossible à égaler avec uniquement un retour visuel. L'auteur analyse le manque de solution matérielle appropriée dans le commerce par la difficulté technologique du dispositif, due en partie aux exigences et à la complexité du phénomène à simuler, le toucher des doigts.

Le geste

Les techniques d'interaction que nous venons de décrire (pointage, sélection) correspondent à une fonction particulière du geste.

Claude Cadoz [Cadoz 94b] [Cadoz 96] propose une description des différentes natures des gestes. Le geste est particulier du fait qu'il sert à la fois à émettre et à recevoir de l'information. On distingue trois fonctions complémentaires : ergotique (action matérielle sur l'environnement), épistémique (connaissance de l'environnement) et sémiotique (émission d'information vers l'environnement).

La fonction épistémique ne s'arrête pas à la perception tactile cutanée (état de surface, température) concentrée sur la face interne des doigts. La main est l'organe du sens tactilo-proprio-kinesthésique. Dans la fonction ergotique, la main est en contact avec la matière, qu'elle va modifier. Les objets reçoivent de l'énergie, et non de l'information, et peuvent en renvoyer une partie en retour. Dans la fonction sémiotique, le geste est un organe d'expression. Elle est naturelle et spontanée. Non seulement les gestes accompagnent la parole mais ils partagent avec elle certaines étapes de traitement. Ces fonctions coexistent dans ce que Cadoz nomme le geste instrumental, en référence à celui du musicien. En frappant sur une touche de piano, on communique de l'énergie à l'instrument pour le faire vibrer mais le but est de s'exprimer, de faire passer une émotion.



Figure 2.7 *l'instrument, médiateur entre l'utilisateur et l'objet, d'après Michel Beaudouin-Lafon*

[Beaudouin 97] propose une autre approche de l'instrument. Il présente un modèle d'interaction instrumentale qui permet de décrire de nombreuses formes d'interaction. L'instrument est fait d'une partie physique (entrée/sortie physique de l'ordinateur) et d'une partie logique qui fait l'interface entre l'objet édité et l'instrument (Figure 2.7). Un principe de ce modèle, la réification, permet de

transformer un concept complexe d'une application en un objet, manipulable par un instrument. Ce modèle permet une approche originale de la description de composants d'une interface.

Nous ne nous sommes pas intéressés au déroulement du geste, tel que le ferait les études en perception action, mais plutôt à comment rendre l'action de l'utilisateur compatible au niveau de l'interface (on s'est intéressé à la manière dont les gens comprennent ces actions, et non à la manière dont ils les perçoivent).

Interaction à deux mains

Le nombre réduit de possibilités d'interaction avec un seul périphérique, et la transposition naturelle de l'utilisation courante des deux mains dans les tâches réelles courantes, nous incitent à utiliser une interaction à deux mains. Certaines expériences, sur ordinateur ou non, ont déjà prouvé le manque d'efficacité d'une tâche réalisée avec une seule main, comparé à une tâche réalisée avec les deux mains. Cependant, les rôles de chaque main doivent être bien spécifiques à leur position, dominante ou non, dans les capacités de l'utilisateur.

[Kabbash 94] met en avant les applications pour lesquelles l'utilisation des deux mains est positive, et n'apporte pas une charge cognitive supplémentaire pour l'utilisateur. En effet, il a constaté que l'utilisation des deux mains peut être plus difficile qu'avec une seule, en particulier si les applications assignent une tâche indépendante à chaque main. Cependant, cela permet aussi dans certains cas de s'adapter beaucoup plus vite, d'aller plus rapidement à l'information ou de manipuler plus facilement l'interface. Pour trouver les applications à deux mains adéquates, on veillera à respecter certaines contraintes qui placent la main gauche (pour les droitiers) comme une main qui sera le référentiel pour l'action, l'initiateur de l'action ou encore celle qui aura la partie de l'action la plus facile.

Bilan

Nous venons de voir que l'on peut spécialiser les périphériques en fonction des tâches. Par exemple, le capteur de position 3D semble le périphérique le plus adapté à des tâches de pointage en trois dimensions. On peut aller jusqu'à utiliser le périphérique le plus adapté pour chaque fonction à réaliser dans un programme. Cette interaction explicite est intéressante mais démultiplie vite le nombre de périphériques nécessaires ! Il convient donc de trouver un ou deux périphériques, les plus adaptés aux tâches de notre environnement. Il faudra donc identifier ces tâches, leur mode d'interaction le plus adapté, et trouver les dénominateurs communs qui feront de notre système d'interaction un outil générique pour les environnements tri-dimensionnels de travail coopératif synchrone.

Les travaux que nous venons de décrire sur l'interaction dans des environnements 3D, et le fait que nous puissions concevoir notre interface sans contrainte sur l'interaction, montrent le chemin vers une interaction plus naturelle, où on cherchera les rapports entre les gestes réels et le sens idéal que l'on peut leur donner dans un environnement virtuel. On essayera de faire en sorte que l'utilisateur puisse reproduire certains comportements réels en virtuel, en identifiant des caractéristiques de l'environnement que nous aurons introduit dans la représentation de l'espace. On espère ainsi que

l'utilisateur pourra assimiler complètement le moyen d'interaction, pour oublier l'outil et se concentrer sur sa seule tâche. Nous voulons définir un modèle d'interaction écologique.



Nous allons maintenant présenter le projet dans sa globalité, en commençant par le contexte et les problématiques générales. Nous décrivons ensuite les premiers travaux réalisés sur Space :

- la partie serveur pour le travail coopératif (Odesca/Scenic)
- la partie interface, SpIn¹, avec la proposition d'organisation spatiale de l'environnement de travail virtuel.

Vient ensuite la présentation de mon sujet, les thèmes abordés et la description des problématiques.

3.1 Présentation du projet

Durant l'année universitaire 1993-1994, l'équipe Graphix du LIFL qui effectue ses recherches sur les images de synthèse temps réel et l'équipe Trigone qui travaille sur le travail coopératif ont mené des réflexions communes montrant qu'il serait intéressant de travailler en commun à l'élaboration de nouvelles interfaces. En effet, les interfaces actuelles, de type bureau, ne permettent pas de prendre en compte dans de bonnes conditions les tâches coopératives.

Une partie du projet a été réalisée dans le cadre d'un contrat avec le CNET intitulé «Interactions hommes/machines multidimensionnelles, multimédia et multiparties pour les services de groupe : conception et évaluation». Les principales actions qui ont été menées dans le cadre de ce contrat sont une veille technologique et un état de l'art, des expérimentations à caractère exploratoire de maquettes d'interfaces, ainsi que la conception et la réalisation d'un prototype d'interface tridimensionnelle et multimédia pour les services de groupe.

Le projet est également soutenu par Ganymède, un pôle régional de recherche sur la coopération. Ce projet réunit des équipes de recherches de la région Nord Pas de Calais autour du thème de recherche «Communication Avancée», et plus particulièrement le Travail Coopératif assisté par Ordinateur. Les laboratoires travaillent sur beaucoup de thèmes différents, des moyens informatiques aux facteurs humains, et en particulier :

- les domaines très techniques des réseaux de communication, de leurs protocoles et de leur normalisation,

1. SpIn : Space Interface. Le projet SpIn regroupe l'ensemble des travaux sur la partie «visible» de Space, c'est-à-dire sur les problèmes de visualisation, d'interaction homme-machine et homme-machine-homme.

- les outils informatiques permettant la coopération et la coordination (objets et multi-agents, supports de groupes, workflow, ...),
- le multimédia et les nouveaux supports de communication,
- les interfaces homme-machines,
- les sciences humaines et cognitives,
- les sciences sociales et l'analyse des organisations.

Enfin, une coopération s'est établie entre l'équipe Graphix du LIFL et la Télévirtualité, équipe de recherche de l'INA. Cette dernière apportait ses connaissances sur la création et l'animation de clones, que les instigateurs du projet Space s'engageaient à intégrer à leur environnement pour tester leur apport dans des tâches de travail coopératif.

3.2 Space

Nous présentons ici le projet Space, ses protagonistes, les travaux qu'il a suscité. Les travaux réalisés sur SpIn en dehors de cette thèse y sont également décrits.

3.2.1 Objectifs

Le projet SPACE consiste à définir un modèle d'interface tridimensionnelle pour le travail coopératif synchrone. Cet environnement s'applique à des réunions de petits groupes. Les utilisateurs sont placés devant leur machine, à distance, et reliés par réseau informatique. Il s'agit de recréer les conditions d'une réunion réelle en mettant à disposition de l'utilisateur suffisamment d'informations sur l'activité pour que le travail se déroule correctement, l'environnement devenant transparent. Space propose à chaque utilisateur d'intervenir dans l'activité au moyen d'un terminal personnel de communication : un poste individuel (station de travail) faisant tourner SpIn.

Il s'agit d'analyser les méthodes et les moyens de fonctionnement des réunions en situation réelle pour reproduire les éléments les plus indispensables dans un environnement virtuel. Nous devons trouver les services qui seront disponibles pour chaque utilisateur, les moyens de communications dont ils disposeront, et les métaphores de l'environnement virtuel par rapport au réel.

3.2.2 Etude du domaine d'application

L'objectif est de médiatiser des réunions de travail de petits groupes, autour de documents partagés, visant à leur présentation, la modification d'un document ou la production d'un nouveau. Il peut s'agir de réunions administratives, de bureau d'étude ou encore d'études de produits.

Les applications dérivées consistent en des activités liées à la formation, l'assistance à distance ou la vente.

3.2.3 Les contraintes

L'utilisateur potentiel est dans un environnement de travail de type «bureautique», c'est-à-dire organisé typiquement autour d'un bureau de travail, d'outils de communications standards (téléphone, etc) et de collègues. Il doit avoir la possibilité de communiquer brièvement avec cet environnement tout en étant en télé-réunion. Il ne doit donc pas être coupé de son environnement de travail réel.

Le système étant destiné à une large population d'utilisateurs potentiels (les gens susceptibles de tenir des réunions dans leur travail), il devra rester à des coûts modérés.

3.2.4 Les actions du monde réel

Les interlocuteurs se voient, ils peuvent discuter, sans interaction directe. En effet, les gens ne rentrent pas en contact direct dans ce type de situation.

Nous nous intéressons à la partie «réunion» proprement dite, la partie où tous les protagonistes sont réunis ensemble. C'est-à-dire la partie synchrone, par opposition aux outils asynchrones qui permettent de programmer une réunion entre les participants, de créer des compte-rendus, de les diffuser, etc.

Les phases

Le schéma typique d'une réunion est le suivant :

- ouverture de la réunion et distribution des documents,
- présentation de l'ordre du jour,
- débat : cette phase peut se décomposer, on peut y trouver des exposés ou des phases de discussions fédérées par une méthode particulière (un brainstorming par exemple),
- compte-rendu par le responsable de réunion.

Les tâches à accomplir

On peut assimiler la tâche principale à la diffusion d'informations et/ou à la création d'un consensus autour d'un problème posé au groupe, i.e. à de la prise de décisions.

Les tâches réalisées par les participants au cours de la réunion sont principalement :

- visualiser des documents, à différentes échelles (de la plus large, comme la projection d'un document sur un écran, à la plus petite, comme la présence du document dans les mains de chaque personne),
- expliquer,
- discuter,
- prendre une décision,
- produire un document.

On peut déduire de cette décomposition des réunions les principaux points à reproduire pour créer un système de communication virtuel à la hauteur du réel. On constate d'ailleurs que ces tâches utilisent en permanence la communication entre les participants.

3.2.5 Les fonctions à réaliser

Dans la réalité, les participants ont la sensation permanente de la présence et de l'activité des autres : ils peuvent constater d'un simple coup d'oeil leur intérêt au débat courant, ou le travail qu'ils exécutent. Ils possèdent une vision globale de l'activité.

On peut ainsi parler pour un système de travail synchrone à plusieurs de la nécessité première d'un système de coopération virtuel, qui implique la communication et l'information visuelle sur l'activité.

Les fonctions primordiales que devra remplir l'environnement de travail virtuel sont les suivantes :

- le partage de documents,
- explication sur les documents (visualisation et désignation sur les documents),
- visualisation des interlocuteurs (pour la téléprésence, l'identification),
- communication vocale avec les interlocuteurs.

3.3 Odesca / Scenic

Trois entités distinctes ont été identifiées dans SPACE. La première est un serveur d'activité centralisé, ODESCA. La seconde, répliquée sur chaque machine cliente, est l'interface visible par l'utilisateur : c'est SpIn. Entre les deux vient se placer ScenIC, dont là aussi une instance existe par machine.

ODESCA est une version remise à jour du serveur d'activité utilisé dans le projet Co-Learn [Viéville 95]. Il centralise les informations sur l'activité (identités et rôles des participants, droits...) et permet la persistance de l'activité et des documents en fin de session. La gestion de l'activité est assurée par le module, appelé ScenIC, chargé de faire le lien entre chaque instance de SpIn et le serveur d'activité ODESCA (Figure 3.1). ScenIC est chargé de fournir à SpIn la partie chargée de la présentation et de l'interaction, une description de l'activité (métaphore de la mise en scène) éventuellement adaptée à chaque utilisateur. Cette description est respectée par l'interface au mieux de ses possibilités, détachant ainsi le noyau fonctionnel collaboratif de la présentation. La description indique un type de décor, quels sont les objets visibles, quelles sont les contraintes spatiales (par exemple, deux objets doivent rester côte à côte, tous les objets ont une place fixe...), quels sont les droits de l'utilisateur sur chaque objet (éditer, sauvegarder...), etc. Elle concerne aussi bien l'environnement sonore que visuel. Elle s'effectue par envois de messages et peut évoluer au fur et à mesure que l'activité se déroule. SpIn construit la scène représentant l'activité sur laquelle l'utilisateur vient agir, ses actions étant répercutées via ScenIC et ODESCA vers les autres participants.

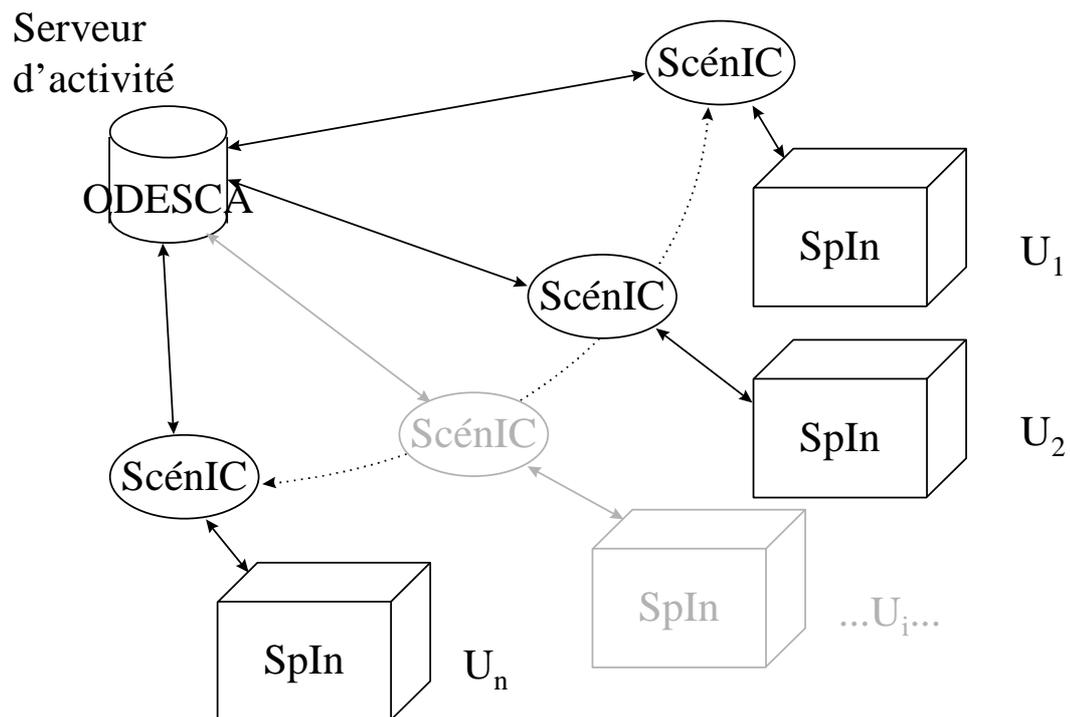


Figure 3.1 architecture du projet SPACE

3.3.1 Odesca

Les informations suivantes sur ODESCA et Scenic sont adaptées d'un document de travail de nos collaborateurs du laboratoire Trigone [Hoogstoel 97]. ODESCA gère l'activité. Ses rôles principaux sont au nombre de quatre :

- Connaissance des participants, des documents et des outils,
- Connaissance du découpage de la tâche,
- Connaissance des droits de chacun,
- Maintenance de la persistance.

3.3.2 Scenic

Scenic est l'interface entre SpIn et ODESCA [Carlier 98]. Il contient une représentation abstraite de la scène et de la tâche qui s'y déroule. Par cette représentation abstraite, Scenic est par conséquent indépendant du rendu visuel exécuté par SpIn ; il peut supporter différentes représentations visuelles pour une même activité. Par contre, il est fortement dépendant d'ODESCA puisque c'est de ce dernier que proviennent des informations essentielles pour l'activité.

Ses rôles principaux sont au nombre de cinq :

- représentation abstraite de la scène et de la tâche associée,
- stockage des contraintes spatio-temporelles,

- point de passage obligatoire entre SpIn et ODESCA,
- stockage des propriétés des différents composants de la tâche,
- gestion améliorée de la qualité de service.

3.4 Spin

Les travaux autour de SpIn concernent particulièrement la partie synchrone de l'activité, la réunion à proprement parler. Les aspects les plus asynchrones du travail coopératif en réunion ont été volontairement évités : nous en avons tenu compte dans la conception, mais nous n'avons pas étudié les problèmes de découpage en phases de l'activité, de mise en place de réunion, de persistance d'une réunion à l'autre, etc (aspect géré par le serveur de travail coopératif). Nous nous sommes intéressés uniquement à la réunion en elle-même.

SpIn est donc un modèle d'interface 3D pour le travail coopératif synchrone. L'étude des possibilités des environnements de travail coopératif sur ordinateur nous a orienté vers les choix suivants :

1. choisir la métaphore de pièce - on pourrait parler plus particulièrement de bureau de réunion,
2. construire une représentation tridimensionnelle de celle-ci à l'écran,
3. pouvoir prendre en compte simultanément plusieurs flux vidéos,
4. utiliser une spatialisation du son pour permettre un meilleur repérage dans l'espace,
5. proposer une interactivité temps réel (fréquence minimum de 10 Hz),
6. orienter l'interface sur l'action à accomplir,
7. concevoir le système d'exploitation permettant de mettre en oeuvre ces concepts.

Après avoir spécifié les objectifs de Spin, nous allons maintenant décrire les premiers travaux autour de cette interface [Saugis 98], qui ont établi un modèle d'organisation spatiale, et jeté les bases de la coopération.

3.4.1 Les métaphores visuelles liées au travail coopératif : les recommandations

Les principaux services que doit remplir un système de travail coopératif ont été tirés d'expériences sur des médiaspaces [Ishii 92][Okada 94][Tanaka 96]. Ils ont été adaptés à la problématique de Spin, mais le contexte de certaines expériences, particulièrement MAJIC, reste le même et on peut y trouver ces éléments indispensables qui sont :

- le contact visuel : lorsque l'on regarde un interlocuteur, celui-ci en a conscience. Si deux personnes peuvent se regarder dans les yeux, ce contact visuel est préservé. L'idéal étant que la représentation de la personne regarde effectivement l'objet concerné (dans l'espace virtuel), plutôt que d'utiliser des liens symboliques (flèches, etc),
- la conscience du regard : quand un utilisateur regarde quelqu'un ou quelque chose, les autres le savent en suivant la direction de son regard,
- la continuité : les personnes et les objets doivent être placés dans un même espace si l'on veut pouvoir montrer les interactions entre personnes et documents,

- assurer une qualité du canal sonore (pas de «trou»), et faire en sorte que la voix et l'image d'une personne proviennent du même endroit de l'espace, et soient synchronisées si sa représentation visuelle est animée,
- favoriser la perception directe en créant une interface explicite plutôt qu'en forçant l'utilisateur à recréer mentalement les liens qui existent entre différents objets, et donc augmenter sa charge cognitive. Ceci sans pour autant tout reproduire, ce qui surchargerait l'interface et noierait l'information pertinente. Il faut reproduire en simplifiant,
- augmenter la densité d'information à l'écran en utilisant la 3D, et permettre ainsi de conserver l'ensemble de toutes les informations visibles afin que l'on puisse suivre dans de bonnes conditions le déroulement de l'activité (éviter les occlusions comme dans beaucoup d'interfaces 2D multi-fenêtrées),
- utiliser l'animation temps réel pour montrer les changements dans l'interface, en évitant les sauts brusques d'un état à un autre d'un objet graphique, qui rendent difficilement compréhensibles ces changements, en particulier dans une interface coopérative où l'on n'est pas seul à agir,
- placer tous les tenants de l'activité dans une scène globale afin de pouvoir représenter les interactions entre les différentes composantes,
- pouvoir connaître le centre d'intérêt de chacun, sans pour autant limiter le système en donnant une même vue à tous.

Ces recommandations ont amené un premier travail de mise en oeuvre sur Spin qui a permis de fournir les premiers éléments essentiels à une interface 3D pour le TCAO synchrone, décrits ci-après.

3.4.2 L'organisation spatiale

Le premier travail sur Spin a consisté à trouver une organisation spatiale des documents qui reprenne les points énoncés ci-dessus et en particulier la visibilité, afin de conserver une vue globale en permanence sur l'activité, et l'interactivité, pour pouvoir naviguer facilement d'un document à l'autre.

La métaphore de l'interface est la table de réunion, qui correspond à une seule et même activité. Nous y opérons un découpage spatial et sémantique en deux parties. La première partie est un bandeau où l'on place les acteurs et les documents partagés (voir ci-dessous). La seconde correspond à l'espace intérieur de ce bandeau - c'est un espace de travail particulier (pour un document occupant une position centrale).



Figure 3.2 *deux niveaux de distorsion dans une première version de Spin*

Nous plaçons donc les acteurs et les documents partagés (2D ou 3D) autour d'une table, sur des «murs». Par commodité, on dispose un document par mur. Pour en représenter autant que l'on veut à la fois, on les répartit suivant un demi-cercle afin de garder l'ensemble des documents visibles. L'utilisateur peut recentrer son point d'intérêt en tournant ce bandeau, action assimilable à celle de tourner la tête dans la réalité. Cette répartition est naturellement uniforme et simule une situation réelle de réunion où plusieurs personnes sont assises autour d'une table. Cela permet d'insérer tout nouvel arrivant (ou un nouveau document) sans qu'il y ait d'a priori sur l'espace qu'il occupe par rapport aux autres interlocuteurs présents dans la scène.

Distorsion

Si le nombre d'objets autour de la table augmente, ils deviendront tous trop fins pour que l'on puisse les utiliser. On définit donc une notion de focus, qui met en valeur le document dans la zone principale de vision, au centre de l'écran. Cela impose une déformation des autres documents sur les côtés (Figure 3.2).

Notre déformation a lieu dans un sous-espace (parallélépipède rectangle), uniquement en abscisse. On considère qu'un point d'abscisse nulle sera au milieu de la boîte. A un point d'abscisse x positif ou nul (la déformation est symétrique par rapport au plan $x=0$), on applique la formule (Figure 3.3) suivante :

$$x' = 1 - (1 - x)^\alpha$$

$$0 \leq x \leq 1$$

Figure 3.3 *formule de déformation*

où x est la demi-largeur de la boîte et α le facteur de déformation. Avec $\alpha=1$, la fonction est l'identité. Pour $\alpha>1$, les points se rapprocheront du bord, avec pour résultat d'étirer les objets centraux, tandis que ceux situés près du bord seront tassés vers l'extérieur. Pour faire varier la déformation (lorsque l'on change le facteur de zoom), nous multiplions α par une constante, plutôt

que de la lui ajouter. Cela permet d'avoir des variations de déformations plus linéaires (et donc une animation plus fluide).

Le tout Visible

Grâce à cette déformation, les parties importantes sont donc parfaitement lisibles, et le reste conserve une place dans l'environnement. Lorsque l'on tourne le bandeau, ce qui disparaît d'un côté réapparaît immédiatement de l'autre (les autres éléments à l'intérieur de la scène ne bougent pas). Cela permet à l'utilisateur de toujours tout percevoir dans l'interface. Il est en effet essentiel en TCAO que chaque artefact et chaque acteur impliqué dans une tâche coopérative soit présent à l'écran.

Un point de focus

La distorsion amène l'utilisateur à placer naturellement au centre, la partie du bandeau avec laquelle il travaille. Cette information sur le centre d'intérêt permet de reproduire, sans dispositif spécial, la sensation de contact visuel et de direction du regard d'un intervenant dans les interfaces de ses interlocuteurs.

En dehors du bandeau, l'espace intérieur est un espace de travail, où l'utilisateur peut apporter (agrandir) le document sur lequel il travaille, y placer des documents (2D ou 3D) pour les éditer ou les manipuler. La navigation de l'utilisateur est donc réduite virtuellement à une rotation (bandeau) et à un déplacement en profondeur (zoom du point de focus).

3.4.3 La téléprésence

Afin d'identifier les interlocuteurs, des avatars ont été placés autour de la table. Ces avatars (Figure 3.4) correspondaient à la première étape d'un système coopératif, ils permettent de savoir qui est présent.

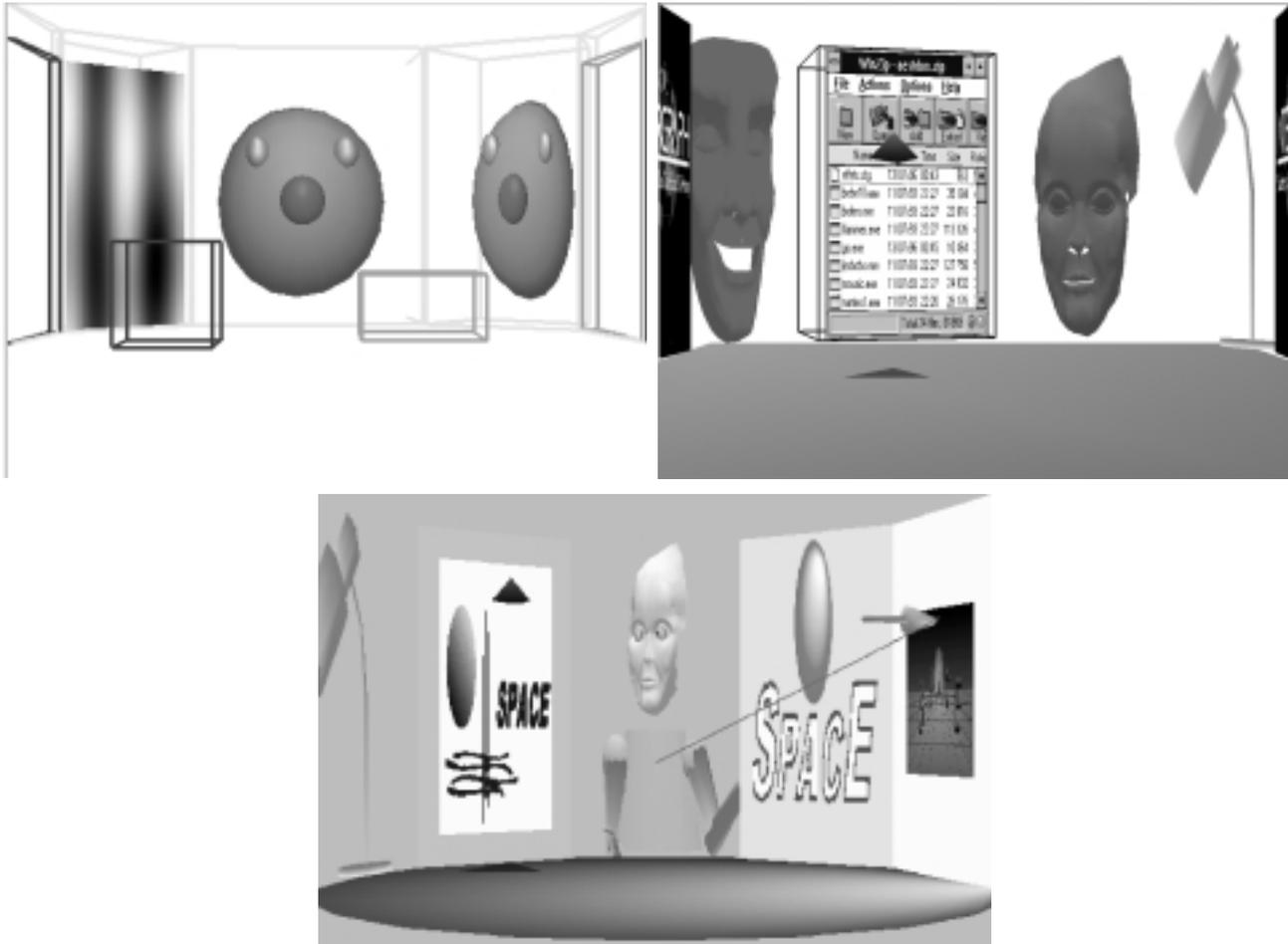


Figure 3.4 *les avatars dans Spin : première (h.g.), seconde (h.d.) et troisième (bas) générations*

3.4.4 La navigation

Afin de conserver en permanence une vision globale sur la scène, on n'autorise pas réellement la navigation. Elle a été adaptée pour faciliter le passage d'un document à un autre, sans pour autant désorienter l'utilisateur, ou lui faire perdre du temps dans des déplacements inutiles. C'est l'organisation même de l'espace qui permet ceci en autorisant la manipulation du bandeau, et le zoom du document au centre.

Ce déplacement des documents peut être assimilé à la rotation de la tête pour regarder une série de documents placés autour d'un intervenant. La déformation sur les bords simule la vision périphérique de l'homme, et permet de recentrer automatiquement l'attention de l'utilisateur sur le document placé au centre du bandeau.

3.5 Mon sujet

Il s'agit de mettre en oeuvre les fondations qui permettront d'interagir dans SpIn. En particulier, nous voulons pouvoir faire correctement de l'interaction et de la communication homme-machine, avant d'entreprendre l'aspect homme-machine-homme (communication à travers les clones).

L'objectif est donc de définir un système global d'interaction dans Spin, qui lui permette de devenir un système interactif homogène [Dumas 98] [Dumas 99].

Cela consiste en une étude des possibilités d'interaction dans un environnement 3D, de manière générale dans un premier temps, puis en se limitant au cadre des interfaces 3D pour le TCAO synchrone. En ne perdant pas de vue cet objectif global, les étapes de développement et de recherche ont été :

- de construire un système interactif autour de notre modèle d'organisation spatiale décrit plus haut. Si la disposition des acteurs et des documents ou la navigation de l'un à l'autre sont essentiels, l'étape suivante consiste à ce que l'utilisateur ne soit plus un spectateur passif mais puisse jouer son rôle dans l'activité coopérative. Pour qu'il puisse agir, il faut rendre les objets, les documents dynamiques, créer les éléments de l'interface qui pourront lui permettre d'accéder aux ressources et de les manipuler. Ceci implique une partie sous-jacente de l'interface qui établit la communication bas-niveau entre tous les éléments de l'environnement. Pour cette tâche, les systèmes de gestion d'environnement 2D ne sont plus adaptés, ni les périphériques, ni les systèmes à fenêtre. Il convient donc de repenser une architecture complète de l'environnement, qui prenne en compte les nouvelles données du problème (l'utilisateur et les interlocuteurs) au sein d'une même interface.

- de proposer à l'utilisateur un système d'interaction adapté à ses possibilités et aux problématiques du travail coopératif. Il devra s'intégrer de façon homogène dans l'ensemble de l'interface. L'idéal étant que l'utilisateur puisse s'abstraire de l'outil et ait l'impression d'être face à ses interlocuteurs, comme dans une situation réelle. Il faut donc allier les contraintes de l'utilisateur (i.e. les problèmes de perception) et les besoins de l'application (faire coopérer plusieurs personnes entre elles, les faire agir sur des documents).

- d'intégrer les interlocuteurs à l'environnement de travail, et les animer pour permettre la communication et la coopération au sein de l'interface. On doit pouvoir montrer ce que font les autres utilisateurs. Mais pour ne pas surcharger l'interface, il faut trouver quelles actions il est essentiel de montrer et comment les représenter dans un environnement de travail virtuel 3D comme Spin.

La question posée, à terme, est de trouver l'incidence et l'utilité des clones dans une situation de travail coopératif synchrone. S'il est possible de justifier leur utilisation d'un point de vue technique, il n'en est pas de même pour l'aspect humain puisque peu d'environnements les intègrent et ont permis des observations sur leur impact sur une situation de travail coopératif. Les questions pourraient porter sur l'aspect graphique, mais la technologie nous apporte de plus en plus de réalisme dans la représentation des clones. Les vraies questions de fond portent davantage sur l'intégration de clones dans l'environnement de travail, sur ce qu'on peut leur faire faire, sur la quantité d'informations qu'ils sont capables de véhiculer, sur leur crédibilité, etc.

L'objectif est de créer un environnement qui justifie de lui-même et de façon suffisante l'emploi de clone. En effet, pour faire de la téléprésence, le support vidéo a montré d'étroites limites fonctionnelles que les améliorations technologiques changeront peu : son manque d'intégration dans un environnement informatique, la difficulté de créer des liens naturels entre une vidéo et les autres

éléments de l'interface, la difficulté que l'on a à manipuler les données (le flux vidéo) pour en extraire les données pertinentes (point de vue, etc), et surtout le manque de maîtrise que l'on a du point de vue de la caméra sur la personne (qui peut la placer comme elle le veut).

La démarche suivie, dans la première partie de cette thèse, a consisté à explorer les domaines s'intéressant aux mêmes questions que nous. On trouve du côté des aspects informatiques : la synthèse d'image temps réel, l'architecture des interfaces homme-machine (système de messages, etc) et la communication réseau. Mais les développements effectués autour de ces thèmes sont toujours motivés par une approche liée aux utilisateurs. C'est donc tout d'abord dans le domaine du travail coopératif assisté par ordinateur que l'on trouve certaines réponses, puis dans celui lié aux problèmes de perception de l'utilisateur, perception de l'espace virtuel, mais aussi perception globale de la scène, des utilisateurs et de l'activité. Les références sur ces sujets sont du domaine de la psychologie et de l'ergonomie des interfaces. Mais on peut regrouper globalement ces problématiques sous le titre d'IHM 3D.

Les questions se poseront donc par la suite dans l'ordre des niveaux d'interaction, sans perdre de vue l'objectif global (définir un modèle d'interface 3D). Elles seront traitées dans la seconde partie de cette thèse :

- la définition d'un système d'interaction bas niveau (chapitre 4) qui permette à l'utilisateur d'agir dans l'interface : il s'agit des primitives d'interaction du système 3D qui constituent la base de toutes les actions.
- l'exposé des différentes parties d'une interface 3D (chapitre 5), à savoir l'organisation globale de l'interface, la navigation (la gestion du point de vue) et les outils disponibles dans l'interface. On pourrait comparer ces deux premiers points à l'approche de X window dans les systèmes Unix : le premier point serait l'équivalent de X window lui même, et le second serait l'équivalent d'un gestionnaire de fenêtre du type Motif avec ses outils.
- la proposition d'un système d'interaction homme-machine-homme (chapitre 6), avec une description des mécanismes d'interaction d'un utilisateur avec ses interlocuteurs, présentant les avantages qu'un espace de travail virtuel peut fournir.
- le dernier chapitre sera consacré à une description fonctionnelle de SpIn et d'une de ses applications mise en oeuvre autour de l'apprentissage de la photo. Nous présenterons les résultats d'une évaluation grandeur nature de notre interface à travers cette application.

Finalement, une partie supplémentaire (en annexe) détaillera l'architecture logicielle qui a permis de développer le système d'interaction précédemment décrit.

2^{ème} Partie

Nous allons décrire maintenant notre modèle. Nous le présentons en fonction des niveaux d'interactions dans l'interface. La figure ci-dessous présente les différents niveaux d'interaction d'une interface de travail coopératif synchrone.

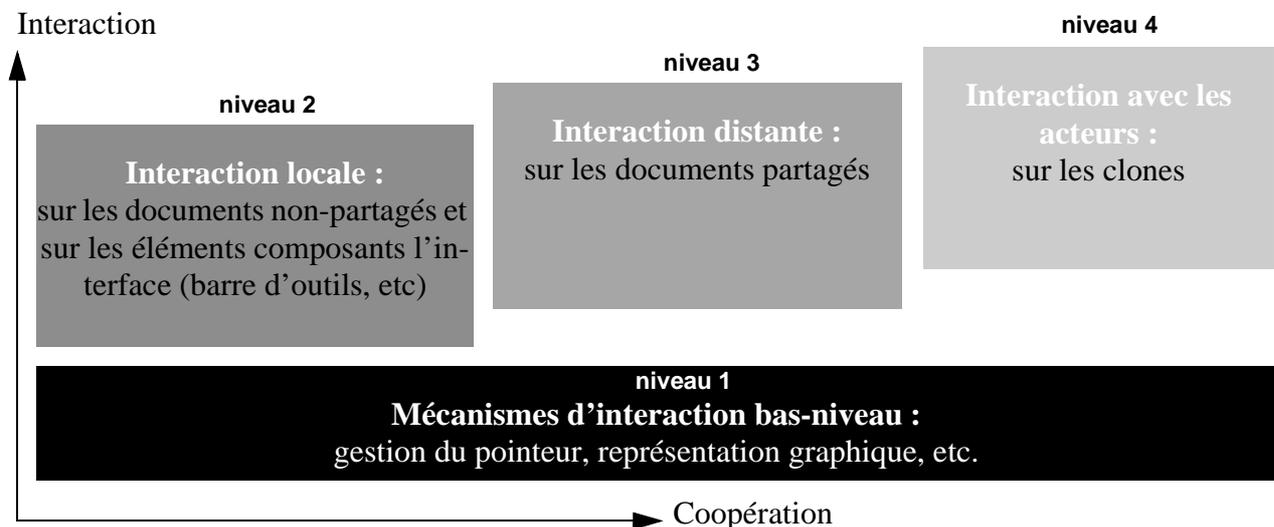


Figure 1 *différents niveaux d'interaction dans une interface de travail coopératif synchrone*

Les trois premiers chapitres de cette seconde partie détaillent ces quatre niveaux d'interaction.

Le premier niveau, exposé dans le chapitre 4, correspond aux mécanismes d'interaction de base de l'interface, en particulier comment désigner et sélectionner un objet. Ce doit être un ensemble cohérent, qui décrit complètement les mécanismes d'interaction, afin que l'utilisateur ne soit pas confronté à différents modes d'interaction d'une application à une autre pour effectuer la même tâche. Il faut qu'il soit également complet et extensible, afin de ne pas brider et pénaliser les possibilités de l'interface. Cette couche de bas niveau sert à construire toutes les autres couches du modèle d'interaction.

Le second niveau, exposé dans le chapitre 5, concerne l'interaction locale, qui correspond aux mécanismes de base de l'interface, c'est-à-dire comment accéder à un document, comment agir dessus. Autrement dit, il s'agit de tous les outils permettant d'exécuter des commandes sur les documents (les menus, les barres d'outils, etc). Ce niveau d'interaction comprend aussi la

visualisation et l'édition des documents privés, c'est-à-dire qui ne concernent pas l'activité du groupe.

Le troisième et le quatrième niveaux sont exposés dans le chapitre 6. Le troisième niveau concerne l'interaction à distance, qui correspond aux actions de manipulation de documents partagés, et à la façon dont ces actions sont montrées aux autres participants. Comme par exemple les modifications d'un document partagé qui sont reportées à distance. On va dans ce cas voir un clone agir sur un document. Cela permet de connaître l'activité des autres utilisateurs dans la réunion. Le quatrième niveau correspond à l'interaction directe avec les acteurs, c'est-à-dire aux actions que l'on va effectuer directement avec un interlocuteur via l'interface. On agit alors avec sa représentation (son clone), à travers des outils correspondant à des actions de coopération (comme une demande d'aparté ou une prise de point de vue sur un document). Dans les autres interfaces, ce type d'action sera vu comme une interaction entre deux clones.

Ces quatre niveaux parcourent l'ensemble des possibilités d'interaction homme-machine dans un environnement de travail coopératif. Des règles de conception appartenant à ces quatre niveaux pourront constituer un modèle d'interaction si elles forment un ensemble complet, cohérent et homogène, permettant à l'interface de répondre au mieux à toutes les situations possibles. Nous suivons cet objectif en présentant notre modèle dans les chapitres suivants. Nos propositions sont fondées sur des analyses et sur des évaluations. L'ensemble est relativement spécialisé sur SpIn, mais nous verrons quelles sont les règles généralisables aux environnements tridimensionnels mono ou multi-utilisateurs.

Le dernier chapitre de cette partie décrit SpIn de façon fonctionnelle. Il permet de récapituler l'ensemble des propriétés de la première version, sur laquelle une application a été développée. Cette dernière y est présentée, ainsi que les résultats de son évaluation «grandeur nature».



Notre objectif est de donner les moyens à un utilisateur d'agir dans une interface 3D composée de documents. Son activité repose sur le travail autour de ces documents. L'environnement proposé est construit explicitement pour recevoir ce type d'activité, et permet à l'utilisateur de se concentrer dessus. Nous allons exposer la démarche qui nous a permis de spécifier un modèle d'interaction, en tenant compte des contraintes de base de SpIn, à qui ce modèle est destiné :

- se placer dans un environnement «bureautique» (coût des matériels, contact avec l'extérieur),
- être adapté à l'aspect travail coopératif de l'activité de l'utilisateur,
- utiliser un environnement de travail tridimensionnel.

Notre but est donc de définir un modèle d'interaction composé d'un ensemble cohérent de règles d'interaction. Nous nous basons sur une démarche de propositions, qui s'appuie sur des raisonnements et des expérimentations.

Notre positionnement nous permet une approche sans contrainte du problème. La conception de notre modèle d'interface est dégagée des limitations des interfaces et des environnements fenêtrés existants. Ceci explique que notre approche ait démarré à partir du niveau de connaissance zéro, c'est-à-dire en remettant en cause tous nos acquis en matière d'interface 2D. Nous nous basons sur l'observation de l'utilisateur, et le moins possible sur une transcription directe des habitudes de la 2D vers la 3D. Nous espérons ainsi pouvoir résoudre certains des problèmes d'interaction par des solutions propres à la 3D.

Les objectifs correspondent aux critères de base d'une interface homme-machine, à savoir être intuitive, et correspondre aux besoins de l'utilisateur. Le critère de prise en main rapide et simple de ce type d'application pourra en effet être déterminant à l'avenir, dans la mesure où les utilisateurs potentiels seront issus de nombreux milieux (l'industrie, le secteur tertiaire, etc), et pas particulièrement de l'informatique.

Nous allons commencer par regarder les applications ayant le même type d'approche, c'est-à-dire possédant un système d'interaction original, basé sur l'analyse de la tâche plutôt que sur les habitudes des interfaces fenêtrées (Windows, Finder Mac, X-window, etc). Dans ces applications, on trouve surtout des programmes liés à la 3D (CAO, etc). Leurs interfaces sont souvent originales (Bryce, SoftImage, etc). Notre objectif n'est pas d'explorer les chemins empruntés par ces logiciels, mais de limiter notre propos à l'observation de certains. Les IHM les plus originales sont souvent

celles des simulateurs, grâce à leur moyen d'interaction correspondant à une réalité. Elles ne cherchent cependant pas à être les plus intuitives possibles mais les plus réalistes (par rapport au phénomène à simuler), les limitations ergonomiques avec l'utilisateur d'un simulateur parfait étant celles du phénomène simulé et non celles de l'interaction.

Pour ce qui est de la navigation dans un monde virtuel, les applications comme *Dive*, *Cosmo Player* ou les *Doom-likes*¹ proposent chacune un mode différent, adapté au besoin de l'application (maniabilité, précision du point de vue, vitesse de déplacement, etc).

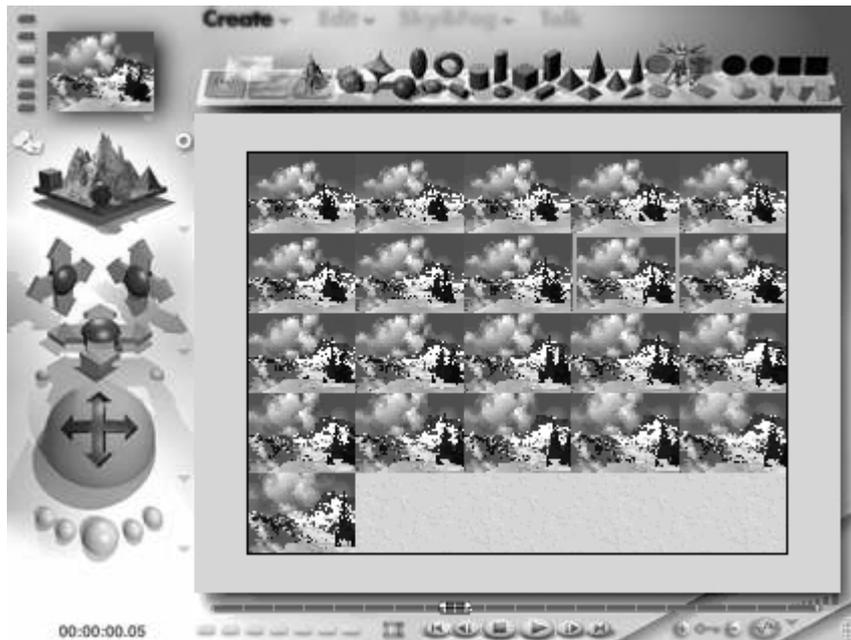


Figure 4.1 *l'interface de Bryce 4 de MetaTools. Les widgets situés sur la gauche permettent de manipuler à la souris le modèle 3D d'un paysage.*

Pour ce qui est de l'interaction dans les interfaces 3D (au sens saisie et manipulation d'objets), *Cosmo Worlds* (Figure 2.5) propose des widgets complètement différents de ceux de *Bryce* (Figure 4.1), un éditeur à l'interface originale permettant de créer des paysages en images de synthèse. L'interface 3D de *Dive* développe une approche fondée sur l'interaction directe à base de gants de données, et les simulateurs médicaux (Figure 4.2) du laboratoire de neurovisualisation de l'université de Virginie permettent d'interagir directement sur des organes en plastique, mais bien réels, «représentés» simultanément à l'écran.

1. on dénomme ainsi les jeux ressemblant à Doom, où l'on se déplace dans des mondes virtuels avec des périphériques classiques.

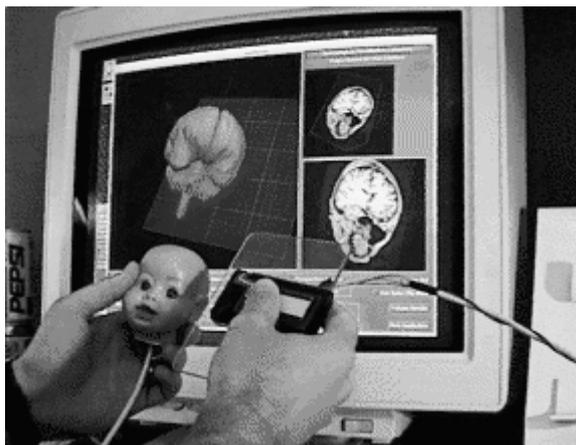


Figure 4.2 *périphériques spécifiques pour la visualisation neurochirurgicale*

Chacune de ces applications propose un mode d'interaction original, qui sont radicalement différents les uns des autres, car elles ont leurs propres contraintes, que la troisième dimension permet d'adapter au mieux sans retomber trop souvent sur un mode d'interaction lié à la forme de l'affichage. Il pourrait donc s'avérer à l'avenir très difficile de proposer un modèle d'interaction générique, adapté à toutes les situations. On peut d'ailleurs remarquer que si en deux dimensions, ce système générique existe avec la souris et le système de fenêtres/menus, quelques domaines d'applications ont apporté une contribution particulière (les outils de dessin comme *Adobe PhotoShop* sont basés sur l'utilisation de palettes d'outils), et certains n'y ont pas adhéré du tout (les modelleurs 3D par exemple).

Cependant notre but est d'offrir à l'utilisateur une interface de travail générale en trois dimensions, qui doit permettre de produire et de manipuler des documents 2D et 3D. Si l'on regarde la manière dont chacune des applications préalablement citées s'est spécialisée, on trouve alors que notre objectif est relativement générique. Peu de travaux se sont attaqués directement à cet aspect «environnement de travail 3D complet» [Roberts 98].

Nous verrons par la suite dans quelle mesure nous avons dû spécialiser notre interface aux applications du travail coopératif synchrone supportées par Spin, et dans quelle mesure notre proposition est extensible.

Il s'agit de donner à l'utilisateur des moyens pour agir sur les documents qui composent son espace de travail. Notre approche de l'organisation de cet espace est en trois dimensions comme nous l'avons vu précédemment. Pour agir sur les composants de l'interface, il nous faut un système d'interaction adapté à un espace 3D.

Dans la mesure où les environnements de travail coopératif synchrone sont basés sur la communication, les interlocuteurs discutent beaucoup. Il paraît dès lors difficile de pouvoir utiliser des commandes vocales. Il faudrait alors explicitement demander à l'utilisateur de rentrer et de sortir du mode de commande vocale. Ce mode d'interaction ne s'intégrerait donc pas naturellement. Nous avons appuyé nos travaux sur le gestuel plutôt que sur le vocal.

4.1 L'essentiel du geste : passage du monde réel au virtuel

L'environnement matériel de SpIn ne permet pas encore de prendre en compte la fonction ergotique du geste [Cadoz 94b]. Quant à la fonction sémiotique, elle fait l'objet d'une étude dans le chapitre 6. C'est pourquoi nous avons préféré réaliser une étude empirique des gestes communs de la réalité, non pas pour juger de leur efficacité ou de leur sens, mais uniquement pour constater la façon dont une personne réalise globalement certaines actions simples (planter un clou avec un marteau, déplacer une souris avec la main, etc.). On s'intéresse donc aux gestes qui permettent à une personne seule de réaliser une action élémentaire. Nous les appelons ici les gestes d'action, pour les opposer globalement aux gestes d'expression (la fonction sémiotique). On peut observer deux niveaux. A un degré assez fin, dans le référentiel de la personne, on regarde ses moindres gestes (la personne agissant sur son objet de travail). Et à un degré plus élevé, dans le référentiel de l'environnement de travail, on observe l'espace dont elle dispose et dans lequel elle évolue (la personne dans son cadre de travail). Il convient pour chacune de ces deux échelles de déterminer ce qui sera utile ou non dans un monde virtuel.

4.1.1 Décomposition du geste d'action

Nous voulons d'abord déterminer ce qui est prédominant dans l'action d'un utilisateur. On peut décomposer une action sous cette forme :

$$\text{Action} = \text{Intention} + \text{Geste}$$

Et le geste se divise en plusieurs phases : Approche / Action / Réaction (Figure 4.3). La main se rapproche de l'objet dans l'espace, elle agit sur cet objet, au besoin en le saisissant, et il se produit un effet, lié à l'intention de l'utilisateur.

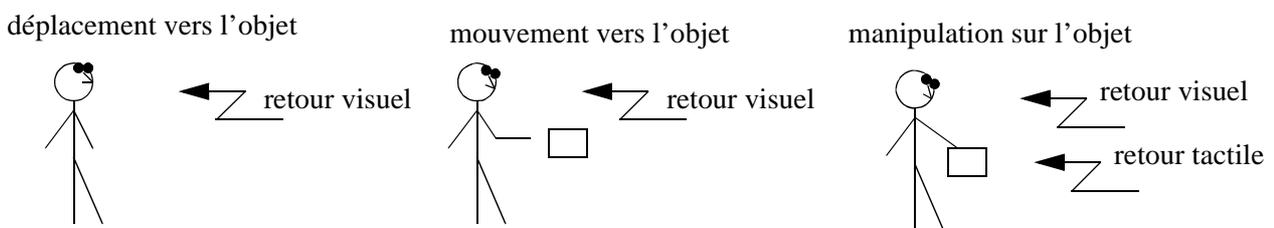


Figure 4.3 *exemple de décomposition d'une action sur un objet simple et appréhensible, et évaluation de l'effet de l'action*

Nous avons commencé par une approche empirique, qui consiste à répertorier des actions de la vie courante, en situation de travail ou non, et à les décomposer : on détaille les mouvements en action/gestes élémentaires, le nombre de mains, les sens utilisés, etc (Figure 4.4).

vous devez avoir : 3 comportements et : nouveau comportement

action	temps	position main	décomposition	réel			virtuel				
				sens	outils	nbr mains	sens (motricité partiel...)	outils	nbr mains	tache rép. / fin del'action	historique
téléphone	brif		composer <i>de la main gauche</i>	contact vue (son)	<i>doigt</i>	1	vue son	<i>plus être opératoire</i>	1		oui x10
	long		communiquer	son	<i>4/5 son</i>	1	son	<i>5/6-7/8</i>	0	poser tel	ind. en temps, le, occasionnel
piano			positionner la main	vue		1	lumière sur touches, vue	<i>assistent ?</i>	1	décalage de piano	partition ?
	<i>rapide</i>	<i>pas précis</i>	frapper touche <i>selon la main</i>	contact son (vue)		1	son	<i>pour séparer les tâches de mains ?</i>	1	note/main	<i>modifier le mode de déplacement - vitesse + rapide</i>
ping pong	rapide		déplacement sur la balle	déplic personnel		1	vue	automatisé on	1	maintenir support position	
push bouton			placement <i>en fonction de la main</i>	vue	<i>bouton</i>	1	vue son			possibilité de gestuel simple	action regroupé auto del
			doigt seul pression	toucher						de base pour un déplacement	utilises
ouvrir une porte	<i>prise, appui, glissement</i>		main vers poignée	vue			son poignée, vue	<i>doigt</i>		geste simple en bq phases	commencé vocale
	<i>appui</i>	<i>pleine</i>	tenir poignée	vue, toucher	poignée	1	vue	<i>de la main et effet de la main</i>	<i>(vidéo)</i>	non macro ? automatisé	
	<i>instinctif</i>		actionner poignée	force						2 mains ? regard ?	
prendre une bouteille	<i>instinctif</i>		prendre la bouteille	force							
			main vers le corps de la bouteille <i>appui sur le corps de la bouteille</i>	vue		1					instinctif, vu de l'œil
poser une bouteille		<i>pleine</i>	pocher/lacher <i>un instant</i>	toucher effet		1	vue, vue, vue	<i>2 blocage</i>		stable, tactile ?	
mettre une cuillère dans une bouteille			placement précis <i>appui sur les bords</i>	vue	<i>cuillère ?</i>	1	vue		1 ou 2 (bouteille)		
tirer un pétale de fleur			prendre le pétale <i>prendre la fleur</i>	vue, toucher		2	vue	<i>deux mains de saisir</i>	2, voir 1	<i>à voir de l'œil - à l'œil</i>	
conduire une			tirer le pétale tourner le volant	vue, force							

Figure 4.4 une des dix pages d'étude empirique des gestes de la vie courante

Ce tableau est donné à titre indicatif sous forme d'extrait. Nous avons ensuite observé les facteurs communs afin de voir si on pouvait dégager un schéma de base pour l'interaction sur des éléments 3D virtuels.

On peut constater que :

- il y a une dynamique importante de la main, voire d'un des objets de l'action,
- chaque geste est initié par un positionnement sur la cible (via l'outil lorsqu'il y en a un),
- les actions où les deux mains agissent simultanément sont fréquentes, mais celles où elles agissent en plus de façon synchrone sont très rares,
- dans les exemples cités, on utilise fréquemment des outils (nous avons pris les actions courantes de la vie de tous les jours), qui sont en général d'une utilisation très simple, nécessitant moins d'apprentissage (outil avec une seule fonction principale, comme un marteau ou une raquette),
- chaque outil a un rôle précis,
- chaque outil est indéformable, et ne change pas d'aspect au cours de l'activité.

Si cette approche n'apporte que des éléments empiriques sur la compréhension du geste, elle a marqué la première étape de notre réflexion, et a commencé à nous faire réfléchir sur les rapports entre les gestes dans l'espace virtuel et dans l'espace réel.

4.1.2 Le geste d'action dans l'espace virtuel

Un environnement de travail coopératif tel que nous le concevons ne peut pas être basé sur des solutions immersives, en effet :

- la navigation dans l'espace de travail est limitée, nous connaissons et gérons la représentation des documents,
- l'environnement «bureautique» interdit les périphériques encombrants, difficiles à calibrer et surtout, qui coupent l'utilisateur de son environnement immédiat.

Les situations mises en place dans l'interface de travail ne sont alors que difficilement des transcriptions exactes de situations réelles. De plus, les actions que nous réalisons dans le monde réel sont pleines de phases «inutiles» dues aux contraintes de l'espace réel, c'est-à-dire de phases qui ne servent pas l'objectif de l'activité en cours. Il n'est donc pas forcément utile de vouloir tout reproduire.

Dans un environnement virtuel, il est possible de ne plus avoir à se déplacer pour saisir tel ou tel objet, mais de se téléporter directement d'une place à une autre (on réalise alors des actions impossibles dans le monde réel). Il s'agit donc de ne garder que l'essentiel des situations que nous voulons «informatiser», et de limiter, voire éliminer les gestes inutiles à l'activité en cours.

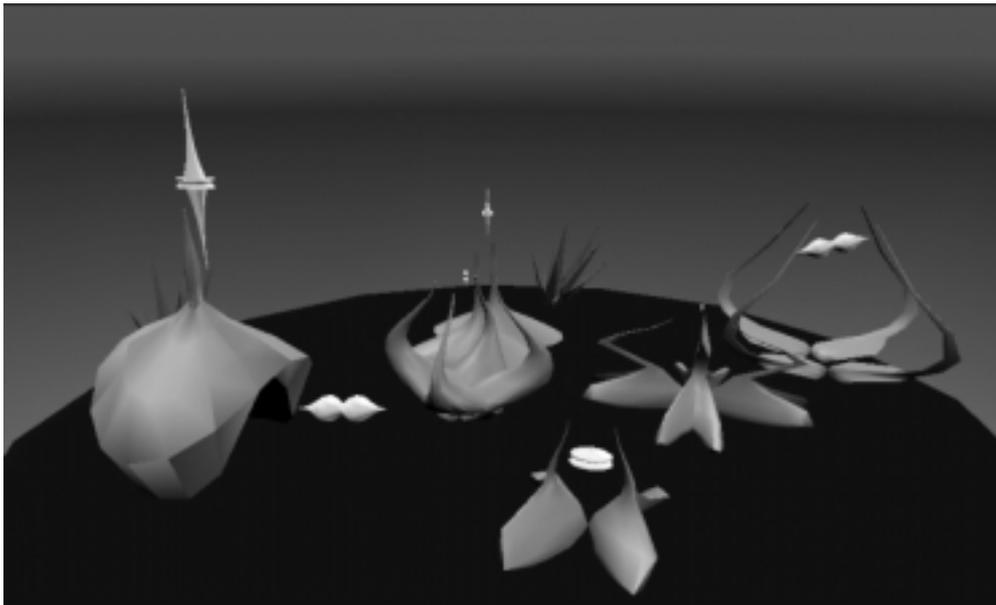


Figure 4.5 *Le monde flottant et sonore de Pagan par Jeff Harrington*

D'un autre côté, il ne s'agit pas non plus de placer l'utilisateur dans un monde sans référence, comme parfois la réalité virtuelle nous en fait imaginer (voir Figure 4.5, le monde flottant de Pagan). L'utilisateur devra pouvoir se repérer, s'orienter facilement de façon naturelle, et surtout garder une certaine confiance dans l'environnement, qui dans ce but contient des références au monde réel. Pour ce qui est de ces référents à reproduire, nous avons fait remarquer au chapitre 2 que certaines sensations ne pouvaient être reproduites que difficilement (la sensation de contact par exemple). Il s'agit donc, d'un autre côté, de modifier ou d'amplifier les autres sens (la vue, l'ouïe) pour compenser ce manque (Figure 4.6), ou de donner un nouvel indicateur (purement virtuel).

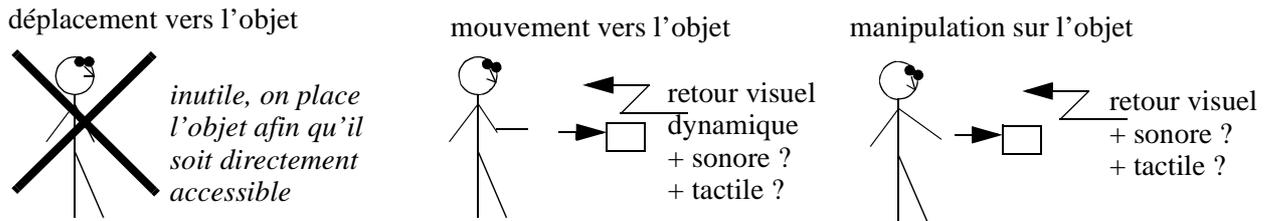


Figure 4.6 simplification de la manipulation d'un objet en virtuel

4.1.3 Bilan

Nous venons d'observer la manière dont un utilisateur agissait de façon générale sur un objet. Nous devons conserver suffisamment d'éléments de l'espace réel dans l'espace virtuel pour qu'il puisse se repérer, en optimisant les mouvements (on conserve les invariants de l'action). D'où trois règles de «simplification du réel» :

1) l'action de déplacement dans l'environnement sera évitée au profit d'une interface où l'on peut directement interagir sur tous les objets (sans navigation). Cette solution induit que le déplacement du point d'action de l'utilisateur puisse se faire dans tout l'espace virtuel à partir de ses seuls mouvements. Cette représentation impose de ne pas se voir dans l'interface, faute d'être à la bonne échelle (en dehors d'autres arguments sur l'immersion et le TCAO). Si l'on a accès directement à tous les points utiles de l'espace, la navigation sera donc limitée, dans SpIn, à la rotation et au zoom du bandeau (voir partie 3.4). Cela permet de ne pas désorienter l'utilisateur à cause de problèmes de déplacements. Il conserve un point de vue fixe sur la scène globale. Ce contexte limite le champ des applications, mais il est idéal pour SpIn et permet ainsi d'éviter à l'utilisateur de devoir se déplacer ou d'utiliser des méthodes du type Go-Go Interaction [Poupyrev 96]. En utilisant un espace délimité, on va donc faire en sorte que la zone d'action du périphérique de pointage permette de déplacer le pointeur dans la totalité de l'espace virtuel.

De plus, le contexte de l'espace 3D est un élément essentiel pour que l'utilisateur réussisse à se créer facilement une image mentale de l'espace (les murs d'une pièce par exemple). Ses dimensions et son aspect sont deux paramètres importants pour aider l'utilisateur à percevoir la profondeur. Un espace délimité mais surtout fermé devrait donc permettre une meilleure perception de la scène globale.

2) pour simplifier la saisie des objets, nous utiliserons un système simplifié pour les désigner/sélectionner. Par simplifié, nous entendons pour le mode et le moyen d'interaction, donc le(s) périphérique(s) d'entrée également. Ce qui évitera les systèmes encombrants, fragiles et compliqués (la calibration en particulier) comme les gants de données. Si on réduit les possibilités d'interaction en diminuant les capacités de la main en virtuel à une sélection d'objet, cela correspond aussi aux situations de travail. Le nombre d'actions possibles sur un document ou une partie de document donnée dans l'environnement virtuel est limité, et dépend du contexte d'utilisation.

3) On peut transformer une action éventuellement compliquée de plusieurs étapes en une seule commande, plus simple à réaliser dans l'environnement virtuel. L'inconvénient étant que l'action n'étant pas la même à réaliser que dans le monde réel, l'utilisateur devra comprendre le nouveau

moyen d'agir mis à sa disposition. C'est ici que le choix de primitives d'interaction simples et intuitives est essentiel.

4.2 Désignation

Si l'on observe les actions que l'on effectue dans un espace de travail réel ou virtuel, la désignation occupe une large part. En effet toute activité sur un document commence par la saisie de celui-ci pour le monde réel, et par son corollaire dans un monde virtuel : l'utilisateur y déplace un pointeur vers le document. Le pointeur est le point d'action de l'utilisateur dans l'environnement virtuel, il est corrélé aux périphériques d'entrée et permet d'agir dans l'interface.

La désignation est à la base de toute action dans un environnement de travail, l'utilisateur doit donc pouvoir désigner facilement et rapidement. Elle est également importante en tant que geste, car elle dévoile en partie l'intention de l'utilisateur : quand par exemple ce dernier va saisir un objet pour agir dessus. Dans un environnement de travail virtuel cela revient à le sélectionner pour effectuer ensuite des actions dessus. On ne prend donc pas le document en virtuel comme en réel, sauf dans le cas d'un environnement tri-dimensionnel avec gant de données. Dans la réalité, en prenant un document pour y travailler, on effectue un déplacement pour se recentrer sur l'objet, puis on le saisit pour porter son attention dessus. Dans les interfaces de travail actuelles, l'action de sélection simple de l'objet est plus économique que dans la réalité. Elle nécessite cependant un apprentissage, assez aisé, pour savoir placer un pointeur sur son centre d'intérêt.

4.2.1 Choix du périphérique

Pour une meilleure efficacité, l'idéal est de conserver un mouvement naturel. Notre objectif sera d'obtenir le plus de corrélations possibles entre l'action réelle et virtuelle pour ce qui est du mouvement. Dans ces conditions, on peut faire les remarques suivantes Table 4.1 sur les interfaces 2D dans leur utilisation courante avec les périphériques 2D les plus significatifs :

périphérique	inconvénients	avantages
souris	déplacement de la main dans un plan orthogonal au plan de l'écran (et donc du pointeur)	découplage du mouvement de la main (dans un plan horizontal) et du pointeur de la souris (dans le plan vertical de l'écran)
tablette tactile transparente	la main masque une partie de l'écran une fatigue due aux mouvements de la main	la main se déplace dans le sens et avec le pointeur (dans le même plan) pas de dispositif à déplacer
glidepoint	décorrélacion du mouvement de la main et du pointeur position difficile pour tenir le dispositif	peu de mouvement de la main (périphérique isométrique)

Table 4.1 *désignation dans les interfaces 2D : exemples les plus courants*

Ce tableau nous montre que la souris n'est pas forcément la plus adaptée, et nous ramène à la remarque du chapitre 2 sur le fait que les périphériques sont plus ou moins proprioceptifs, et ce en fonction de leur mode d'utilisation.

On peut, par exemple, penser objectivement qu'une tablette graphique transparente posée sur un écran est plus proprioceptive que la souris sur des tâches de manipulation d'objets : le déplacement du bras est celui du pointeur ou de l'objet à l'écran (la souris ne se déplace pas dans le même plan que l'écran), le doigt vient pointer directement sur l'objet, la souris est déplacée avec la main, alors que le pointage avec le bout du doigt possède plus de précision dans le mouvement. La souris a elle aussi ses avantages, mais ces remarques montrent que son utilisation est dans ce cas moins intuitive, car l'effort cognitif de compréhension des relations entre les mouvements à effectuer et les objets à manipuler seront plus importants pour un adulte novice qu'en utilisant la tablette (la différence de performance risque cependant d'être mineure dans cet exemple, car d'autres facteurs interviennent, l'utilisateur risque en effet de corriger automatiquement les déplacements vers une cible avec l'un ou l'autre des périphériques).

Il s'agit donc de prendre en compte pour le choix d'un périphérique adapté, en plus des critères techniques, des critères liés à la perception de l'utilisateur, à ses capacités motrices et sa compréhension des déplacements spatiaux. En particulier, en ce qui concerne les environnements 3D, il paraît plus naturel, si l'utilisateur navigue ou interagit en 3D, d'utiliser des périphériques 3D eux aussi, afin de prendre en compte un nombre suffisant de mouvements, pour les mettre en corrélation avec autant de déplacements possibles dans les trois dimensions de l'interface. Cette démarche est plus écologique et doit permettre de faire des interfaces plus faciles à appréhender.

On peut donc faire des remarques Table 4.2 sur les différents types de périphériques 3D pour déplacer un objet virtuel en trois dimensions :

périphérique	inconvénients	avantages
capteur 3D à six degrés de liberté (type Polhemus)	fatigue relative de la main	corrélation directe du mouvement de la main et du pointeur
«trackball 3D» (type Spacemouse)	décorrélation du mouvement de la main et du pointeur	peu de mouvement de la main les axes de déplacement correspondent aux translations et rotations autour des 3 axes
Joystick à 3 degrés de liberté au minimum (type Microsoft SideWinder 3Dpro)	décorrélation du mouvement de la main et du pointeur les axes de déplacement ne correspondent pas aux translations et rotations autour des 3 axes	peu de mouvement de la main le périphérique se recentre seul dès qu'on le lâche (élastique)

Table 4.2 *désignation dans les interfaces 3D : exemples significatifs*

Pour conserver un mouvement naturel, nous nous sommes orientés vers les capteurs de position 3D. Ces derniers permettent une meilleure proprioception, ils sont plus écologiques dans le sens où il y a une transposition de l'outil physique vers le virtuel.

J'ai effectué avec Patricia Plénacoste une étude de comparaison sur le pointage dans une version simple de l'interface pour valider notre choix. Il s'agit d'étudier la compatibilité des périphériques classiques et isotoniques au cours d'une activité motrice, nécessitant rapidité et précision et ce, quel que soit le niveau d'expertise des utilisateurs.

Dispositif expérimental

Trente sujets adultes volontaires ont participé à l'expérience. Ils étaient répartis dans le groupe novice ou dans le groupe expert en fonction des réponses données au questionnaire pré-expérimental concernant le niveau de familiarité avec l'outil informatique. Les paramètres suivants ont été contrôlés : la main dominante et leur vision (normale-corrective).

Le matériel expérimental est constitué d'un écran de 21 pouces qui sert de support à l'interface, interface correspondant à une pièce en 3D composée de deux murs, d'un plancher et d'un plafond, et d'un éclairage dont la luminosité est constante (Figure 4.7). A l'intérieur de cet espace sont présentés un cube et un pointeur.

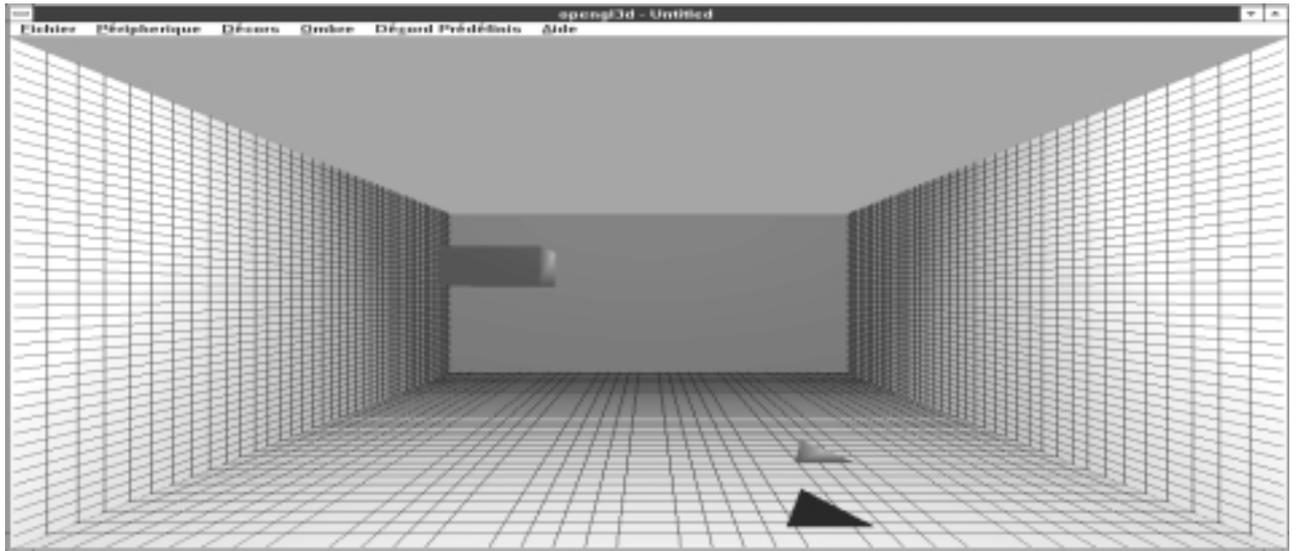


Figure 4.7 *une configuration expérimentale*

Les périphériques d'entrée sont de deux types, une souris 3D et une FreeD. Il a été ajouté à la souris un degré de liberté au moyen des flèches du clavier, ainsi l'axe y s'effectue à partir des flèches (haut-bas). L'axe z (profondeur) et x (gauche-droite) s'effectue à partir de la souris. La FreeD est constitué d'un émetteur à ultrasons fixé sur une bague et un capteur placé autour de l'écran (Figure 1.10). Il comporte trois degrés de liberté (x,y,z).

La configuration du contexte 3D dépend des trois variables indépendantes suivantes :

la texture T comprenant deux modalités, quadrillé et non quadrillé,

l'ombre du cube C comprenant deux modalités, écho cube et sans écho cube,

l'ombre du pointeur P comprenant deux modalités, écho pointeur et sans écho pointeur.

Procédure

L'expérience était composée de deux sessions espacées d'au moins deux semaines. Chacune d'elle débutait par une courte phase de démonstration de la manipulation du périphérique d'entrée - soit la souris 3D, soit la FreeD.

Les sujets étaient placés face à l'écran à une distance d'environ 70 cm. La tâche consistait à pointer le cube le plus rapidement possible. Quand le sujet pensait avoir atteint la cible, il validait sa réponse par un «clic». Un bip sonore était émis en cas de réponse correcte. Suite au bip sonore, le sujet devait repositionner le pointeur dans sa position initiale (en bas au milieu de l'écran).

L'expérimentateur sélectionnait dans les menus de l'interface les propriétés correspondantes à la condition choisie. Dès que le cube et le pointeur apparaissaient sur l'écran, un chronomètre était déclenché. La condition d'arrêt de ce dernier était l'émission du bip sonore. Pour chaque essai de la condition, la position du cube était aléatoire. L'intervalle de

temps entre le bip sonore et l'apparition du cube suivant était de deux secondes.

Chaque session expérimentale comportait huit conditions de vingt essais, conditions issues de la combinaison des variables présentées précédemment, à savoir : la texture, l'ombre du cube, l'ombre du pointeur et le type de périphériques. Le contrebalancement des huit conditions permettait de contrôler un éventuel effet d'apprentissage.

Données recueillies et traitement

Pour chaque sujet, le nombre de clics jusqu'au signal sonore confirmant la réponse correcte, les coordonnées (x,y,z) du cube, les positions (x,y,z) successives du pointeur depuis l'apparition du cube jusqu'à l'apparition du bip, et le temps mis par le sujet pour chaque clic jusqu'au bip de validation étaient enregistrés.

Les verbalisations spontanées du sujet étaient également enregistrées. A la fin de chaque expérimentation, un entretien a été réalisé : on demandait au sujet d'expliquer ce qu'il essayait de faire pour atteindre la cible et de se rappeler les conditions qui lui avaient paru le plus difficile ou le plus facile en fonction du périphérique assigné.

Les traitements statistiques ont été effectués à l'aide d'analyses de variance (ANOVA) par Patricia Plénacoste. Deux ANOVA ont été réalisées, la première en prenant comme variable dépendante (VD) les temps de réaction, et la seconde en prenant comme VD le nombre de clics.

Ces deux ANOVA suivent le plan : $S9 * T2 * C2 * P2 * O2$ avec S pour les sujets, T pour la texture (texturé vs non texturé), C pour l'ombre de la cible ou du cube (écho vs sans écho), P pour l'ombre du pointeur (écho vs sans écho) et O , pour le type de périphérique (souris 3D vs FreeD).

Résultats

L'ANOVA réalisée à partir des temps de réponses nous révèle la présence d'un effet principal significatif du périphérique ($F = 40.035$; $F(1,180) = 6.81$, $p < .01$), et du degré d'expertise ($F = 26.46$; $F(1,180) = 6.81$, $p < .01$). Par contre, l'effet principal de texture, d'ombre de la cible et d'ombre du pointeur n'est pas significatif.

Comme nous pouvons le constater (Figure 4.8), les utilisateurs sont plus rapides pour pointer la cible en utilisant la FreeD (10543.94 ms) qu'en utilisant la souris 3D (17639 ms).

Les sujets experts pointent plus rapidement (Figure 4.9) la cible (11412 ms) que les sujets novices (16775 ms).

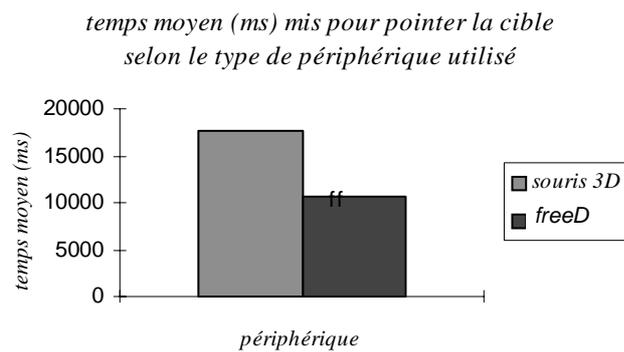


Figure 4.8 *temps moyen de pointage / périphérique*

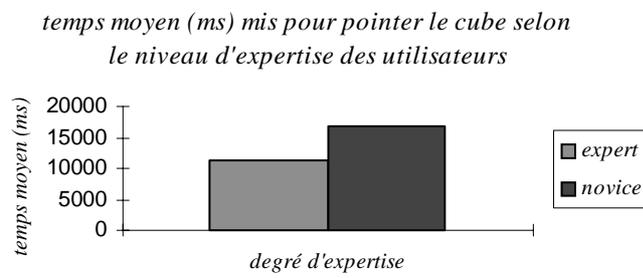


Figure 4.9 *temps moyen de pointage / niveau d'expertise*

L'ANOVA réalisée en prenant en compte le nombre de clics émis pour délivrer la réponse correcte, c'est-à-dire pour pointer correctement le cube, montre qu'il y a un effet principal significatif du niveau d'expertise ($F = 4.68$, $F(1,180) = 3.9$, $p < .05$), de l'ombre du pointeur ($F = 3.37$, $F(1,180) = 2.75$, $p < .10$) et de l'ombre de la cible ($F = 5.72$, $F(1,180) = 3.9$, $p < .05$), et qu'il n'y a pas d'effet significatif de la texture.

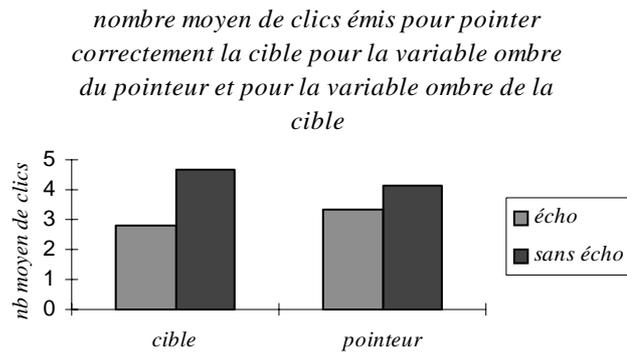


Figure 4.10 nombre de clics avec une ombre sur le pointeur et sur la cible

On note sur la figure ci-dessus que les utilisateurs délivrent moins de clics lorsqu'il y a une ombre au pointeur (3.319 clics vs 4.109 clics) et également lorsqu'il y a une ombre à la cible (2.793 clics vs 4.635 clics).

De même, on constate (Figure 4.11) que le nombre de clics émis par les experts pour atteindre la cible (2.765 clics) est inférieur à celui émis par les novices (4.66 clics).

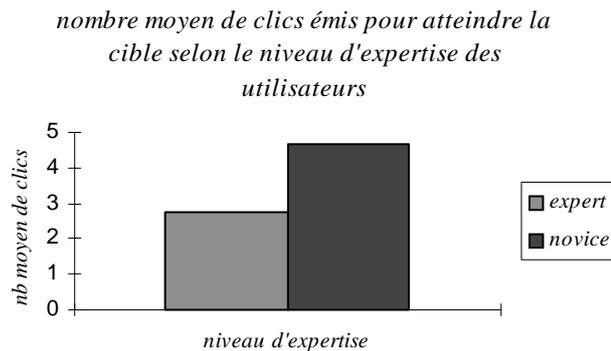


Figure 4.11 nombre de clics / niveau d'expertise

L'interaction ombre de la cible * ombre du pointeur est significative ($F = 3.67$, $F(1, 180) = 2.75$, $p < .10$). Cela signifie que l'effet de l'ombre du pointeur diffère selon l'ombre de la cible (écho vs sans écho) et vice versa. Pour affiner ces résultats, nous avons réalisé un certain nombre de contrastes (Table 4.3).

contraste	DDL	Rapport ^F
C/P1	(1,180)	9,28*
C/P2	(1,180)	0,112 ns
P/C1	(1,180)	7,04*
P/C2	(1,180)	0,0033 ns
<u>variables indépendantes :</u> P : ombre au pointeur (P1: écho, P2: sans écho) C : ombre de la cible (C1: écho, C2: sans écho)	<u>valeurs tables :</u> F (1,180).01 = 6.81* ns : non significatif	

Table 4.3 contrastes

Dans le cas où il y a une ombre au pointeur (Figure 4.10), on observe des performances différentes en situation d'ombre à la cible par rapport à la situation où il n'y a pas d'ombre à la cible ($F = 9.28$, $F(1,180) = 6.81$, $p < .01$). En effet, les sujets émettent moins de clics pour atteindre la cible lorsqu'il y a une ombre sur la cible (2.043 clics) que lorsqu'il n'y en a pas (4.59 clics). Par contre, on ne remarque pas ce fait quand il n'y a pas d'ombre au pointeur.

En ce qui concerne la présence d'ombre à la cible, les performances diffèrent selon qu'il y a une ombre au pointeur ou pas ($F = 7.04$; $F(1,180) = 6.81$, $p < .01$). On constate, Figure 4.12, que les utilisateurs cliquent moins de fois en présence d'une ombre du pointeur (2.043 clics) qu'en son absence (3.54 clics).

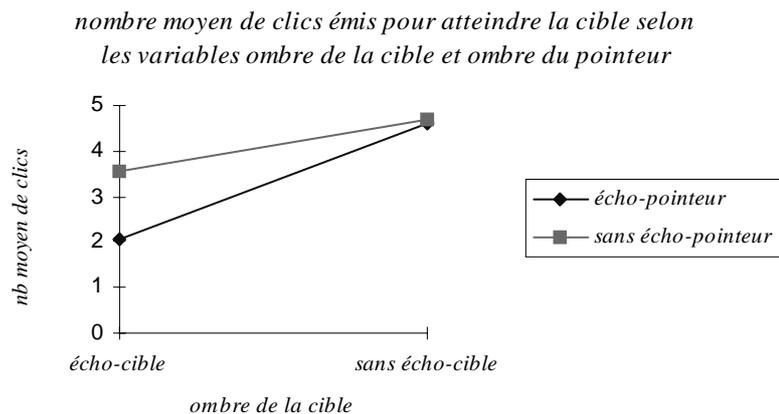


Figure 4.12 nombre de clics / ombres du pointeur et de la cible

Bilan

En résumé, on constate une supériorité des performances lors de la tâche de pointage lorsque l'ombre est présente, lorsque le périphérique utilisé est la FreeD et lorsque les sujets sont des experts. De plus, on peut noter que, dans le cas où il y a une ombre à la fois au pointeur et à la cible, les sujets sont plus précis pour pointer la cible. En effet, ceux-ci émettent moins de clics donc commettent moins d'erreurs.

4.2.2 Précision

La précision du mouvement à effectuer en fonction des déplacements est paramétrable en fonction de l'utilisateur, cela revient à définir une zone d'action dans l'espace où les déplacements du capteur correspondent aux déplacements du pointeur dans l'interface.

Des travaux ont été réalisés par Patricia Plénacoste en collaboration avec Yann Coello, de l'Université de Lille 3, sur la spécificité du contexte visuel dans des tâches manuelles impliquant une coordination visuo-motrice. Leur étude [Plénacoste 98] s'intéresse à l'influence de l'éloignement de la cible et de la quantité d'informations contextuelles, sur la précision spatiale du mouvement. Elle montre que la structuration de la scène visuelle a une influence déterminante sur la précision terminale du mouvement.

Par ailleurs, l'évaluation des distances séparant l'individu des éléments de l'environnement s'effectue en référence à une métrique spatiale relative, construite à partir de l'ensemble des informations disponibles dans la scène visuelle de l'acteur.

Bilan

Une des applications de ce travail concerne la structure de l'interface utilisée dans les interactions avec les environnements virtuels. Il semble important que la distance motrice, relative à la manipulation physique du périphérique, coïncide avec la distance métrique virtuelle de la scène visuelle. Pour que la distance parcourue par le pointeur dans l'espace virtuel soit donc la même que celle de la main, le déplacement du pointeur devra être calculé à partir des paramètres suivants :

- la taille de l'écran (en mettant l'interface en mode plein écran),
- la distance de l'utilisateur à l'écran,
- l'indice de perspective utilisé dans l'environnement virtuel et les caractéristiques de son repère 3D,
- la relation de coordonnées/distance parcourue du capteur 3D.

Si ce choix peut paraître judicieux pour renforcer la sensation de mouvement entre la main et le pointeur, il impose une contrainte forte et difficile à respecter, qui consiste à demander à l'utilisateur de rester à une distance donnée et fixe de l'écran. Elle pourrait être résolue dans d'autres circonstances par l'utilisation d'un casque de visualisation.

4.2.3 Désignation dans l'interface

Elle se fait au moyen d'un pointeur 3D. Si sa forme a une importance, elle doit être suffisamment simple pour être sans ambiguïté, et petite pour ne pas masquer de grandes parties de la scène. Le choix peut en être laissé à l'utilisateur, mais nous utilisons par défaut une flèche 3D comme pointeur. Elle a le mérite d'être un symbole simple et connu de tous.

Le pointeur

L'utilisateur peut avoir du mal à évaluer la position du pointeur par rapport aux autres objets. Ce dernier se déplace en trois dimensions en fonction des mouvements de l'utilisateur avec le capteur de position. Il s'agit d'ajouter les bons indices afin que l'utilisateur puisse au mieux évaluer les distances relatives, entre le pointeur et les autres objets. Nous avons vu précédemment que l'utilisateur pouvait désigner directement quelque chose dans tout l'espace de travail : l'espace est fermé pour faciliter l'appréhension globale de la scène.

Transparence

Le premier critère consiste à ne pas perdre de vue le pointeur. Ce dernier doit donc se déplacer dans les limites de l'espace de travail. En utilisant les transparences, un objet traversé devient transparent, ce qui permet de ne pas perdre le pointeur, il reste toujours visible. De plus, la transparence agit comme un indice perceptif [Zhai 96] puisqu'elle permet de découvrir des aspects cachés d'une scène.

Nous verrons par la suite que cela peut également permettre à l'utilisateur de ne pas perdre la vue de l'objet sur lequel il travaille.

Dynamique du pointeur

On déplace le pointeur suivant les 3 axes (x,y,z) pour lui faire décrire l'espace. L'utilisateur a un point de vue fixe sur la scène au moment de déplacer le pointeur, on laisse donc le pointeur orienté dans le même sens, le sens qui correspond au déplacement vers les objets, dans le sens de la profondeur.

On se sert de la rotation du pointeur pour renforcer sa dynamique : on fait varier légèrement l'orientation de celui-ci en fonction du mouvement. Le pointeur tourne sur l'axe de son déplacement. Si le pointeur va de gauche à droite, il tourne sur l'axe vertical de plus ou moins 15°, et s'il va de haut en bas, il tourne sur l'axe horizontal de plus ou moins 15°.

Enfin, l'animation du pointeur lui-même, c'est-à-dire le faire changer de forme, permet d'informer sur le contexte de la tâche. Par exemple, lorsque dans une application de dessin, le pointeur se transforme en pinceau.

Taille du pointeur

On remarque empiriquement que l'interaction semble facilitée lorsque le pointeur est gros. La sensibilité du pointeur est identique (son extrémité), mais l'espace qu'il occupe dans la scène est plus important. La taille ne peut cependant pas être très élevée, typiquement on pourrait la choisir

égale au plus petit objet sélectionnable. Ce paramètre est lié au type de document visualisé, mais trouve sa limite dans la précision du mouvement de l'utilisateur, par rapport à la taille de l'écran et à la sensibilité du périphérique de pointage. Cette limite est bien inférieure à une taille raisonnable du pointeur, qui le laisse aisément visible dans toute l'interface. Sa taille est donc fixée arbitrairement au vingtième de la taille de la scène.

On peut faire varier la taille du pointeur de façon plus importante que la perspective, afin de renforcer l'effet de mouvement suivant l'axe de la profondeur. Cette information est intéressante, mais doit être utilisée avec précaution, le pointeur ne devant pas être un élément gênant lorsqu'il est devant, et ne doit pas être perdu lorsqu'il est derrière. Nous avons fixé la variation de taille empiriquement à 1,1 x la taille dans les deux directions au seuil maximum.

[Wang 99] a réalisé une étude qui compare la précision de l'utilisateur lorsque la taille du périphérique, du pointeur et de l'objet pointé varient. Bien que l'expérience se situe en vision stéréo (ce qui est différent de notre environnement), il en déduit qu'une taille équivalente du contrôleur et du pointeur améliore la vitesse d'exécution de l'utilisateur, et qu'une taille équivalente entre le pointeur et l'objet pointé améliore la précision du mouvement. On se rend compte au vu de cette étude, comme à celle sur la précision ci-dessus, que la maîtrise de la distance moyenne de l'utilisateur à l'écran et la connaissance de la taille de ce dernier peuvent être des facteurs importants d'efficacité, dont il faut tenir compte dès que les conditions le permettent.

Calibration

Ensuite, bien que nous ayons vu qu'une correspondance en terme de distance était souhaitable entre l'environnement réel et virtuel, il faut laisser le soin à l'utilisateur de choisir la sensibilité du périphérique de désignation. On voit donc des utilisateurs débutants choisir une sensibilité faible, cela permet des mouvements moins précis et plus amples. Au fur et à mesure de l'apprentissage du capteur de position 3D, on travaille dans des espaces de déplacement de la main de plus en plus réduits, pour limiter la fatigue, et ne laisser la précision du pointage qu'au seul poignet et à la main, plutôt qu'au bras entier, moins précis.

4.2.4 Bilan

Nous utilisons de préférence un capteur de position 3D, le plus proprioceptif, pour la corrélation directe qu'il y a entre le mouvement de la main et celui du pointeur (le problème de la fatigue est pris en compte dans le reste du modèle d'interaction). Nous l'utilisons en mode absolu, avec la possibilité pour l'utilisateur de le recentrer et de faire varier sa sensibilité. On utilise les trois degrés de liberté de ce capteur, correspondant à la translation, pour déplacer un pointeur 3D dans l'interface.

4.3 Indices visuels

Lors de la conception d'environnements informatiques virtuels, se pose très souvent la question de la structuration visuelle d'une scène 3D pour rendre l'interaction plus écologique, c'est-à-dire plus proche de la représentation de l'action du point de vue de l'utilisateur. Outre les difficultés

techniques liées aux modes d'entrée/sortie que nous venons d'aborder, la simulation d'un espace 3D sur un écran 2D d'ordinateur pose des problèmes. On n'y retrouve pas l'information de profondeur fournie par la disparité rétinienne dans le monde réel. Si nous avons écarté les solutions de visualisation stéréographique pour des raisons techniques, de commodité et d'applications, nous allons devoir d'autant plus concevoir l'espace de travail virtuel pour aider l'utilisateur.

L'espace de travail peut contenir un certain nombre d'indices perceptifs. Tout d'abord l'éclairage sur les objets grâce au mode de rendu, puis des textures représentant des gradients, comme un quadrillage sur le sol, permettent d'aider l'utilisateur à percevoir la profondeur. Ces indices visuels peuvent se classer suivant trois critères (Table 4.4).

OU	
permanent l'indice est affiché en permanence	temporaire l'indice est affiché à un moment donné pendant un instant déterminé, il intervient donc en fonction d'actions précises d'un utilisateur
statique l'indice ne se déplace pas	dynamique l'indice se déplace dans l'espace, pour suivre un objet en mouvement par exemple
passif l'indice ne change pas de forme	actif l'indice change de forme

Table 4.4 *trois catégories d'indices perceptifs*

La présence d'une ombre, au dessous d'un objet par exemple, sera un indice permanent, dynamique parce qu'il suit l'objet, passif s'il ne change pas de couleur ou de forme, ou actif si sa taille et son opacité varient en fonction de la hauteur de l'objet (cas d'une ombre réelle). On trouve d'autres exemples Table 4.5.

indices

permanent		temporaire	
statique	dynamique	statique	dynamique
textures sur les murs quadrillage sur le sol	ombres portées sur les objets	une boîte englobante autour d'un objet sélectionné	les boîtes englobantes progressives lors de l'approche d'un document

Table 4.5 *exemples d'indice par catégorie*

Si l'environnement réel ne réagit qu'à un contact ou à une manipulation directe, il en va autrement pour un espace virtuel, qui peut avoir un comportement actif pendant le déroulement d'un geste (qui

n'a donc pas encore amené à une action). On a donc le modèle approche/action avec une réaction active du système, due à la mesure permanente que l'on a des mouvements de l'utilisateur.

Ainsi la proximité du pointeur à l'objet est une information préliminaire qui peut être utilisée pour anticiper les approches d'objet et aider l'utilisateur à savoir où il en est. Si l'objet s'éclaire de plus en plus lorsque l'on s'en approche par exemple, il s'agit d'un indice temporaire et actif.

Les indices actifs permettent de créer des comportements implicites de l'espace de travail, qui va réagir en fonction de l'utilisateur pour l'aider à se repérer dans l'espace de travail 3D.

Les textures

Pour évaluer au mieux la position du pointeur, on peut ajouter des indices visuels statiques permanents, comme des repères jalonnant régulièrement le sol, les murs, etc. Ces indices peuvent prendre la forme de lignes de fuite (verticale, horizontale ou en profondeur sur les éléments du décor comme le sol), de textures répétées à différentes fréquences sur les trois axes, ou simplement d'effet d'éclairement. En effet, en faisant varier artificiellement la lumière en fonction de la profondeur d'un objet particulier (le sol par exemple), c'est-à-dire de manière plus forte que le calcul d'éclairement de la scène ne l'impose pour le rendu global, on renforce l'effet de profondeur. Cependant, il faut que tout reste visible et net en permanence, en évitant les zones d'ombres qui risqueraient de masquer des informations. On a donc pu constater qu'en environnement de travail, il est difficile de jouer beaucoup sur les effets de lumière.

L'expérimentation décrite en 4.2.1 nous a permis de tester l'efficacité de certains périphériques mais également la qualité de certains indices. Les conditions comprenaient la présence de murs éclairés (donc d'un gradient de texture qui s'assombrissait avec la distance), d'un quadrillage et d'ombres sur le pointeur ou la cible. Les résultats ont montré que l'effet d'autres indices que celui de l'ombre sont négligeables. [Wanger 92] a également observé un effet significatif des ombres pour la localisation d'un objet. Elle permet de fournir un ensemble de relations sur la position spatiale, la localisation d'un objet et sur la profondeur.

L'ombre

Nous avons opté pour l'ombre sur le pointeur 3D et les documents à manipuler. Ici il s'agira d'un indice dynamique. Il est essentiel, voire indispensable. L'ombre des objets ou ombres portées fournit des indicateurs sur les relations spatiales et sur la profondeur. Elles permettent aux utilisateurs d'inférer à la fois la position et la localisation d'un objet. L'écart sur la surface de la scène entre un objet et son ombre indique sa hauteur. La localisation de l'ombre sur la surface indique la distance de l'objet.

Les sources lumineuses ne sont pas visibles en elles-mêmes en rendu projectif, ni les rayons lumineux. On peut matérialiser les lampes dans la scène, mais là n'est pas l'essentiel, il faut surtout que toutes les ombres soient cohérentes et semblent provenir de la même source. Sachant que l'on éclaire surtout la scène par le dessus pour des raisons de réalisme et de netteté, on projette l'ombre à partir d'une source lumineuse fixe, placée en hauteur.

Pour déterminer la position et la taille idéale à utiliser pour les ombres des objets virtuels, nous avons réalisé une expérimentation portant sur une tâche de pointage d'objets suivant plusieurs types d'ombres. Cette manipulation avait aussi pour objectif de déterminer quel algorithme de calcul d'ombre portée était suffisant, certains d'entre eux pouvant être assez coûteux en matière de temps de calcul (voir chapitre 1).

La plupart des auteurs [Kersten 97] [Mamassian 98], qui travaillent sur l'influence des ombres sur la perception de la profondeur ou sur la reconnaissance des objets naturels, se focalisent sur des tâches de discrimination ou de jugement d'objets en mouvement, en fonction de l'ombre des objets. Dans notre situation, nous nous centrons sur une situation d'action, situation de pointage telle que l'utilisateur la réaliserait dans son interface de travail. Nous étudions surtout l'effet de la forme de l'ombre (carrée, octogonale, réelle, sans) pour la réussite de l'action selon la zone d'action dans la scène 3D.

Dispositif expérimental

13 sujets, droitiers et de vision normale ou corrigée, ont participé volontairement à l'expérience. Le dispositif expérimental est constitué d'un écran de 21 pouces. A l'intérieur d'un espace 3D, trois objets et un pointeur sont présentés (Figure 4.13).

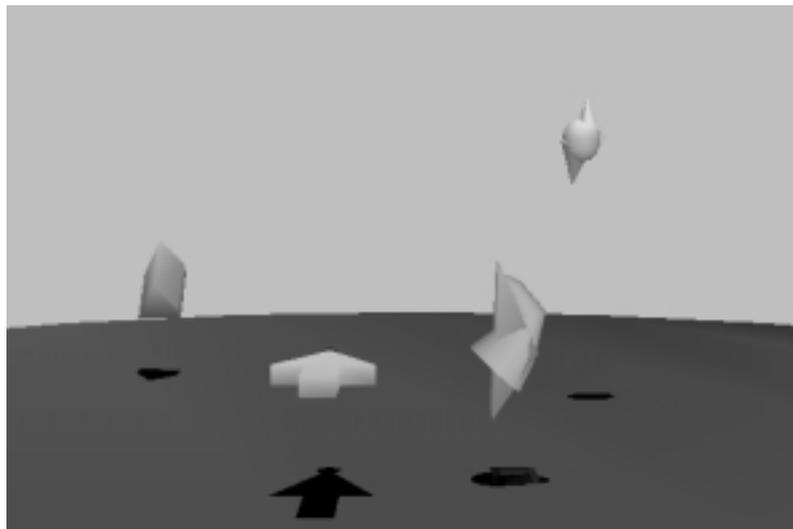


Figure 4.13 *une configuration expérimentale, le pointeur et trois objets géométriques.*

L'espace a été décomposé en 12 zones (Figure 4.14). Quatre modalités de l'ombre sont manipulées : carrée, octogonale, réelle et sans.

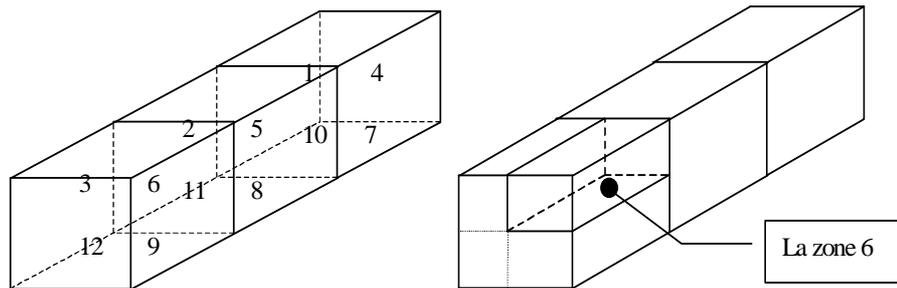


Figure 4.14 vue des 12 zones (détail à droite)

Procédure

Une phase de démonstration de la manipulation du capteur de position 3D précède la phase expérimentale. Les sujets sont placés face à l'écran à une distance d'environ 70 cm. La tâche consiste à pointer un des trois objets, le plus rapidement possible. Pour débiter, il faut cliquer sur la boîte de positionnement «départ». Les trois objets sont alors présentés. L'objet à pointer apparaît un court instant entouré d'une boîte englobante. Chaque objet est tiré au hasard dans une des zones. Quand le sujet atteint la cible, la boîte «départ» réapparaît. Un chronomètre est déclenché quand le sujet clique sur la boîte départ, il s'arrête quand il a cliqué sur l'objet à pointer. Dix pointages sont effectués dans chaque zone et selon le type d'ombre. Ainsi, pour chaque condition, 120 essais sont réalisés. La présentation des objets est aléatoire. Le type d'ombre est sélectionné par l'expérimentateur. Pour éviter l'effet d'apprentissage, les types d'ombre ont été contrebalancés. Pour les quatre conditions d'ombre, le nombre de clics émis et le temps mis pour cliquer l'objet sont enregistrés. Une ANOVA a été réalisée sur les clics et les temps de réactions.

Résultats

Les résultats ont été analysés par Patricia Plénacoste. L'ANOVA réalisée montre, à partir des temps de réponse pour cliquer sur l'objet, un effet principal de l'ombre ($F=2,541$, $p<0.00671$) et des zones ($F=25,671$, $p<0.0001$). L'interaction entre les zones et l'ombre est significative ($F=3.930$, $p<0.0001$). Les résultats montrent que, quelle que soit la forme de l'ombre et en fonction des zones, les performances des utilisateurs varient. Ils mettent moins de temps en présence de l'ombre que sans ombre pour cliquer sur l'objet (Figure 4.15).

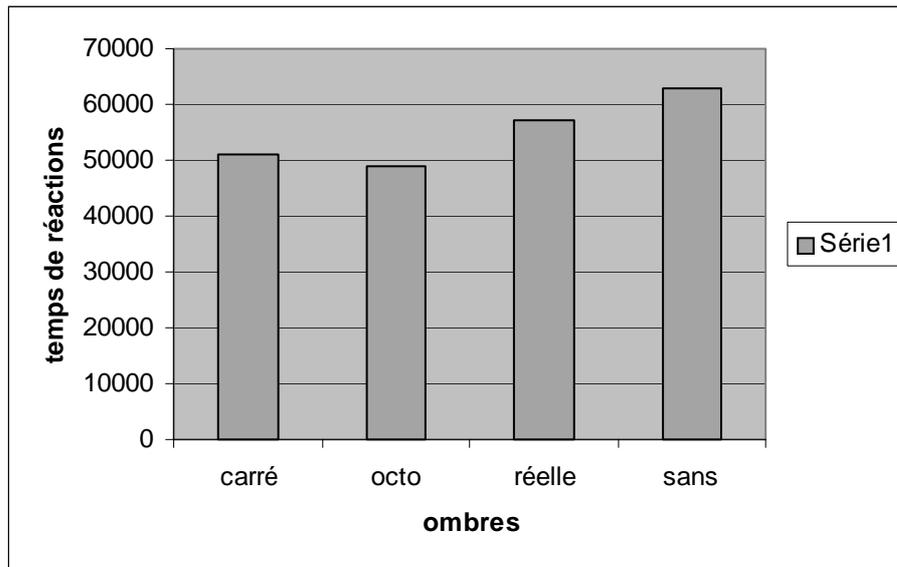


Figure 4.15 temps moyen en fonction des ombres

L'ANOVA révèle une tendance du facteur ombre ($F=1,683$, $p<0,1003$) à partir du nombre de clics émis pour cliquer sur l'objet. Par contre, nous avons un effet très significatif du facteur zone ($F=33,606$, $p<0,0001$). L'interaction entre les deux facteurs ombre et zone est significative ($F=3,929$, $p<0,001$). Là encore, la performance du sujet varie en fonction de la zone et de la forme de l'ombre. Ils cliquent moins en présence d'ombre que sans ombre (Figure 4.16).

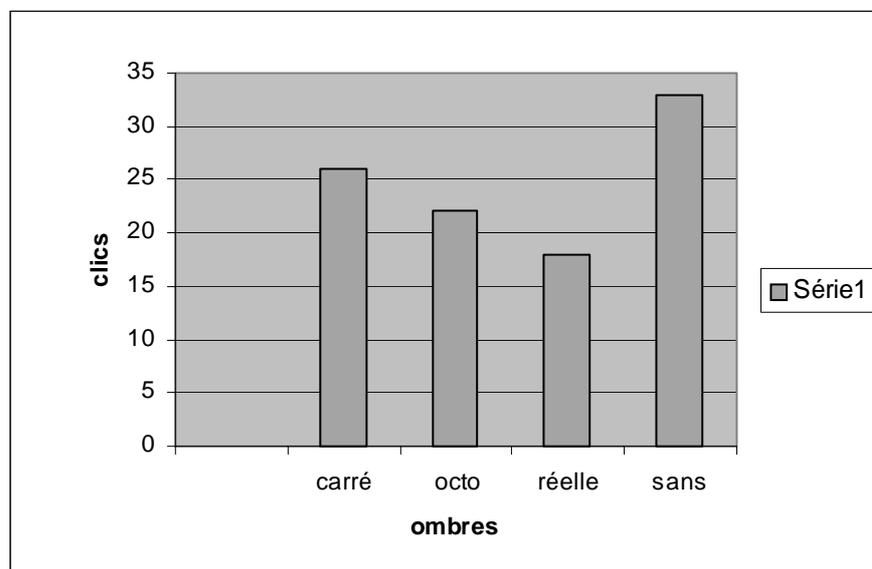


Figure 4.16 nombre moyen de clics en fonction de l'ombre

Conclusion

On constate que, lors d'une tâche de pointage d'objet, les performances des sujets sont meilleures en présence d'ombre et ce, quelle que soit sa forme. On en déduit que les sujets traitent les ombres

comme une «affordance d'action» (terme que G.A. Boy traduit par capacité suggestive d'action). L'ombre ne serait donc pas traitée comme une propriété intégrante à l'objet mais bien comme une heuristique. On peut donc choisir la forme d'ombre la moins coûteuse en terme de calcul, il suffit juste de respecter l'échelle entre un objet et son ombre portée.

Les boîtes englobantes

Les indices dynamiques peuvent être une aide précieuse pour mieux percevoir les objets sur lesquels on travaille. La proximité du pointeur avec un objet particulier est facilement détectable. Nous nous en servons pour donner des indications sur la position du pointeur par rapport à un document.

On peut déterminer l'endroit d'un clic, s'il n'a pas lieu à l'intérieur d'un objet, comme la volonté de se rapprocher ou de sélectionner l'objet le plus proche. On pourrait donc mettre en place des règles d'optimisations permettant d'accéder directement à l'objet le plus proche en (XoY) ou en (oZ), le plus proche dans la direction du pointeur dans le cas d'un périphérique à 6 degrés de liberté. Cette technique est cependant sensible à des clics parasites, ou à une volonté bien réelle de cliquer dans une zone «neutre», il faut activer cette possibilité seulement à proximité de l'objet et indiquer à l'utilisateur ce qu'il va sélectionner.

Nous utilisons donc des boîtes englobantes pour indiquer quel objet nous sommes en train de pointer. Nous faisons varier la taille de ces boîtes au fur et à mesure que l'on s'éloigne de l'objet (Figure 4.17), en atténuant leur couleur, jusqu'à ce qu'elles disparaissent à une distance donnée (1,3 fois la taille de l'objet par expérience). Ce système permet d'animer uniformément les transitions du pointeur entre les objets tout en conservant une information sur ce qui est pointé. Ce sont des boîtes englobantes progressives.

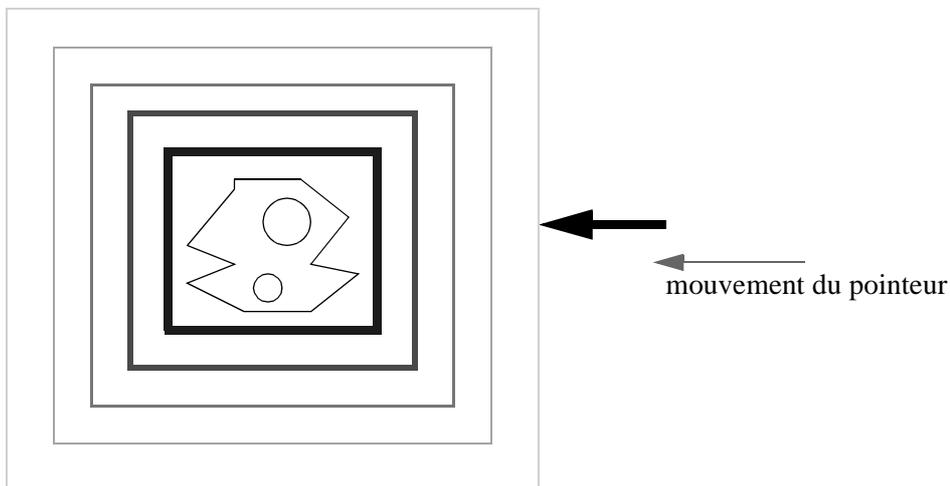


Figure 4.17 boîte progressive à l'approche d'un objet

La boîte de l'objet, qui va être sélectionné, est affichée en bleu (Figure 4.19). Si d'autres boîtes sont affichées en même temps, elles le sont de façon très discrète, en transparence. Ce système permet de sélectionner un objet, même si l'on ne se trouve pas exactement à l'intérieur. Ce qui permet de

compenser les éventuelles erreurs d'appréciation de la profondeur, et de faciliter l'approche des objets, des petits en particulier.

Lorsque l'utilisateur clique, c'est la boîte englobante dont le pointeur est le plus au centre qui est sélectionnée (Figure 4.18). Cela permet de toujours pouvoir sélectionner une sous-partie d'un objet, même si elle est comprise dans une plus importante.

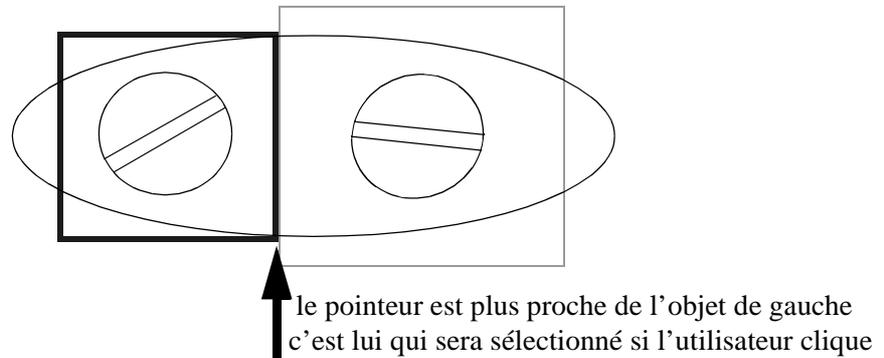


Figure 4.18 *sélection des objets*

Un document peut être sélectionné dans sa globalité, mais on peut également sélectionner des sous-parties de ce document. C'est le cas dans les textes 2D, dans les dessins vectoriels, mais également avec les documents 3D, composés de plusieurs parties, activables si le document a été conçu dans ce sens (une sous-partie géométrique d'un document ne peut être sélectionnée que si l'on peut agir dessus, effectuer une action particulière).

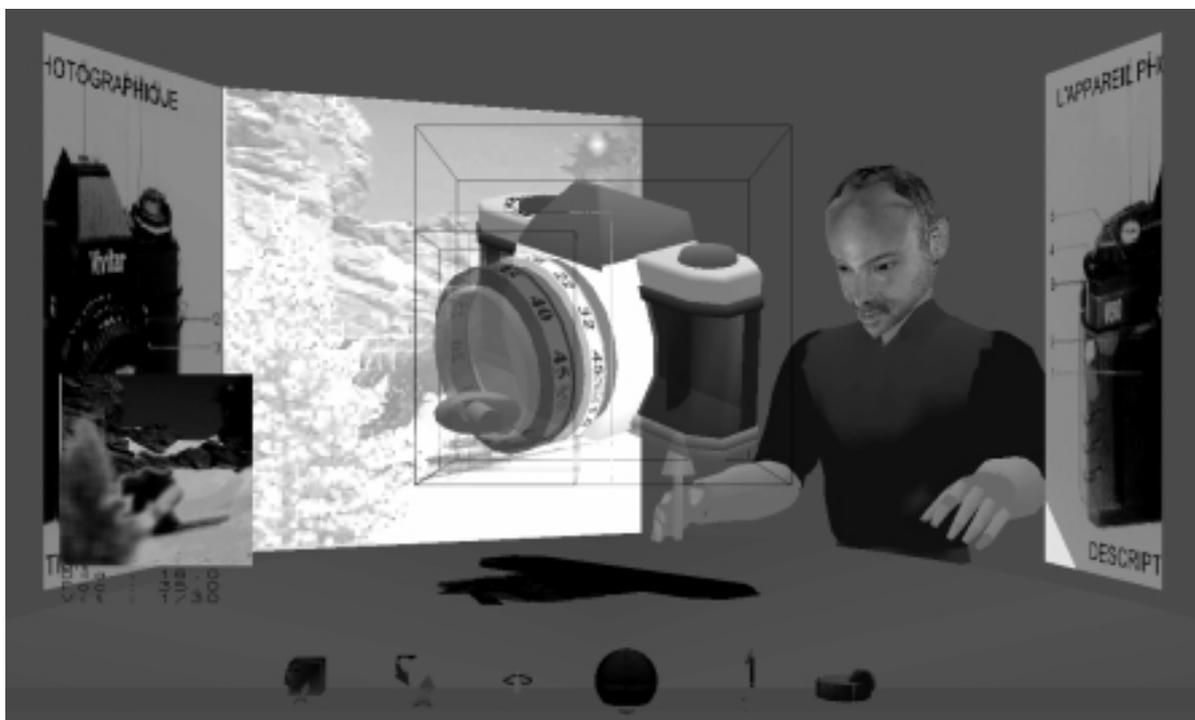


Figure 4.19 *les boîtes englobantes progressives dans Spln*

La mobilité relative

La mobilité relative de la scène permet, lorsque l'on observe un objet dans l'avant plan, de déplacer légèrement son point de vue sur ce dernier afin de pouvoir décaler l'angle de vision (Figure 4.20). Cela aide à percevoir globalement l'objet, les faces cachées de l'objet. Ce mouvement, que l'on applique à l'objet, correspond dans la réalité aux mouvements du regard que l'on effectue pour scruter un objet. Ces mouvements sont effectués par des légers déplacements de la tête et par le mouvement de scrutation des pupilles pour parcourir un objet. En virtuel, lorsque l'on observe un objet, ce mouvement se réalise avec un périphérique isométrique (comme la Spacemouse montrée Figure 4.20), ce qui permet de décaler légèrement la scène autour de l'axe des abscisses et ordonnées. Lorsque l'utilisateur relâche le périphérique, la scène revient à sa position initiale.

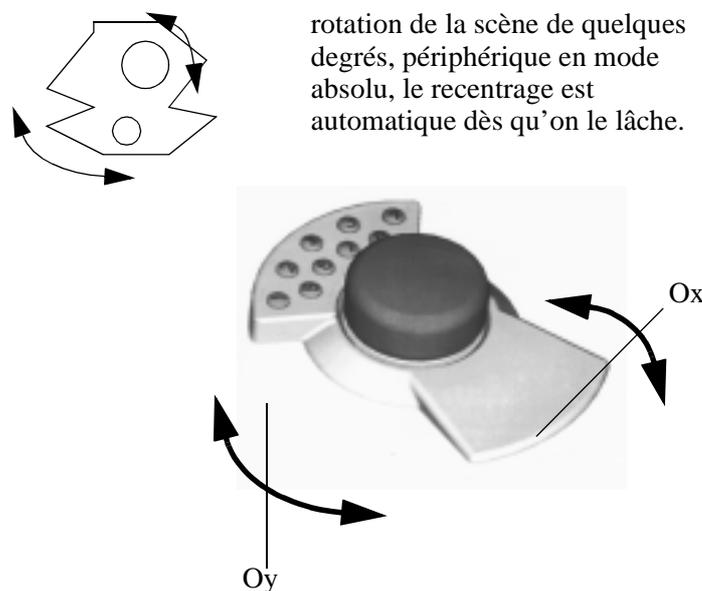


Figure 4.20 *observation de la scène*

Cette action ne correspond pas à une manipulation d'un document mais juste à une rotation limitée de la scène complète, afin de mieux la percevoir dans son ensemble. Cette action permet de mieux appréhender l'objet placé au devant de la scène, c'est donc lorsque l'on est positionné à l'intérieur de ce dernier que l'on peut décaler légèrement son point de vue.

Il ne s'agit pas à proprement parler de navigation puisqu'on ne se déplace pas par rapport aux objets, on ne modifie que légèrement et temporairement notre point de vue global pour mieux appréhender la scène.

Dans le cas de SpIn (Figure 4.21), la rotation sur l'axe Oy est difficile à comprendre car le bandeau tourne suivant le même axe, c'est donc assez perturbant. L'information donnée par le changement de point de vue en hauteur est plus pertinente (on voit la scène d'un peu plus haut), on pourra donc se contenter d'une rotation de quelques degrés autour de l'axe des abscisses.

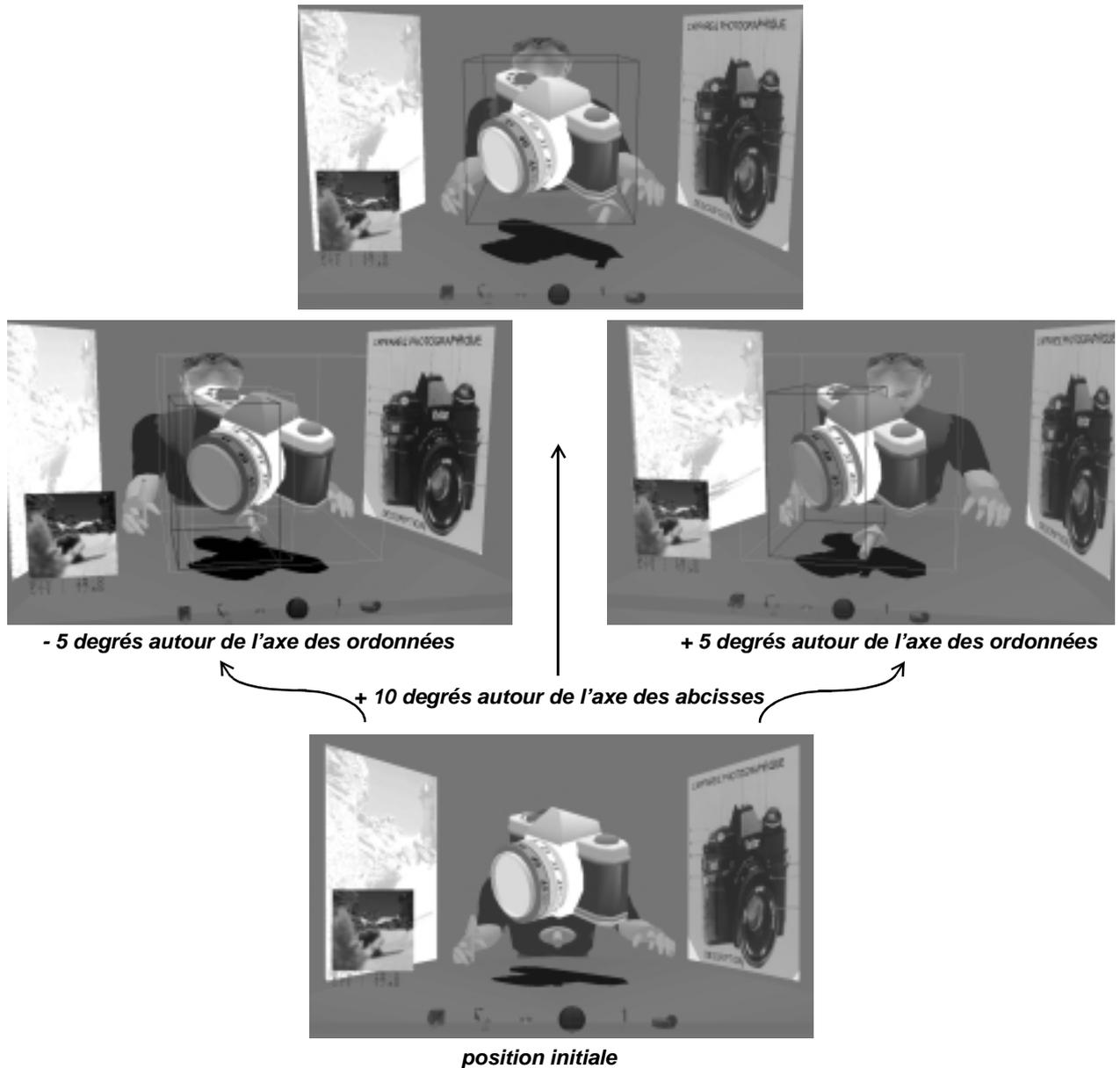


Figure 4.21 *la mobilité relative vue dans Spln*

Le fait de changer de point de vue sur l'axe des abscisses modifie notre perception de la scène. Il est difficile d'évaluer l'impact sur l'activité globale et sur la perception des autres, que l'on voit alors avec ce point de vue décalé légèrement vers le haut. Si la rotation de la scène sur l'axe des abscisses permet donc une meilleure perception des mouvements du pointeur par rapport au document sur l'avant-plan, on est en droit de se demander si cela ne va pas changer non plus notre perception des autres (d'un point de vue social). Si les utilisateurs ont réservé un accueil positif à la mobilité relative en matière d'interaction 3D, nous n'avons pas eu la possibilité de l'évaluer dans une situation de réunion réelle.

Bilan

Nos recommandations pour la désignation sont d'utiliser un capteur de position 3D pour déplacer un pointeur 3D (recentrable, sensibilité réglable). L'utilisateur doit pouvoir pointer directement à travers tout l'espace de travail, qui doit être fini. Des indices visuels doivent être placés sur le pointeur et sur dans la scène, de manière à aider l'utilisateur, mais sans surcharger l'interface. Il faut donc les choisir, en particulier en utilisant les ombres sur les documents et sur le pointeur. L'approche des objets doit être facilitée et ne pas poser d'ambiguïté avec un mécanisme du type des boîtes englobantes progressives.

4.4 Sélection de l'objet : interaction bimanuelle

La sélection de l'objet se fait, premièrement, par sa désignation au pointeur. Si on est à l'intérieur de la boîte englobante de l'objet, on peut le sélectionner, en cliquant par exemple. Il faut veiller à conserver une taille minimale de boîte englobante afin de pouvoir sélectionner les petits objets. Quant à la sélection de l'objet lui-même, elle peut se faire explicitement ou implicitement.

La sélection d'un document peut être implicite. C'est-à-dire que le seul passage du pointeur sur une zone du document peut déclencher un événement. Etant donné que l'on peut passer à proximité d'un objet uniquement pour déplacer le pointeur d'un point à un autre sans rien vouloir sélectionner, ce mode n'est pas exploitable pour effectuer une sélection. Cependant l'idée d'avoir des zones réactives peut servir lors de la désignation, si un document est réactif lorsqu'on le traverse, on peut renseigner l'utilisateur sur l'endroit où il se trouve (en mettant en transparence l'objet que traverse le pointeur), ou sur le document lui-même.

4.4.1 Périphérique

Compte tenu de la simplicité de l'action, du manque de retour (sensitif) des périphériques utilisés, et de l'historique de l'informatique (utilisation du bouton comme moyen de sélection avec succès depuis longtemps), on utilise un dispositif avec un bouton pour marquer son attention sur l'objet.

La sélection s'effectue au moyen du bouton placé sur le périphérique de pointage, comme avec une souris. Afin que l'interface soit plus intuitive, on n'utilise autant que possible qu'un seul bouton. L'utilisateur n'a pas à se poser la question au moment d'agir. En effet, dans le cas de plusieurs boutons sur les périphériques, l'utilisation de tel ou tel bouton relève de l'apprentissage, et non d'une certaine logique ou d'un côté intuitif, leur fonction étant générique. De même, on utilise toujours le même mode pour sélectionner, ou désélectionner si besoin est, un document, en cliquant sur le bouton du périphérique.

On peut remarquer que la facilité d'interaction avec un périphérique 3D vient d'abord de son ergonomie. Pour les périphériques actuels, on notera que les deux fabricants de capteur 3D électromagnétique fournissent des capteurs 3D à boutons particulièrement difficiles à utiliser. Polhemus propose une boule de billard (la 3Dball Figure 4.22) munie d'un bouton circulaire, dont la notion de pression n'est que très peu sensible au doigt, et qui génère des rebonds (plusieurs

impulsions envoyées à l'ordinateur pour un seul clic). Ce dernier n'est donc pas utilisable. Mais Ascension ne propose guère mieux avec un capteur de position en forme de souris (Figure 4.22), quasiment de la taille d'une souris, avec un côté plat. Elle est particulièrement difficile à manier dans l'espace, de par sa taille, sa forme et la position des boutons. D'autres marques de capteurs 3D, acoustique cette fois, ont fait évoluer leurs modèles, comme Pegasus Technologies, qui a équipé son capteur initial FreeD, en forme d'anneau à passer au doigt, d'une poignée à saisir à la main (comme un manche) avec deux boutons positionnés sur le dessus.

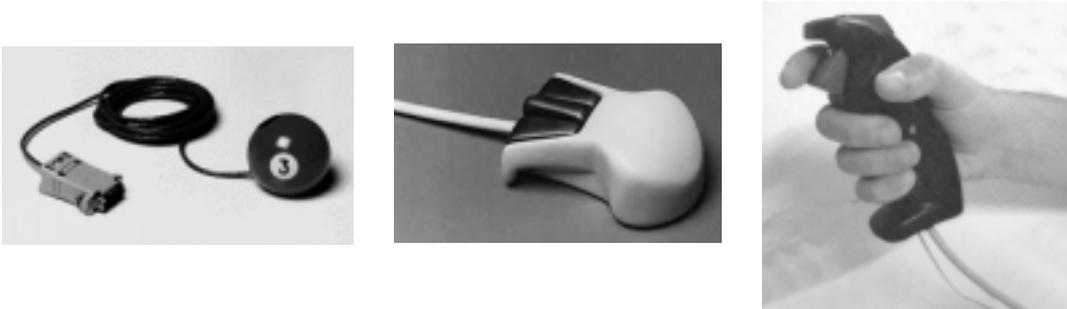


Figure 4.22 *la souris 3D de Polhemus (à gauche), d'Ascension (au milieu) et un prototype pour Spline (à droite)*

Une des solutions les plus simples consiste également à adapter un capteur de position 3D de type Polhemus (un cube d'un centimètre de côté) dans un corps de périphérique, avec un bouton simple et facile à manipuler. Nous avons réalisé un capteur à bouton de ce type en le plaçant dans un manche de joystick (Figure 4.22), l'ensemble ayant une bonne prise en main tout en restant léger.

4.4.2 Visualisation de la sélection

La visualisation de la sélection d'un objet peut se faire de différentes manières :

- en laissant affichée une boîte englobante autour de l'objet tant qu'il est sélectionné. Ce mode a l'inconvénient d'utiliser le même artefact graphique que la désignation, assez abstrait, il reste assez peu esthétique quand l'objet est longtemps sélectionné,
- en sur-éclairant l'objet. Ce mode est pratique mais peut être assez peu discriminant visuellement lorsque l'objet est déjà clair,
- en affichant le profil de l'objet avec une couleur qui contraste. Ce mode n'est pas très visible non plus, en particulier lorsque l'on sélectionne une sous-partie d'un document (cela nécessite de plus beaucoup de calculs),
- en modifiant la couleur, par exemple, en saturant une des couleurs de l'objet. Ce mode est le pendant du texte sélectionné dans les traitements de texte, que l'on affiche en couleur inverse. C'est le mode, qui selon nous, crée le moins d'équivoque. Nous affichons en bleu les objets sélectionnés. Cela peut poser un problème pour une réunion où les couleurs du document ont une importance dans l'activité.

Le fait de modifier la structure du document pour indiquer un changement d'état lié à l'interaction (objet sélectionné) peut ne pas paraître idéal ni approprié. On préférera l'illumination, le changement de couleur, pour faire ressortir l'élément concerné.

4.4.3 L'instrumentation

Lorsqu'un objet est sélectionné, on veut pouvoir appliquer une action dessus. C'est le principe même d'un système interactif. On divise ces actions possibles en deux catégories :

- la manipulation du document, c'est-à-dire la modification du point de vue sur le document ou du document lui-même (voir paragraphe suivant),
- les actions qui ne sont pas de l'ordre de la manipulation (opération non géométrique, comme sauvegarder le document dans un fichier).

Dans ce dernier cas, on peut accéder à ces actions à travers différents moyens d'interaction : commande en ligne, commande lancée interactivement à travers un menu, utilisation d'un widget, utilisation d'un outil.

Dans la mesure où l'environnement de travail 3D se prête bien à l'utilisation de similitude entre un espace de travail réel et virtuel, d'un point de vue visualisation et interaction, nous privilégions l'utilisation d'instruments, au sens de [Beaudouin 97], et plus particulièrement d'outils.

Les outils sont des objets 3D de l'interface capables d'agir sur des documents de type particulier en interaction directe. Ils remplacent le pointeur 3D une fois sélectionnés, et permettent d'agir directement sur le document. Ils évitent le décalage spatial entre l'instrument logique et le document, que l'on rencontre avec les widgets et qui obligent l'utilisateur à basculer son attention entre le document et l'instrument logique. L'accès aux outils se fait à travers un regroupement de type palette d'outils (voir chapitre suivant).

4.4.4 La manipulation

En travail coopératif synchrone, après la désignation, c'est la manipulation d'objet qui est la plus fréquente. Dans les situations de réunion, cibles de SpIn, les interlocuteurs discutent autour de documents : communication, désignation, manipulation et parfois édition d'un document. On a donc souvent des documents comme support de la réunion, que l'on se contente de manipuler. Il s'agit alors de modifier son point de vue sur le document ou sur une partie du document. Par exemple, faire défiler du texte ou faire tourner un objet sur lui-même. La manipulation va donc être l'action unique définie pour beaucoup de documents si l'activité ne nécessite pas de les éditer. Le fait de sélectionner un document donne donc souvent directement la possibilité de bouger son point de vue sur les documents.

Manipulation directe

On préfère agir directement sur les documents plutôt qu'avoir une manipulation déportée sur des menus ou des widgets. Les widgets obligent à un mouvement différent du mouvement réel. On doit faire l'effort d'associer mentalement le mouvement du widget à celui qui est effectivement appliqué au document, ce qui fait passer notre attention de l'un à l'autre. Nous préférons l'utilisation des outils (voir paragraphe précédent), qui permettent d'interagir directement avec les documents.

Nous voulons conserver le lien avec le réel dans la représentation de l'espace virtuel : si on ne peut toucher l'objet virtuel pour le manipuler, on recherche un rapport visuel direct entre les gestes

d'action de l'utilisateur et l'objet visé, avec une animation continue de l'objet en rapport avec les gestes (i.e. le mouvement du périphérique). Ce mode de manipulation est plus naturel.

Les documents dans SpIn

Comme nous le verrons dans la partie sur l'architecture de SpIn, la notion d'application n'existe pas de la même façon que dans les interfaces 2D standards. SpIn est une interface graphique qui gère l'espace de travail de l'utilisateur, elle se substitue complètement à ce qui peut exister sur la machine où elle est exécutée, et n'en utilise pas les mécanismes. On peut ainsi voir SpIn comme l'interface de travail possible de n'importe quel système d'exploitation. C'est dans cette optique que SpIn a été développé, en particulier pour s'astreindre de l'existant.

SpIn permet de visualiser des objets suivant une organisation spatiale bien précise. Il reproduit une structure d'interface où les documents peuvent être manipulés grâce à des actions associées à des événements particuliers. SpIn gère plus particulièrement :

- des acteurs, qui montrent les actions des interlocuteurs distants, ils agissent dans l'interface.
- des documents, qui réagissent à une liste d'événements particuliers (définie pour chaque document) auxquels sont associées des actions agissant localement sur le document ou une de ces parties. C'est cet ensemble qui peut être vu comme une application, mais l'architecture comme l'interface sont centrées sur les documents. Ces actions sont génériques et définies dans le système (SpIn), comme celles pour manipuler, ou sont développées pour un document particulier.

SpIn a donc une approche centrée sur les documents, qui sont affichés «librement», sans cadre ou délimiteur (comme les fenêtres des applications dans les interfaces 2D). Cet agencement permet de donner toute son importance au document lui-même, sans noyer sa représentation autour d'autres éléments graphiques de l'interface. On peut agir directement sur le document.

Quel moyen pour manipuler les documents?

Pour appliquer une action, en particulier celles qui permettent de manipuler une représentation en trois dimensions, il faut agir d'une certaine manière sur le document. Une fois le document sélectionné, on a donc le choix du mode d'action pour le manipuler : utilisation du clavier, du capteur de position 3D, etc.

Nous pourrions utiliser le capteur de position 3D recommandé pour le pointage et la sélection de l'objet. Nous continuerions ainsi à faire de la manipulation directe. Associé à un autre capteur de position 3D dans l'autre main (non-dominante), on manipulerait dans ce cas directement le document des deux mains, d'une façon très proche de la réalité. L'utilisateur se trouve alors dans une situation connue et qui nécessite peu d'apprentissage. Par exemple, le fait de prendre l'objet par deux points et de déplacer les capteurs permet de déplacer l'objet. On peut se servir également du déplacement d'une main par rapport à l'autre pour mesurer les translations et les rotations appliquées à l'objet.

Cependant, le mouvement des mains dans l'espace pour définir les mouvements sur l'objet, s'il présente l'avantage d'être intuitif et naturel, présente des défauts notables :

- il n'y a pas de retour tactile, ni d'effort sur l'objet, empêchant d'avoir des mouvements de mains cohérents autour d'un volume fixe, ce qui oblige à corriger les trajectoires. L'utilisateur doit s'habituer à l'effet de ses actions pendant tout leur déroulement, du contrôle au résultat,
- les mains positionnées dans l'espace ne sont pas précises, les gestes sont sensibles aux tremblements et aux vibrations,
- la fatigue est un facteur important qui limite l'utilisation de capteur 3D positionné sur la main, surtout quand on doit l'utiliser longtemps,
- le mouvement complet des mains impose deux capteurs de position 3D mais celui des doigts impose des gants de données, encombrants et incompatibles avec notre contexte.

L'utilisation de deux périphériques peut cependant être intéressante. Le nombre de possibilités réduites de sélection et de manipulation d'un seul capteur, la transposition naturelle de l'utilisation courante des deux mains dans les tâches réelles, nous incitent à utiliser une interaction à deux mains dans l'interface. Si l'approche à deux périphériques isométriques (deux capteurs 3D) pose trop de contraintes, la fatigue en particulier, on doit choisir une autre voie.

Le problème de la fatigue se pose en fait également avec la simple utilisation d'un capteur de position 3D dans la main dominante. Si cela est indispensable pour désigner les objets, on peut s'en passer pour le reste : on peut utiliser un périphérique isométrique dans la main non dominante pour toutes les autres tâches. On profite ainsi des avantages des périphériques isotoniques et isométriques. Le rôle de chaque main est donc distribué ainsi : on utilise le capteur de position 3D (isotonique) dans la main dominante pour la désignation et la sélection et un périphérique isométrique à au moins trois degrés de liberté (comme un Trackball 3D) dans l'autre main pour réaliser les autres actions (manipulation, choix dans un menu, etc). On conserve ainsi l'intérêt du capteur de position 3D, en limitant au maximum son utilisation pour résoudre les problèmes de fatigue, et en permettant des actions synchrones des deux mains avec les deux périphériques.

En raison de sa fonction (choix dans des menus, manipulation d'objet, etc) et de son maniement dans la main non dominante, le périphérique isométrique est naturellement utilisé en mode relatif. Avec un périphérique possédant une bonne sensibilité, on peut réaliser des déplacements précis. En effet, c'est l'extrémité du bras, les doigts, qui ont la plus grande précision. En mode relatif, sur un périphérique précis et facile de prise en main comme la *SpaceMouse* (un trackball 3D), on peut réaliser des manipulations de document de façon bien plus précise qu'avec un capteur de position, et ce même avec la main non dominante.

Des expériences d'interaction bimanuelle ont déjà été menées sur d'autres modes et avec d'autres périphériques [Balakrishnan 99] [Hinckley 97b]. Elles montrent l'efficacité accrue des utilisateurs en mode bi-manuel dans les tâches de manipulation d'objets qui y sont étudiées. La réussite du mode bimanuel y est conditionné par le découpage des tâches affectées à chaque main.

Pour Paul Kabbash, la main non-dominante est capable de faire plus qu'approximer les mouvements de la main dominante [Kabbash 93], et peut être aussi efficace pour des tâches de manipulation si elles ne sont pas trop précises. Il utilise un périphérique «à point fixe» pour la main non-dominante,

comme un joystick, pour compenser les différences de vitesse d'exécution des deux mains. C'est un argument de plus pour notre répartition isotonique/isométrique.

On peut donc noter que l'utilisation de deux périphériques donne de manière générale un gain d'efficacité si l'on suit les critères énoncés au premier chapitre, comme la répartition adéquate des tâches à chaque main [Kabbash 94]. En ce qui concerne notre système d'interaction, nous attribuons un rôle précis et complémentaire à chacune d'elles, le pointage et la sélection au moyen d'un capteur pour l'une, puis l'action sur l'objet sélectionné pour l'autre, au moyen d'un trackball 3D. Les tâches des deux mains sont donc liées dans une même action, et ne sont donc pas indépendantes. Et la difficulté d'exécution est répartie en fonction de la capacité de chaque main (la tâche de pointage est réservée à la main dominante).

Bilan

Nous venons de présenter la démarche qui nous a amené à choisir une interaction bimanuelle : le choix d'un périphérique isométrique pour la main non dominante permet de limiter les efforts fournis par l'autre main avec le capteur 3D, utilisé désormais uniquement pour la désignation. Ce choix permet également de profiter de la dominante manuelle de l'homme, de ses capacités à travailler à deux mains dans le monde réel. En améliorant les moyens de contrôle de l'utilisateur sur l'ordinateur, nous lui donnons plus de moyens pour gérer les quantités d'information à sa disposition, et nous permettons aux tâches réalisées dans SpIn d'être décomposées au mieux en fonction des périphériques.

4.5 Bilan sur le modèle d'interaction

Nous venons de poser les règles de notre système d'interaction, qui constituent les fondations pour une interaction de plus haut niveau. Pour tirer partie des possibilités complémentaires des périphériques isométriques et isotoniques, nous avons choisi une approche hétérogène. Pour des raisons d'efficacité et pour un accès facile à l'ensemble de l'espace de travail, l'interaction, bimanuelle, utilise un capteur de position 3D pour déplacer un pointeur 3D dans toute l'interface suivant les trois axes, donc sans aucun décalage spatial ni temporel, et avec un degré d'intégration maximum (égal à un). Le capteur 3D ne sert qu'à désigner, le reste étant effectué avec un périphérique isométrique dans la main non-dominante. Pour la manipulation d'objets, ce dernier devra posséder au moins trois degrés de liberté homogènes (i.e. dont l'accès se fait suivant le même mode).

Nous utilisons un certain nombre d'indices perceptifs pour aider l'utilisateur, les ombres en particulier. Certains sont des indices dynamiques qui peuvent renseigner progressivement l'utilisateur, comme les boîtes englobantes progressives.

Cet ensemble de règles de conception peut être appliqué à tout environnement de travail tridimensionnel où l'on manipule des documents, sachant que :

- nous évitons les dispositifs trop chers ou trop encombrants (au sens du contexte bureautique de notre environnement),

Cette condition en ayant amené deux autres pour conserver un environnement appréhendable par l'utilisateur :

- nous limitons la navigation,
- nous interagissons à l'intérieur d'un espace fermé.

Ces trois conditions, issues du domaine d'application de SpIn, restreignent le domaine d'application de notre modèle aux environnements «bureautiques» centrés sur l'interaction avec des documents. Mais nous avons pu voir que notre modèle n'a pas été influencé au niveau du mode d'interaction par son domaine d'application, le travail coopératif synchrone.

Notre modèle est donc adapté aux environnements 3D, SpIn en particulier, dont l'objectif principal est de travailler sur des documents (et respectant les trois critères ci-dessus).



Maintenant que nous avons établi les propriétés de base du système d'interaction, nous allons nous intéresser à l'interaction à plus haut niveau, celle qui compose les éléments graphiques d'une interface de travail virtuel, comme les boutons, les widgets ou les menus dans une interface 2D classique.

Les différents éléments qui composent les métaphores d'interaction interne d'une interface 3D sont des transcriptions de l'espace 2D adaptées à la 3D, ou bien des éléments nouveaux, que seule la troisième dimension permet de mettre en oeuvre. Nous présentons ici ceux qui sont essentiels, en donnant leur adéquation avec SpIn, qui conditionne que nous les ayons développés et intégrés à notre système d'interaction.

Ces éléments essentiels sont ceux que nous estimons être les éléments d'interaction élémentaires d'une interface 3D. Il s'agit des moyens d'accès aux documents (*comment insérer un document dans l'interface de travail ?*) et des possibilités de réaliser des actions sur ces documents (*comment exécuter telle commande sur un document ?*). Lorsqu'une interface de travail coopératif comme SpIn a répondu à ces deux questions, elle est en mesure de répondre à toutes les exigences possibles d'une tâche quelconque développée dans son contexte. En effet, elle fournit alors les moyens d'accès et d'interaction avec les documents.

Nous commençons tout d'abord par un bref retour sur les concepts de SpIn (paragraphe 1). En effet, la prise en compte de l'interaction dans le modèle d'organisation de SpIn (§ 3.4.2 - p. 65) nécessite de revoir et de compléter certains concepts, comme l'espace attribué et mis en valeur pour interagir sur un document donné.

Puis, nous présenterons les éléments d'interaction essentiels de SpIn. Ceux qui permettent de gérer l'interaction sur les documents, c'est-à-dire qui permettent de sélectionner des actions à appliquer dessus, sous formes d'outils (paragraphe 2) ou sous forme de commandes (paragraphe 3). Et ceux qui permettent avant tout d'insérer un document dans SpIn (paragraphe 4).

5.1 retour sur les concepts de SpIn

Il s'agit de préciser et de compléter les principes de SpIn définis avant notre proposition de modèle d'interaction. Nous allons préciser comment rendre compatible le modèle d'organisation de l'espace de travail et le système d'interaction adopté.

Tout d'abord, nous allons aborder le problème de l'animation du bandeau, c'est-à-dire de la navigation dans SpIn.

On complétera ensuite le modèle d'organisation spatiale pour le rendre compatible avec notre système d'interaction, et permettre de manipuler les documents sans problème. Nous reverrons donc la notion de focus, liée directement à l'organisation de l'espace de travail.

5.1.1 la navigation

Comme nous l'avons vu au chapitre précédent, nous recommandons un modèle où l'utilisateur pourra aller pointer partout dans l'espace de travail. Cela nous engage donc à construire un espace fini. L'utilisateur devra avoir une vision globale de la scène. En limitant ainsi la navigation, on retire un paramètre important du changement de la composition de la scène. L'utilisateur perd moins de temps à essayer de se créer une représentation mentale de l'environnement virtuel.

Cette recommandation est en adéquation avec le modèle d'organisation spatiale des documents et des acteurs, autour d'une table de réunion, où la navigation est limitée au bandeau. L'utilisation d'un périphérique analogique est indispensable pour assurer la continuité de l'animation de ce dernier.

Un périphérique isométrique peut permettre un déplacement précis, sans mouvement «violent» de la main gauche, qui viendrait perturber celui de la main droite. Un trackball 3D paraît adapté à ces actions de type rotation/zoom (Figure 5.1). On peut imaginer qu'un périphérique dessiné spécialement pour l'occasion pourrait tout aussi bien remplir ces fonctions (comme un joystick équipé d'un potentiomètre ou d'un trackpoint).

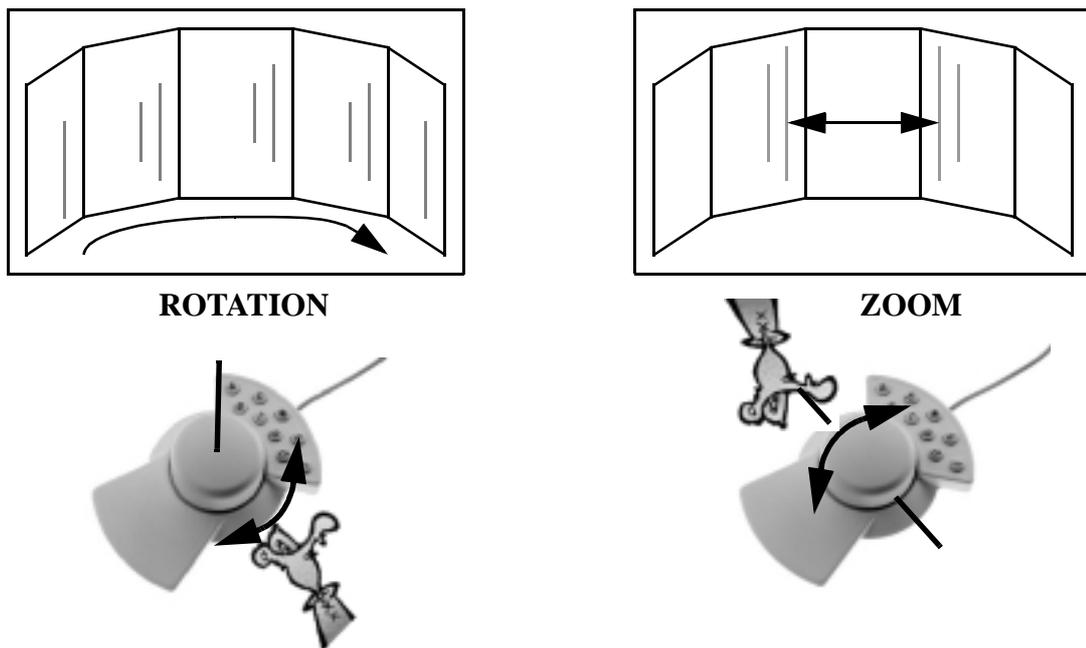


Figure 5.1 *la navigation dans SpIn*

Comme nous l'avons vu précédemment, le bandeau est manipulé avec la SpaceMouse. On utilise l'axe de rotation (Oy) pour le faire tourner, et l'axe de rotation (Ox) pour zoomer.

5.1.2 l'organisation spatiale de l'espace de travail

Lorsqu'un utilisateur veut manipuler un document, il doit pouvoir recentrer rapidement son attention sur ce dernier. Le document doit ensuite occuper une position centrale dans l'activité de l'utilisateur.

Nous plaçons les documents devant l'utilisateur, sur la table de réunion. Ce dernier peut prendre n'importe quel document placé dans le bandeau pour l'amener devant lui grâce à un outil d'échange (voir la barre d'outils ci-dessous). Ce nouvel espace d'interaction situé devant le bandeau, donne tout son sens à la métaphore de table de réunion.

Figure 5.2, on voit un appareil photo placé sur la table. L'utilisateur peut le manipuler et le manier à son aise. Le mode d'interaction privilégie l'interaction sur des documents tri-dimensionnels. Cependant, lorsque l'on observe les situations dans lesquelles SpIn pourra être utilisé, les réunions de petits groupes, il existe souvent un produit, un objet au centre des discussions. Cet objet ayant une réalité, il pourra donc avoir une représentation virtuelle dans SpIn. On pense par exemple à des réunions sur le design, la conception ou la vente de produits. Ce n'est plus la représentation sous formes de photos ou de maquettes statiques qui sera privilégiée, mais bien un modèle dynamique du produit.

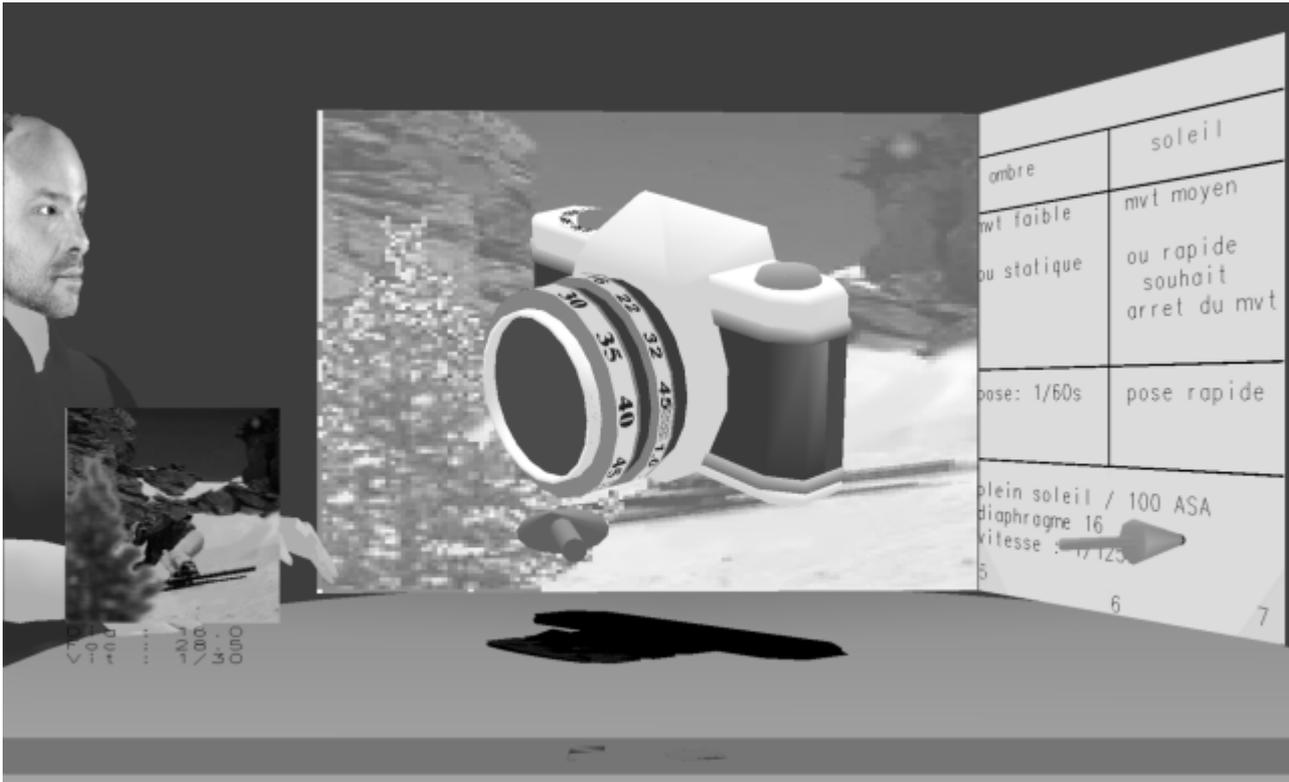


Figure 5.2 un objet est placé sur la scène

La table peut permettre également de placer un objet qui est au centre des discussions, c'est-à-dire qu'on ne manipule pas forcément soi-même, mais que l'on désire pouvoir observer en détail.

5.1.3 La notion de focus

Nous avons vu précédemment que le focus, ou centre d'intérêt d'un utilisateur, était essentiel à la compréhension globale de l'activité dans une réunion (§ 3.4.2 - p. 67). Dans SpIn, la déformation du bandeau amène l'utilisateur à positionner naturellement le document auquel il s'intéresse au centre du bandeau et à l'agrandir pour le consulter. Nous étendons cette notion de centre d'intérêt au document que l'utilisateur manipule, puisqu'il s'agit alors de son centre d'intérêt principal. Le focus peut donc être le document placé sur la table s'il y en a un.

Pour connaître le focus d'un utilisateur (bandeau ou table), il existe deux modes, l'un explicite, où l'on va demander à l'utilisateur de choisir, et l'autre implicite, où l'on va détecter automatiquement sur quel objet l'utilisateur est positionné.

mode explicite

On peut demander (indirectement) à l'utilisateur de choisir si il regarde le document placé sur la table ou au fond (au centre du bandeau). On définit alors deux positions d'observations, en corrélation avec la notion de mobilité relative définie précédemment :

- quand l'utilisateur regarde (ou manipule) l'objet posé sur la table, on le place en position haute sur la scène, il a alors un point de vue centré sur l'objet posé sur la table (Figure 5.3). Il peut le sélectionner et agir dessus. Le document est toujours visible, c'est le focus.

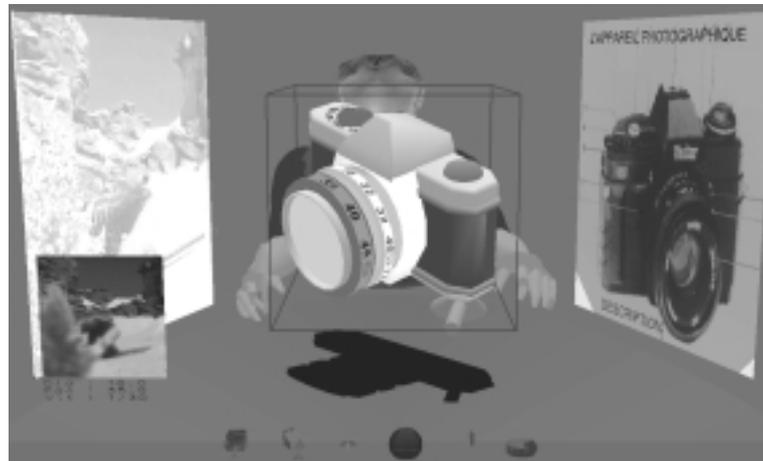


Figure 5.3 *point de vue sur la table de réunion, avec un angle de 10 degrés autour de l'axe des abscisses (par rapport à l'horizontale)*

- quand l'utilisateur n'observe plus l'objet posé sur la table, il regarde le bandeau, on le place donc en face de ce dernier (Figure 5.4). Le document posé sur la table devient alors partiellement transparent. L'utilisateur ne peut plus le manipuler ni le sélectionner, même si il distingue d'éventuelles manipulations d'autres utilisateurs dessus; c'est le document situé au centre du bandeau qui est en focus.

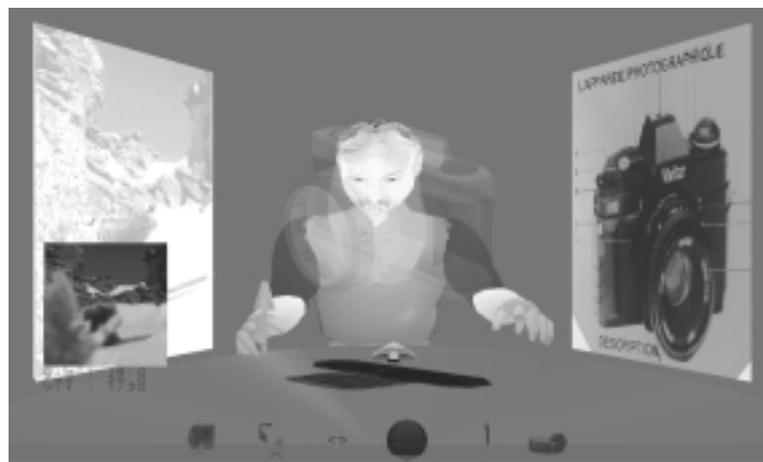


Figure 5.4 *point de vue sur le bandeau, on distingue le document sur la table en transparence*

L'utilisateur doit spécifier quand il veut passer d'un mode à l'autre. On utilise le trackball 3D placé dans la main non-dominante. L'utilisateur peut changer de focus quand aucun document n'est sélectionné, sinon il doit finir sa manipulation. En poussant la Spacemouse vers le haut, le focus se place sur le document posé sur la table. En poussant la Spacemouse vers le bas, le focus va sur le document ou l'acteur situé au centre du bandeau.

mode implicite

On peut plus simplement détecter le focus de l'utilisateur en conservant le même point de vue sur la scène. Il lui suffit pour cela de sortir son pointeur de l'objet posé sur la table pour que ce dernier devienne transparent (Figure 5.5), au point de ne plus gêner la lecture du document placé au centre du bandeau. L'effet de transparence permet à l'utilisateur de continuer à voir le document dans son champ de vision, de conserver un certain niveau d'attention dessus, ce qui lui permet de refaire passer son focus sur ce dernier si l'activité du groupe le lui suggère. Lorsque l'utilisateur replace son pointeur dans l'objet placé sur la table, ce dernier redevient opaque. Il n'y a donc pas de manipulation spécifique à faire, ce mode est plus naturel que le précédent, qui nécessite une manipulation spécifique.

Dans un soucis de simplification de l'interaction, nous essayons de limiter les manipulations de l'utilisateur liées à l'interface. Le mode explicite ajoute une action supplémentaire dans l'interface. Nous ne préconisons donc pas son utilisation tant qu'elle n'est pas jugée indispensable.

Dans un soucis de simplicité (pour l'utilisateur), nous avons donc préféré le mode implicite.



Figure 5.5 *Focus sur le bandeau, l'appareil photo est en transparence, l'utilisateur regarde les clones*

bilan

Nous venons d'étendre la notion de centre d'intérêt à tous les documents (et acteurs) de l'espace de travail, qu'on les édite ou que l'on se contente de les consulter. La transparence permet de profiter des capacités de l'attention visuelle de l'utilisateur [Harrison 95], qui se partage entre son centre d'intérêt à un instant précis, et celui qu'il vient de quitter (mais où il peut se passer encore quelque chose qu'il pourra continuer à suivre).

5.2 gestion des outils

Une interface de travail a besoin d'éléments qui permettent d'accéder aux fonctions principales du système, comme un sélecteur de fichiers ou des outils de coopération, composants essentiels de SpIn. En continuant sur l'idée de la manipulation directe, nous utilisons des outils vus sous forme d'icônes. Les icônes sont souvent formés de l'outil lui-même vu sous une forme réduite. Ils sont placés dans un gestionnaire d'outils, une palette, élément graphique qui permet d'organiser, d'accéder et de sélectionner un outil en particulier. Plus précisément, on a les critères suivants :

- l'accès aux différents icônes devra être aisé (donc des icônes suffisamment gros et éloignés les un des autres),
- la palette devra être accessible dès qu'un outil est susceptible d'être utilisé. On prend donc le parti de laisser la palette visible en permanence et de changer éventuellement son contenu, en fonction du contexte,
- la palette ne devra pas gêner graphiquement, c'est-à-dire laisser libre les zones de travail, la table de réunion en particulier.

La palette devra de plus être utilisable avec les mécanismes d'interaction de SpIn. On étudie ici les différentes solutions possibles :

5.2.1 palette d'outils

Des solutions basées sur des palettes d'outils réelles font leur apparition depuis quelques années. Il s'agit de placer une palette rectangulaire dans la main non dominante de l'utilisateur, et d'y afficher les outils disponibles en fonction du contexte. L'utilisateur peut aller chercher ces outils en cliquant dessus de l'autre main.



Figure 5.6 *palette bimanuelle du programme de modélisation de Chapel Hill*

palette bimanuelle

En situation immersive [Mine 96], Figure 5.6, la palette d'outils présente un grand intérêt puisque l'on peut donner la représentation voulue à la palette. On se sert de repères proprioceptifs pour agir sur la palette puisque la main gauche agit alors comme un référentiel pour la main droite, mais aussi de la vision puisque la position des mains dans l'espace (qui tiennent la palette et le pointeur) est connue, il suffit de positionner la palette à l'endroit exact dans le champ de vision en virtuel.

Cette approche n'est pas utilisable pour SpIn. L'utilisateur aurait alors une vue des deux palettes. L'une serait la palette réelle, utile à la manipulation, mais n'ayant pas de contenu sémantique. L'autre serait la palette virtuelle, qui permet de placer des outils. Ce type de solution a l'avantage de donner des informations tactiles puisque la palette a une consistance, et d'être manipulable facilement puisque placée dans chacune des mains. On utilise cette palette bi-manuelle également en réalité augmentée. Dans [Szalavari 97], le PIP (Personal Interaction Panel) supporte toute l'interaction de l'interface (Figure 5.7). L'auteur conclut que les utilisateurs s'approprient vite ce mode d'interaction, et n'éprouvent aucune fatigue. Le fait de maintenir dans ses mains une tablette de contrôle et un stylet, équipés chacun d'un capteur de position 3D, semble pourtant fatiguant (dans la mesure où maintenir un seul capteur en permanence l'est déjà).



Figure 5.7 *le PIP, système d'interaction entièrement basé sur une palette bi-manuelle*

bilan

Pour réaliser des actions entre les documents ou les documents et les acteurs, l'interaction directe au moyen d'outils paraît plus appropriée qu'une palette. Cependant, le principe de la palette bimanuelle pourrait être intéressant, toutefois la gestuelle (dans le dialogue) risque d'être encombrée par la palette. De même la fatigue, une de nos pré-occupation, risque d'intervenir souvent. Et enfin, si le caractère proprioceptif en fait un outil très adapté à l'immersion ou à la réalité augmentée, on devine plus difficilement son intérêt et son aspect pratique dans notre modèle d'interface où un affichage plus simple pourrait suffire.

Toolglass

Bill Buxton et son équipe ont étendu le concept de palette d'outils à celui de *Toolglass* [Bier 93] [Kurtenbach 97]. Le périphérique placé dans la main non dominante permet de déplacer une palette d'outils transparente. Le *Toolglass* (Figure 5.8) est utilisé dans des applications de dessin 2D, où l'utilisateur déplace et tourne la palette afin de placer un outil au dessus de la zone à modifier. Il n'a alors qu'à cliquer sur l'outil en question pour réaliser la transformation (le remplissage d'une zone par exemple).



Figure 5.8 *Toolglass*, un exemple où l'on choisit une forme (a), avant de l'appliquer (b)

Une extension de ce type d'outils, adaptée à l'interaction 3D, pour créer des palettes spécialisées dans SpIn pourrait être intéressante. Il faut cependant des activités adaptées à la gestion de type *Toolglass*, comme le dessin.

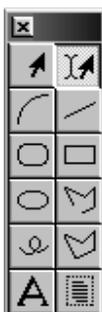
Ce type d'outils pourrait être proposé à l'avenir dans notre environnement pour être utilisé dans certaines activités développées sur SpIn.

5.2.2 le conetree

Dans ce cas précis, il ne s'agit pas réellement de *conetree* (voir paragraphe 5.4) puisque la structure des éléments à afficher ne possède qu'un niveau. Cependant, une version adaptée du *conetree* permettrait de ranger des outils vus sous forme d'icônes en quantité importante, autour d'un cône.

Mais cela ne correspond pas au premier critère énoncé : la manipulation du cône se fait en effet en plusieurs étapes, ce qui la rend moins facile que des solutions basées sur une simple palette. Le peu d'outils à afficher simultanément nous incite donc à choisir une structure simple. Le problème de la quantité d'information ne se posant pas, une structure tri-dimensionnelle comme le *conetree* n'est donc pas souhaitable ici.

5.2.3 barre d'outils



Comme palette d'outils ayant une structure simple, nous connaissons celles que propose couramment les applications 2D (voir ci-contre).

On se propose d'utiliser dans SpIn une palette toujours visible contenant les outils adaptés au contexte de l'activité (Figure 5.9), afin qu'ils puissent être toujours accessibles facilement. Elle est composée des différents icônes des outils, que l'on a simplement à sélectionner en cliquant dessus pour utiliser l'outil en question. Lorsque ce dernier est sélectionné, il prend alors la place du pointeur pour permettre à l'utilisateur d'agir sur un élément particulier.

Ces outils sont surtout ceux liés à la coopération. On peut citer en exemple :

- l'échange du document sur la scène avec un document placé dans le bandeau (afin de manipuler celui-ci),
- les outils de manipulation du point de vue sur un document (rotation et zoom).

Ou encore les outils de coopération (voir le chapitre suivant) :

- la prise de point de vue (d'un autre utilisateur sur le document placé sur la scène),
- l'envoi de document (à un interlocuteur).

La forme de la barre est un alignement des icônes placé en une ou deux dimensions. Elle ne doit pas gêner, donc nous la positionnons devant la table de réunion. Sa position assez basse et avancée au devant de la scène évite toute manipulation accidentelle.

Nous avons étudié plusieurs formes possibles, afin d'optimiser l'espace occupé, et d'améliorer l'esthétique. Sur la Figure 5.9, la barre d'outils et ses icônes sont aplatis sur le devant de la scène. Lorsque le pointeur arrive sur la barre (bas de la figure Figure 5.9), elle se lève automatiquement et on peut alors sélectionner un icône. Nous avons jugé cette forme trop lente d'accès (temps de montée de la barre), et pas assez visible (icônes aplatis).

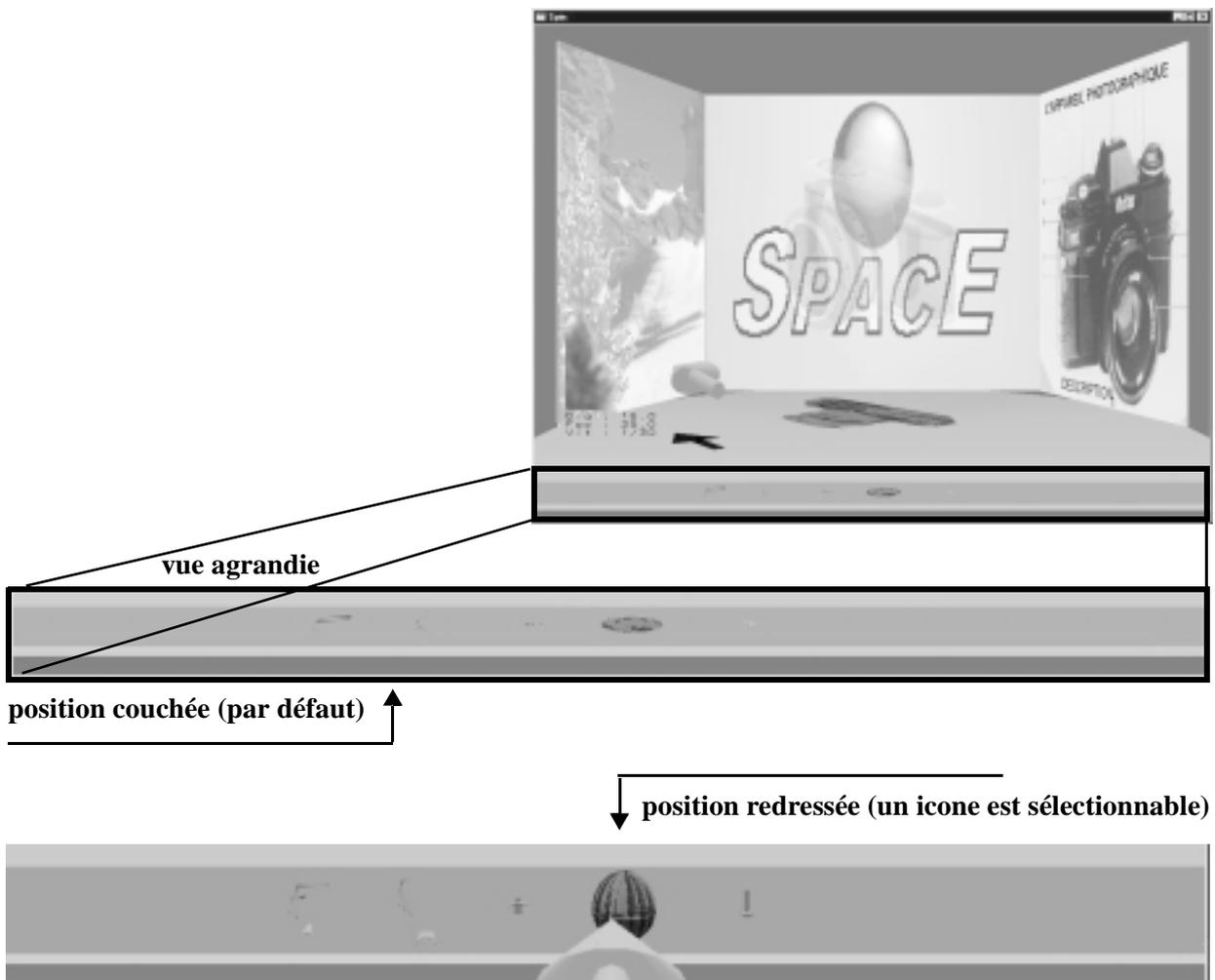


Figure 5.9 *la barre d'outils de Spln (deux positions)*

Nous sommes donc passés à une forme de barre où les icônes des outils sont visibles en permanence, de façon réduite (Figure 5.10). Lorsque le pointeur arrive sur un icône particulier, celui-ci grossit et s'illumine, on peut alors le sélectionner.



Figure 5.10 *la forme définitive d'outil dans SpIn, les icônes sont juste positionnés les uns à côté des autres*

Il faut remarquer qu'on ne place dans cette barre que les outils génériques de l'interface SpIn, en particulier ceux liés à la coopération. Les outils particuliers à un document sont accessibles à travers un «menu» contextuel d'actions (voir le paragraphe suivant).

5.3 gestion des actions

Si la barre d'outils dont nous venons de discuter est adaptée à des outils génériques affichés en permanence, elle ne l'est pas pour tous. Lorsqu'il s'agit d'appliquer des actions sur un document particulier, qui n'ont pas forcément de sens en tant qu'outil, il convient néanmoins de pouvoir matérialiser ces actions (pour pouvoir les sélectionner et les appliquer).

On connaît les menus classiques des interfaces 2D qui pourraient être ré-adaptés en trois dimensions. Comme on peut le voir Figure 5.11, les formes classiques de menu sont mal intégrées à la représentation des documents dans SpIn. De haut en bas et de gauche à droite, on peut voir un menu avec barre de menus, un menu contextuel, un menu contextuel transparent, et un menu circulaire (empruntés au logiciel *Maya* de Alias-Wavefront).

Ces représentations sont découplées du mode de rendu, elles ne profitent donc pas de l'éclairage et semblent découpées de l'espace de travail. Leur positionnement masque des informations, partiellement dans le cas de la transparence. Cela peut être particulièrement ennuyeux lorsqu'il s'agit du document auquel s'applique le menu. Etant dans une interface de travail coopératif, il est souhaitable que le centre d'intérêt d'un utilisateur ne puisse pas être masqué. De plus, si elles remplissent bien leur rôle en proposant une liste d'actions, elles deviennent difficile à utiliser lorsque la liste est importante, en particulier au niveau du temps de parcours de la liste. Si les menus circulaires permettent d'accéder à chaque option en parcourant la même distance, et donc dans les mêmes délais, ils semblent moins efficaces à utiliser [Dix 98]. Nous retiendrons tout de même qu'il est plus intéressant d'utiliser des affichages contextuels pour conserver l'attention de l'utilisateur autour de l'objet.

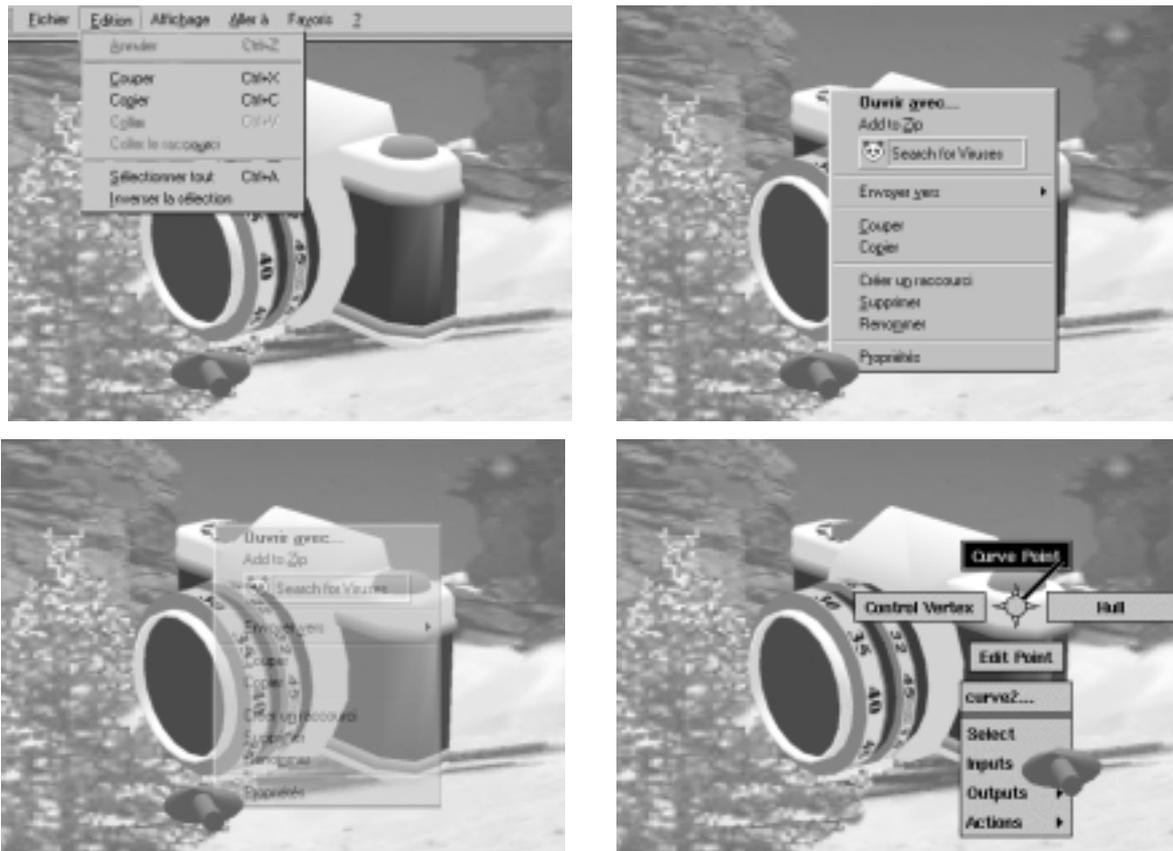


Figure 5.11 *menus classiques 2D sur un document 3D de Spln*

5.3.1 Les menus 3D

Le manque d'intégration dans l'environnement des menus 2D nous pousse à abandonner cette solution, et nous dirige vers les menus 3D. Dans ce cas, l'affichage du menu se fait en trois dimensions, au côté de l'objet, pour bénéficier du même contexte d'éclairage (phénomène propre à la 3D). On conserve ainsi un environnement homogène.

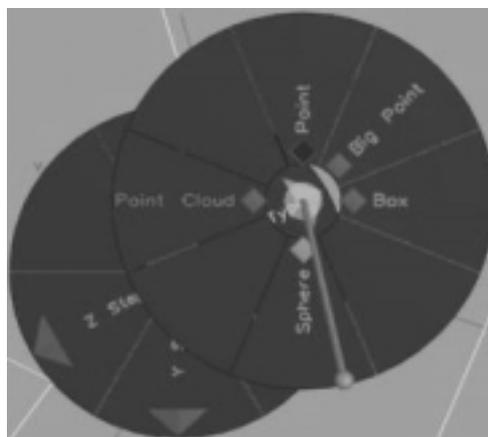


Figure 5.12 *Sundial menu*

[Shaw 97] utilise des menus 3D (Figure 5.12) dans son interface 3D de visualisation d'informations. Son système a l'inconvénient des *pie-menus* (menus circulaires) dans les interfaces WIMP, ils occupent beaucoup de place. L'idée du menu circulaire est pourtant intéressante puisque l'information est placée autour de l'objet. Ensuite, on peut utiliser une représentation tridimensionnelle adaptée dans le cas d'un nombre important d'éléments à placer dans le menu. Enfin, l'animation et l'utilisation des deux périphériques peut nous permettre d'améliorer la vitesse et la facilité de sélection dans le menu.

Le fait de placer le menu autour de l'objet évite toute ambiguïté sur l'objet concerné (car placé au centre), et ne masque pas l'objet tout en lui laissant un espace assez grand pour l'afficher. Il peut contenir des icônes 3D ou 2D, et éventuellement un texte d'information (on reprend la fonctionnalité des Toolbars 2D dans les menus 3D en utilisant des symboles plutôt qu'exclusivement du texte). Les sous-menus peuvent s'emboîter autour de la sélection d'une première option, de la même façon, en circulaire.

5.3.2 Les menus 3D dans les jeux

On peut remarquer que beaucoup de jeux sur ordinateur utilisent maintenant des menus circulaires (à une dimension), composés d'icônes 3D répartis le long d'un cercle en perspective. Si l'effet esthétique est indéniable, c'est souvent le seul objectif puisque l'interactivité n'est pas toujours au rendez-vous. Il s'agit toujours de menus à crans, où l'objet placé sur l'avant (ou sur le côté) est sélectionnable. Enfin la vitesse de rotation, si elle permet des effets animés intéressants, ne permet pas une sélection très rapide. On gagne rarement en visibilité sur ce type d'objet, le placement du menu en perspective masquant une partie de son contenu.

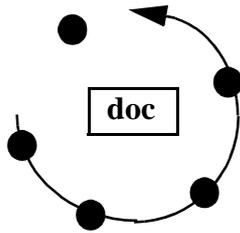
5.3.3 Propositions

Une forme circulaire pour les menus se prête bien à une sélection à l'aide d'une rotation. On contrôle mieux une situation de rotation suivant un axe qu'un mouvement de pointage, même si ce dernier est plus direct. Encore une fois, on pourra coupler les deux modes pour obtenir une action bimanuelle, la *Spacemouse* permettant de tourner le menu vers l'option que le pointeur du capteur 3D peut alors aller sélectionner.

On utilise des icônes dans nos exemples de menus, mais il pourrait tout aussi bien s'agir d'étiquettes textuelles. L'apparence des icônes, s'ils sont utilisés à la place ou en complément de champs textuels, a une importance cruciale, puisqu'aidant l'utilisateur dans la compréhension de l'action associée. On espère donc à l'avenir voir apparaître des bibliothèques d'icônes 3D animés aussi riches que celles en deux dimensions.

Suivant la quantité d'actions à placer dans le menu, on peut l'organiser suivant plusieurs dimensions :

Une dimension



On place circulairement les icones, dans un plan parallèle à l'écran (Figure 5.13). Le champ du menu sélectionnable est celui situé au milieu à gauche, en rouge. Le menu se manipule simplement avec un axe de rotation du trackball 3D (O_y).

On peut placer raisonnablement une dizaine de champs dans ce menu.

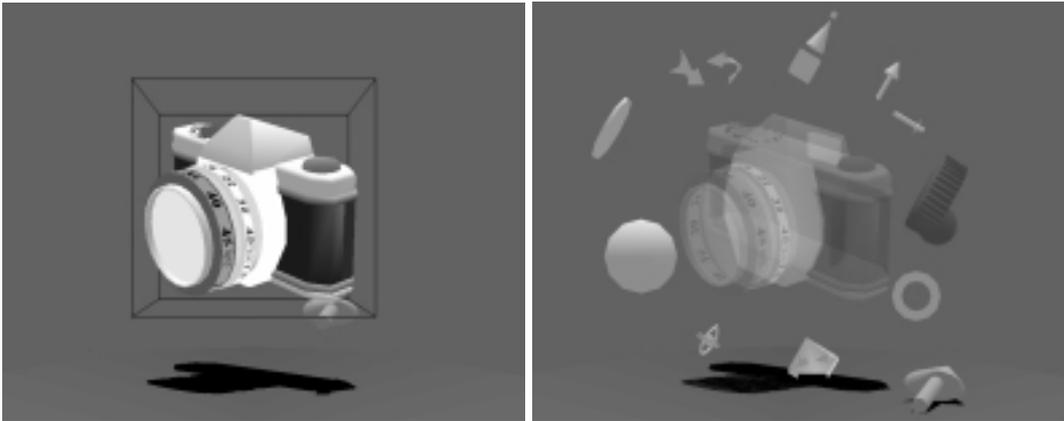
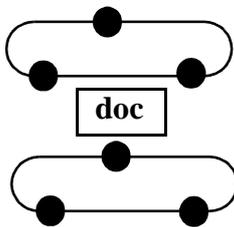


Figure 5.13 *Menu 1D, on sélectionne l'objet pour faire apparaître un menu autour*

Deux dimensions



On place les icones autour de deux anneaux superposés, orthogonaux au plan de l'écran, et situés de part et d'autre du document (Figure 5.14). L'utilisateur peut tourner les anneaux ensemble avec le trackball 3D (utilisation de l'axe de rotation O_y). Il peut pousser ce dernier vers le haut pour faire monter l'icone se situant en dessous afin de le sélectionner, ou vers le bas pour l'icone du haut. L'icone sélectionnable s'affiche en rouge.

L'animation aide l'utilisateur à comprendre l'organisation des icones. L'interaction n'utilisant toujours qu'un degré de liberté, elle est toujours évidente à appréhender. On peut placer deux fois plus d'icones dans ce type de menus que dans le précédent.

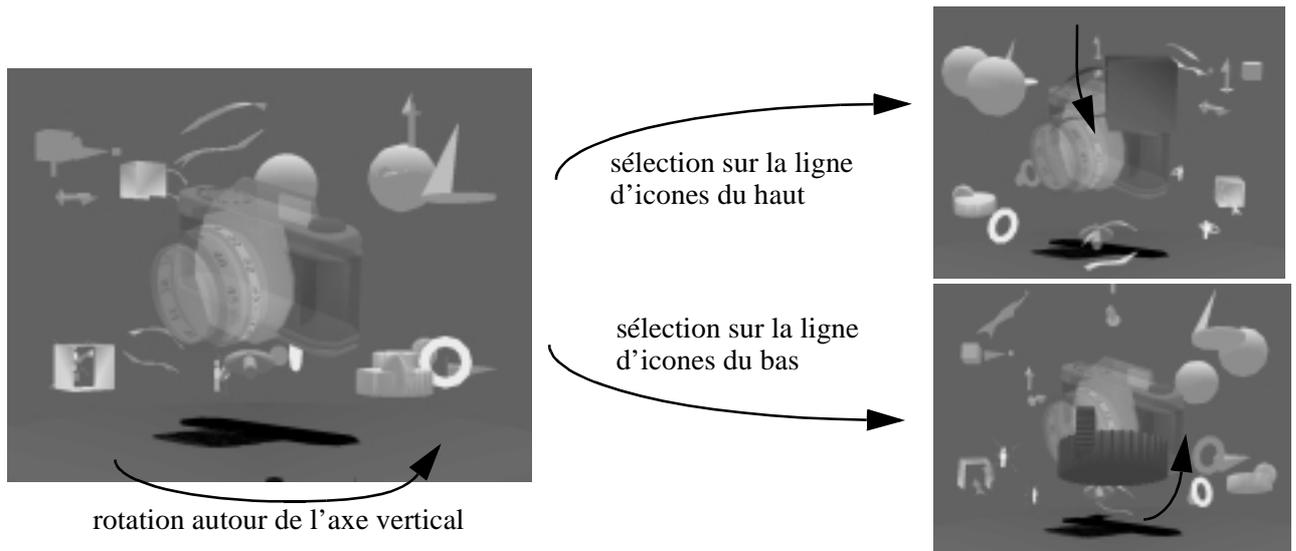
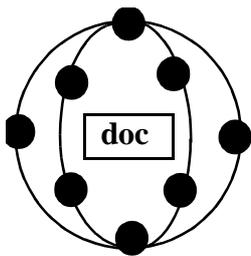


Figure 5.14 *Menu 2D, les icônes sont positionnés circulairement dans deux plans horizontaux*

Trois dimensions



On répartit les icônes autour d'une sphère. On utilise les trois degrés de liberté du trackball 3D pour tourner à sa guise la sphère composée de champs, jusqu'à l'option désirée. C'est le champ situé (arbitrairement) au sommet de la sphère (Figure 5.15) qui est sélectionnable, et affiché en rouge.

Ce menu devient vite inutilisable lorsqu'il contient trop de champs, les icônes risquant de se masquer entre eux et de devenir trop petits pour tous tenir autour de la sphère. Cependant, avec un système de visualisation adapté, et l'utilisation de couleurs pour regrouper les champs, on peut réussir à en placer plusieurs dizaines autour d'un document. C'est le cas de ce menu sphérique, en trois dimensions.

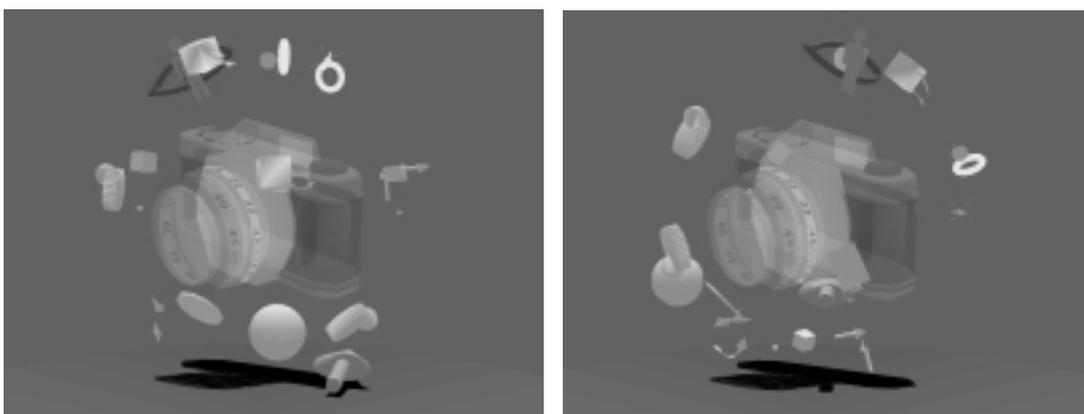


Figure 5.15 *Menu 3D, les champs sont placés le long d'une sphère englobant le document*

5.3.4 Bilan

Il existe beaucoup de possibilités de menu contextuel en trois dimensions. Nous en avons montré ici quelques exemples utilisés dans SpIn. Tout tient dans l'organisation globale des champs. Lorsque le menu en contient beaucoup, on se rapproche de modèles de visualisation d'information, en reprenant leur idée de dynamique et d'organisation spatiale. Cependant les menus contiennent généralement peu d'actions, ou sont alors organisés sous formes de menus hiérarchiques.

L'idéal étant en général d'avoir peu d'option dans un même menu, le menu à une dimension sera généralement suffisant ! L'utilisation des trois dimensions ne se justifie donc pas, en particulier pour le dernier modèle, en forme de sphère, moins évident à manipuler que les deux autres.

L'interaction à deux périphériques permet une sélection rapide dans ce type de menu. L'essentiel est désormais l'apparence du menu, avec des champs bien identifiables par l'utilisateur.

5.4 Accès aux documents

Dans une interface de travail informatique, pour accéder à un document il faut pouvoir charger le fichier correspondant. La gestion des fichiers est donc importante puisqu'elle permet d'accéder aux éléments de base d'une activité quelconque : les documents. Nous nous sommes donc intéressés aux possibilités offertes par la troisième dimension pour ouvrir un fichier.

5.4.1 Visualisation

Le but est de trouver et de sélectionner un fichier dans une hiérarchie de répertoire. Si on observe le système d'ouverture d'un fichier (Figure 5.16), commun aux interfaces 2D, on peut faire les remarques suivantes :

- on ne peut voir qu'un seul niveau de la hiérarchie à la fois,
- tous les fichiers et répertoires ne sont pas visibles, il faut faire défiler la liste horizontalement,
- on peut sélectionner un fichier en cliquant sur «Ouvrir» ou directement dessus.

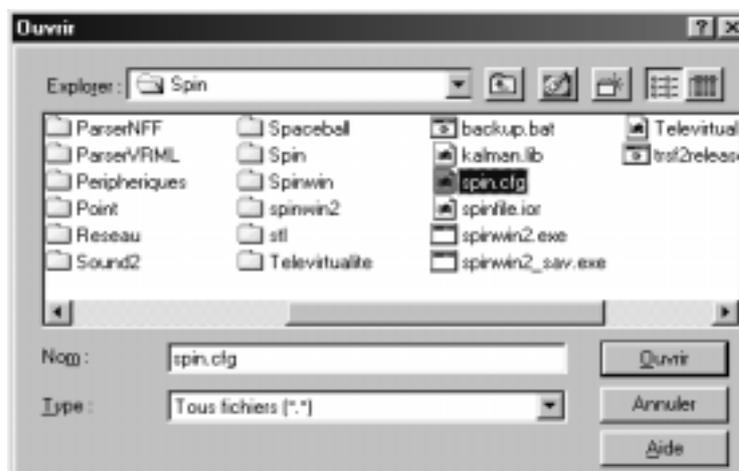


Figure 5.16 sélecteur de fichier classique de Windows

Ces problèmes de lisibilité nous engagent à observer les gestionnaires de fichiers. Si leur fonction est de manipuler les fichiers, et pas seulement de les ouvrir, ils peuvent proposer des modes de visualisation intéressants.

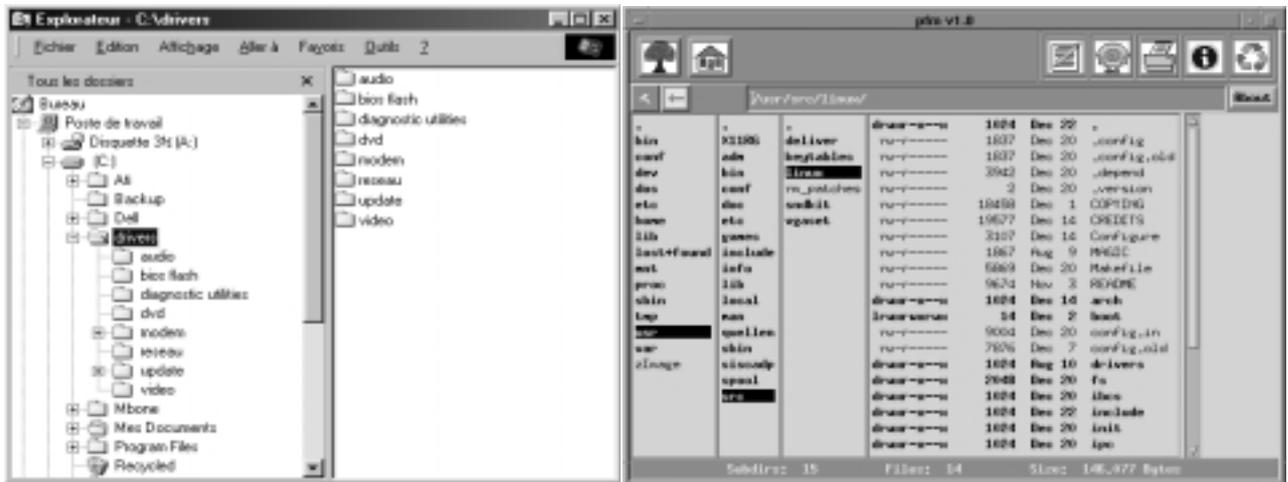


Figure 5.17 gestionnaires de fichiers Windows et Pfm (Unix)

Le gestionnaire de gauche (Figure 5.17) présente les répertoires sous la forme d'une liste écartée de répertoires (à gauche). On peut ainsi développer à sa guise telle ou telle branche, mais on utilise les ascenseurs pour voir toute la hiérarchie, et l'information est partiellement dupliquée avec la fenêtre de droite qui montre les répertoires et fichiers. *Pfm* (gestionnaire de droite Figure 5.17) reprend le style du gestionnaire de fichier de *NextStep* en utilisant une liste en colonne de répertoires, qui a l'avantage de montrer toute la hiérarchie (3 niveaux maximum affichables) de répertoire jusqu'au fichier sélectionné. Il possède également un système de signets (icône en haut). Cependant, aucun des deux ne résout le problème de visibilité puisque l'on retrouve les listes défilantes munis d'ascenseur.

Le *Conetree* [Robertson 91] est issu des travaux de Xerox sur la visualisation d'information, et permet l'affichage d'arborescences (Figure 5.18) où chaque nœud comporte de nombreux fils (comme le cas qui nous occupe ici). Tous les fils d'un même nœud sont représentés «en rond» en profondeur, minimisant ainsi la place occupée sur l'écran. On remarque qu'en présence de beaucoup de nœuds, des informations sont masquées par les données placées au premier plan; ce qui nous amène à dire que le problème de visualisation n'est pas complètement résolu. Mais l'affichage est dynamique et interactif : l'utilisateur peut facilement et instantanément faire tourner les fils d'un nœud pour faire apparaître ceux qui sont cachés. Le cône permet de visualiser directement la taille de la liste, sans widget supplémentaire. Il s'agit d'une solution efficace pour visualiser interactivement le contenu d'un répertoire. On peut noter qu'il existe d'autres travaux dérivés sur la visualisation d'arborescence de listes [Munzner 97] [Najork 95].

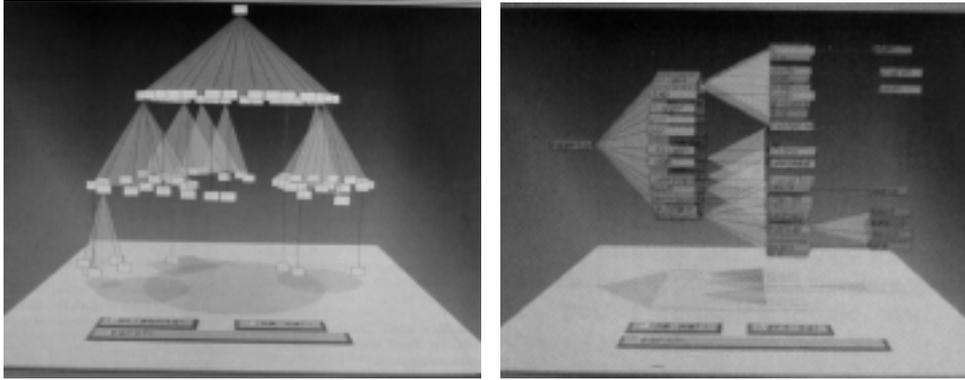


Figure 5.18 *Conetree*, deux vues différentes

Tout comme dans le système courant d'ouverture des documents dans les interfaces WIMP (Figure 5.16), il est suffisant d'afficher un seul niveau de la hiérarchie, l'objectif n'étant pas de réaliser des opérations entre fichiers mais d'en sélectionner un seul (Figure 5.19).

5.4.2 Interaction

La manipulation n'est pas très simple, le *conetree* possède deux degrés de liberté, mais complètement dissociés. On privilégie à nouveau l'interaction directe, en essayant de simplifier au maximum et en tirant profit de l'interaction bimanuelle.

On utilise le trackball 3D pour faire tourner le cône. Un fichier est sélectionnable par défaut, il apparaît en jaune. Il s'agit de celui placé sur l'avant du cône. On peut plutôt en choisir un autre en pointant dessus avec le capteur 3D. Il suffit dans tous les cas de cliquer avec le bouton du capteur pour le sélectionner.

Pour développer un nouveau répertoire, on fait un drag'n drop très court d'une étiquette. On laisse, de même que pour le pointage d'une étiquette, la possibilité à l'utilisateur d'utiliser le trackball 3D avec un déplacement latéral (axe Ox) de la SpaceMouse pour développer un niveau.

Le chemin parcouru est visualisé sous la forme d'étiquettes successives sur lesquelles on peut cliquer pour revenir en arrière. On peut également à nouveau utiliser le trackball avec un mouvement latéral en sens inverse pour reculer d'un niveau et remonter l'arborescence de fichier.

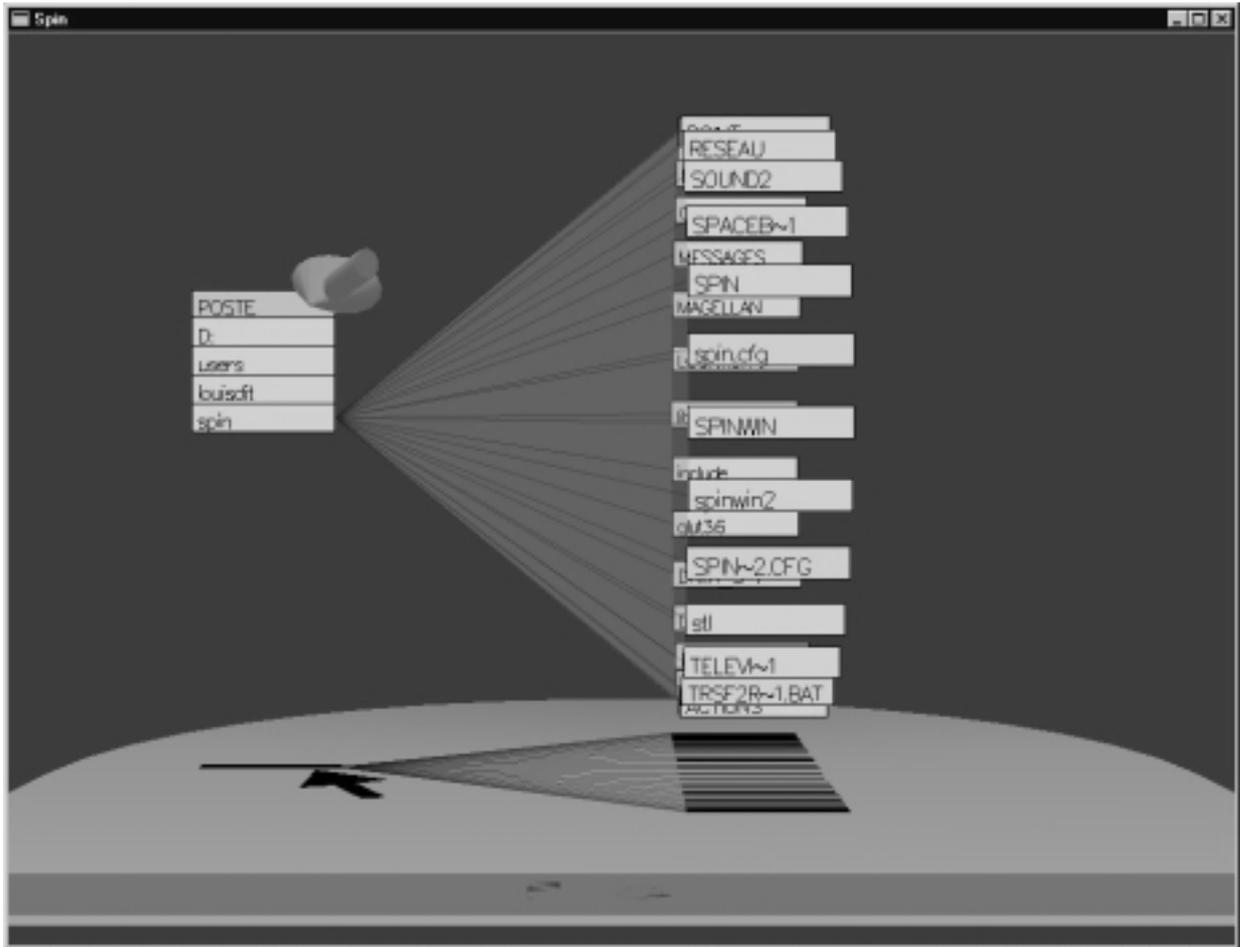


Figure 5.19 Sélecteur de fichier de SpIn

Le sélecteur de fichier est accessible à partir de la barre d'outils. C'est un outil générique qui permet de sélectionner un fichier pour le placer dans le bandeau ou encore pour le partager avec d'autres utilisateurs.

5.5 Bilan

Nous venons de décrire :

- la gestion des documents (ouverture),
- la gestion des outils, composants de SpIn qui permettent d'agir sur les documents,
- les menus 3D contextuels qui permettent de regrouper les actions applicables à un document.

Cependant, lorsque l'on sélectionne un objet (i.e. on clique dessus), l'utilisateur a souvent directement accès à la seule fonction particulière d'un document. Dans le contexte de SpIn, il s'agira souvent de manipuler le document ou une de ses parties : c'est l'action la plus fréquente. Les autres actions applicables dans le cadre de l'activité de la réunion, sont rendues accessibles à travers des outils, des outils de coopération, permettant de distribuer un document ou prendre le point de vue d'un utilisateur. Il n'y a alors pas besoin de menus. C'est grâce à ce principe que l'on simplifie

l'interaction en limitant l'usage des périphériques avec un seul bouton en tout (par exemple, Windows 9x utilise un bouton pour sélectionner un objet et un bouton pour afficher un menu contextuel).

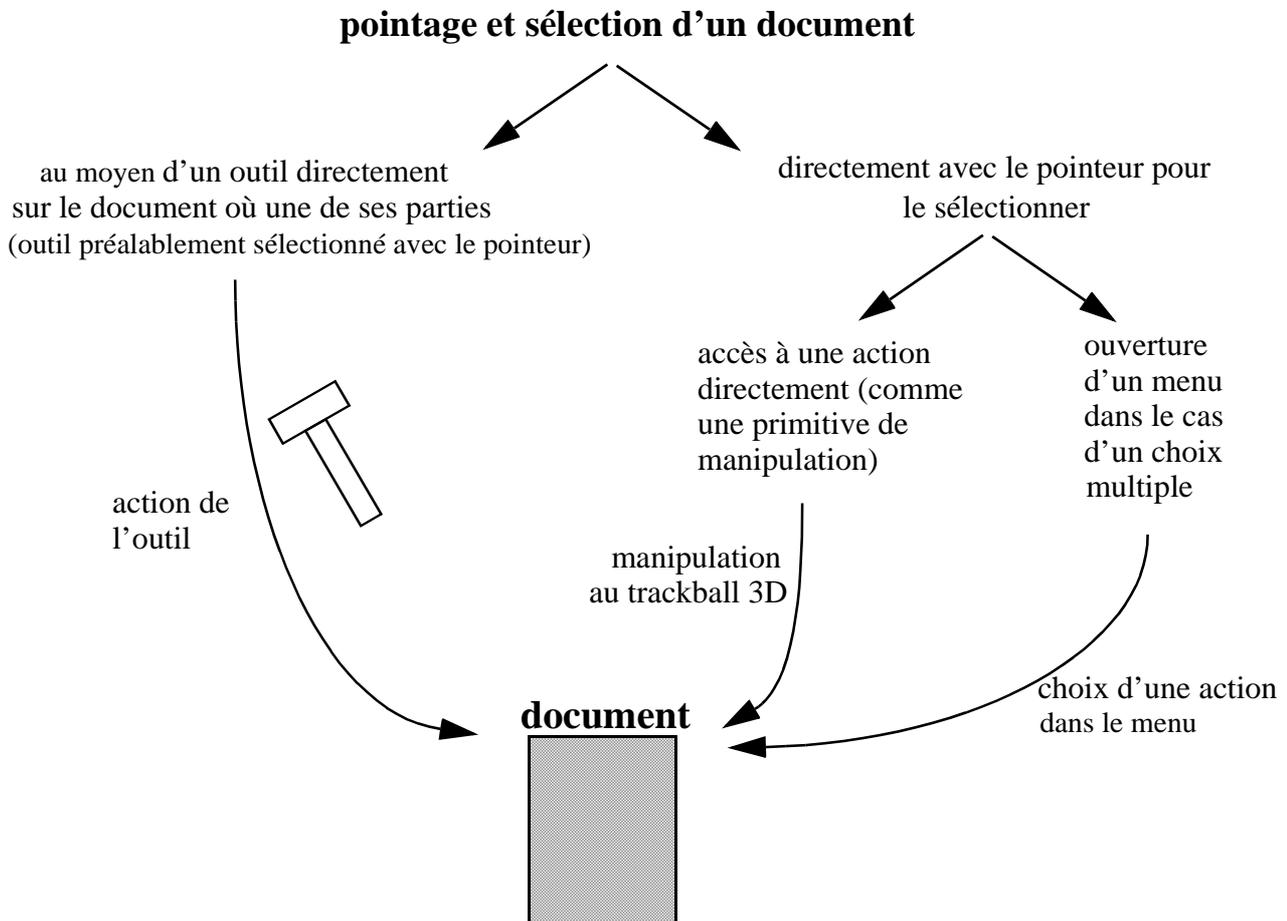


Figure 5.20 *découpage de l'interaction sur un document dans Spln*

On peut se contenter souvent des deux modes d'interaction directe dans la gestion de l'interaction d'une réunion, et essayer de se passer des outils d'interaction indirecte, en particulier ceux à choix multiples comme les menus (Figure 5.20). L'activité de l'utilisateur n'en sera que plus naturelle et se trouvera facilitée. C'est un point important pour l'impliquer dans la réunion, et tenter de lui faire oublier l'outil pour qu'il se concentre complètement sur sa tâche.



Une meilleure intégration des éléments dans un espace 3D permet à l'utilisateur d'avoir une cohérence et une facilité d'approche de l'environnement de travail coopératif plus importante qu'en 2D (cohérence dont on a assez peu besoin pour des applications résolument mono-utilisateur). Cet objectif passe à travers l'utilisation et la mise en valeur des clones 3D, en montrant qu'ils renforcent la sensation de présence, mais surtout de collaboration entre les participants.

Nos situations de travail coopératif autour de SpIn correspondent à des réunions ayant un but précis, étayées par des documents. On doit pouvoir dériver ce modèle en télé-enseignement, en maintenance, en commerce, etc. La condition est donc de réaliser des actions à l'intérieur du groupe, vers des personnes du groupe ou des documents partagés, dont les principales sont exposées Figure 6.1.

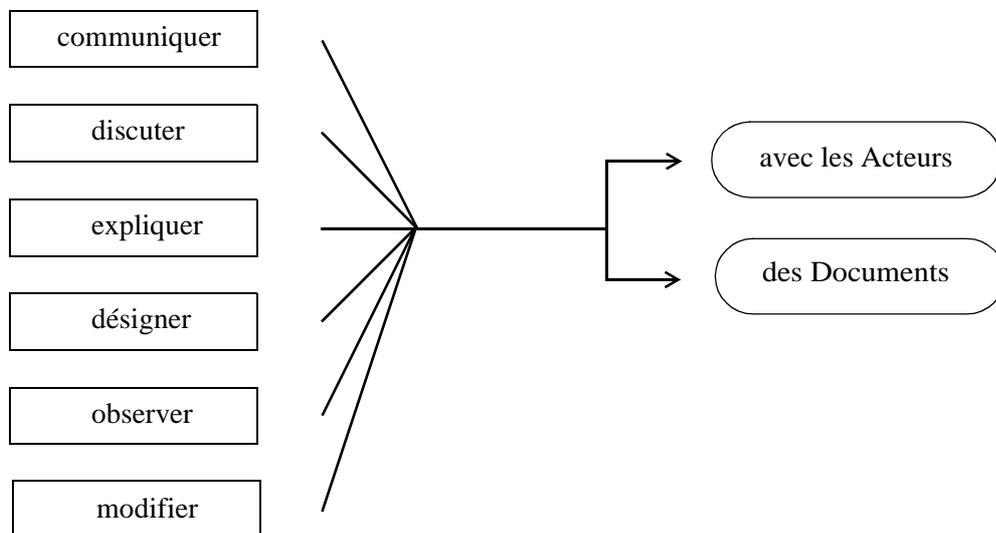


Figure 6.1 *primitives de coopération*

Cette figure montre les primitives indispensables à un système de travail coopératif. On peut les utiliser pour décrire l'activité de la réunion, et en déduire, à l'aide des principes tirés de notre analyse sur le travail coopératif synchrone (chapitre 3), les éléments qui devront faire partie de SpIn :

- la présence visuelle et sonore des interlocuteurs,
- le contact visuel, savoir qui regarde qui,

- les expressions faciales et les gestes, qui donnent des informations sur ce que perçoivent les gens,
- le centre d'intérêt, donné par la direction du regard et qui indique sur quel document l'utilisateur travaille,
- le point de vue sur un document, c'est à dire quelle partie il regarde en particulier,
- le point d'action sur un document, c'est à dire sur quelle partie il agit,
- montrer les modification du document en temps réel.

Nous allons maintenant décrire comment ces éléments de coopération sont pris en compte dans SpIn.

6.1 La téléprésence

La téléprésence décrit le phénomène de la projection d'un utilisateur dans un environnement virtuel distant. Il s'agit de **la définition simple** de la téléprésence d'après [Draper 98], qui en dénombre deux autres :

la définition cybernétique : la téléprésence est une mesure du degré de qualité dans l'interface homme-machine. Ce qui correspond à la compatibilité, au niveau comportemental et physique, entre l'homme et l'ordinateur ou l'homme et le robot piloté (en télé-opération). Dans ce sens, nous avons essayé dans SpIn de fournir un système d'interaction adapté à l'activité de l'utilisateur.

la définition expérimentale : la téléprésence décrit un état psychologique qui intervient lorsque l'utilisateur se sent physiquement présent dans l'environnement virtuel distant.

Le modèle de visualisation de SpIn positionne l'utilisateur au centre des informations, de l'activité, et lui permet d'interagir, sans le représenter à l'intérieur de cet espace. La table de réunion virtuelle simule un contexte de réunion réel. C'est cet aspect simulation qui donne conscience à l'utilisateur de sa position dans le groupe et l'activité, en recomposant une situation réelle. On ne le place pas pour qu'il ait la sensation d'agir à distance ou d'être ailleurs, il reste sur place, physiquement mais dans son activité également. L'utilisateur est en effet conscient de la présence des autres chez les autres grâce aux facteurs suivants :

- la simulation de l'organisation spatiale réelle d'une réunion (autour de la table),
- le contexte bureautique qui permet de conserver le contact avec son environnement immédiat,
- l'environnement non-immersif, avec la réunion se déroulant sur un terminal individuel de communication (un écran d'ordinateur).

On s'est donc intéressé à la structure de l'espace de travail virtuel pour donner la sensation de réunion à l'utilisateur, ce qui passe par la représentation de ses interlocuteurs. L'objectif étant de construire un environnement qui donne la sensation de travailler à plusieurs sur les mêmes documents en temps réel. Cela rejoint le concept de CVE (Collaborative Virtual Environment) de Steve Benford, auquel nous nous attachons dans SpIn. Il consiste à renforcer la sensation de collaboration entre les interlocuteurs pour qu'ils se sentent immergés dans la réunion, notion différente de la téléprésence au sens expérimental, et plus proche de la définition simple.

Pour approcher au mieux cet état, il convient de choisir une représentation des utilisateurs adaptée à nos contraintes. Nous avons pris le parti d'utiliser dans SpIn une représentation anthropomorphique des utilisateurs, pour plusieurs raisons :

- un des éléments importants dans la réunion est l'identification sans équivoque, de ses interlocuteurs,
- Il faut simuler un certain nombre d'éléments de la réalité pour conserver une facilité d'interprétation de l'utilisateur qui pourra retrouver des repères connus, en particulier :
- une représentation réaliste permet de reproduire les éléments essentiels à la communication, les expressions faciales en particulier (voir paragraphe suivant),
- le clone permet de re-crée de façon naturelle les primitives servant à informer sur l'activité d'un utilisateur (citées ci-dessus), comme la direction du regard.

Lorsque l'on propose à des professionnels un système qui leur permet de créer une réunion en virtuel, ils sont soucieux, comme dans la réalité, de leur image et de son impact. Si les solutions de visio-conférence reproduisent exactement la réalité, ce n'est pas le cas des environnements virtuels qui peuvent proposer des solutions alternatives, ne véhiculant que les informations essentielles à la tâche, comme dans Massive [Greenhalgh 95], mais pas de représentation réaliste des intervenants. Pour réaliser une adaptation future de SpIn dans le monde professionnel, et en se basant sur les améliorations de la représentation des humains virtuels dans les systèmes de synthèse d'images temps réel, nous utilisons des Clones 3D.

Il s'agit de la représentation synthétique d'une personne, la plus fidèle possible, d'où le terme de clone. On rencontre surtout ce type de représentation sous forme d'objets 3D animés. Ces clones peuvent ne pas être ceux d'un humain (clone d'un personnage de dessin animé par exemple) ou ne pas être celui de la personne désignée (utilisation du clone d'un tiers). En ce qui nous concerne, nous travaillons avec des clones réalistes identifiant exactement les personnes réelles, ce qui correspond au besoin du TCAO synchrone.

Nous utilisons les clones (Figure 6.2) créés avec le système semi-automatique développé dans l'équipe Télévirtualité [Télévirtualité] de l'Institut National de l'Audiovisuel (chapitre 1).

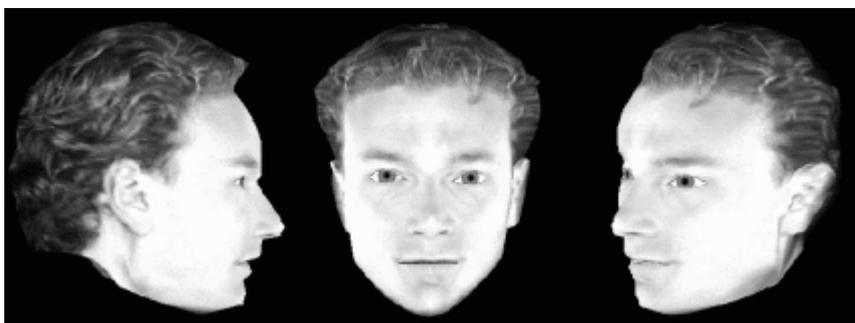


Figure 6.2 *un clone crée à l'Institut National de l'Audiovisuel*

6.2 Communication

Un système de communication viable doit comprendre la parole, mais également les mouvements du corps, des bras en particulier, et les expressions faciales [Cassel 94]. On peut certes citer le cas de petites réunions où les gens se connaissent et s'entendent bien, et pourront se contenter d'un système de communication dégradé (comme le téléphone...). Mais l'objectif de SpIn est de remplacer des réunions réelles, justifiant la présence de multiples personnes. Notre environnement pourra être un outil pour des gens situés à distance, mais également localement car si on peut réussir à faire un système de communication équivalent à un contact direct, le virtuel ouvre des possibilités plus importantes au niveau de la gestion des interactions.

Dans la mesure où nous pouvons évaluer l'impact d'un système de communication basé sur des clones grâce au système de l'INA, nous n'avons pas développé l'animation faciale ni les gestes liés à la communication dans SpIn. Nous sommes cependant conscient que c'est une fonction indispensable d'un système de travail coopératif synchrone, et qu'un tel élément devra faire parti un jour de SpIn. Mais notre travail sur les clones s'est situé au niveau de la coopération, en partant du principe que la communication était possible.

6.2.1 La voix

Notre système permet de diffuser la voix de chacun vers les autres. Le son est positionné par un effet stéréo en fonction de l'emplacement du clone à l'écran. La voix est un outil indispensable, qui doit pouvoir être transmise à distance avec une garantie de qualité. La gestion de la voix dans SpIn est dérivée des outils de diffusion MultiCast de Mbone.

Les problèmes de prise de parole n'étant pas dans les problématiques de départ de SpIn, cet aspect n'a pas été étudié. Nous préférons nous placer dans le cas critique d'une situation réelle, et laisser les prises de parole se régler naturellement sans contraindre et frustrer l'utilisateur.

On peut imaginer à l'avenir que pour les modèles dérivés, comme pour le télé-enseignement, on spécifie des droits inhérents à la prise de parole, avec une personne pour diriger la conversation.

6.2.2 L'animation faciale

Un clone dans un environnement de coopération, et donc de communication comme SpIn, doit disposer d'un système d'analyse-synthèse pour permettre l'animation faciale. L'équipe Télévirtualité de l'INA dispose d'un tel système, décrit au premier chapitre. Ce n'est qu'à travers l'amélioration de la puissance des processeurs, pour l'analyse de l'image, et de la puissance des dispositifs graphiques, pour l'animation faciale, que l'on améliorera la qualité de l'animation et du rendu du visage.

Pour perfectionner l'animation des lèvres, l'INA n'utilise plus de dispositif d'analyse basé simplement sur une caméra placée au dessus de la machine, mais explore des systèmes s'appuyant sur l'utilisation de la parole pour générer le mouvement.

Un des avantages qui permet de valoriser ce système d'animation, vient de son utilisation très réduite des réseaux. En effet, à l'heure où la vidéo nécessite le maintien de flux de plus de 10 ko/s dans le meilleur des cas (le pire pour la qualité d'image), l'envoi des paramètres d'animation faciale ne prend que quelques octets par seconde. Ce facteur est important à une époque où l'on voit les capacités des ordinateurs se démocratiser plus vite que celles des réseaux.

Le système n'ayant été disponible que tardivement sur la plate-forme matérielle de SpIn, aucune expérimentation grandeur nature n'a pu être envisagée.

6.2.3 Le geste

La fonction sémiotique du geste est produite par un clone grâce à un système de capture des mouvements. Il existe plusieurs solutions techniques pour capter le mouvement des bras. Nous allons ici livrer une première analyse du problème de la qualité de la détection des gestes dans les environnements virtuels, sachant que cet aspect n'a pas encore été évalué dans SpIn.

Une première solution est d'équiper l'utilisateur de plusieurs capteurs pour détecter le mouvement des bras ou des mains. Ce système est onéreux, demande un calibrage assez important, et impose d'équiper l'utilisateur de matériel encombrant pour un environnement comme SpIn.

Toutefois il est possible de réduire le nombre de capteur de position. On peut recomposer le mouvement des deux bras à partir de deux capteurs de position à six degrés de liberté, placés sur chaque main. Avec un système de calcul adapté, comme avec la cinématique inverse, on calcule les positions de chaque articulation. Avec seulement deux capteurs, on laisse une incertitude sur certaines valeurs, celle du buste par exemple, pour laquelle il faudrait un troisième capteur.

Ce système pose également l'inconvénient de ne pas donner la position des doigts. Mais a-t-on besoin de mouvements complets ? Dans le cas du geste utilisé pour le dialogue, on pourrait se permettre certaines approximations, qui permettraient de n'équiper que très faiblement l'utilisateur, et de ne pas le gêner. Une étude du geste sémiotique en environnement virtuel déterminerait le système le plus adapté. En particulier, on pourrait imaginer qu'à partir de la seule position de la main droite (et donc du bras droit), on puisse interpréter les positions des mains et du bras gauche, grâce, par exemple, à une base de données de gestes indexés par la position d'un seul capteur et de la prosodie. Cependant, le geste est un mode d'expression en soi et pas seulement un ajout à la parole. Il faudrait donc mesurer finement la dégradation de l'information pour savoir quel mode adopter en fonction de l'application.

Une deuxième solution consiste en un système de détection de la position des bras grâce à une ou plusieurs caméras placées autour de l'utilisateur [Blake 94]. Ce système a l'avantage de laisser libre de tout périphérique l'utilisateur, mais requiert une chaîne de traitement vidéo assez lourde en terme de calcul, et de matériel s'il faut plusieurs caméras.

Le système actuellement utilisé à l'Institut National de l'Audiovisuel et dans SpIn est un système à un capteur de position à six degrés de liberté par main. On pourrait imaginer à l'avenir que l'image vidéo qui permet l'extraction des paramètres d'animation faciale, fournisse également des informations sur la position des bras pour compléter l'information des capteurs.

6.2.4 Bilan

L'avantage du clone dans un système de communication sur d'autres formes d'avatars, est le rapprochement qu'il va effectuer entre la réalité et le monde virtuel :

- il augmente la conscience de l'utilisateur de la présence d'autres participants, par sa facilité d'identification,
- il limite les efforts de compréhension des mouvements de ses interlocuteurs, ceux-ci étant les mêmes que ceux d'un être réel.

Bien que nous le reconnaissons comme indispensable, nos travaux n'ont pas mis l'accent sur l'animation des clones pour la communication. Cet aspect est géré par le système de l'INA. Cette phase est en effet principalement un problème technique dans la mesure où l'on cherche à simuler la réalité. Nous avons orienté nos travaux sur les aspects homme-machine, c'est-à-dire l'intégration des clones dans un système de TCAO synchrone et leur valorisation à travers les informations qu'ils peuvent donner sur l'activité.

6.3 Coopération

Le clone est un modèle 3D sur lequel on possède un contrôle complet. Si ce modèle peut-être piloté directement par les mouvements de l'utilisateur en phase de communication directe, il peut également être une aide précieuse pour renseigner un utilisateur sur l'activité des autres.

En effet, les gestes de manipulation de l'interface d'un utilisateur ne constituent pas une information en soi, l'information pertinente est ce sur quoi ils agissent. Par exemple, si quelqu'un clique sur un bouton avec sa souris, son clone ne va pas reproduire le geste, qui n'aurait peut-être aucun sens dans les environnements distants, mais plutôt s'animer en fonction de ce que le clic a déclenché. Par une interprétation des actions de l'interlocuteur sur sa machine, on ajoute un niveau intermédiaire d'analyse pour simplifier et faire comprendre au mieux l'activité des participants. Nous préférons donc l'expression interaction homme-machine-homme, plutôt qu'interaction homme-homme médiatisée.

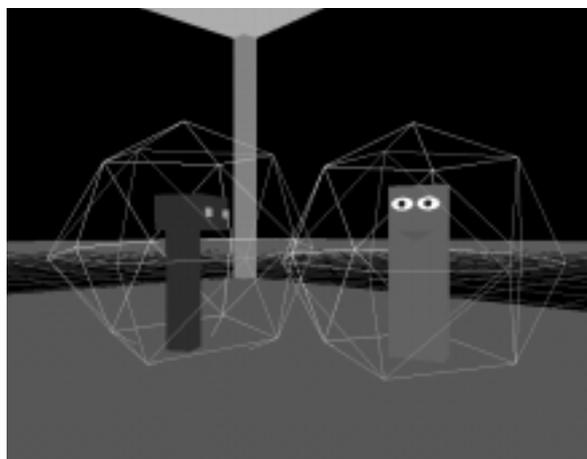


Figure 6.3 *les blockies de DIVE, avatars vecteurs d'informations*

On doit donc transmettre les actions d'un utilisateur liées à l'activité, pour animer son clone. Nous avons vu en introduction quelles sont les informations pour un système de travail coopératif synchrone. Nous pouvons recouper cette étude avec les analyses de référence des systèmes DIVE et MASSIVE [Benford 95][Benford 97], qui décrivent les informations que peuvent exprimer des avatars 3D (Figure 6.3) dans de tels environnements virtuels collaboratifs. Les paragraphes suivants reprennent les différentes formes d'informations qui sont exposées dans ces analyses, et expliquent la manière dont elles ont été gérées pour SpIn :

6.3.1 La présence

C'est la forme la plus élémentaire d'activité dans un CVE. Il s'agit d'une information qui indique la présence de quelqu'un dans l'environnement virtuel. Nous en avons discuté dans le paragraphe précédent en déterminant le mode de téléprésence de SpIn.

6.3.2 La position

Il s'agit de connaître la position et l'orientation de ses interlocuteurs dans l'environnement virtuel. Dans SpIn, l'utilisateur ne perd jamais de vue ses interlocuteurs du fait de l'organisation spatiale. Les clones sont placés autour de la table de réunion, et donc toujours orienté vers l'utilisateur. Ils sont de plus toujours visibles (grâce à la rotation du bandeau qui permet à chaque objet disparaissant d'un côté de ré-apparaître de l'autre).



Figure 6.4 *Un clone de SpIn (sur la gauche) regardant des diapositives (sur la droite)*

6.3.3 Le point de vue

Le point de vue donne une information sur les objets proches d'un utilisateur, souvent lié à la direction de son regard. SpIn intègre cette notion à travers la notion de focus. On connaît le centre d'intérêt d'un utilisateur (chapitre 4), c'est une information qui permet de diriger le regard du clone, en tournant sa tête et ses yeux :

- vers le document partagé placé dans le bandeau ou sur la scène,

- vers la table si le document regardé par l'utilisateur est un document privé.

Figure 6.4, on voit un participant regardant les diapositives placées sur la droite de l'écran. Le clone tourne les yeux et la tête vers le document.

6.3.4 Le point d'action

Il représente l'endroit où l'utilisateur agit, il n'est pas forcément couplé au point de vue. C'est un facteur important dans le cadre d'une application de travail coopératif synchrone, puisque la désignation, support de l'explication, est un des gestes principalement rencontré. Nous en discuterons au paragraphe 6.4.

6.3.5 Disponibilité et degré de présence

Il s'agit de savoir si un utilisateur est disponible pour une activité donnée, et savoir à quelle tâche il est occupé. Il est en effet utile de pouvoir juger du degré de présence d'un interlocuteur pour planifier son activité. Cette information est d'autant plus importante dans les environnements connectant beaucoup d'utilisateurs, ou sur une durée importante. Ce n'est pas le cas de SpIn qui relit un petit groupe de personnes pour la durée d'une réunion déterminée, en temps et en objectifs. Il peut cependant être intéressant d'être informé lorsque l'utilisateur se détache de la réunion, à savoir :

- qu'une défaillance matérielle s'est produite (ordinateur ou réseau),
- qu'il est en contact avec son environnement réel (téléphone, collègues, etc).

On peut noter par exemple la performance du logiciel ICQ (prononcer en anglais «je te cherche») de la société Mirabilis, dont le succès est révélateur. C'est un petit logiciel de communication asynchrone qui permet principalement d'envoyer des messages textuels courts à une liste de contacts, qui sont d'autres utilisateurs d'ICQ. Le grand intérêt est la mesure que donne ICQ du degré de présence des gens d'une liste de contacts. Le logiciel les fait passer entre plusieurs modes comme par exemple :

- *Available* (prêt à répondre),
- *Away* (qui se déclenche automatiquement après plusieurs minutes d'inactivité sur la machine),
- *Extended Away* (qui se déclenche quelques temps après un *Away* ou lorsque la machine passe en veille),
- *Do Not Disturb* (qui permet à l'utilisateur d'être tranquille et de ne plus rien recevoir ou tout du moins de ne plus le voir),
- *Absent*.

Ce type d'information permet de savoir immédiatement si on peut avoir une réponse, et sinon, laisser un message pour une réponse ultérieure. Ce renseignement n'est pas présent dans notre environnement mais rejoint les trois grandes étapes de l'activité d'un utilisateur de SpIn :

- l'utilisateur discute avec ses interlocuteurs, il est en mode de communication directe, ses gestes, voix et expressions sont transmis directement à son clone,

- l'utilisateur agit sur un document partagé, explique quelque chose. Les informations enregistrées par SpIn doivent être interprétées pour que son clone renseigne au mieux sur son activité,
- l'utilisateur travaille sur un document privé, ou rentre dans un des deux cas cités au début de ce paragraphe.

Si l'on sait générer le comportement d'un clone pour chacune de ces phases de l'activité d'un utilisateur, on ne sait pas comment détecter correctement et systématiquement le passage de l'une à l'autre de ces phases. Si certaines phases sont aisées à détecter, par exemple lorsque l'utilisateur manipule un document privé ou public, il n'en va pas de même lorsqu'il ne bouge plus devant sa machine.

Il faut se servir de tous les systèmes d'entrée de l'utilisateur pour détecter automatiquement, autant que possible, le passage d'une phase à l'autre, et il ne paraît pas raisonnable de pouvoir se passer d'un seul : la voix, les périphériques d'interaction des deux mains, la caméra qui permet l'analyse faciale, et d'autres interventions comme le clavier, ou celles données par le système comme la mise en veille de la machine.

6.3.6 Les gestes et les expressions faciales

Les expressions faciales peuvent être provoquées, en pilotant son avatar. C'est le cas dans *MASSIVE* ou dans le logiciel de discussion *MSchat* de Microsoft, où les utilisateurs sont représentés sous forme de personnages de bandes dessinées dans des cases, ils peuvent piloter l'expression et quelques gestes de leur avatar (Figure 6.5).

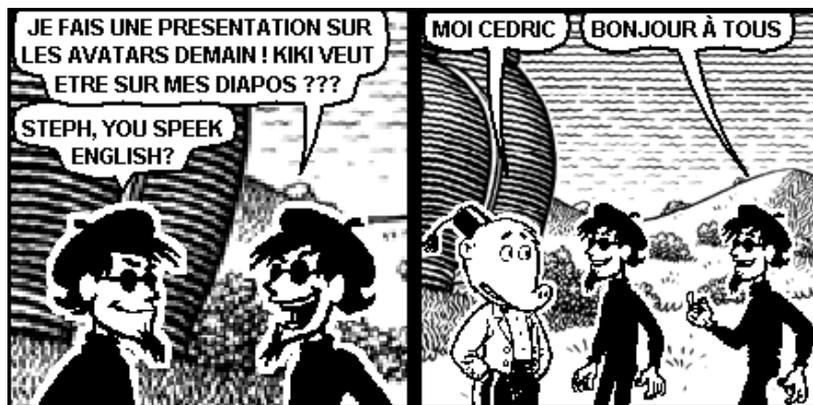


Figure 6.5 *MSchat où l'on choisit l'expression de son avatar*

Comme nous l'avons vu précédemment, le chaîne d'analyse/synthèse de la Télévirtualité permet d'analyser et de reproduire les expressions faciales. Dans la mesure où le système est également capable de donner aux clones des expressions faciales exprimant des sentiments (joie, tristesse, anxiété, etc), il pourrait être intéressant d'en intégrer un certain nombre dans un environnement de travail coopératif, sans exagérer les expressions pour ne pas les caricaturer. Les utilisateurs auraient alors à disposition un outil permettant de forcer légèrement leur attitude.

6.3.7 Liens avec les objets et informations

Ce critère met en avant l'intérêt de créer des liens entre objets dans les systèmes permettant d'en visualiser un grand nombre. Cela permet de créer, par exemple, des liens entre les utilisateurs et les documents qu'ils utilisent lorsqu'ils sont très nombreux.

Ces liens ne doivent cependant pas se spécialiser aux systèmes de réalité virtuelle à large échelle. Un des grands intérêts de l'utilisation de la troisième dimension pour les environnements de travail coopératif synchrone est justement la facilité pour créer des liens entre les documents et les objets. Les documents et les acteurs étant dans le même espace, on peut donner des informations sous formes visuelles, comme faire manipuler un document directement par un clone. Cela a l'avantage d'être explicite, aussi facile à interpréter et à comprendre que dans la réalité. C'est ce type d'indice qui permet de mettre en valeur l'aspect virtuel de SpIn.

6.3.8 Historique de l'activité

Il s'agit de conserver une trace de la présence et de l'activité d'un utilisateur dans un environnement virtuel, les endroits qu'il a fréquenté, les tâches accomplies, etc. Cela permet de donner une consistance dans le temps à un utilisateur dans l'environnement virtuel.

Dans le cadre de SpIn, la persistance est gérée par un serveur de TCAO (chapitre 3). Même si nous étions conscients de certains aspects asynchrones des réunions, nous nous sommes focalisés dans SpIn sur la conception de l'interface pendant le déroulement d'une réunion, et non sur des aspects plus asynchrones (planification de réunions, etc) ou à la supervision de la réunion (gestion de droits particuliers, organisation sur une période assez longue).

6.3.9 Bilan

On retrouve naturellement un certain nombre de paramètres essentiels entre notre approche et les critères de Steve Benford. Nous gérons cependant principalement les aspects purement synchrone dans SpIn, pendant le déroulement de la réunion.

SpIn est un environnement centré autour d'une tâche précise à réaliser grâce à un certain nombre de documents. C'est donc l'aspect coopération que nous avons développé, avec des éléments en plus, particulièrement au niveau des documents, comme les modes de partage ou la gestion du point de vue.

Nous allons donc décrire maintenant les principales primitives de coopération, qui renseigne un utilisateur sur l'activité des autres et lui permet de coopérer.

6.4 Désignation à distance

La désignation à distance est réalisée dans SpIn à travers l'utilisation d'un télépointeur. C'est un pointeur distant. Il permet d'indiquer le point d'action d'un utilisateur. C'est un facteur de téléprésence.

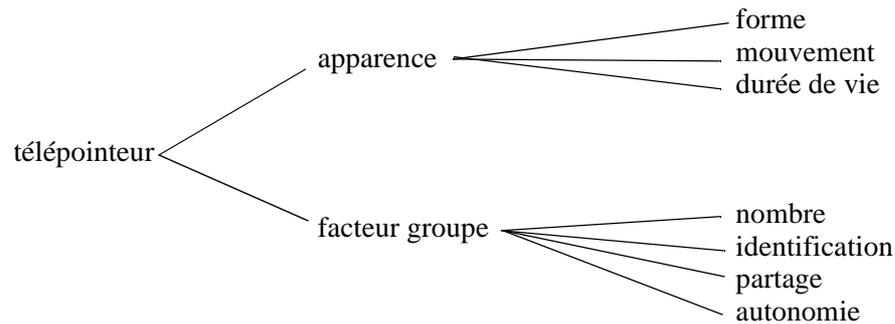


Figure 6.6 *cadre d'analyse d'un télépointeur de Pascale Primet*

Pascale Primet propose un cadre conceptuel (Figure 6.6) pour décrire les télépointeurs [Primet 94]. Nous allons suivre ce cadre pour décrire l'approche que nous avons suivie.

6.4.1 Analyse de l'apparence (la forme)

Le télépointeur doit avoir une forme explicite. Dans la mesure où il s'agit de la prolongation d'un doigt (dans le monde réel), on pourrait naturellement penser à une forme de main, avec un index pointé. Cette forme est largement répandue dans les interfaces 2D, mais dans la mesure où il existe dans SpIn les clones des utilisateurs, représenter une main supplémentaire dans l'espace ne donne pas un effet très satisfaisant.

Nous avons donc choisi une flèche, indicateur traditionnel. Nous ne changeons pas sa forme en fonction du contexte, le télépointeur n'étant utilisé ici que pour désigner un document ou une de ses parties pendant une explication. Figure 6.4, on voit le clone du télépointeur, vert tout les deux, placé sur les diapositives à droite de l'image.

Pour identifier sans équivoque le télépointeur, il est représenté de la couleur du corps du clone auquel il appartient. De même, le pointeur local d'un utilisateur est de la couleur de son télépointeur pour aider la coordination verbale des utilisateurs dans une réunion.

Nous avons utilisé un indice supplémentaire, reliant le clone au télépointeur par une courbe spline (Figure 6.7).

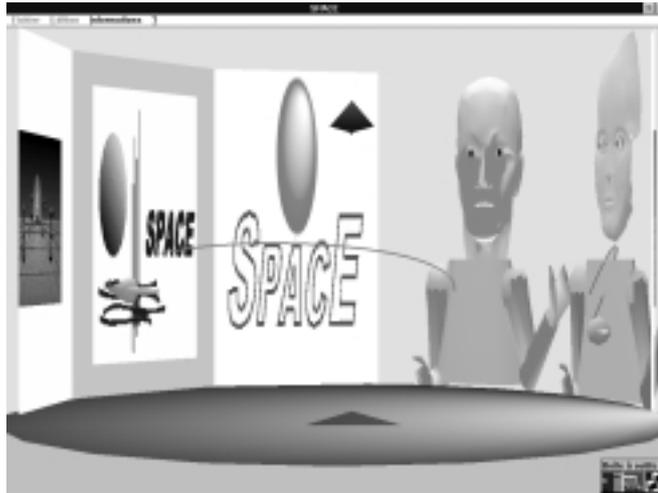


Figure 6.7 Télépointeur relié par une courbe à son clone (version 1997 de SpIn)

Le problème provenait de l'occupation permanente de l'espace par les courbes. Elles pouvaient traverser des objets et se déplaçaient continuellement au dessus de la table. Le système était finalement perturbant, et assez peu utile, dans la mesure où pour un petit groupe, les télépointeurs restent facilement identifiables juste avec l'information de couleur.

La taille du télépointeur doit être la plus grande possible en respectant deux critères :

- la taille doit être inférieure à celle du pointeur, et la forme doit si possible être différente,
- le télépointeur doit être suffisamment petit pour ne pas masquer l'information qu'il désigne.

6.4.2 Le mouvement

Le déplacement est continu. Le télépointeur se déplace de la même manière que le pointeur, dès que ce dernier désigne un document partagé, le fonctionnement est implicite, l'utilisateur n'a rien à spécifier pour que son télépointeur se déplace. Auquel cas il reste dans la main de son clone : il s'agit d'un indice supplémentaire pour connaître l'appartenance d'un télépointeur. La compréhension se fait naturellement lorsque l'on voit le télépointeur partir de la main d'un clone.

Si chaque utilisateur possède son propre point de vue sur un document, i.e. s'il a modifié l'orientation d'un document dans son environnement de travail, il faut veiller à replacer le télépointeur dans de bonnes conditions à distance, y compris en orientation. C'est le cas de SpIn, qui génère des calculs en local sur la machine d'un utilisateur, et transmet les résultats aux clones de cet utilisateur sur chaque machine pour que le télépointeur soit replacé dans une position et une orientation correcte.

Etant donné le petit volume de données transmises, la diffusion (multicast sur réseau ethernet) ne génère quasiment pas de temps de latence, aussi peu que pour le son. Il n'y a donc pas eu de problème remarquable au niveau de l'utilisation du télépointeur.

6.4.3 Durée de vie

Lorsque nous avons commencé à travailler sur les télépointeurs, nous avons tout de suite intégré la notion de marqueur. Un marqueur est une croix placée au niveau d'un document pour signaler un point particulier. N'ayant pas de moyen simple d'enlever les marqueurs, nous n'avons pas développé plus en avant cette notion.

6.4.4 Analyse du facteur de groupe

Nombre de pointeurs

SpIn est conçu pour des réunions de petits groupes, de 3 à 5 personnes. Pour davantage d'utilisateurs, il faut savoir que tout le monde n'explique pas forcément quelque chose en même temps que son voisin. Nous avons donc opté pour un télépointeur par utilisateur (par clone).

Le fait de gérer un seul télépointeur oblige à insérer des mécanismes de contrôle, contraignant pour les utilisateurs. Notre optique est au contraire depuis le début de laisser un maximum de liberté, pour que la réunion se déroule «comme elle devrait», sans contrainte forte du système. On espère ainsi que les utilisateurs retrouvent leurs comportements sociaux dans l'environnement de travail virtuel.

Identification du manipulateur

Comme nous l'avons décrit dans l'analyse de l'apparence, l'identification est assurée par deux éléments :

- la couleur du télépointeur est la couleur de son clone,
- la couleur du télépointeur est la couleur du pointeur local de l'utilisateur représenté par le clone,
- le télépointeur part et revient de la main droite de son clone,

Le clone permet d'ajouter encore un élément d'identification :

- un bras du clone bouge avec le télépointeur, pointant dans sa direction.

L'association personne/télépointeur est donc facile à faire. Un paramètre supplémentaire qui pourrait aider, non pas directement à l'identification, mais à la différenciation, serait d'attribuer un télépointeur de forme différente à chaque utilisateur.

Politique de partage

N'ayant pas de télépointeur unique, l'accès de chacun est traité en permanence et en parallèle pour chaque clone.

Degré d'autonomie

Dans les interfaces fenêtrées, beaucoup de problèmes se posent avec les télépointeurs dus à la nature des systèmes WIMP : un utilisateur peut redimensionner, déplacer les fenêtres, le télépointeur peut avoir à passer dans des zones non partagées (et donc être invisible à ce moment). On retrouve également les problèmes de masquage d'information avec les icônes et les superpositions de

fenêtres. De manière général, des problèmes se poseront sur la compréhension de l'activité d'autres utilisateurs s'ils ne sont pas dans notre champ de vision, eux et les documents dont ils parlent.

SpIn propose un modèle «tout visible» ou au moins du «tout perceptible» lorsqu'il y a beaucoup de documents dans la scène, et résout donc tous ces problèmes liés à la disparition des télépointeurs, toujours visibles dans SpIn.

6.4.5 Bilan

La troisième dimension permet des mouvements continus du télépointeur entre les clones et les documents, dans le même espace, avec le même type de rendu. La continuité est assurée et permet une compréhension rapide.

La formalisation de Pascale Primet nous a permis de donner une définition complète de notre modèle, mettant en avant les avantages de notre organisation spatiale pour la désignation à distance. De plus, le télépointeur est souvent vu comme un vecteur des gestes de l'utilisateur, en particulier les gestes volontaires, dans le cas d'un dessin par exemple. SpIn complète cet aspect du télépointeur par la présence des gestes du clone.

6.5 Manipulation à distance

On entend par manipulation à distance, la manipulation d'un utilisateur sur un document partagé et ses répercussions dans les environnements de ses interlocuteurs. Il est indispensable que les modifications apportées à un document partagé soient reportées à distance. S'il existe des difficultés techniques, le plus dur reste le problème de l'interface homme-machine.

On peut en effet verrouiller les objets pendant la modification d'un document par un utilisateur pour que personne d'autre ne puisse y accéder. Ce système est très contraignant et impose un mécanisme de prise de pouvoir sur un document. Il peut paraître adapté à des situations du type enseignement, où l'enseignant peut piloter les droits des participants, mais il n'en va pas de même lors de la réunion d'un groupe de personnes où tous les utilisateurs voudront légitimement avoir les mêmes droits (le cas critique en fait).

Il existe des systèmes de prises de tour, où chacun dispose dans une phase de l'activité, de l'accès à un document, avec une rotation vers tous les utilisateurs. Ce système est également contraignant, et on a montré l'avantage des systèmes possédant un mode libre [Shu 94].

C'est le parti que nous avons choisi, par défaut, pour SpIn, afin de permettre à nouveau aux comportements sociaux habituels de se mettre en place. Un utilisateur peut donc accéder à n'importe quel partie d'un document, on ne verrouillera que la plus petite partie du document possible (afin de ne pas créer de comportements incohérents du système). La partie du document n'est verrouillée que le temps de la manipulation de l'utilisateur concerné, et n'empêche pas ses interlocuteurs de manipuler le reste du document. Lorsqu'un utilisateur essaye de sélectionner une partie verrouillée d'un document, la partie en question devient rouge, et il lui est impossible d'y accéder.

A distance, la manipulation d'un objet produit le même effet qu'en local. Les utilisateurs sont informés de la manipulation d'un document par un des participants :

- par la modification du document,
- par le regard du clone,
- par le télépointeur placé sur le document;

et éventuellement par l'utilisateur lui-même qui peut commenter oralement ses actions.

6.6 Gestion du point de vue

Il ne faut pas confondre l'édition, la manipulation d'un document, avec sa visualisation. L'édition consiste à modifier un document, à changer sa structure, comme modifier un texte, ou ajouter des atomes à une molécule. La visualisation consiste à modifier son point de vue global sur le document sans en changer la structure, comme faire défiler les pages d'un texte, ou regarder un objet 3D sous tous les angles.

La gestion du point de vue est très importante, car un utilisateur peut avoir une vue différente des autres d'un document donné, et ne pas comprendre ce que lui explique un interlocuteur parce qu'ils ne voient pas les choses sous le même angle. Le point de vue d'un document doit donc pouvoir être partagé. Dans ce cas, si un utilisateur modifie ce point de vue, il changera alors chez les autres.

Ce mode a cependant le défaut d'être contraignant, et n'est pas très adapté à la liberté de compréhension de chacun. Certaines formes de réunion préféreront donc un modèle de réunion plus souple, en laissant à chacun un point de vue libre sur les documents partagés. Chacun peut alors visualiser un document à sa guise et sous tous les angles.

Cela impose cependant d'intégrer un outil de prise de point de vue. Chaque utilisateur pourra, à travers cet outil, prendre le point de vue d'un autre utilisateur :

- sélection de l'outil dans la barre de menu,
- sélection avec l'outil d'un clone pour prendre son point de vue sur l'objet sur la table (l'utilisateur conserve alors le nouveau point de vue).

On pourra envisager également de créer un outil pour prendre le point de vue sur n'importe quel document partagé, mais SpIn ne permet ensuite de manipuler que le document placé au centre de la table.

La gestion du point de vue sur les documents est indispensable à la compréhension et à la coordination des acteurs entre eux. Elle est encore à l'étude dans SpIn.

6.7 Partage et mise à disposition de documents

SpIn ne spécifie pas un cadre rigide au niveau des partages de documents. Même si l'on peut préconiser la coopération directe, c'est à la personne qui organise la réunion, ou à celui qui distribue un document particulier de choisir le mode de partage.

Ce dernier paragraphe étudie donc les différents modes de partage de documents possibles. L'objectif est ensuite de spécifier les métaphores visuelles en rapport avec ces différentes situations. Ce travail est issu de nombreuses réunions sur SpIn, et doit être vu comme une ouverture vers les prochaines versions.

6.7.1 La mise à disposition de document

Avec la possibilité de charger des fichiers dans SpIn (chapitre 5), il devient aisé dans un tel environnement de permettre la diffusion des documents. Ainsi, lorsqu'un utilisateur ouvre un document avec le *conetree*, au moment de la sélection du fichier, celui-ci se retrouve au bout du pointeur. L'utilisateur a alors deux solutions :

- le placer dans le bandeau pour charger localement le fichier,
- placer le document sur un clone pour lui envoyer automatiquement le fichier.

Si ce mode de fonctionnement est simple, il ne permet pas les envois d'un même fichier vers plusieurs participants. Il faudra créer un outil de coopération capable de réaliser une telle opération.

Pour la réception, nous enregistrons automatiquement le fichier sur la machine de l'utilisateur distant afin de ne pas interrompre son activité. De manière générale, il est d'ailleurs souhaitable de ne pas diriger l'interface par le système, nous conservons l'aspect interactif pour l'utilisateur en évitant de donner à SpIn un comportement dirigiste. Ceci est vrai pour les interfaces homme-machine en général, laisser l'initiative à l'utilisateur, mais encore plus pour un environnement de travail coopératif synchrone, où l'utilisateur peut être affairé à une autre tâche que celle qui va l'interrompre.



Figure 6.8 *un document est disponible auprès de ce clone (l'icone placé devant)*

Nous n'utilisons donc pas de fenêtre de dialogue pour informer l'utilisateur. Il suffirait de placer un icône devant le clone de l'intervenant qui a envoyé le fichier, indiquer l'arrivée par un message sonore, et laisser l'utilisateur venir prendre l'icône pour le placer dans son bandeau quand bon lui semble. Un exemple est montré Figure 6.8.

6.7.2 Les modes de partages

Nous exposons ici les différents modes de partage. Le mode est à choisir en fonction de la tâche à accomplir, des intervenants et du type de document disponible.

Mode dupliqué

Une copie du document partagé pour chacun (coopération lâche). Il n'y a pas de coopération au niveau de l'interaction avec le document dans ce mode. La seule coopération est au niveau de la tâche globale. Ce mode favorise la communication verbale, chacun travaillant «de son côté» sur une instance d'un document, sans voir celle des autres.

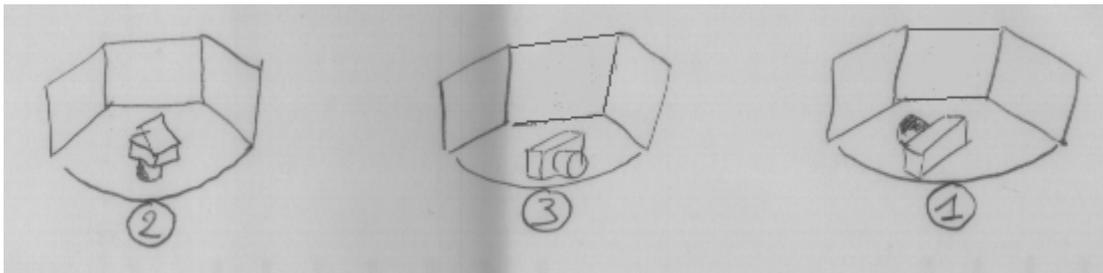


Figure 6.9 *Chacun possède une version du document (document de travail SpIn)*

On peut légèrement étendre ce modèle en créant un mécanisme qui permet de visualiser les documents des autres chez soi, on est alors en mode maître-esclave puisqu'on ne fait que visualiser la version d'un autre. Un outil pourrait alors permettre de copier cette version dans son interface de travail, en écrasant sa copie personnelle. Des icônes de couleur devant les clones peuvent permettre facilement de savoir de quelle version chacun dispose (la sienne ou celle d'un autre).

Mode partagé complet

Un document partagé par tous: une seule copie, les actions se répercutent de tous vers tous. Il faut donc insérer des mécanismes de prise de contrôle (coopération stricte).

Si SpIn est dérivable dans n'importe quel mode, nous avons particulièrement utilisé ce mode. C'est celui qui nous semblait induire le plus de coopération et le moins de contraintes pour l'utilisateur, à savoir un mode partagé complet où chacun est tout de même maître du point de vue sur chaque document (voir le dernier chapitre sur l'évaluation). On peut suivre dans ce cas en permanence les évolutions d'un document et intervenir à tout moment.

Modes intermédiaires

Il existe un certain nombre de possibilités entre un partage complet et pas de partage (dupliqué).

- Un document partagé par tous, avec la possibilité de faire des copies locales pour le modifier. Il faut insérer un mécanisme de mise à jour de la version partagée, et un mécanisme de copie de la version partagée en version locale.
- Un document chacun avec la possibilité de partager sa version avec un ou plusieurs autres (N versions dont plusieurs partagées par certains). C'est un mode mixte où certains ont une copie locale

et d'autres partagent une même version (Figure 6.10). Il semble difficile avec autant d'instances du document de maintenir une cohérence des différentes vues, la coordination revient alors aux utilisateurs.

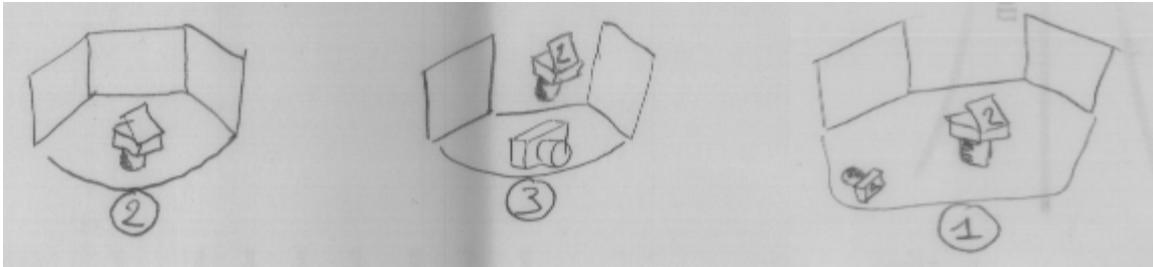


Figure 6.10 *Chacun possède une version, mais peut la partager avec qui il veut (document de travail Spln)*

Figure 6.10, UN décide de prendre la version de DEUX pour la manipuler en même temps que lui, la version de DEUX est alors partagée par UN et DEUX, et visualisée par TROIS.

- Un document chacun avec la possibilité de le partager pour tous (N versions) mais un seul document partagé à la fois (N versions dont une partagée par tous). Cela revient à insérer un mécanisme pour passer facilement d'un mode dupliqué à un mode partagé complet à partir d'une version donnée par un des utilisateurs (Figure 6.11).

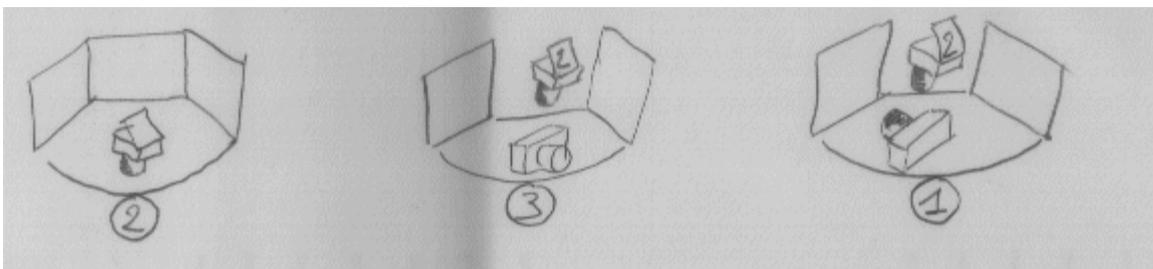


Figure 6.11 *Un document par personne dont une version partagée (document de travail Spln)*

Figure 6.11, DEUX décide de partager sa version, elle est alors affichée dans le bandeau de UN et TROIS, qui peuvent alors visualiser la version de DEUX.

- Un document chacun plus une version commune mise à jour par tel ou tel utilisateur (N+1 versions). L'inconvénient de ce mode de partage est l'existence de deux versions d'un même document dans l'espace de travail de l'utilisateur. Il faut alors travailler sur l'information donnée aux utilisateurs pour une bonne compréhension de l'activité, l'espace unique de travail doit permettre d'en donner une visualisation cohérente avec, par exemple, la version commune placée dans le bandeau.

Ces modes particuliers imposent de créer des outils de coopération spécifiques pour les actions suivantes :

- prendre la version du document d'une autre personne,
- proposer une nouvelle version d'un document à quelqu'un,

- partager son document avec quelqu'un ou avec tous,
- prendre le contrôle d'un document,

et de manière générale, on doit pouvoir passer d'un mode de partage à un autre en fonction des phases de la réunion.

Récapitulatif

On reprend dans le tableau ci-dessous l'ensemble des modes de partage possibles avec les métaphores visuelles qu'il faut prévoir.

Table 7: modes de partages pour N utilisateurs

Mode	nombre totale d'instances du document	nombre d'instances du document dans une interface	nombre de vues différentes du document	mécanismes de partage	métaphores visuelles
dupliqué	N (chacun sa version)	1	N	-	-
dupliqué avec prise de vue	N	1 (+ 1)	N + 1 partagée pour voir une autre instance	- prise de point de vue sur un autre document - copie sur la version locale de la version partagée	- savoir à qui appartient la version visualisée en plus chez soi
dupliqué avec une version partagée par tous	N dont une partagée par tous	1 + 1 d'un autre utilisateur partagée par tous	N	- proposer sa version à mettre en commun - mettre à jour sa version locale avec la version partagée	- savoir la version de qui on partage
dupliqué avec versions partageables entre plusieurs utilisateurs	1..N	1 (personnelle ou partagée)	1..N	- partager son document avec d'autres - demander sa version à un autre - reprendre le document en local	- savoir qui travaille sur quelle version - savoir qui copie la version partagée en local
partagé + 1 version en copie locale	1+N	1+1 copie locale	1+N	- mécanisme pour faire une copie locale - mécanisme pour proposer aux autres une mise à jour de la version commune - mécanisme pour prendre la main sur la version partagée	- savoir qui possède une copie locale - savoir qui a la main sur la version partagée - savoir l'origine de la version partagée
partagé	1	1	1	- prise de contrôle du document	- savoir qui possède la main sur le document

6.8 Bilan

Notre modèle d'interface a un rôle de proposition, il ne verrouille pas les problématiques, mais essaye de fournir une réponse à chacune, adaptée au cadre de SpIn. Les primitives de coopérations présentées ici permettent de proposer des services de travail coopératif synchrone. Ils forment un ensemble de base complet indispensable pour créer une situation de travail coopératif.

Il pourrait être ajouté à l'avenir à ces outils de coopération des services plus spécifiques comme :

- savoir qui partage quoi avec qui,
- les informations d'aparté entre interlocuteurs, savoir qui discute avec qui,
- un outil de vote : public et secret,
- des outils de visualisation de données particulières,
- les outils d'édition de documents avec vues dupliquées, mécanisme de mise à jour sur parties modifiées en concurrence ou non.

SpIn est une proposition d'interface, nous traitons des problèmes de métaphores visuelles une fois les caractéristiques de la coopération établies. Pour faire de SpIn un outil de coopération complet, il faut continuer à développer des outils de coopération, mais surtout l'activité du clone, pour que ce dernier informe au mieux de l'activité de son utilisateur. C'est dans ce sens que le virtuel peut se substituer aux réunions réelles et améliorer leur efficacité.



SpIn est un environnement de travail coopératif synchrone, il doit pouvoir remplacer n'importe quelle situation de réunion. Nous l'avons donc conçu de façon aussi générique que possible. Pour l'utiliser dans des conditions réelles, il faut définir le cadre d'une situation particulière, c'est-à-dire créer un contexte, de la même façon que l'on prépare une réunion ou un cours dans la réalité.

Nous avons donc mis en place quelques situations permettant l'utilisation de SpIn dans un contexte particulier, et une principalement, qui vise l'apprentissage de la photographie. C'est une situation de contexte externe, qui ne correspond pas à une situation de travail réel. En effet, nous n'avons pas encore pu adapter de situation de terrain dans le contexte de SpIn.

Avant toute chose, nous commençons ce chapitre par un récapitulatif des fonctionnalités de SpIn par rapport à l'ensemble des descriptions des chapitres précédents. Puis nous présentons le contexte de la situation d'apprentissage du fonctionnement d'un appareil photo. Enfin, nous exposons les résultats et l'analyse de l'évaluation de cette situation.

7.1 L'interface actuelle

La première version utilisable de SpIn a marqué une étape. Elle a été présentée à Imagina 99 et a permis d'élaborer une expérimentation, qui est exposée plus loin dans ce chapitre. Avant cela, nous décrivons ici les fonctionnalités principales de SpIn afin de résumer et d'illustrer les principes établis précédemment (dans les chapitres 4, 5 et 6).

7.1.1 Les périphériques



Comme périphérique d'entrée, on utilise un trackball 3D dans la main non-dominante.

Il s'agit ici de la *Spacemouse* de *Logitech* (connue aussi sous le nom de *Magellan*). C'est un périphérique isométrique à six degrés de liberté. Nous n'utilisons pour l'instant que les 3 axes de rotations.



Dans la main dominante, on utilise un capteur de position 3D. Il s'agit ici d'un capteur acoustique *FreeD* de *Pegasus Technologies* (système à ultrasons), anciennement appelé *Owl*. Il possède trois degrés de liberté, qui donnent la position du capteur dans l'espace. On l'utilise couramment en raison de son prix peu élevé (résolution de 100 dpi, pour une fréquence de 50 Hz).

Ce capteur est cependant directionnel, il nécessite donc de conserver sa main dirigée en permanence vers le récepteur placé sur l'écran. La poignée du capteur (que l'on voit sur la photo) est ergonomique et permet de conserver naturellement cette même direction.

Pour ce qui est du son en entrée/sortie, on utilise des enceintes avec micro, ou un casque-micro si l'environnement est bruyant.

L'affichage se fait en plein écran sur des moniteurs de stations de travail de 17 pouces au moins. On peut noter que l'espace d'affichage de SpIn a une importance. On ne conçoit pas une visualisation sur un écran de petite taille, pour des raisons de visibilité et d'impact de l'image. On ne le conçoit pas non plus sur un écran trop grand, il faut que l'extérieur de l'écran puisse continuer à occuper le champ de vision périphérique de l'utilisateur, celui-ci servant de référentiel pour l'espace réel et donnant des repères stables à la visualisation de l'écran. On a pu également remarquer empiriquement que les écrans à forte dominante horizontale (écrans 16/9^{ème} ou 16/10^{ème}) étaient plus adaptés au mode de représentation de SpIn (le bandeau) que les écrans au format 4/3.

7.1.2 Le point d'action

Nous utilisons un pointeur 3D (Figure 7.1) dont le mouvement est directement lié à celui du capteur 3D placé dans la main dominante. Ses rotations sont calculées à partir de ses déplacements. Il peut se mouvoir dans tout l'espace de travail virtuel et possède une ombre portée.

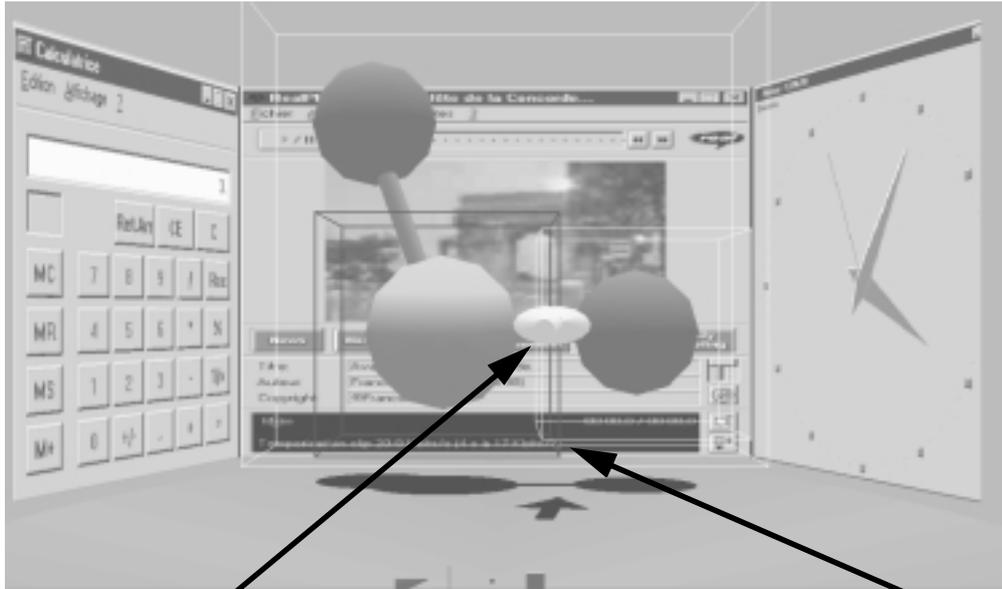


Figure 7.1 le *pointeur 3D désigne l'objet sur la table* (une molécule d'eau), on voit les *boîtes englobantes progressives s'afficher*.

On utilise des boîtes englobantes progressives pour sélectionner les différentes parties du document sur lesquelles l'utilisateur peut agir. Le pointeur permet de manipuler des objets sur la table de réunion, la barre d'outils, ou les objets du bandeau (en particulier pour les ramener sur la table).

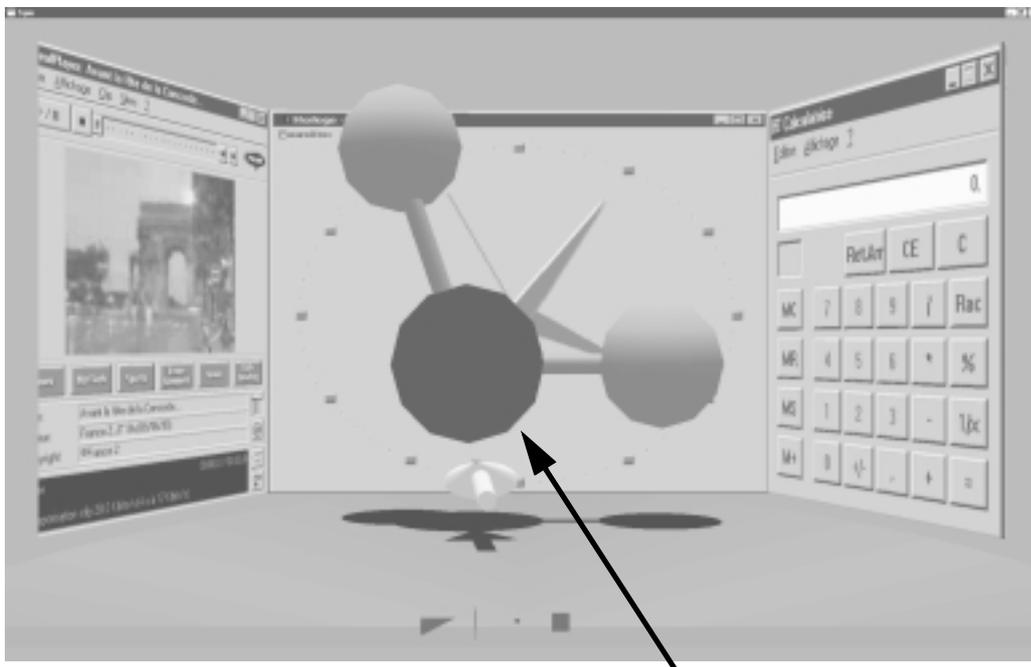


Figure 7.2 *l'atome de la molécule d'eau vient d'être sélectionné, il est affiché en bleu*

Lorsqu'un objet est sélectionné, il apparaît en bleu (Figure 7.2). On a alors accès à une action associée à cet élément, comme par exemple l'accès à un menu (qui permet de sélectionner une action parmi un ensemble). Figure 7.2, l'utilisateur peut déplacer en translation l'atome dans les trois directions à l'aide de la *Spacemouse*.

7.1.3 L'organisation des documents

Les documents sont placés autour de la table. On peut faire tourner le bandeau, et zoomer le document au centre à l'aide de la *SpaceMouse*. L'utilisateur peut amener n'importe quel document du bandeau vers le centre, où il peut alors le manipuler.

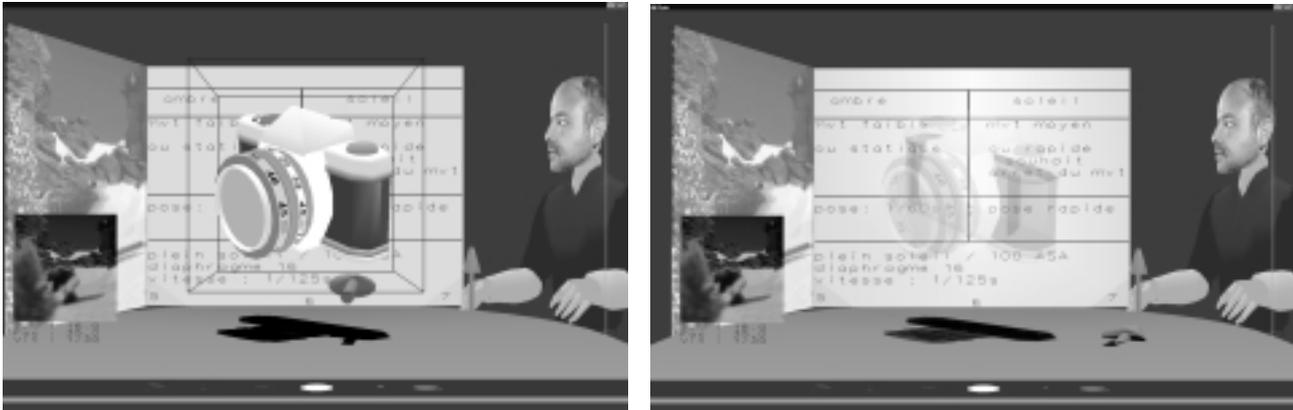


Figure 7.3 *focus* : l'utilisateur se focalise sur le document placé sur la table (à gauche) ou sur le document au centre du bandeau (à droite)

Le document placé sur la table (Figure 7.3) est le centre d'intérêt de l'utilisateur si il est affiché en opaque (cas où le pointeur est à l'intérieur), sinon c'est le document au centre du bandeau (cas où le pointeur n'est pas à l'intérieur du document placé sur la table, alors affiché en transparence). Ce focus permet d'orienter le regard des clones à distance. Nous avons en effet jusqu'ici utilisé le mode implicite, décrit chapitre 5, pour la détection du focus.

7.1.4 Les clones

Les utilisateurs distants sont représentés par leurs clones. Leur regard est dirigé par le focus de leur utilisateur respectif, il possède un télépointeur pour désigner à distance (Figure 7.4).



Figure 7.4 *Le clone* regarde et désigne les diapositives avec son télépointeur (à gauche) le clone regarde l'utilisateur (à droite)

Lorsqu'un utilisateur agit sur un document partagé (Figure 7.5), ses modifications sont reportées à distance et éventuellement ses changements de point de vue si ce dernier est le même pour tous (point de vue unique).

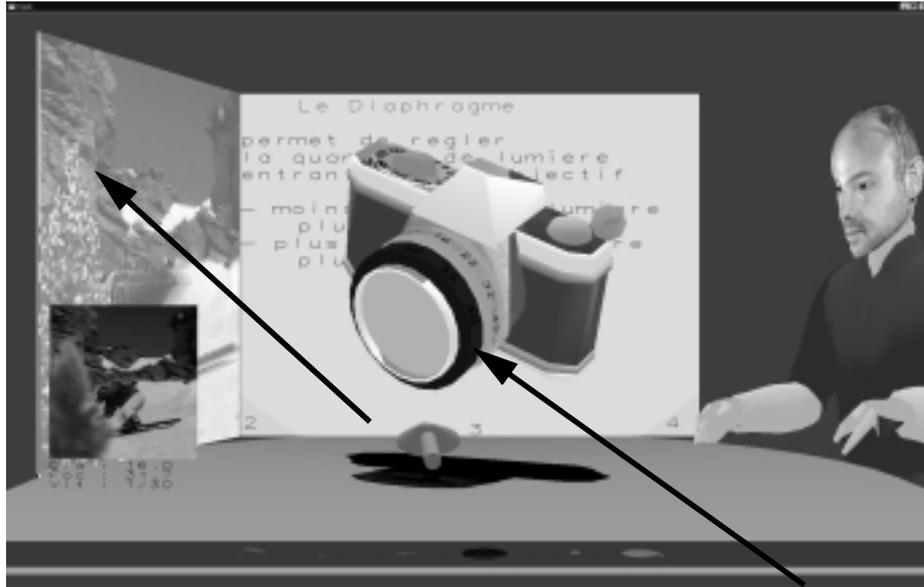


Figure 7.5 *partage* : si l'appareil photo situé sur la scène est partagé, la rotation de la bague (en bleu) effectuée par l'utilisateur, est reportée à distance. Le clone regarde l'appareil sur la table.

Si l'utilisateur veut manipuler un élément déjà utilisé par un autre, un flash rouge le prévient qu'il y a un verrou et qu'il ne peut pas le sélectionner.

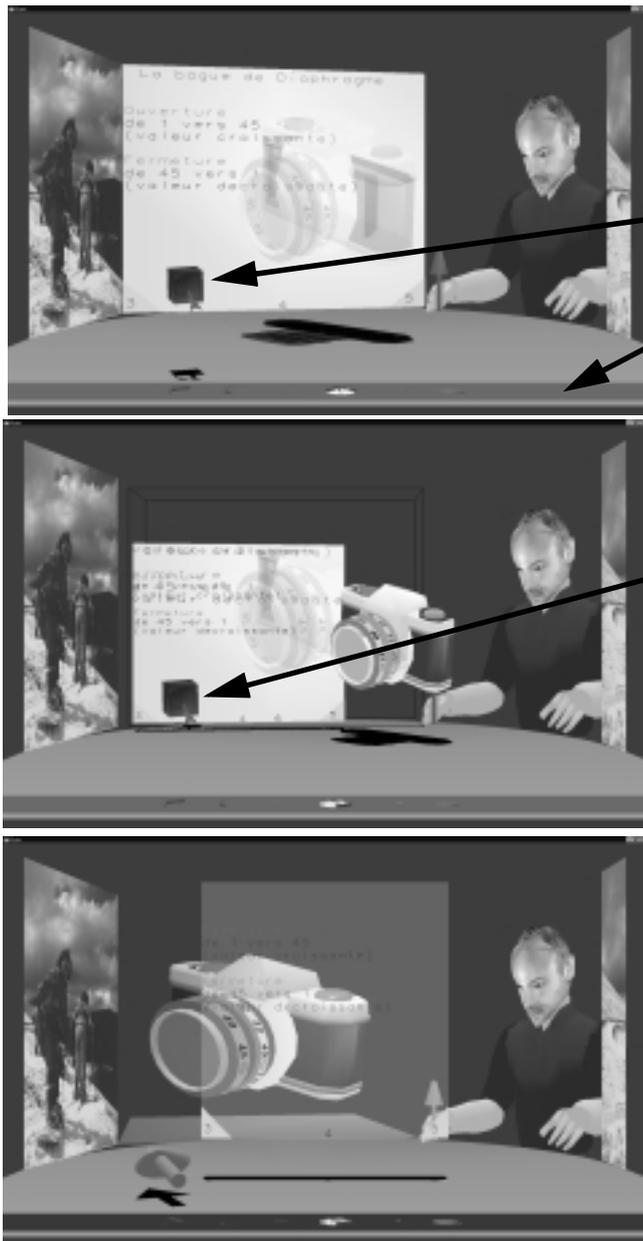
7.1.5 Les outils

Tous les outils développés actuellement dans SpIn sont accessibles dans la barre située au devant de la scène. On peut y sélectionner un icône ayant la forme de l'outil, puis utiliser ce dernier pour agir sur un document ou un acteur.

L'outil d'échange, par exemple, permet de permuter un document du bandeau avec celui placé sur la scène (Figure 7.6).

Il existe d'autres outils dans SpIn comme la possibilité d'envoyer un document à un autre utilisateur à travers le *conetree*. Le *conetree* permettant la sélection d'un fichier dans une arborescence. Chaque situation de réunion peut exiger le développement d'actions particulières, liées à des documents spéciaux, comme nous le verrons dans le paragraphe suivant.

Un des inconvénients vient du manque de sens de la représentation graphique de chaque outil. Elle permet pourtant à l'utilisateur de construire sa représentation mentale de l'outil, de comprendre l'action de l'outil et son effet. Il faudrait penser chacune de ces représentations, et que des graphistes travaillent avec les concepteurs d'environnements comme le notre, comme on a pu le voir pour le projet *Realplaces* chez IBM ou le projet *Capucine* au CNET.



*première étape :
on a sélectionné
l'outil d'échange
dans la barre d'outils*

*deuxième étape :
en allant cliquer avec l'outil
sur un document du bandeau
on échange sa position avec
celle de l'objet sur la table.
Une animation continue permet
de comprendre ce qui se passe.*

*finalement les deux
documents ont changé
de position (les diapositives
sont sur la table).*

Figure 7.6 échange d'un document entre le bandeau et la scène

7.2 L'appareil photo

Une situation particulière a été mise en oeuvre sur SpIn. L'objectif était de créer un contexte qui ressemble à une réunion réelle. Afin de mettre en valeur le mode de représentation de SpIn, nous avons centré l'activité sur un objet complexe dans la réalité. Nous avons choisi le thème de l'apprentissage du fonctionnement d'un appareil photo, de par son côté grand public, la relative difficulté de son fonctionnement et les manipulations qu'il nécessite.

Le schéma typique de cette situation met en présence trois intervenants, deux novices et un expert en matière de photo. Ce dernier va devoir apprendre aux autres le fonctionnement de l'appareil, en leur faisant prendre des photos.

7.2.1 Les documents

Il a donc fallu réunir les éléments nécessaires à l'activité pour mettre en oeuvre cette situation : les documents.

Un appareil photo

Le document utilisé est un appareil photo modélisé en trois dimensions (Figure 7.7). Il s'agit d'un modèle reflex classique que l'on peut manipuler. Il possède trois bagues pour régler la focale, la vitesse et le diaphragme, et un bouton pour prendre une photo. L'utilisateur peut observer l'appareil sous tous les angles en modifiant son point de vue lorsqu'il est placé sur la table.

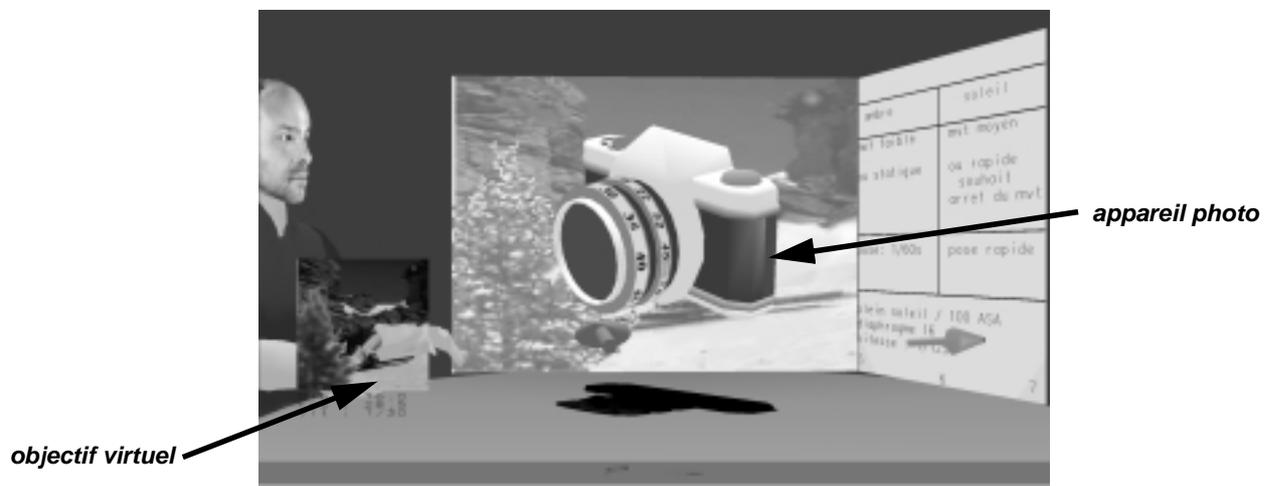


Figure 7.7 composition de la scène

Un objectif virtuel

C'est le complément indispensable de l'appareil photo. C'est une petite fenêtre qui montre une scène animée, en l'occurrence un skieur descendant une piste (Figure 7.7). L'image devient floue si la focale est mal réglée, et le déplacement du skieur permet de faire une estimation de la vitesse de la prise de vue. Plutôt que de placer cet objectif directement sur l'appareil (comme dans la réalité), nous l'avons placé à côté de l'appareil photo. L'utilisateur peut donc le voir en permanence sans avoir besoin de tourner l'appareil dans le bon sens.

Un simulateur de prise de photo

Après avoir réglé les paramètres de sa prise de vue, l'utilisateur doit pouvoir en observer les résultats. Nous avons donc développé un simulateur de prise de photos, tenant compte des paramètres réglés sur l'appareil (Figure 7.8). Lorsqu'un utilisateur déclenche une photo (en appuyant sur le bouton), une image est générée et placée dans le bandeau. Elle représente bien sûr la

même scène que celle de l'objectif virtuel, mais transformée par le manque de lumière ou la vitesse du skieur.

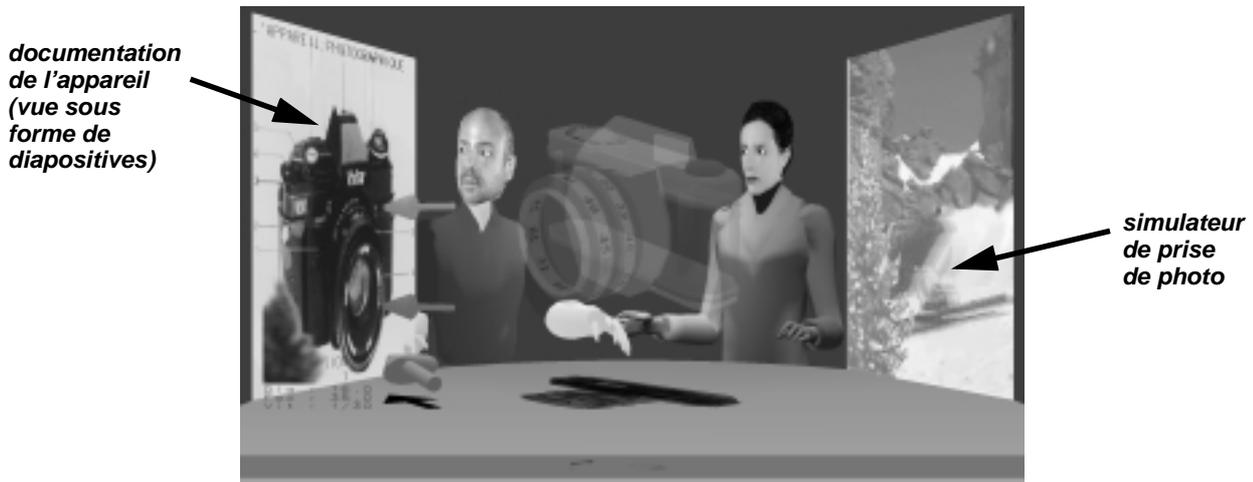


Figure 7.8 *composition de la scène*

Une documentation

Pour renseigner les utilisateurs sur les principes de base de la photographie, une documentation succincte composée d'une dizaine de diapositive est placée dans le bandeau (Figure 7.8). L'utilisateur peut les faire défiler en cliquant simplement dessus.

Une fois que les documents nécessaires à l'activité sont établis, il faut définir les modes de partage et les droits, en fonction de l'objectif et du rôle de chacun.

7.2.2 Modes de partage

Nous avons établi les règles de partage qui nous semblaient les plus adaptées à cette situation d'apprentissage; afin qu'une collaboration puisse s'établir, que les interlocuteurs s'entraident, nous avons adopté un fonctionnement le plus souple possible. Nous avons donné les mêmes droits à tous les interlocuteurs (les modes de partage sont plus ou moins lâches en fonction de la situation de travail envisagées).

L'appareil photo est un document partagé. Afin que chacun puisse voir les modifications des réglages des autres, c'est le même objet présent chez chacun, toutes les modifications (tourner les bagues) sont reportées à distance. Par contre, le point de vue sur le document est laissé libre, chacun peut le tourner à sa guise pour l'observer ou le manipuler sous le bon angle.

Lorsqu'un utilisateur prend une photo, le résultat est affiché chez tout le monde, afin de permettre une discussion commune des problèmes rencontrés et de les résoudre ensemble.

Enfin, la documentation est présente chez chaque intervenant, mais chacun peut la consulter comme il le souhaite. Si un utilisateur fait défiler les diapositives, elles ne bougeront pas chez les autres.

Ces règles définissent un mode assez ouvert de coopération, dépendant de la coordination verbale des interlocuteurs, qui devrait s'établir par le petit nombre de personne et par l'essence même de la tâche à accomplir : chacun doit comprendre comment l'appareil fonctionne.

SpIn permet de définir n'importe quel mode de partage de documents, ce qui peut permettre de créer pour chaque nouvelle situation un contexte de réunion adapté.

Cette application a permis de réaliser une évaluation que nous allons maintenant voir en détails.

7.3 Evaluation

La mise en oeuvre de la première situation de travail coopératif avec SpIn nous a permis de mener une évaluation. Il s'agissait d'une situation de laboratoire dans la mesure où c'est nous-même qui avons choisi une situation de travail qui pourrait être applicable dans SpIn.

7.3.1 Les objectifs

L'objectif de cette expérimentation était de tester la façon dont les sujets utilisaient l'interface, d'essayer d'appréhender leur représentation de la coopération et du système d'interaction.

Il s'agissait donc d'une expérimentation visant à orienter la conception de SpIn. Le but n'était pas de valider la situation de travail dans SpIn. On a donc limité les contraintes sur l'utilisateur au maximum. Dans cette optique, le contexte initial de l'apprentissage de l'appareil photo n'a pas été modifié (voir paragraphe précédent).

7.3.2 Les conditions

Les tests utilisateurs se sont déroulés au CNET à Lannion au sein du laboratoire DIH/PEPS/CES, en collaboration avec le LIFL et le CUEEP où le scénario a été écrit. Il consistait à mettre les utilisateurs en situation d'utilisation de SpIn au travers de la réalisation d'un scénario préétabli. Patricia Plénacoste a conçu les consignes, le déroulement et les documentations pour l'utilisateur.

Huit passations impliquant à chaque fois trois utilisateurs ont été organisées. À chaque passation, les trois utilisateurs étaient répartis dans trois salles distinctes équipées d'un PC SpIn et des périphériques nécessaires.

Les tests se sont organisés par plages horaires de 1 heure 30 environ qui se décomposent de la manière suivante :

- une présentation du test par l'observateur.

- une phase de familiarisation avec les périphériques et les éléments de l'application (1/4 d'heure environ). Au cours de cette phase les utilisateurs apprenaient à réaliser l'ensemble des actions qu'il est possible de faire dans l'interface SpIn à l'aide des périphériques mis à disposition, l'objectif n'étant pas d'évaluer les périphériques eux-mêmes. Une documentation sur l'application rappelant les différents principes d'utilisation leur était remise afin qu'ils puissent s'y référer tout au long du test.

- la passation des tests et réalisation d'un scénario préétabli (3/4 d'heure environ). Le scénario leur demandait d'apprendre ensemble à régler un appareil photo manuel afin de prendre une photo correcte d'un skieur en action. L'un des trois utilisateurs se voyait assigner un rôle d'expert chargé d'expliquer aux autres les mécanismes de la photo et d'initier la collaboration.

- réponse à un questionnaire lors d'un entretien avec l'utilisateur (1/4 d'heure environ). Le questionnaire final (cf. Annexe 2) porte sur le ressenti des utilisateurs, les difficultés éprouvées, les suggestions d'amélioration, etc.

Trois expérimentateurs ont effectué les observations et ont conduit l'entretien permettant de remplir le questionnaire final auprès de chaque utilisateur. Un enregistrement audio-vidéo a permis une analyse a posteriori du comportement des utilisateurs.

Personnes concernées

24 personnes ont réalisé ces tests. Elles sont pour la plupart utilisatrices de micro-ordinateurs (stations de travail PC, Mac ou Sun) et ont toutes des connaissances en télécommunications.

Parmi ces utilisateurs, un quart avait l'expérience des médiaspace. La plupart des utilisateurs ont entre 25 et 30 ans et sont employés au CNET. Ils n'ont généralement pas l'habitude d'utiliser des applications 3D sauf pour certains via la pratique de jeux sur micro-ordinateur.

La configuration particulière de l'application testée mettant en scène deux clones hommes et un clone femme, chaque trinôme était donc constitué de deux hommes et une femme.

Matériel utilisé

Chaque utilisateur avait à sa disposition :

- un micro-ordinateur muni de l'application SpIn. Il s'agissait d'un PC, équipé de deux processeurs de type Pentium II, et d'une carte graphique Titan II équipée du jeu de composants GMX de 3DBlabs, doté d'une prise en charge de toutes les primitives géométriques d'OpenGL et de capacités de placage de texture suffisantes pour SpIn.

- un téléphone en conférence à trois. Bien que SpIn dispose d'une fonction de communication sonore intégrée, utilisable avec un microcasque, la solution téléphonique a été privilégiée pour ce test, notamment pour pouvoir écouter et enregistrer ce que disaient les utilisateurs.

- un capteur 3D à ultrasons, le *FreeD*.

- une *Spacemouse*.

Deux clones d'homme et un clone de femme standards étaient utilisés pour tous les groupes. Il n'était pas possible de créer les clones de tous les intervenants.

7.3.3 Résultats des questionnaires

Les résultats présentés ici sont issus des éléments des résultats des questionnaires analysés par Karine Grein et Olivier Thunin du CNET de Lannion. Ils sont décrits dans la note technique interne

6238 du CNET, nous en avons extrait uniquement les résultats, que nous avons reproduit ici. Ils concernent les réponses aux questions que les sujets ont pu donner dans le questionnaire.

Familiarisation

Lors de la phase de familiarisation, les utilisateurs arrivent rapidement à se servir de la *Spacemouse* pour manipuler l'appareil photo, les bagues de l'appareil ainsi que le bandeau autour de la table. Les opérations de pointage utilisant le *FreeD* pour diriger le télépointeur sont plus difficiles à maîtriser. Sont principalement en cause la difficulté pour l'utilisateur d'agir en tenant compte de la profondeur et des difficultés liées à la sélection des différents dispositifs de l'appareil photo.

Dans l'ensemble, à l'issue d'1/4 d'heure réservé à la familiarisation, les utilisateurs arrivaient à maîtriser suffisamment les périphériques et à utiliser convenablement les concepts de l'interface.

Les périphériques

Il apparaît que les utilisateurs n'ont pas été gênés par la nécessité d'utiliser conjointement le *FreeD* et la *Spacemouse*. Cependant, il faut noter que la réalisation du scénario ne nécessitait pas d'utiliser le clavier ou de prendre des notes. Les utilisateurs avaient donc leurs deux mains libres pour utiliser les périphériques.

Les utilisateurs comprennent bien le principe du *FreeD*. Cependant, ils trouvent que ce périphérique n'est pas facile à utiliser. Il est jugé comme contraignant et fatiguant à manier. Il faut toujours être bien en face des capteurs (intégrés en haut à gauche de l'écran de l'ordinateur). Le maintien du bras levé pour s'en servir est fatigant.

Les utilisateurs affirment ne pas avoir eu de problème à apprendre son fonctionnement.

Les utilisateurs ne trouvent pas le *FreeD* très précis, ils ont souvent du mal à pointer les éléments de l'interface. L'utilisation du bouton du *FreeD* pour sélectionner ou désélectionner les objets est bien comprise.

Les utilisateurs trouvent la *Spacemouse* facile à utiliser, confortable et facile à apprendre. Dans l'ensemble ils trouvent ce périphérique agréable à utiliser.

Les clones

Un certain nombre d'inconvénients techniques ont modifié la perception que les utilisateurs avaient des clones :

- du fait que les voix des participants étaient véhiculées par le téléphone et non pas par l'interface SPIN elle-même, la présence «physique» d'un participant dans la réunion, signifiée par la présence de son clone, pouvait être décorrélée de sa présence «orale». Un participant, bien qu'il ait quitté la réunion, pouvait continuer à entendre les autres participants et à leur parler. Normalement, le son proposé est spatialisé ; ce qui permet de savoir qui est à droite et à gauche, et si un participant quitte la réunion, on ne l'entend plus.

- les trois clones utilisés pour les tests n'étaient pas, sauf exceptions fortuites, physiquement ressemblants aux utilisateurs. Seul le genre était respecté, c'est-à-dire que

dans chaque trinôme la femme était représentée par le clone féminin et les deux hommes par les deux clones masculins.

Ces deux particularités inhérentes à l'organisation des tests doivent être gardées à l'esprit pour l'interprétation des résultats concernant les clones.

Utilisation générale

Au lancement de l'application, les clones constituent un premier centre d'intérêt important pour les utilisateurs. Ils cherchent à savoir qui est représenté par chacun des clones visibles, et comment eux-mêmes prennent part à la scène. Les deux participants masculins n'ont pas trop de difficulté à comprendre qui sont les clones représentés à l'écran, l'un étant masculin et l'autre féminin. Par contre, la participante qui, elle, est en face des deux clones masculins (un bleu, un vert) a beaucoup de difficultés à savoir lequel représente chacun des deux hommes.

Ces problèmes d'identification des appartenances des clones sont évidemment liés à leur non-ressemblance avec les participants, à des problèmes relatifs à la couleur des clones et de leur pointeur, et à la décorrélation du son. Ces problèmes sont explicitement évoqués par les utilisateurs au cours du test par 7 trinômes sur 8. Les utilisateurs ne peuvent comprendre que leur pointeur n'a pas, pour les autres participants, la même couleur. Cela donne lieu à des dialogues du type «-Je vois ta flèche rose. - Mais quelle flèche rose, elle est bleu-vert la mienne ?». A ce problème s'ajoute le fait que deux des clones ont des couleurs trop proches l'une de l'autre.

Rapidement les clones sont délaissés, et ne semblent plus constituer un centre d'intérêt pour les participants.

Tout au long du test les utilisateurs ont surtout recours à la voix pour coordonner leur activité et ne se basent apparemment pas sur les informations véhiculées par les clones.

Intérêt perçu des clones et leur animation

Neuf utilisateurs sur 24 reconnaissent pendant l'entretien l'intérêt des clones comme indicateur de présence. Parallèlement, ils sont 17 à affirmer ne pas s'en être servi et ne pas les avoir regardé. Certains pensent que les clones prennent trop de place et que de simples étiquettes (photo ou icône) seraient suffisantes pour représenter les personnes. D'autres pensent fondamentalement que la visiophonie apporte plus de choses : visualiser en réel les personnes, montrer des documents, etc.

La majorité (22/24) des utilisateurs affirme avoir remarqué les jeux de regard et les mouvements de bras des clones. Beaucoup pensent qu'a priori la direction du regard peut apporter une aide dans la communication. Cependant, quasiment tous les utilisateurs (23/24) affirment ne pas avoir utilisé ces informations.

Six utilisateurs évoquent comme raison le fait qu'ils ne savent pas à partir de quelles informations comportementales le système déduit la direction du regard de leur clone. Ils ne pouvaient donc pas faire confiance à ce type d'information, puisqu'ils n'en connaissent pas l'origine. Bien que non interrogée sur cette question une personne a spontanément émis l'hypothèse que la direction du regard ne pouvait être déduite que par la

position du *FreeD*, puisque c'est le seul «point d'entrée». De même une personne s'interroge pendant les tests : «-Mais comment je peux bouger la tête ?».

Concernant le mouvement des bras, 22 d'entre eux n'ont pas compris ce qu'il signifiait et n'y trouvent aucune utilité dans la version actuelle (le bras suit la position du télépointeur).

L'organisation spatiale

Pendant les tests, la majorité des utilisateurs se sert convenablement du bandeau (rotation et élargissement). Son utilisation semble être assez intuitive. Les utilisateurs, globalement, jugent satisfaisante l'idée de positionner des objets sur un bandeau autour d'une table (18/24). Cinq personnes évoquent spontanément l'intérêt de pouvoir faire passer un document du bandeau au premier plan.

L'objet central

La possibilité de mettre un objet au centre est très bien perçue par l'ensemble des utilisateurs. Ils trouvent que cela permet de se focaliser sur l'objet sur lequel on travaille : «-on porte tout l'intérêt dessus».

Domaine privé/public

Au début des tests, les utilisateurs pensent que tout le monde à la même vue de l'objet et que s'ils le changent de position pour eux-mêmes, ils le font aussi forcément pour les autres. Par exemple, on peut relever les échanges verbaux suivants : «-Est-ce que vous voyez l'appareil tourner ?», «-J'ai sélectionné : tu peux faire tourner». Ensuite, ils s'aperçoivent que l'orientation de l'objet est personnelle à chacun.

Les avis concernant ce principe sont partagés : certains préfèrent que tous aient la même vue et d'autres pensent qu'à partir du moment où les actions (travail de réglages) sont communes et visibles, cela suffit.

Le fait que l'on ne voit pas lorsqu'un élément de l'objet central est sélectionné par un des participants est plus problématique. Cela donne lieu en début de test à de nombreux échanges peu appropriés, comme par exemple «-Vous voyez j'ai sélectionné la bague, elle doit apparaître en bleu chez vous», «-Qui a sélectionné ?», «Pourquoi on a pas vu que le bouton est sélectionné ?», «Qui tourne le diaphragme ?».

Transparence

Les utilisateurs déclarent être très souvent gênés par la transparence de l'objet central qui ne leur permet pas de bien visualiser ce qui se trouve sur le bandeau (17/24). Au cours du test, plusieurs positionnent le bandeau de manière à ce que le document qui les intéresse ne soit pas masqué par l'objet central ; ils le placent alors sur les côtés de l'écran et, de ce fait, le voient de biais (ceci a aussi des conséquences sur l'information fournie par le regard des clones).

De plus, ils rencontrent des difficultés à sélectionner les objets du bandeau car ils font malencontreusement apparaître l'objet central en faisant passer leur télépointeur involontairement dans la zone centrale.

Certains utilisateurs trouvent pénible de devoir pointer en permanence sur l'objet central afin de le laisser visible, par

exemple pour pouvoir observer les actions des autres participants.

Mode de sélection

Concernant les boîtes englobantes, les utilisateurs reconnaissent l'importance d'indiquer les éléments qui peuvent être sélectionnés. Cependant, la majorité des utilisateurs ont des difficultés, selon l'orientation de l'objet central, à en sélectionner efficacement les éléments. Ils leur arrivent assez souvent de ne pas sélectionner celui qu'ils voulaient. Il est également gênant de devoir chercher l'angle sous lequel l'objet doit être orienté pour pouvoir sélectionner un élément pourtant apparent. Au cours de l'entretien certains utilisateurs évoquent le manque de précision des boîtes englobantes et leur superposition sans utilité apparente (il s'agit en fait des changements de taille).

Les télépointeurs

La majorité des utilisateurs considère les télépointeurs comme étant a priori le moyen de communication le plus utile de l'interface et ceci même par rapport aux clones. Cependant, au cours des tests, ils ne semblent pas en tirer parti pour comprendre qui agit dans l'environnement. Les utilisateurs sont de ce fait obligés de coordonner verbalement leur activité. Plusieurs limites liées aux pointeurs ou à d'autres caractéristiques de l'interface peuvent expliquer cet état de fait :

- les problèmes liés à la couleur des pointeurs et plus largement aux difficultés d'identification des propriétaires des clones qui empêchent les utilisateurs de comprendre qui est en train d'agir.
- les télépointeurs ne représentent pas réellement les actions des participants. En effet, une fois qu'un élément est sélectionné et que l'utilisateur le manipule à l'aide de la *Spacemouse*, ni son télépointeur, ni l'élément sélectionné lui-même n'indiquent qu'il est en train d'agir dessus.
- sur le diaporama, du fait que les participants peuvent chacun afficher des pages différentes, l'élément pointé par l'un d'eux ne sera pas forcément celui que verront les autres.
- le manque de stabilité dans la trajectoire des télépointeurs, en partie liée au périphérique de commande (le *FreeD*), fait que la scène est un peu confuse. Un groupe a parlé de «guerre des étoiles» avec les flèches qui se croisent dans tous les sens. Par ailleurs, un trinôme a testé la manière dont les télépointeurs se positionnent sur chacun de leur écran. Il leur a semblé que la position des télépointeurs par rapport à l'objet central n'est pas toujours identique sur tous les postes.

L'appareil photo

Les utilisateurs trouvent que l'appareil et les possibilités qu'il offre sont des concepts intéressants et ludiques auxquels ils ont bien adhéré bien qu'ils rencontrent quelques difficultés à le manipuler avec le *FreeD*. Ils ont du mal à sélectionner les zones et ce notamment concernant les boutons de vitesse et de prise de photo. Pour sélectionner un bouton ou une bague de l'appareil, ils trouvent assez désagréable de devoir orienter l'objet sous un angle presque parfait et être bien en face ; d'autant plus que le bouton ou la bague est déjà visible

«tout ce qui est visible doit être cliquable». Par contre, une fois que la sélection est faite, les utilisateurs trouvent très facile la manipulation des réglages et de l'objet avec la *Spacemouse*.

Concernant la prise de photo à proprement parlé, les utilisateurs ne savent pas s'ils ont ou non appuyé sur le déclencheur. Aucun retour d'information ne leur indique s'ils ont pris la photo. De plus, si une photo a déjà été prise, la deuxième se superpose à la première sur le bandeau mais ceci n'est quasiment pas visible étant donné que les photos se ressemblent beaucoup. Les rendus finaux de la photo pour deux réglages proches sont trop similaires, les utilisateurs n'arrivent pas à voir les différences.

La documentation de l'appareil photo

Les utilisateurs pensent que l'affichage des pages du diaporama est le même pour tous. Ils n'imaginent pas spontanément qu'un document du bandeau qui semble forcément être visible de tous n'ait pas forcément le même contenu pour tous. Cela donne lieu à des comportements inadéquats pour 4/8 trinômes au moins tels que «-Vous voyez ça ?», en désignant un élément sur une page à l'aide de son télépointeur. Les autres participants n'ayant pas la même page affichée sur leur écran voient le télépointeur de ce participant désigné un élément qui occupe la même place dans la page affichée, mais qui est sans rapport avec celui-ci.

Les utilisateurs doivent donc recourir à une coordination verbale explicite dans la désignation à l'aide des télépointeurs pour s'assurer qu'ils parlent bien de la même chose.

Concernant la documentation, les utilisateurs ont souvent beaucoup de mal à se servir de la numérotation des pages et à naviguer dans le diaporama. Ils prennent souvent le numéro de la page suivante pour celui de la page courante.

Ceci est rendu encore plus difficile par la place de l'objet central qui «cache» souvent et interfère trop avec le positionnement des numéros.

Bilan

Nous venons d'exposer les résultats des manipulations des différents sujets ayant passé l'expérience. Le premier constat que l'on peut faire est le manque de préparation technique du prototype ayant servi à la manipulation. Les perturbations générées ont masqué certains concepts de SpIn, et les sujets se sont focalisés sur de «faux problèmes».

Les problèmes techniques

Le rendu d'un modèle de carte graphique à l'autre n'étant pas identique, certains problèmes techniques étaient uniquement liés à la plate-forme de l'évaluation :

- le fait qu'un clone et son télépointeur associé n'aient pas tout à fait la même couleur,
- régler le niveau de transparence de l'objet sur la table afin qu'il ne gêne pas la vue du bandeau.

Ensuite, d'autres problèmes techniques connus sont venus perturber la manipulation, en particulier :

- les boîtes englobantes. Le système utilisé dans SpIn exploite des boîtes englobantes parallèles aux axes, ce qui génère des boîtes beaucoup trop grandes une fois l'objet tourné dans certaines positions. Pour rendre le système utilisable, il faut modifier ce calcul pour afficher les vraies boîtes englobantes des objets.

- le *FreeD*. Ce périphérique est sensible aux perturbations de l'environnement, il se met en veille au bout d'une minute, et exige de garder la main dirigée vers le récepteur. Cela fait beaucoup de contraintes pour une utilisation courante de SpIn. Il sera donc souhaitable à l'avenir d'utiliser des capteurs de position 3D moins contraignants (comme le *Flock of Bird* d'Ascension Technologies), qui conservent toujours le mouvement naturel de la main pour déplacer le pointeur.

- le manque de retour sonore. La version utilisée alors de SpIn ne permettait pas de déclencher des sons à des moments particuliers. C'est une aide parfois indispensable, dans le cas du déclenchement de la prise de photo par exemple, les sujets auraient pu entendre un clic caractéristique. Avec ce type d'information, ils n'auraient pas douté du fait que la photo ait pu être prise ou non.

- les clones. Le fait que les sujets n'aient pas leur vrai clone dans SpIn a limité énormément l'information de téléprésence, cela a même pu la perturber par de fausses images de gens qui se connaissaient en dehors de l'évaluation. De même le fait que le système de communication ne soit pas présent (animation faciale et gestuelle des clones) a perturbé l'évaluation des primitives de coopération et de communication qui nous intéressent. Pour finir, le son positionné de SpIn n'était pas utilisé. Toutes les remarques à propos des clones faites dans cette évaluation n'ont donc aucune valeur. Des manipulations ultérieures dans d'autres conditions pourront nous permettre d'évaluer l'impact des clones dans SpIn.

Nous venons d'apporter les explications d'un certain nombre de problèmes techniques liés à l'avancement de SpIn au moment de l'évaluation. Les commentaires des utilisateurs nous ont permis de donner un poids à l'urgence de chaque évolution apportée depuis à notre environnement.

Les problèmes de fond

Les réflexions des utilisateurs, en dehors de l'évolution technique, nous donnent les priorités à établir à l'avenir dans l'orientation de SpIn. La plus flagrante est liée à la coopération. Certains utilisateurs ne comprenaient pas, par exemple, les modes de partage des documents. Pour que l'utilisateur soit à même de comprendre l'ensemble de l'activité sans avoir recours à des explications préalables, nous devons renforcer les informations véhiculées par l'interface. Autrement dit, nous devons donner plus d'informations sur l'activité en cours, afin que chacun comprenne les actions de ses interlocuteurs, à travers des métaphores visuelles simples impliquant les clones.

Un autre élément, découvert grâce à l'expérimentation, est la nécessité d'avoir le pointeur et le télépointeur d'un utilisateur de la même couleur. Il est apparu en effet que pendant la phase de coordination au début de l'activité, les sujets repéraient l'appartenance des télépointeurs grâce à leur couleur. On peut cependant penser que ce phénomène est un effet dû au manque de conformité des clones utilisés avec les sujets pendant cette évaluation, plutôt qu'un besoin dans SpIn (dans le cas où les clones représentent réellement les gens présent dans la réunion).

7.3.4 Résultats de l'analyse vidéo

Nous présentons ici les résultats de l'analyse des cassettes vidéos tournées pendant l'ensemble des expérimentations. Elles ont été dépouillées par Patricia Plénacoste, du laboratoire Trigone de Lille. On retrouve les problèmes techniques posés ci-dessus (que nous ne reprendrons donc pas ici), mais l'analyse vidéo révèle d'autres questions liées à SpIn.

Coopération

On a remarqué que les utilisateurs mettaient du temps à comprendre que les diapositives étaient visibles de tout le monde mais privées dans l'usage. De même, SpIn n'ayant pas d'outil de prise de point de vue dans cette version, il leur était difficile de comprendre que le point de vue était privé. Les sujets finissaient cependant par comprendre les modes de partage après quelques discussions de coordination. On rejoint ici la conclusion de l'analyse précédente, il manque des indices sur les modes de partage de document. De manière général, nous devons aller dans le sens de plus d'informations sur l'activité des utilisateurs, sur la coopération. Par exemple, les sujets discutaient beaucoup entre eux, négociaient verbalement, la coordination était faite en mode verbal. Ce qui était voulu et nécessaire de toutes façons, le système ne pouvant pas tout faire, la coordination verbale devra toujours être possible.

Dans le même sens, les utilisateurs avaient du mal à identifier qui était qui, faute de ressemblance des clones avec leurs interlocuteurs. Cela nous confirme dans le rôle indispensable de la téléprésence et des clones.

L'expérimentation montrait également un manque d'habitude du travail en coopération des différents sujets. Cela peut être dû à leur manque d'intérêt de l'apprentissage de la photo. Il faut en effet cibler la situation en fonction des personnes passant l'évaluation, mais l'appareil photo était la seule manipulation disponible sur SpIn. L'idéal étant de travailler sur une situation réelle, qui concernent un peu les gens qui passent l'évaluation.

De même, le type de coopération devrait être choisi en fonction de la situation de travail réel. Dans l'expérimentation, nous avons pu observer différents modes de médiation de l'expert : le directif (qui explique et montre aux autres qui regardent), le médiateur (qui regarde, analyse et qui montre si les autres n'y arrivent pas) et le passif (chacun fait ce qu'il veut de son côté). Toutefois ici, l'objectif n'était pas d'analyser la coopération.

Interaction

Lorsqu'ils devaient aller désigner ou sélectionner une partie de l'appareil photo avec le pointeur, les sujets, sachant qu'ils pouvaient l'atteindre directement, ne cherchaient pas à modifier leur point de vue pour se faciliter la tâche. On peut interpréter ce manque d'adaptation au système d'interaction, qui permet de modifier son point de vue sur l'appareil photo, par l'habitude des interfaces 2D où l'on agit d'une seule main sur des objets statiques. On peut donc s'interroger sur l'intérêt de tester SpIn avec des sujets conditionnés par leurs habitudes des interfaces 2D classiques, a fortiori de leur demander de faire des propositions. Il conviendrait mieux à l'avenir de choisir une proportion équivalente de gens ne maîtrisant pas les outils informatiques et d'experts (au sens de l'usage) de ces

outils, c'est-à-dire des gens qui ne soient ni des informaticiens, ni des gens habitués aux collecticiels classiques (basés sur la vidéo). Il faut en effet garder à l'esprit le secteur d'activité industriel visé à terme par un tel environnement, les besoins et les spécificités de ses utilisateurs, qui ne sont pas des informaticiens.

Lorsqu'un utilisateur agissait sur l'appareil photo, avec son télépointeur en désignant ou en modifiant la position d'une bague, les sujets ne voyaient pas bien ce qui se passait si l'appareil photo était en transparence. Ils positionnaient donc leur pointeur dans l'appareil photo pour le maintenir opaque. Si cette situation informe bien sur le focus de l'utilisateur, elle a deux inconvénients :

- si l'utilisateur ne maîtrise pas bien l'utilisation du capteur 3D et de son recentrage, il risque de devoir maintenir la main en l'air, d'où un problème de fatigue,
- mais surtout, le mouvement du pointeur dans l'appareil photo génère des mouvements du télépointeur associé, qui ne sont pas dans l'intention de l'utilisateur.

On en déduit que le meilleur choix n'est pas le mode implicite de détection du focus de l'utilisateur (exposé paragraphe 5.1.3), basé sur le fait que le pointeur est situé ou non à l'intérieur du document placé sur la table. Il faudra donc utiliser le mode explicite (proposé section 5.1.3 on page 110), où l'utilisateur spécifie lui-même s'il travaille sur le bandeau ou sur la scène, dans une prochaine utilisation de SpIn.

7.3.5 Bilan

Si nous avons principalement détaillé ici les problèmes, il faut aussi retenir que les expérimentations se sont bien déroulées et qu'elles ont rempli leur objectif, lié à l'apprentissage de la photo. Les utilisateurs comprenaient bien ce qu'il fallait faire avec l'appareil.

Au cours de cette étude empirique de SpIn, on a donc pu vérifier la compréhension de l'interface auprès des sujets. Les trois points principaux qui ont été dégagés sont :

- l'accent sur certains problèmes techniques,
- la confirmation qu'il est nécessaire de développer la coopération dans SpIn. Nous devons donner d'autres informations sur l'activité aux intervenants, les renseigner sur les documents et sur les autres. Cela passe par une mise en valeur des clones plus importante, à travers la communication et la coopération. En particulier, il faut améliorer la représentation de l'action des autres participants. Afin de pouvoir observer une partie de l'action d'un autre utilisateur et d'inférer correctement son objectif, l'utilisation des clones devrait se révéler cruciale, par leur rapport direct au réel, et donc à la facilité de compréhension de leurs animations (il faut une cohérence entre l'action et sa représentation),
- la remise en cause du principe de détection de focus implicite. Nous allons donc faire évoluer ce mode, et tester le mode de détection explicite, qui permet d'utiliser plus simplement la mobilité relative (§ 4.2.4 - p. 98), le positionnement vertical de la scène étant similaire.

Dans une prochaine évaluation, nous devons essayer de choisir une tâche où chacun ne peut pas se contenter de manipuler de son côté, comme il était possible de le faire pour l'appareil photo, mais où il est nécessaire d'aller retirer de l'information auprès de ses interlocuteurs. Il faudra choisir une

manipulation de terrain, où l'on pourra observer avant la situation réelle (faire une analyse de l'activité) afin de choisir au mieux les métaphores dans SpIn.

CONCLUSION

l'intérêt de la troisième dimension

Nous venons d'exposer en détails les primitives d'interaction et de coopération de SpIn. Nous avons pu conserver notre positionnement tout au long de sa conception en respectant nos contraintes de départ, à commencer par la première avec la mise en valeur de la troisième dimension dans SpIn grâce au niveau d'intégration de l'environnement de travail. La troisième dimension nous permet de créer des primitives de coopération en créant des relations visuelles directes entre les documents et les acteurs, impossible à réaliser de façon aussi intuitive dans d'autres formes d'environnements. SpIn justifie aussi son approche 3D grâce à l'intégration de documents 3D, objets virtuels au centre des débats de beaucoup de réunions d'entreprises, mais qui ne sont pas forcément accessibles dans une réunion réelle (bâtiments, produits en projets, etc).

le contexte bureautique

La deuxième contrainte, l'environnement bureautique, a également été respectée dans la mesure où à travers ses évolutions, l'utilisation de SpIn permet toujours de garder le contact avec son environnement de travail immédiat (collègues, téléphone, etc). De même, les avancées technologiques en matière de cartes graphiques 3D ont donné raison à notre approche, en démocratisant et en développant les capacités d'affichage de scènes tridimensionnelles de tous les micro-ordinateurs, du PC en particulier. SpIn est un environnement qui fonctionne sur des machines de plus en plus économiques.

l'interaction dans SpIn

Notre environnement s'intéresse à l'activité de l'utilisateur, ce qui met en avant la notion de document. On lui fournit les éléments nécessaires pour agir directement avec toutes les sortes de documents possibles qui peuvent composer l'espace de travail. La notion d'application au sens des interfaces classiques n'apparaît pas explicitement. L'environnement de travail virtuel permet de visualiser l'ensemble des documents, sans barrière ni cadre pouvant détourner l'attention de l'utilisateur. On les place autour de la table de réunion, sur laquelle l'utilisateur peut amener un document particulier afin d'interagir dessus.

On peut agir sur un document en le sélectionnant, lui ou une de ses parties. Si plusieurs actions sont possibles sur cet élément sélectionné, un menu apparaît, sinon on accède directement à une action particulière, qui peut être simple, comme un changement de point de vue, ou plus compliquée, comme une édition.

Un document dans SpIn est donc constitué d'un ensemble de données -le document lui-même- et d'un ensemble de comportements qui lui sont associés. Un comportement est en fait une réaction par une action à un évènement donné (par exemple : on accède à l'action de changement de point de vue en sélectionnant le document avec l'outil ad-hoc). Si un document ou une de ses parties n'a pas de comportement spécifié, on ne peut bien sûr pas le sélectionner ni agir dessus.

Les actions génériques que l'on peut appliquer à tous les documents, en particulier celles liées à la coopération, sont accessibles à travers une barre d'outils située devant la table.

Nous recommandons dans SpIn d'utiliser des actions sous forme d'outils. Ils permettent une interaction directe avec un document, un acteur, ou les deux.

***l'interaction
à distance***

Ces actions peuvent concerner une activité coopérative se déroulant dans SpIn. L'environnement permet donc de reporter à distance toutes les actions effectuées par un utilisateur sur un document partagé. Ses interlocuteurs interpréteront ses modifications à l'aide de son clone, la représentation tridimensionnelle et réaliste de cet utilisateur. Le clone permet de connaître le point d'intérêt d'un utilisateur grâce à la direction de son regard. Il possède un télépointeur pour désigner à distance et montrer leur point d'action aux autres.

Voilà ce qui constitue l'interface utilisateur de SpIn. Ces composants interactifs, tout comme ceux de toutes les activités que l'on voudra mettre en place dans SpIn (comme celle de l'appareil photo vue au chapitre précédent), doivent respecter les primitives d'interaction de base de SpIn.

***les
mécanismes
d'interaction
de base***

Ces primitives de base constituent un modèle d'interaction 3D, dont les principaux éléments sont les suivants :

- l'utilisation d'un capteur de position 3D (trois degrés de liberté) dans la main dominante pour déplacer un pointeur dans l'ensemble de l'espace de l'interface. Ce pointeur sert à désigner et à sélectionner les objets,

- l'utilisation d'un périphérique isométrique dans la main non-dominante (au moins trois degrés de liberté) pour réaliser les actions sur les documents,

- le respect d'un certain nombre d'éléments pour aider l'utilisateur à percevoir l'effet de profondeur généré dans les images de synthèse et à se construire facilement une représentation mentale de la scène globale :

- l'utilisation d'un espace tridimensionnel fermé dans lequel l'utilisateur puisse déplacer partout son pointeur suivant les coordonnées spatiales (X,Y,Z) du capteur de position. La rotation du pointeur n'étant pas donnée par celle de la main, mais par un calcul visant à renforcer l'effet de dynamique du mouvement, basé sur ses déplacements,
- limiter la navigation au maximum,
- utiliser des indices perceptifs dont les principaux sont l'éclairage des objets, l'utilisation d'ombres portées, et d'indices dynamiques permettant de positionner facilement le pointeur par rapport aux autres objets, comme les transparences, les boîtes englobantes progressives ou la mobilité relative.

évaluations

Pour aboutir à cette proposition, nous avons réalisé plusieurs évaluations : pour préciser quel type de périphérique utilisé, pour mesurer quels indices perceptifs étaient pertinents, pour savoir quelles caractéristiques les ombres portées devaient avoir; mais surtout, la manipulation de l'appareil photo, première réalisation basée sur un scénario construit, a permis de réaliser une évaluation sur SpIn, de tester la manière dont les utilisateurs travaillent avec l'interface, d'essayer de comprendre la façon dont ils appréhendent la coopération et l'interaction dans SpIn.

Cela nous a montré que malgré des développements lourds et poussés, des détails techniques comme le "dysfonctionnement" du capteur de position acoustique *FreeD* pouvait déformer le sens des manipulations.

Mais cette étude empirique nous a également révélé quelques revers dans notre système coopératif, en particulier le passage en transparence de l'objet sur la scène de façon automatique lorsque le pointeur n'est plus à l'intérieur (pour pouvoir voir le bandeau). Ce qui nous permet de faire d'autres propositions, permettant d'avoir des modes plus explicites que l'utilisateur ne peut pas détourner, ne perturbant plus l'information transmise.

perspectives Il faut donc continuer à tester SpIn, si possible en conditions réelles. La meilleure solution serait de trouver une entreprise prête à utiliser une version de SpIn, adaptée à leurs activités de réunions.

Une autre solution, qui a commencé à être mise en oeuvre au LIFL, est un client passif pour SpIn, permettant de voir un interlocuteur possédant une version interactive (normale). Développé en VRML97 et en Java, il pourrait permettre à n'importe quelle machine sur Internet, sans périphérique particulier, de visualiser une activité se déroulant dans SpIn. Il pourrait être utilisé pour diffuser des cours, notre environnement étant un support riche en possibilités de présentation et d'explications interactives.

Quelle que soit la façon dont les tests sont réalisés, il faudra qu'ils permettent à la version actuelle d'évoluer. Le modèle d'interaction pourra être remis en cause sur certains aspects, comme les périphériques, où le toucher n'a pas été exploré faute de correspondre aux contraintes actuelles de SpIn. Des changements pourraient être envisagés si la technologie évolue et se démocratise. Il sera intéressant par exemple d'utiliser des périphériques à retour d'effort dans leur version bureautique, comme le *Phantom Desktop* de Sensable Technologies, pour donner des informations à l'utilisateur sur ses déplacements, les objets qu'il rencontre, sans toutefois entraver ses mouvements.

De manière générale, ce sont également les clones, et à travers eux les possibilités de coopération de SpIn, qui devront être étendus. L'intérêt de notre environnement réside dans la représentation des acteurs et des documents, dans le même espace. Il faudra continuer à concevoir des outils de coopération et des métaphores visuelles qui permettront de renseigner plus encore un utilisateur sur l'activité de ses interlocuteurs. En particulier sur les droits et les possibilités d'interaction qu'il a sur un document, les utilisateurs doivent en effet comprendre simplement ce qu'ils peuvent faire d'un document, et même plus, qui le voit et ce que les autres en font. A terme, l'objectif de SpIn devra être de pouvoir remplacer pour les utilisateurs n'importe quelle réunion (réelle), en ajoutant une plus value grâce au virtuel et à ses possibilités, comme la visualisation de produits non existants, les modes particuliers de partage de documents ou les outils de coordination de l'activité. Ce qui implique l'intégration d'un système de communication complet. SpIn ne peut pas être utilisé en conditions réelles en entreprise sans les premiers éléments de support des gestes et de l'animation faciale.

Il faudra alors savoir déterminer quand l'utilisateur communique simplement et quand il travaille dans SpIn, et même détecter quand il s'échappe temporairement de la réunion (pour répondre à ses collègues par exemple). Il s'agit de détecter le passage d'un mode à l'autre avec tous les éléments de l'activité de l'utilisateur. On devra peut-être demander à l'utilisateur de spécifier explicitement quand il passe en mode communication directe. Le passage d'un

mode de communication à l'autre entre dans la conception d'un gestionnaire intelligent de l'activité des clones.

La recherche autour de SpIn va s'élargir de plus en plus pour toucher des thèmes généraux aux interfaces 3D. En particulier, des questions restent en suspens à propos des documents en deux dimensions, et leur intégration dans les interfaces 3D : quelle est leur place ? comment les intégrer et les représenter ? faut-il des modes d'interaction particuliers ? L'organisation spatiale reste également un thème d'actualité, avec les évolutions que l'on peut donner au mode de visualisation du bandeau.

plus proche des standards et de l'industrie Sur les aspects techniques, pour que SpIn puisse se développer, il faudra aussi une plus grande adéquation aux standards actuels. Pour éviter des développements inutiles, profiter des avancées d'autres domaines de recherche en informatique, mais surtout pour pouvoir supporter le plus grand nombre de documents possibles. Ainsi, les prochaines versions de SpIn devront supporter des standards établis comme HTML ou VRML97. Elles devront également supporter et utiliser d'autres standards prometteurs pour gérer l'aspect réparti et la diffusion de flux multimédia, comme CORBA et MPEG4; sans quoi, la mise en oeuvre liée à l'évolution de SpIn risque de s'encombrer de problèmes complexes qui ne sont pas directement liés aux IHM 3D ou au TCAO synchrone.

ouvertures La réalité virtuelle a ouvert de nouveaux champs d'applications dans l'informatique, comme la télé-opération. Elle peut permettre également à d'autres domaines de l'informatique de sortir de solutions trop conventionnelles, qui par manque de recul ou de moyen n'ont pas exploré les voies qui leur permettraient de donner une solution acceptable à tout utilisateur novice. C'est ce que SpIn commence à démontrer en s'appuyant sur un environnement de communication et de coopération entièrement tridimensionnel adapté à l'entreprise. Si les possibilités sont encore trop limitées, le raisonnement est là. Même si ce n'est pas par un CVE tel que SpIn, les outils de travail du futur passeront très probablement par une approche similaire à la notre. On peut d'ailleurs constater que d'autres grands groupes de télécommunications que France Telecom (le CNET en ce qui nous concerne) s'intéressent sérieusement aux environnements de communication avancée. Quand ces projets devront s'adapter aux possibilités des entreprises, on devrait voir apparaître dans le milieu professionnel des outils de réalité virtuelle adaptés à la bureautique, supportant, intégrant et surtout centralisant tous les supports de communication actuels. La convergence des moyens de communication devra passer par une adaptation à l'utilisateur, afin de lui proposer un environnement de travail où toute l'information est à sa disposition, et où ses collaborateurs sont également présents. Ce qui passe par une redéfinition des environnements de travail tels que nous les connaissons, en faisant un certain nombre de propositions comme nous les avons faites. Les interfaces homme-machine en trois dimensions ont donc encore beaucoup à apporter aux interfaces de travail; en particulier dans les domaines d'applications où l'outil utilisé a une incidence sur les méthodes de conception, comme en architecture ou, de manière générale, dans les bureaux d'études. Le besoin en matière de travail coopératif synchrone assisté par ordinateur est déjà présent dans ces secteurs, et la complexité de leurs programmes pourrait gagner en ouverture et en utilisabilité à travers une approche entièrement 3D.

Bibliographie

Les références données ci-dessous sont classées par ordre alphabétique.

- [Apple] Apple Inc.
Apple Human Interface Guidelines
Addison Wesley, 1987.
- [Balakrishnan 97] Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, George Fitzmaurice
The Rockin' Mouse : Integral 3D Manipulation on a Plane
Actes de la conférence CHI, ACM Press, p. 311-318, 1997.
- [Balakrishnan 99] Ravin Balakrishnan, Gordon Kurtenbach
Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces
Actes de la conférence CHI, ACM Press, p. 56-63, 1999.
- [Barne 96] Laurent Barne
Le son dans les systèmes informatiques pour le travail coopératif
Thèse de doctorat de l'université de Lille I, le 30 avril 1996.
- [Beaudouin 94] Michel Beaudouin-Lafon
L'usage du son dans les systèmes interactifs
Actes de la conférence IHM, p. 119-124, décembre 1994.
- [Beaudouin 96] Michel Beaudouin-Lafon
Les habits neufs du travail en équipe
La Recherche, 285, p. 48-52, mars 1996.
- [Beaudouin 97] Michel Beaudouin-Lafon
Interaction instrumentale : de la manipulation à la réalité augmentée
Actes de la conférence IHM, p. 97-104, septembre 1997.
- [Bell 96] Gavin Bell, Anthony Parisi, Mark Pesce
The Virtual Reality Modeling Language, Version 1.0 Specification
<http://www.vrml.org/VRML1.0/vrml10c.html>
janvier 1996.
- [Benford 93] Steve Benford, Lennart E. Fahlén
A Spatial Model of Interaction in Large Virtual Environments
Actes de la conférence ECSCW, Kluwer Academic, 1993.

-
- [Benford 95] Steve Benford, Chris Greenhalgh, John Bowers, Dave Snowdon et Lennart E. Fahlén
User Embodiment in Collaborative Virtual Environments
Actes de la conférence CHI, ACM Press, p. 242-248, 1995.
- [Benford 97] Steve Benford, John Bowers, Lennart E. Fahlén, Chris Greenhalgh, Dave Snowdon
Embodiments, avatars, clones and agents for multi-user, multi-sensory virtual worlds
Multimédia Systems, Springer Verlag, 5, p. 93-104, 1997.
- [Bier 93] E.A. Bier, M.C. Stone, K. Pier, W. Buxton, T.D. DeRose
Toolglass and Magic Lenses: The See-Through Interface.
Actes du Siggraph, p. 73-80, 1993.
- [Blake 94] Andrew Blake, Michael Isard
3D position, attitude and shape input using video tracking of hands and lips
Actes du Siggraph, p.185-192, 1994.
- [Bonnet 82] Claude Bonnet
La perception visuelle des formes
Traité de psychologie cognitive, Dunod, p. 2-82, 1989.
- [Bregler 97] C. Bregler, M. Covell, M. Slaney
Video Rewrite : Driving Visual Speech with Audio
Actes du Siggraph, p. 353, 1997.
- [Brookshire 92] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, Andries van Dam
Three-Dimensional Widgets
SIGGRAPH Symposium on Interactive 3D Graphics, p. 183-188, 1992.
- [Burdea 96] Grigore Burdea, Paul Richard, Philippe Coiffet
Multimodal Virtual Reality: Input-Output Devices, System Integration, and Human Factors
International Journal of Human-Computer Interaction, 8(1), p. 5-24, 1996.
- [Buxton 92] William A. S. Buxton
Telepresence: Integrating Shared Task and Person Spaces
Actes de la conférence Graphics Interface, Morgan Kaufmann, p. 123-129, 1992.
- [Cadoz 94a] Claude Cadoz
Les Réalités virtuelles
collection Dominos, Flammarion, 1994.
- [Cadoz 94b] Claude Cadoz
Le geste canal de communication homme/machine - la communication instrumentale
Technique et Science Informatiques, 13(1), p. 31-61, 1994.
- [Cadoz 96] Claude Cadoz
Réintroduire les sensations physiques
spécial La Recherche, 285, p. 80-84, 1996.
- [Carlier 98] Patrice Carlier
ScenIC : une proposition pour l'assistance à la composition spatio-temporelle des interfaces des environnements de TCAO
Thèse d'informatique de l'Université des Sciences et Technologies de Lille, le 12 octobre 1998.
- [Cassel 94] Justine Cassel, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Becket, Brett Douville, Scott Prevost et Matthew Stone
Animated Conversation : Rule-based Generation of Facial Expression, Gesture and Spoken Intonation, for Multiple Conversational Agents
Actes du Siggraph, p. 413-419, 1994.
- [Chen 88] M. Chen
A Study in Interactive 3D Rotation Using 2D Control Devices
ACM Computer Graphics, 22(4), p. 121-127, août 1988.

-
- [Coomans 97] M.K.D. Coomans, H.J.P. Timmermans
Towards a Taxonomy of Virtual Reality User Interfaces
Actes de la conférence Information Visualisation (IV), 1997.
- [Demarey 98] Catherine Demarey, Patricia Plénacoste
Effect of 2D versus 3D spatialisation on the identification of everyday sounds: a preliminary study.
International Conference of Auditory Display, Glasgow, novembre 1998.
- [DIVE] le projet DIVE
<http://www.sics.se/dce/dive/dive.html>
environnement de réalité virtuelle multi-utilisateurs.
- [Dix 98] Alan Dix, Janet Finlay, Gregory Aboud, Russel Beale
Human Computer Interaction
Prentice Hall, deuxième édition, 1998.
- [Draper 98] John V. Draper, David B. Kaber, John M. Usher
Telepresence
Human Factors, 40(3), p. 354-375, 1998.
- [Driver 90] Jesse Driver, Rob Read, Eric Blough, Kenneth Seah
An Evaluation of the Polhemus, Spaceball, and Mouse Devices for 3D Cursor Positioning
Computer Science Department, Université du Texas (Austin), août 1990,
rapport technique numéro TR-90-29.
- [Dumas 98] C. Dumas, S. Degrande, G. Saugis, C. Chaillou, M.-L. Viaud
A 3-D Interface for Cooperative Work
Actes de la conférence CVE, Université de Manchester, UK, juin 1998.
- [Dumas 99] Cédric Dumas, Samuel Degrande, Grégory Saugis, Christophe Chaillou, Marie-Luce
Viaud, Patricia Plénacoste
SpIn : a 3-D Interface for Cooperative Work
Virtual Reality Society Journal, Edition Springer-Verlag, 1999.
- [EAI] External Authoring Interface
<http://cosmosoftware.com/developer/eai/java2.html>
explication complète sur le site CosmoSoftware de Silicon Graphics.
- [Edgar 95] Graham K. Edgar, Peter J. Rex
Vision and displays
Simulated and Virtual Realities, édition Taylor & Francis, chapitre 4, p. 85-101, 1995.
- [Emmerik 90] Maarten J.G.M. van Emmerik
A Direct Manipulation Technique for Specifying 3D Object Transformations with a 2D Input Device
Automatika, the journal of the Yugoslav Committee ETAN, 31(1-2), p. 95-99, 1990.
- [Fuchs 96] Philippe Fuchs
Les Interfaces de la Réalité Virtuelle
édition de l'Association des journées Internationales de l'Informatique de Montpellier-District, 1996.
- [Furnas 86] George W. Furnas
Generalized fisheye views
Actes de la conférence CHI, ACM Press, p. 16-23, 1986.
- [Gaver 97] William Gaver
Auditory interfaces
dans Handbook of Human-Computer,
de M.G. Helander, T.K. Landauer, P. Prabhu (Eds.)
Interaction, deuxième édition. Amsterdam, The Netherlands: Elsevier Science, 1997.
- [Gigante 93] Michael A. Gigante
Virtual Reality: Definitions, History and Applications
dans Virtual Reality Systems, Academic Press, 1993.

-
- [Gomez 94] Julian E. Gómez (avec Dan Venolia, Andries van Dam, Tony Fields, Rikk Carey)
Why is 3d interaction so hard and what can we really do about it ?
Actes du Siggraph, panel, ACM Press, p. 492, 1994.
- [Greenhalgh 95] Chris Greenhalgh, Steve Benford
MASSIVE : a Virtual Reality System for Tele-conferencing
ACM Transactions on Computer Human Interface, 2(3), p. 239-261, 1995.
- [Hand 97] Chris Hand
A Survey of 3D Interaction Techniques
Computer Graphics forum, 16(5), p. 269-281, 1997.
- [Harrison 95] Beverly L. Harrison, Hiroshi Ishii, Kim J. Vicente, William A.S. Buxton
Transparent Layered User Interface : An Evaluation of a Display Design to Enhance Focused and Divided Attention
Actes de la conférence CHI, ACM Press, p. 317-324, 1995.
- [Henderson 86] D. A. Henderson, Jr. et S. K. Card
Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface
ACM Transactions on Graphics, 5(3) , p. 211-243, Juillet 1986.
- [Henry 93] D. Henry, T. Furness
Spatial Perception in Virtual Environments: Evaluating an Architectural Application
Actes de la conférence VRAIS, 1993.
- [Hinckley 94] Ken Hinckley, Randy Pausch, John C. Goble, Neal F. Kassel
A Survey of Design Issues in Spatial Input
Actes de la conférence UIST, p. 213-222, 1994.
- [Hinckley 97a] Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt et Neal Kassell
Usability Analysis of 3D Rotation Techniques
Actes de la conférence UIST, p. 1-10, 1997.
- [Hinckley 97b] Ken Hinckley, Randy Pausch, Dennis Proffitt, James Patten et Neal Kassell
Cooperative Bimanual Action
Actes de la conférence CHI, ACM Press, p. 27-34, 1997.
- [Hinckley 99] Ken Hinckley, Mike Sinclair
Touch-Sensing Input Devices
Actes de la conférence CHI, ACM Press, p.223-230, 1999.
- [Hoogstoel 97] Frédéric Hoogstoel, Patricia Plénacoste, Jean-Claude-Tarby
ODESCA/SCENIC/SPIN : Rôles et fonctions
document de travail, laboratoire Trigone.
- [Hudson 92] Scott E. Hudson
Adding Shadows to a 3D Cursor
ACM Transactions on Graphics, 11(2), p. 193-199, Avril 1992.
- [IBM a] IBM Guidelines
les règles de conception d'interface homme machine
<http://www.ibm.com/ibm/easy/i/index.html>
- [IBM b] IBM Guidelines
Realplaces Elements for Creating 3D User Environments
<http://www.ibm.com/ibm/guidelines/hci/design/3d/>
- [Ishii 92] Hiroshi Ishii, Minoru Kobayashi
ClearBoard: a seamless medium for shared drawing and conversation with eye contact
Actes de la conférence CHI, ACM Press, p. 525-532, 1992.
- [Kabbash 93] Paul Kabbash, I. Scott MacKenzie, William Buxton
Human performance using computer input devices in the preferred and non-preferred hands
Actes de la conférence CHI, ACM Press, p. 474-481, 1993.

-
- [Kabbash 94] Paul Kabbash, William Buxton et Abigail Sellen
Two-Handed Input in a Compound Task
Actes de la conférence CHI, ACM Press, p. 417-423, 1994.
- [Kersten 97] D. Kersten, P. Mamassian, D. C. Knill
Moving cast shadows induce apparent motion in depth
Perception, 26(2), p. 171-192, 1997.
- [Kurtenbach 97] Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, Bill Buxton
The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency
Actes de la conférence CHI, ACM Press, 1997.
- [Kuzuoka 94] Hideaki Kuzuoka, Toshio Kosuge et Masatomo Tanaka
GestureCam; a Video Communication System for Sympathetic Remote Collaboration
Actes de la conférence CSCW, ACM Press, p. 35-43, 1994.
- [Mackinlay 91] J.D. Mackinlay, G.G. Robertson et S.K. Card
The perspective wall: detail and context smoothly integrated
Actes de la conférence CHI, ACM Press, p. 173-180, 1991.
- [Mamassian 98] P. Mamassian, D.C. Knill, D. Kersten
The Perception of Cast Shadows
Trends in Cognitive Sciences, 2(8), p. 288-295, 1998.
- [Manitiu 98] Andrei Manitiu, Jean-Pierre Jessel, René Caubet
Métaphores pour l'interaction 3D dans les mondes virtuels
Actes de l'AFIG, p. 165-174, 1998.
- [Mine 96] Mark Mine
Working in a Virtual World: Interaction Techniques Used in the Chapel Hill Immersive Modeling Program
UNC Chapel Hill Computer Science, rapport technique numéro TR96-029, 1996.
- [Morineau 96] Thierry Morineau
Adaptation Cognitive à un environnement virtuel lors de premières immersions
thèse de psychologie de l' université d' Anger, soutenue le 25 octobre 1996.
- [Morningstar 90] C. Morningstar, F.R. Farmer
The Lessons of Lucasfilm's Habitat
publication en ligne <ftp://sunsite.unc.edu/pub/academic/communications/papers/habitat/lessons.txt>
publié également dans : M.Benedikt (éd.) Cyberspace: First Steps, MIT Press, 1991.
- [MSS] Department of Speech, Music and Hearing, Royal Institute for Technology, Suède
<http://www.speech.kth.se/multimodal/>
Multimodal Speech Synthesis project.
- [Munzner 97] Tamara Munzner
H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space
Actes de la conférence IEEE Symposium on Information Visualization, p. 2-10, 1997.
- [Najork 95] Marc A. Najork, Marc H. Brown
Obliq-3D: A High-Level, Fast-Turnaround 3D Animation System
IEEE Transactions on Visualization and Computer Graphics, 1(2), p. 175-193, 1995.
- [Noser 96] H.Noser, T. Çapin, I. Pandzic, N. Magnenat Thalmann, D. Thalmann
Playing Games Through the Virtual Life Network
Actes de la conférence Alife, Nara, Japon, 1996.
- [Obeysekare 96] U. Obeysekare, C. Williams, J. Durbin, L. Rosenblum, R. Rosenberg, F. Grinstein, R. Ramamurthi, A. Landsberg, W. Sandberg
Virtual Workbench - A Non-Immersive Virtual Environment for Visualizing and Interacting with 3D Objects for Scientific Visualization
Actes de la conférence Visualisation, San Francisco, USA, 1996.

-
- [Okada 94] Ken-Ichi Okada, Fumihiko Maeda, Yusuke Icikawaa et Yutaka Matsushita
Multiparty Videoconferencing at Virtual Social Distance: MAJIC design
Actes de la conférence CSCW, ACM Press, p. 385-393, 1994.
- [Olsen 98] Jr. Dan R. Olsen
Developping User Interface
édition Morgan Kaufmann, 1998.
- [Peters 94] Sophie Peters Van Deinse
Devenir Ingénieur en communication grâce à la visioconférence
L'Usine Nouvelle, 2448, p. 126, mars 1994.
- [Plénacoste 98] Patricia Plénacoste, Yann Coello, Pierre Magne
Structuration du champ visuel et performance motrice: le problème des interfaces graphiques
VII^{ème} colloque de l'association pour la recherche cognitive, Saint Denis, p. 401-409 , décembre 1998.
- [Poupyrev 96] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, Tadao Ichikawa
The Go-Go Interaction Technique : Non-linear Mapping for Direct Manipulation in VR
Actes de la conférence UIST, p. 79-80, 1996.
- [Primet 94] P. Primet, S. Akkouche
Réflexions autour du Concept de Télépointeur et Réalisation d'un Outil coopératif de Désignation
Actes de la conférence IHM, Lille, 1994.
- [Regan 95] E.C. Regan
An investigation into nausea and other side-effects of head-coupled immersive virtual reality
Virtual Reality, 1(1), p. 17-32, 1995.
- [Roberts 98] D. Roberts
RealPlaces, 3D interface for Office Applications.
IEE colloquium on The 3D Interface For The Information Worker, 98(437), mai 1998.
- [Robertson 91] G.G. Robertson, S.K. Card et J.D. Mackinlay
Cone Trees: animated 3D visualizations of hierarchical information
Actes de la conférence CHI, ACM Press, 1991.
- [Robertson 93] G.G. Robertson, S.K. Card et J.D. Mackinlay
Information Visualization using 3D Interactive Animation
Communications of the ACM, 36(4), avril 1993.
- [Roll 98] Jean-Pierre Roll
Les muscles, organes de la perception
Pour la science, p. 92-99, juin 1998.
- [Rosenberg 96] Louis Rosenberg, Scott Brave,
Using Force Feedback to Enhance Human Performance in Graphical User
Actes de la conférence CHI, ACM Press, 1996.
- [Saugis 98] Grégory Saugis
Interface 3D pour le travail coopératif synchrone, une proposition
Thèse d'informatique de l'Université des Sciences et Technologies de Lille, le 30 janvier 1998.
- [Scapin 86] Dominique L. Scapin
Guide Ergonomique de Conception des Interfaces Homme-Machine
INRIA, Rapport Technique numéro 77, octobre 1996.
- [Schneiderman 83] B. Schneiderman
Direct manipulation : A step beyond programming languages
IEEE Computer, 16, p. 57-69, 1983.
- [Shaw 97] Chris Shaw and Mark Green
THRED: A Two-Handed Design System
Multimedia Systems Journal, 5(2), ACM/Springer Verlag, 1997.

-
- [Shoemake 92] K. Schoemake
ARCBALL : A User Interface for Specifying Three-Dimensional Orientation Using a Mouse
Actes de Graphics Interface, Morgan Kaufmann, p. 151-156, 1992.
- [Shu 94] Li Shu, Woodie Flowers
Teledesign: Groupware user experiments in three-dimensionnal computer-aid-design
Collaborative Computing, 1(1), p. 1-14, 1994.
- [Stoakley 95] Richard Stoakley, Matthew J. Conway, Randy Pausch
Virtual Reality on a WIM: Interactive Worlds in Miniature
Actes de la conférence CHI, ACM Press, p. 265-272, 1995.
- [Szalavari 97] Zsolt Szalavari, Michael Gervautz
The Personnal Interaction Panel - a Two-Handed Interface for Augmented Reality
Computer Graphics Forum, the international journal of the Eurographics association,
Blackwell Publishers, 16(3), p. 335-246, 1997.
- [Tanaka 96] Shunsuke Tanaka
Desktop conferencing system using multiple still-pictures: Desktop-MAJIC
Actes de la conférence ECSCW, Kluwer Academic, 1996.
- [Teleface] le projet Teleface
<http://www.speech.kth.se/teleface/>
animation de clone à partir d'une simple voix.
- [Télévirtualité] équipe Télévirtualité de l'INA
<http://www.ina.fr/TV/>
un système de création et d'animation de clones.
- [Ultima On Line] Origin Ultima
<http://www.origin.ea.com/>
le site du premier jeu commercial considérant plusieurs centaines de joueurs simultanément.
- [Vander 89] Arthur J. Vander, James H. Sherman, Dorothy S. Luciano, Jean R. Gonthier
Physiologie humaine
Mc Graw-Hill, seconde édition, 1989.
- [Venolia 93] Dan Venolia
Facile 3D Direct Manipulation
Actes de la conférence CHI, ACM Press, p. 31-36, 1993.
- [Viéville 95] Claude Viéville
Structuring conversation in asynchronous communication systems to support collaborative learning at a distance
ED-MEDIA, Graz, Autriche, 21 juin, 1995.
- [Virtual Friends] Hapték Inc.
<http://www.hapték.com/>
logiciel de communication textuelle avec animation faciale.
- [Wang 99] Yanqing Wang, Christine L. Mac Kenzie
Object Manipulation in Virtual Environments : Relative Size Matters
Actes de la conférence CHI, ACM Press, p. 48-55, 1999.
- [Wanger 92] Leonard R. Wanger, James A. Ferwerda, Donald P. Greenberg
Perceiving Spatial Relationships in Computer-Generated Images
IEEE Computer Graphics and Applications, p. 44-57, mai 1992.
- [Ware 88] Colin Ware, Danny R. Jessome
Using the Bat : A Six-Dimensional Mouse for Object Placement
IEEE Computer Graphics and Applications, 8(6), p. 65-70, novembre 1988.

-
- [Watlington 95] John A. Watlington, Mark Lucente, Carlton J. Sparrell, V. Michael Bove, Jr., Ichiro Tamitani
A hardware architecture for rapid generation of electro-holographic fringe patterns
Actes de la conférence SPIE, 2406(23), 1995.
- [Williams 77] Ederyn Williams
Experimental Comparisons of face-to-face and mediated communication
Psychological Bulletin, 84(5), p. 963-976, 1977.
- [Woo 98] Mason Woo, Jackie Neider, Tom Davis
OpenGL Programming Guide, the official guide to learning OpenGL, version 1.1
Addison-Wesley, deuxième édition, 1998.
- [Worlds Chat] Worlds Inc.
<http://www.worlds.net/>
une application de discussion dans un monde 3D.
- [Yoshimura 94] Tetsuya Yoshimura, Yasuhiro Nakamura, Masataka Sugiura
3D Direct Manipulation Interface : Developement of the Zashiki-Warashi System
Computer and Graphics, 18(2), p. 201-207, 1994.
- [Zhai 94] Shumin Zhai, William Buxton et Paul Milgram
The Silk Cursor: Investigating Transparency for 3D Acquisition
Actes de la conférence CHI, ACM Press, p. 459-464, 1994.
- [Zhai 96] Shumin Zhai, William Buxton et Paul Milgram
The Partial-Occlusion Effect: Utilizing Semitransparency in 3D Human-Computer Interaction
ACM Transactions on Computer Human Interaction, 3(3), p. 254-284, 1996.
- [Zhai 98] Shumin Zhai
User Performance in Relation to 3D Input Device Design
ACM Computer Graphics, 32(4), p. 50-54, novembre 1998.



Nous présentons ici les éléments de l'architecture interne de SpIn. Ces développements ont été effectués dans l'objectif de répondre aux critères du modèle d'interaction de SpIn. Même si nous avons pu observer le fonctionnement de systèmes d'exploitations interactifs, nous n'avons pas repositionné notre travail dans un contexte plus large. Nous ne nous sommes pas positionné non plus par rapport aux modèles classiques d'architecture d'interface homme-machine, tant au niveau conceptuel qu'au niveau implémentation. Pour ces raisons, ces explications sont situées en annexe. L'étude de l'architecture logicielle de la prochaine version de SpIn s'inscrit dans une étude approfondie du domaine, et justifiera son approche par rapport aux systèmes existants.

La conception de l'architecture de SpIn a été réalisée une fois le fonctionnement global de l'interface défini. Les spécifications du modèle d'interaction (voir en particulier le chapitre 4) ont permis d'exprimer les différents besoins.

La partie visuelle de SpIn est composée des éléments suivants :

- un espace de travail tridimensionnel composé du bandeau et de la table de réunion
- la représentation de chaque document dans un espace délimité
- un pointeur 3D qui permet de désigner et de sélectionner les documents ou une de leurs parties

La structure de ces éléments est basée sur la définition de la représentation interne des documents (paragraphe 1) et de l'interaction (paragraphe 2). Nous décrivons ensuite la gestion des entrées/sorties (paragraphe 3) et les informations qui sont interprétées dans SpIn (paragraphe 4).

Les diagrammes de classes sont notés en UML par la suite :

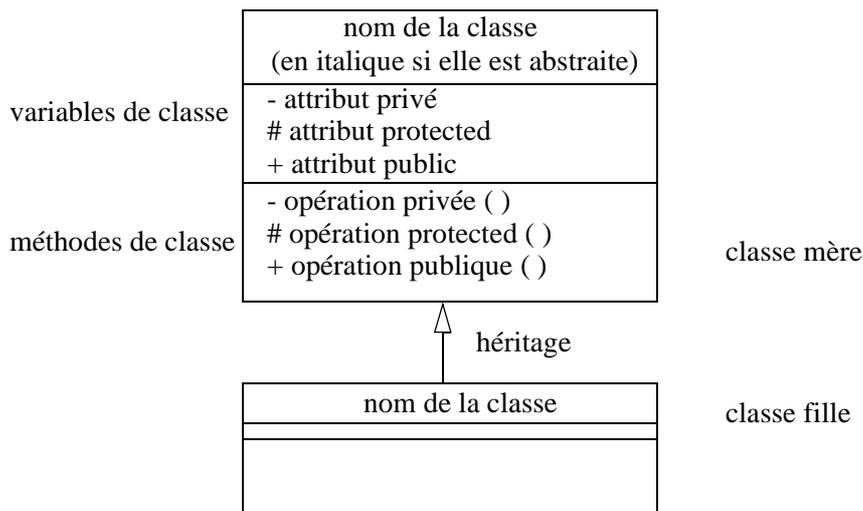


Figure 1.1 *rappel sur la notation UML*

1.1 Structure de données

L'ensemble de l'espace de travail est tridimensionnel. Les documents, quelque soit le nombre de leurs dimensions, y sont représentés en trois dimensions. Afin de maintenir la cohérence et la généralité des mécanismes d'interaction, on utilise la même représentation pour tous les documents. Il a donc fallu choisir une structure de données commune pour les placer en mémoire.

Il existe un certain nombre de formats d'objets 3D, liés à des formats de fichier. Beaucoup sont propriétaires et liés à des applications particulières, surtout des modeleurs (3D Studio Max d'Autodesk, Maya d'Alias-Wavefront, etc). Les formats publiques, comme NFF (Neutral File Format), se contente de décrire des objets géométriques sous forme de facettes, à plat, et sans transformation géométrique. D'autres sont des standards normalisés, comme VRML.

1.1.1 Le format VRML

VRML 1.0 propose un format de description d'objets 3D complet pour les paramètres géométriques avec des quadriques, les objets facettisés et leurs matériaux [Bell 96]. Il permet en outre d'inclure des transformations géométriques dans la description de l'objet.

Ce format se compose d'un graphe, qui permet de regrouper les objets par sous-ensembles de l'espace. Les propriétés (matériaux, transformations géométriques) se propagent dans le graphe de haut en bas et de gauche à droite. Elle peut également permettre de regrouper les objets en sous-graphes sur d'autres critères que la géométrie (à la discrétion du concepteur du fichier). Le concepteur d'une molécule complexe pourra par exemple regrouper les atomes par type (oxygène, hydrogène, etc) si il le juge utile.

Les différents noeuds qui composent le graphe peuvent être nommés (on place alors un *DEF nomDuNoeud* devant le noeud en question).

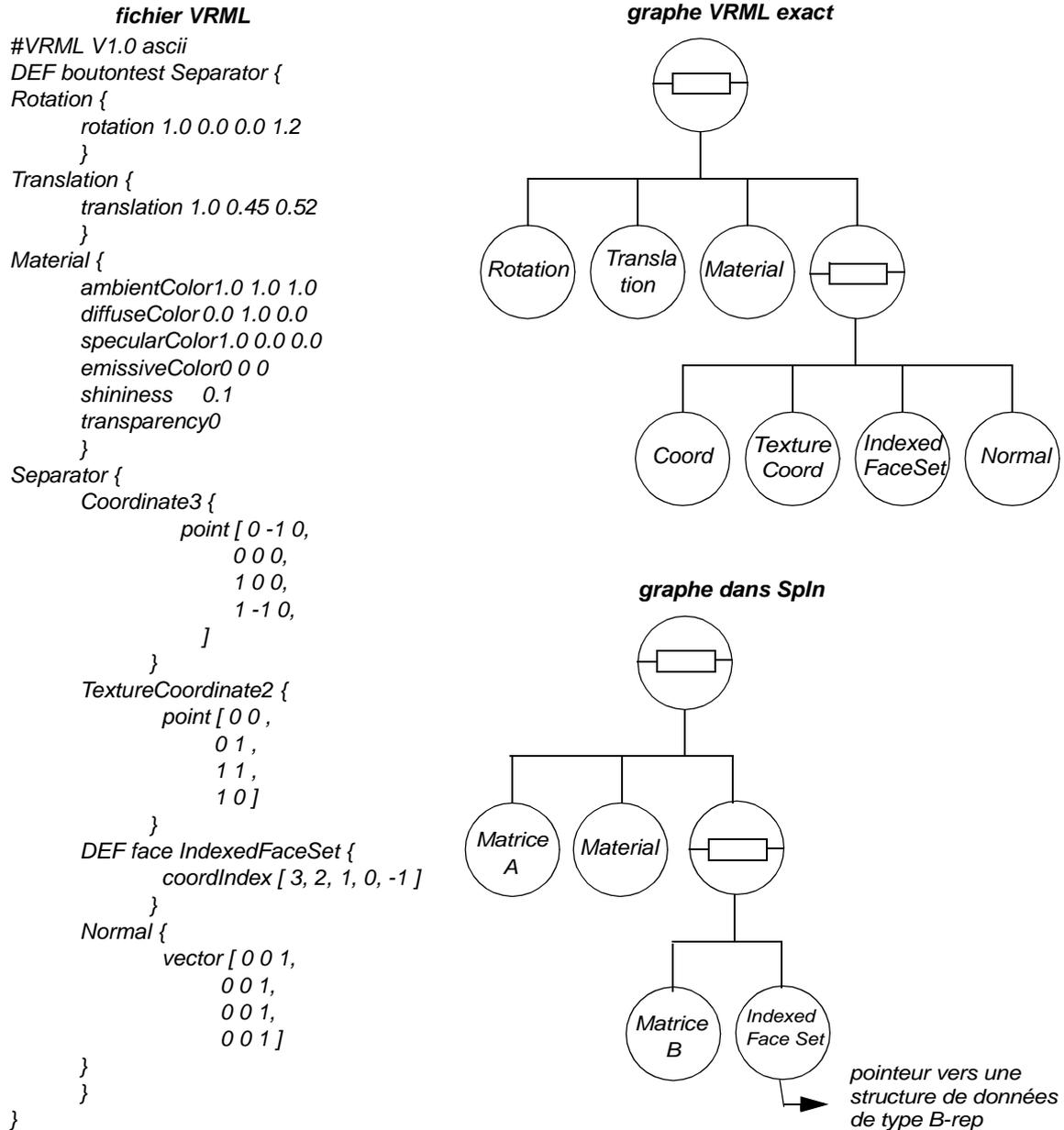
Le format VRML 1.0 a l'avantage d'être complet, standard, et de proposer une structure de graphe. Il est suffisant en ce qui concerne SpIn à l'heure actuelle puisqu'il est utilisé comme format de base de document, les aspects interactifs sont gérés séparément comme nous le verrons plus loin. Le format VRML 97 apporte en plus une certaine forme d'interaction, mais principalement adaptée à l'animation et à la navigation dans une scène 3D, et n'est pas suffisante pour interagir dans une scène. Il pourrait cependant être intéressant de suivre ce format à l'avenir, les descriptions d'objets s'améliorant, en particulier en ce qui concerne la géométrie. Son adaptation permettrait également à SpIn de suivre l'évolution des standards.

1.1.2 Le format VRML dans SpIn

Nous avons donc adopté le format VRML 1.0, en conservant sa structure de graphe qui permet un regroupement de sous-partie du document. Cependant, après chargement du graphe, la géométrie est optimisée afin de réduire la taille du graphe et de faciliter son traitement ultérieur :

- on conserve un seul noeud, *IndexedFaceSet*, pour les objets facettisés (on ne conserve pas de noeud pour les normales, les coordonnées de points et de texture comme dans VRML 1.0), tout est regroupé dans une seule structure de type B-rep. Il n'y a en effet aucun intérêt à conserver plusieurs noeuds pour un seul objet géométrique (les concepteurs de VRML 2 l'ont compris).
- les transformations géométriques consécutives dans un même niveau de graphe sont concaténées en une seule matrice de transformation.
- les objets géométriques sont recentrés en zéro afin de faciliter les manipulations ultérieures sur l'objet (une translation est insérée devant l'objet). Cela permet en particulier de réaliser des rotations d'un objet sur lui-même.
- le document complet subit une mise à l'échelle pour avoir la bonne dimension dans SpIn.
- les boîtes englobantes de chaque objet géométrique et de chaque arbre (noeud *Separator* en VRML) sont calculées.

On peut voir Figure 1.2 un exemple de fichier VRML 1.0 affichant une facette à quatre points. On peut voir sur la droite du fichier une représentation du graphe tel qu'il est dans le fichier VRML 1.0, puis en dessous, tel qu'il est interprété et utilisé par SpIn après le traitement post-chargement du fichier. La matrice A est la matrice concaténant les matrices de transformations présentes (ici une rotation et une translation) dans le graphe et la matrice de calibration du document complet. La matrice B est la matrice qui permet de positionner la facette à sa place d'origine, les coordonnées de cette dernière ayant été recentrées en (0,0,0) pour des manipulations ultérieures.



1.1.3 Structure objet

Ce sont les objets qui composent le graphe VRML, il s'agit de tous les noeuds de la norme VRML, les noeuds qui gèrent la structure d'arbre (Separator, Switch, etc), les noeuds qui gèrent les transformations géométriques (Translation, Scale, etc), les noeuds qui gèrent les objets graphiques (IndexedFaceSet, Sphere, Bitmap, Texte, etc), etc.

Ces objets de la structure de données permettent de composer les documents. On y a ajouté deux éléments particuliers pour simplifier la gestion de l'interaction. Une classe *SwitchPointeur3D* qui hérite de la classe VRML *Switch*. Et une classe *SeparatorClone* qui hérite de la classe VRML *Separator*, pour gérer les clones, leur affichage, et les messages particuliers qu'ils reçoivent et émettent. Ces ajouts correspondent surtout à une adaptation de code importé dans SpIn, comme celui de l'INA. Pour éviter leur portage sous la forme d'actions et de données séparées, on a créé des noeuds spéciaux qui ont permis une adaptation plus rapide.

La Figure 1.3 propose une vue de la structure objet du graphe des données. La classe *MessageHandler* permet la gestion des messages pour tout le graphe; les méthodes *Idle()* et *TraiterMessage()* n'ont pas besoin d'être redéfinies ailleurs. Chaque nouvel objet n'a besoin que de la méthode *Afficher()*. La méthode *EnregistrerAssociationActionEvenement()* de *MessageHandler* permet de constituer la liste d'associations décrite dans le paragraphe suivant.

La classe *NoeudVisible* contient tous les noeuds ayant une réalité dans l'espace 3D. Elle permet la gestion de toutes les primitives de détection de collision (les boîtes englobantes hiérarchiques et les inclusions). La classe *NoeudGenerique* est la classe mère de tous les objets non-visibles en VRML, comme les lampes ou les définitions de matériaux.

Le graphe du document est composé de noeuds arbres et de noeuds feuilles. Il faut des primitives pour accéder au graphe, l'augmenter ou le réduire, se déplacer à l'intérieur, chercher des noeuds particuliers suivant un type ou un nom. La classe *Arbre* contient donc les primitives de gestion des arbres du graphe, tandis que la classe *ObjetGenerique* contient les primitives de gestion des feuilles du graphe.

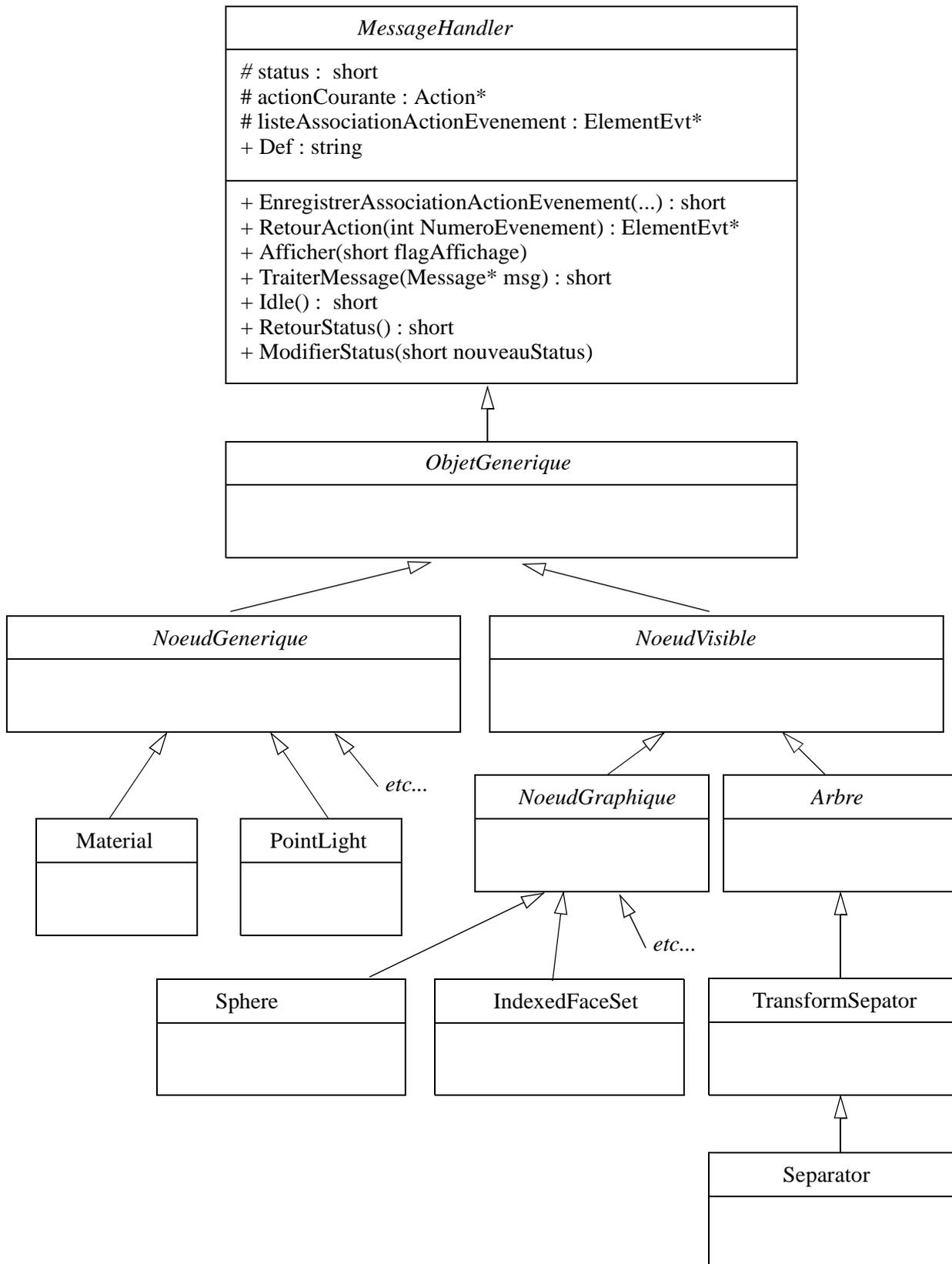


Figure 1.3 **graphe de classe** simplifié de la structure de données des documents (notation UML)

1.2 Gestion de l'interaction

SpIn est une interface multi-utilisateurs, on doit donc gérer l'interaction de l'utilisateur local, mais également de ses interlocuteurs. Il faut donc un système conçu pour réagir à plusieurs sollicitations simultanément, sur différents documents (ou sur le même). Notre environnement est centré sur les documents au niveau de l'utilisateur. Nous conservons logiquement la même approche dans le fonctionnement interne de l'interface que dans l'aspect visuel : un noyau se charge de l'affichage des documents dans le bandeau ou sur la table de réunion. Il faut ensuite rendre ces documents réactifs.

Dans la mesure où l'on utilise l'interaction directe, nous avons utilisé une interface pilotée par évènements, ie par messages [Olsen 98]. L'utilisation systématique des messages n'est pas obligatoire, on peut par exemple utiliser aussi des mécanismes d'objets distribués (comme Corba) pour gérer la répartition des documents partagés. Nous avons préféré un système homogène qui n'utilise qu'un seul mode de propagation des informations, les évènements.

1.2.1 Dynamique des documents

Comme nous l'avons vu, la notion d'application au sens des interfaces 2D n'existe pas explicitement dans SpIn. L'architecture de l'interface est centrée sur les documents : nous avons un graphe qui représente l'ensemble des données des documents présents dans la réunion.

Pour que l'on puisse agir dessus, il faut tout d'abord détecter la présence du pointeur dans un document. Nous utilisons des boîtes englobantes hiérarchiques (Figure 1.4). La présence du pointeur dans la boîte la plus profonde (dans l'arbre du graphe) permet à l'objet en question de commencer à recevoir tous les messages de SpIn, en particulier ceux du pointeur.

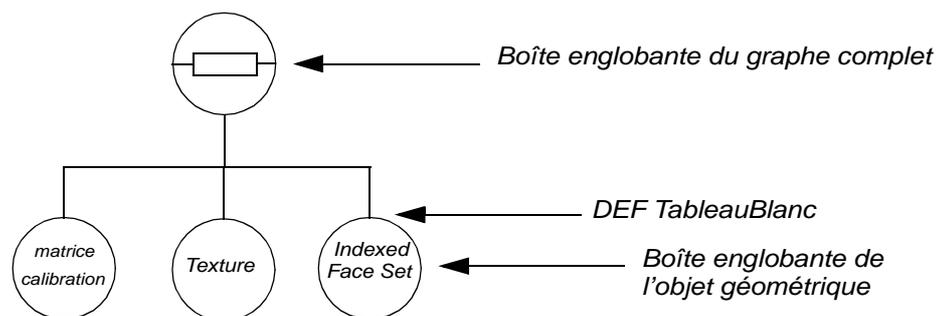


Figure 1.4 exemple de document dans SpIn

De plus, seuls les noeuds du graphe qui sont nommés reçoivent des messages (donc ceux qui possèdent le *DEF* du langage VRML). Auquel cas, il peut être sélectionné.

1.2.2 Evènements

Il existe trois types de messages dans SpIn qu'un objet peut recevoir :

- un message Global, qui va être envoyé à tous les objets de SpIn,
- un message direct, qui contient le nom (*DEF*) d'un noeud particulier et qui est donc redirigé

directement vers ce noeud,

- un message 'classique' qui va être dirigé vers l'objet sélectionné si il en existe un, sinon vers l'objet courant (l'objet pointé).

Pile de messages

SpIn possède une pile de message (Figure 1.5), provenant des périphériques (de l'utilisateur), du réseau (des autres utilisateurs) ou de la communication interne entre objets. Les messages sont empilés, et donc traités dans leur ordre d'arrivée. Leur traitement est séquentiel afin d'éviter d'éventuels problèmes de synchronisation.

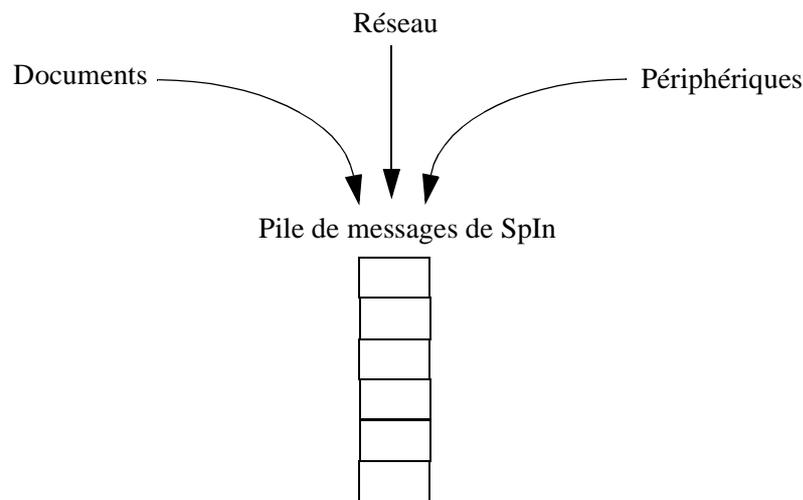


Figure 1.5 *pile de messages de SpIn*

SpIn gère donc ses propres messages, la manière de les répartir, la fonction `Idle()` lorsqu'il n'y a pas de message, mais également l'affichage et la gestion du réseau. L'interface reprend et modifie donc tous les fonctionnements du système d'exploitation sur lequel il s'exécute. On comprend donc mieux les principes de SpIn en l'imaginant tel qu'il a été conçu, comme une interface de travail tridimensionnelle autonome.

Il existe une autre pile de message dans SpIn, au niveau des envois sur le réseau. Dès qu'un message a besoin d'être envoyé sur le réseau, il est empilé dans une file d'attente spéciale. Un *Thread* qui scrute cette file se charge de construire et d'envoyer les messages vers les autres interfaces. Un autre *Thread* en attente sur les ports de communication gère l'envoi des messages. Lorsqu'un nouveau message arrive par le réseau, il est envoyé directement sur la pile de messages de SpIn.

Structure objet

La structure d'un message dans SpIn est simple, un type et des données. La classe générique des messages est dérivée en de nombreuses classes qui facilitent la manipulation des informations contenus dans les types de messages connus.

On montre Figure 1.6 la structure de la classe abstraite qui donne son interface à tous les messages. `MessageEnvoiTelepointeur` est un exemple typique de message de SpIn, identifié par son type et ne contenant pas de donnée propre. `MessagePériphérique` au contraire contient des données

(stockée dans le champ `dataMessage` de la classe `Message`), et pose l'interface des messages venant des périphériques.

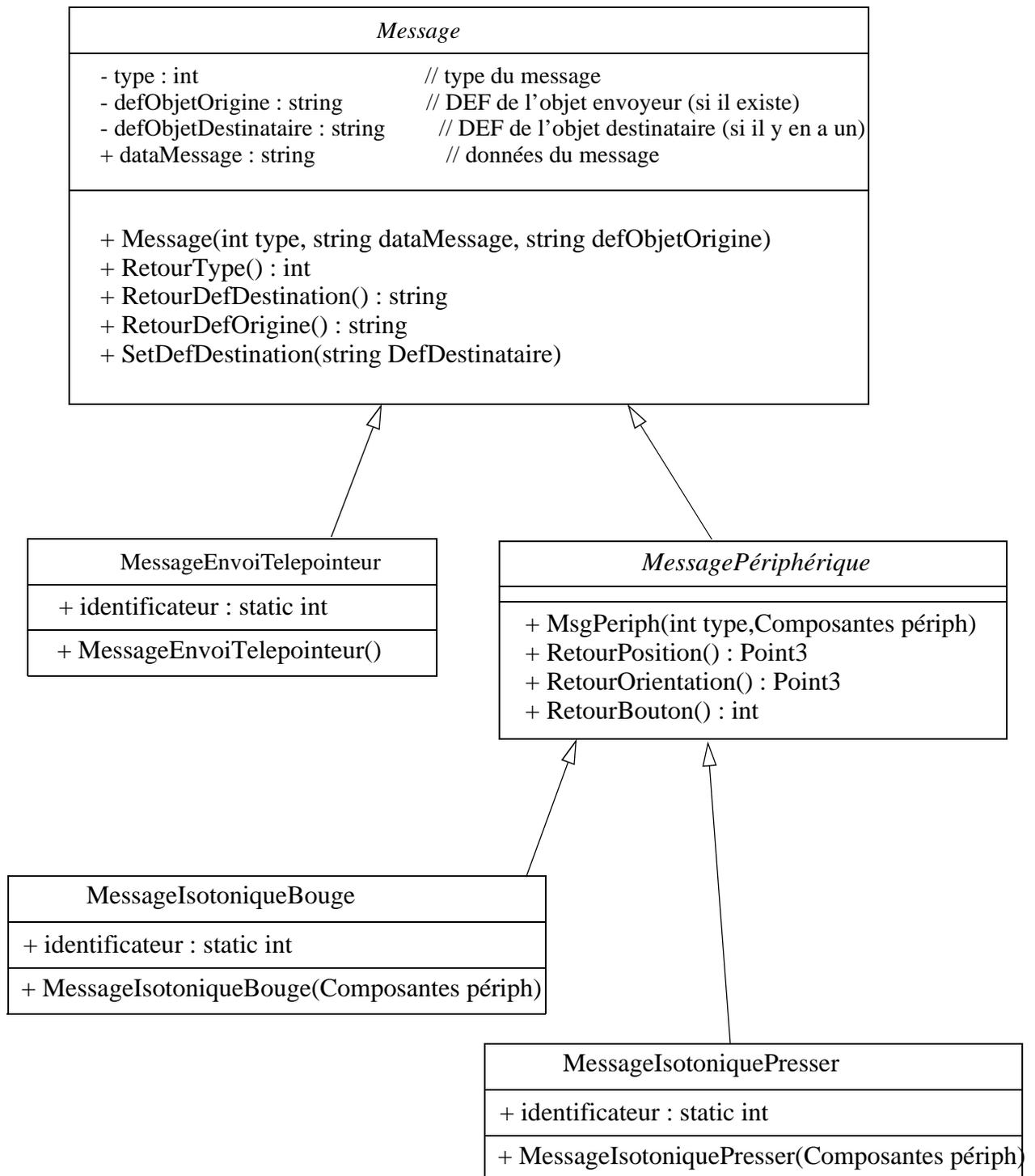


Figure 1.6 description des principaux messages de Spln (format UML)

La classe *Composantes* est une structure de données contenant toutes les informations propres au périphérique. Les classes *MessageIsotoniquePresser*, *MessageIsotoniqueRelacher*, *MessageIsométriqueBouge*, etc... sont dérivées de cette classe abstraite.

Chaque classe est identifiée par une valeur statique *id* (conservée pour chaque nouvelle instance de la classe) et initialisée à une valeur unique. Ainsi, on peut tester la réception d'un message *MessageIsotoniquePresser* grâce à l'identificateur *MessageIsotoniquePresser:id*. Cet identificateur est passé obligatoirement (et automatiquement) en paramètre au constructeur d'un message, afin de remplir le champ *type* de la classe *Message*. Ce qui permet à n'importe quel objet d'accéder au type du message.

Ainsi, la structure générique des messages permet à n'importe quel objet de SpIn de les identifier et de les manipuler.

1.2.3 Actions

Une fois qu'un objet du graphe des documents a reçu un message, il faut le traiter. A cette fin, nous avons créé un mécanisme générique de déclenchement d'une action lors de la réception d'un message.

On définit une action comme une entité intelligente agissant sur un document. Pour un document particulier, on associe donc à un message donné, une action donnée, qui se déclenche lorsque l'objet en question reçoit ce message. L'action reçoit alors, à travers l'objet sélectionné, tous les messages de la pile. C'est elle-même qui décide de se terminer. On utilise à cette fin le statut que possède chaque objet pour que chaque action informe le noyau de SpIn sur l'état de l'objet.

Une phase d'action dans SpIn

Une phase d'action de SpIn se déroule ainsi :

- l'utilisateur pointe un document ou une de ses parties, son statut prend alors la valeur *COURANT*. A ce niveau, on voit des indices comme les boîtes englobantes progressives s'afficher.
- l'utilisateur clique pour le sélectionner. Si l'objet en question possède un *DEF*, son statut passe alors à *SELECTIONNER*. L'objet change alors de couleur.
- l'objet reçoit alors les messages de la pile de SpIn. Si un de ces messages est présent dans la liste d'évènements-actions de l'objet, l'action associée à l'évènement est déclenchée et recevra désormais automatiquement tous les messages. Le statut de l'objet prend la valeur de l'identificateur de l'action (*ACTION_ROTATION*, etc). L'objet est alors géré directement par l'action, qui décidera elle-même de s'arrêter en faisant passer le statut de l'objet à 0. SpIn détruit alors l'action en question, qui est terminée.

Ce schéma de représentation, séparant les données de ce qui peut s'y appliquer permet de découper dans SpIn les documents et leur comportement. On peut voir Figure 1.7 un noeud réactif du graphe d'un document (nommé *partiedroite*) et la liste d'associations évènement-action qui lui est associée.

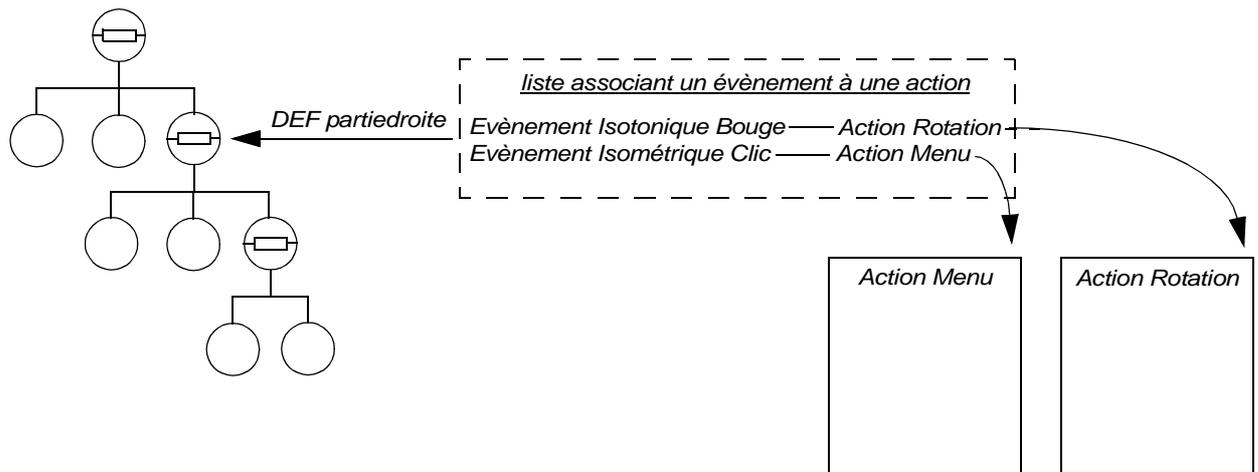


Figure 1.7 gestion des Actions dans SpIn

Une action peut donc servir à plusieurs documents. Elle peut également avoir différents types de réaction en fonction des objets sur lesquels elle s'applique. Au niveau de l'implémentation, l'idéal est d'écrire ces actions sous forme de scripts ou de code téléchargeable dans SpIn (ce qui sera le cas dans la prochaine version). Les actions ont jusqu'ici été programmées 'en dur' dans SpIn.

Action persistante

Afin d'améliorer les possibilités des actions dans SpIn, une action peut continuer à tourner en arrière plan (et donc continuer à recevoir les messages). C'est ce qui permet de réaliser, par exemple, des moteurs d'animation. Une action possède donc quatre méthodes qui lui permettent d'agir :

- une méthode Agir() qui reçoit les messages quand elle s'applique normalement à un document
- une méthode AgirIdle() qui permet de continuer à exécuter l'action lorsque la pile de messages est vide
- et deux méthodes AgirArrière() et AgirArrièreIdle() qui lui permettent de continuer à s'exécuter dans les mêmes conditions que les deux méthodes précédentes, une fois que l'action a indiqué au système qu'elle continuait à tourner en arrière plan.

Description du comportement

La liste des associations action-événement est contenue dans un fichier séparé du fichier VRML, portant le même nom, mais avec une extension .DEF (Figure 1.8). Il permet d'associer à chaque partie du document, ou au document entier, un comportement.

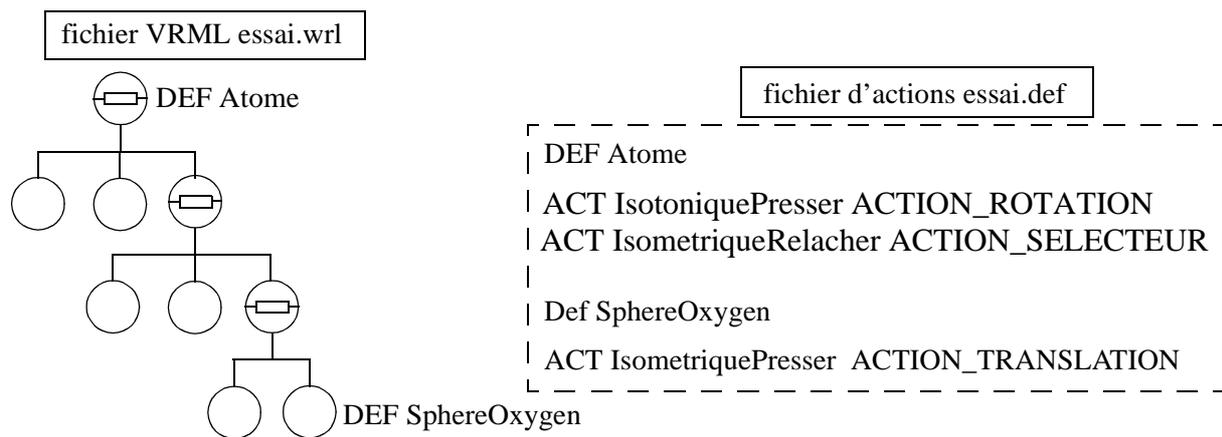


Figure 1.8 un document 3D au format VRML (essai.wrl) et son fichier de comportements associé (essai.def)

Dans l'exemple de la Figure 1.8, on peut voir des associations événement-action précédées de ACT. Il existe plusieurs suffixes possibles en fonction du type de gestion de message désiré :

ACT messageA ActionA

Lorsque l'objet reçoit le *messageA*, l'action *ActionA* est initialisée.

OBJ messageA DefA

Lorsque l'objet reçoit le *messageA*, il est automatiquement renvoyé vers l'objet *DefA*. Aucune action n'est initialisée à ce niveau. Cette fonction sert à effectuer des renvois automatique d'objets en objets. Ce type de règles sert par exemple à faire des boutons.

AEO messageA ActionA DefA

Lorsque l'objet reçoit le *messageA*, l'action *ActionA* est exécutée sur l'objet *DefA*. Cette fonction permet de modifier un objet à partir d'un autre. Cela permet d'exécuter dynamiquement des actions sur des parties de documents passives.

ACS ActionA

Cette fonction permet d'associer l'action *ActionA* à un l'objet, sans qu'il soit réactif à un événement particulier. Cela permet d'associer beaucoup d'actions à un document, en particulier pour y accéder à travers les menus.

Structure objet

Le fonctionnement des actions est assez simple vu de l'extérieur de SpIn. Leur initialisation se fait automatiquement, la gestion de la fin de l'action également. Pour développer une nouvelle action, on a juste besoin d'écrire sa méthode *Agir()* de gestion des messages.

SpIn comprend un certain nombre d'actions élémentaires qui limitent le développement de nouvelles actions, en particulier pour les transformations géométriques comme les rotations autour d'un ou plusieurs axes; mais également la gestion des boutons ou des menus.

On peut voir Figure 1.9 la description de la classe virtuelle abstraite *Action*. Les méthodes *Agir()*, *Idle()* et *Afficher()* sont appelées en même temps que celles du document sur lequel l'action s'applique.

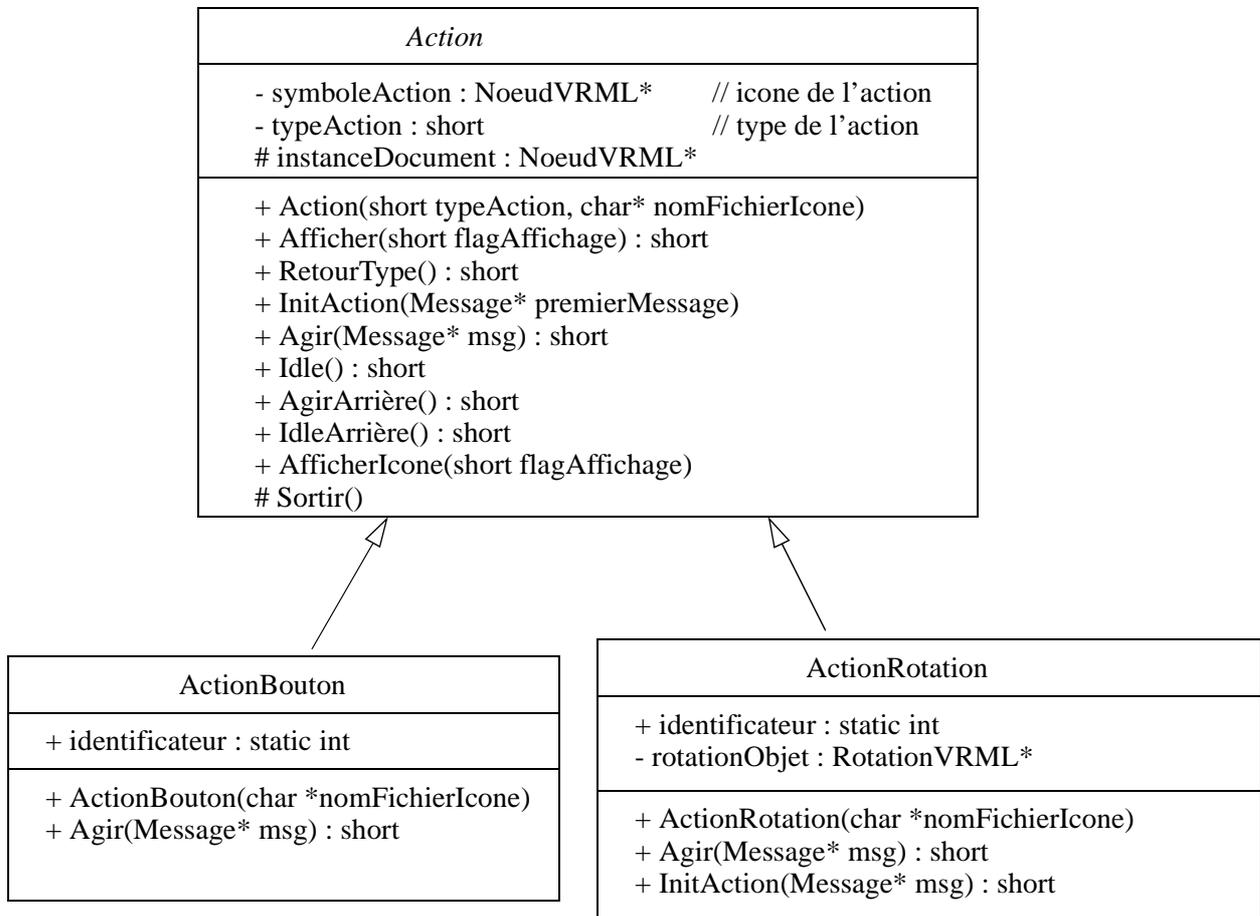


Figure 1.9 *structure objet des Actions avec deux classes en exemple (notation UML)*

Une nouvelle action peut implémenter au minimum une méthode *Agir()*. La méthode *InitAction()* sert à une initialisation de données éventuelle avant qu'*Agir()* soit appelée pour la première fois. Lorsque l'action se termine, il suffit qu'elle appelle la fonction *Sortir()*.

Chaque action est identifiée par un type particulier, enregistré dans un identificateur statique *id*, et accessible à travers le champ *type* de la classe *Action*, comme pour les messages.

1.2.4 La barre d'outils

La barre d'outils est un élément interne de *SpIn*, au même titre que le bandeau ou la table. On y sélectionne un icône qui correspond à un outil particulier. Une action particulière est associée à l'icône de l'outil.

Une fois l'icône sélectionné, cette action transforme alors le pointeur 3D en un autre objet 3D, représentant l'outil. On peut alors aller cliquer sur un document ou un acteur, l'action lancée exécutera la fonction de l'outil sur l'objet sélectionné. A ce moment, le pointeur reprend sa forme initiale.

Pour gérer plusieurs représentations pour le pointeur 3D, on utilise un arbre de type Switch, un noeud VRML qui n'affiche qu'un seul de ses fils à la fois. On peut remarquer que la gestion des outils, comme pour les menus ou d'autres éléments interactifs de SpIn, n'entraîne aucune modification dans la structure de SpIn. Tout est créé avec la structure de graphe VRML et les actions.

1.2.5 Bilan

SpIn est donc centré sur des documents, composés de graphe d'objets graphiques de bases. La connaissance de leurs propriétés géométriques permet d'y détecter la présence du pointeur. Un mécanisme de message permet alors de déclencher une action sur l'objet pointé (Figure 1.10).

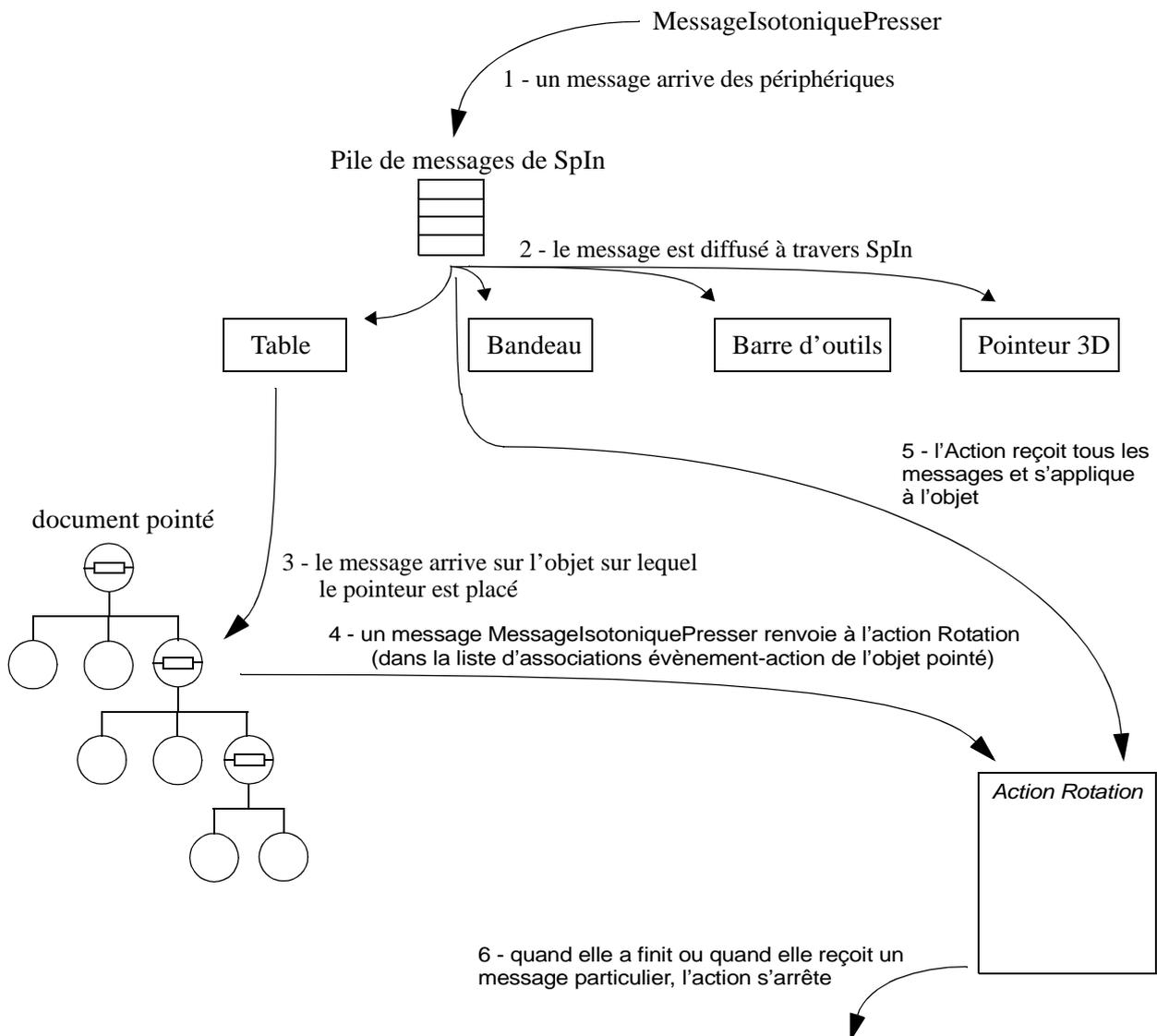


Figure 1.10 *fonctionnement des messages dans SpIn*

Ce mécanisme permet de réaliser ce que l'on veut sur un document donné, sans limitation jusqu'ici. Cette structure permet de composer des objets VRML dont toutes les parties sont réactives, comme

par exemple un menu 3D, où chaque icône est un objet 3D du graphe du menu, qui peut réagir à plusieurs sortes d'évènements.

1.3 Entrée/Sortie

Nous allons décrire maintenant les techniques d'entrée/sortie, en particulier celles qui nous ont amenées à développer des solutions spécifiques, faute de bibliothèque ou de fonction système adaptées.

1.3.1 Graphique

La bibliothèque OpenGL permet de gérer l'affichage bas-niveau en trois dimensions. Lorsque l'on affiche l'espace de travail, on parcourt les différents graphes des documents pour les afficher. On réalise ce parcours à partir des deux éléments de l'espace de travail : la table de réunion, qui peut gérer un document, et le bandeau, qui contient le reste des documents.

Les actions retournent des valeurs qui conditionnent ou non un nouvel affichage. Les différents noeuds VRML étant représenté dans le graphe par des objets, ils possèdent chacun leur propre méthode d'affichage.

Les systèmes d'exploitation actuels, et les bibliothèques de développement qui les accompagnent, sont centrés sur les interfaces de type WIMP. Dans le cas de la synthèse d'image temps réel, telle qu'elle existe dans nos interfaces 3D, on redessine tous les objets à chaque image, particulièrement à cause des phénomènes d'occlusion et de lumière. Les mécanismes permettant de ne pas recalculer la totalité de l'image sont souvent plus coûteux que le calcul complet de l'image lorsque l'on possède des cartes graphiques puissantes. Au niveau des calculs géométriques, l'utilisation de *Display Lists* [Woo 98] permet d'éviter de refaire les calculs des positions des objets qui ne bougent pas d'une image à l'autre. Cela constitue techniquement une différence assez importante par rapport aux systèmes fenêtrés.

1.3.2 Son

Le son entre les participants est diffusé grâce à une adaptation des outils de MBone à SpIn. MBone, pour Multicast BackBone, est le réseau virtuel de «diffusion multipoint» s'appuyant sur Internet. Mbone permet de diffuser de la vidéo, de l'audio et d'autres formats vers des groupes.

On peut également jouer des sons systèmes au format PCM (Pulse Code Modulation) pour marquer des évènements particuliers dans l'interface. Ces sons sont mixés avec le canal audio utilisé par MBone : la voix provient du réseau sous la forme de paquets, elle est mixée à son arrivée avec les sons systèmes puis le tout est placé dans les buffers de la carte son (les paquets de la voix sont toujours émis, même si la voix est nulle, c'est à dire si la personne ne parle pas).

Les sons systèmes peuvent être associés à des évènements particuliers de SpIn ou à des lancements d'Actions.

1.3.3 Périphériques

Les périphériques qui sortent du commun ne sont pas gérés directement par les systèmes d'exploitation actuels. Et lorsqu'ils le sont, on ne maîtrise pas leur taux d'échantillonnage. Ce dernier peut-être élevé dans les interfaces 2D, car on ne redessine que partiellement l'écran. Les techniques de dessins en deux dimensions sont très optimisées dans ce sens. Le pointeur de souris est souvent affiché dans des plans d'*overlay* (quand la carte graphique de la machine en possède), ce qui permet une gestion de l'affichage du pointeur spécifique. Or nous avons vu ci-dessus que tous les objets sont redessinés à chaque image dans les scènes 3D.

Le déplacement du pointeur avec un capteur 3D nécessitant un ré-affichage à chaque nouvelle position du pointeur, il est utile de maîtriser la vitesse d'envoi des messages de ce dernier si l'on ne veut pas insérer des mécanismes de régulation. Ainsi, par faute d'existence ou d'adaptation des pilotes de périphériques 3D, on développe souvent des solutions de lecture spécifiques.

Cette solution a l'avantage, si elle prend plus de temps, de donner une maîtrise complète de l'envoi de messages, sans passer par le système d'exploitation de la machine et sans un mécanisme coûteux de filtrage supplémentaire. On gère donc les différents périphériques en scrutation (*polling*) sur les ports de l'ordinateur. Ce qui nous permet d'utiliser une fréquence de lecture en relation avec la fréquence de rafraîchissement maximale de l'affichage. Cela évite de générer des images ou des lectures inutiles, et permet de rendre le processus dynamique afin de s'adapter au mieux aux performances de la machine.

La lecture des périphériques consistant principalement à attendre le nouveau moment où l'on a besoin de ses coordonnées, on utilise des *Threads* qui scrutent les ports où ils sont connectés. Une nouvelle valeur est lue à interval fixe. Si elle a changé par rapport à la précédente, un message est envoyé dans la pile de messages de SpIn.

SpIn utilise une classe abstraite indépendante du type de périphérique pour gérer l'interaction et envoyer des messages. Il s'agit de la classe *Peripherique* (Figure 1.11). Le seuil de détection permet de ne prendre en compte les valeurs lues qu'à partir d'un certain niveau. Les paramètres du périphérique sont donnés par position, orientation, bouton (PRESSER, RELACHER, DRAG, RIEN), la fréquence de lecture (*tauxDeRaffraichissement*) et le facteur de correction sur les positions suivant les trois axes. *EnvoyerMessage()* permet d'empiler un message du périphérique dans SpIn si les paramètres du périphérique ont changés depuis la dernière lecture (fonction *PositionDifferente()*).

Seules les fonctions *Flush()* et *Lire()* ont besoin d'être développées pour chaque périphérique, ainsi que le constructeur qui permet l'initialisation du périphérique.

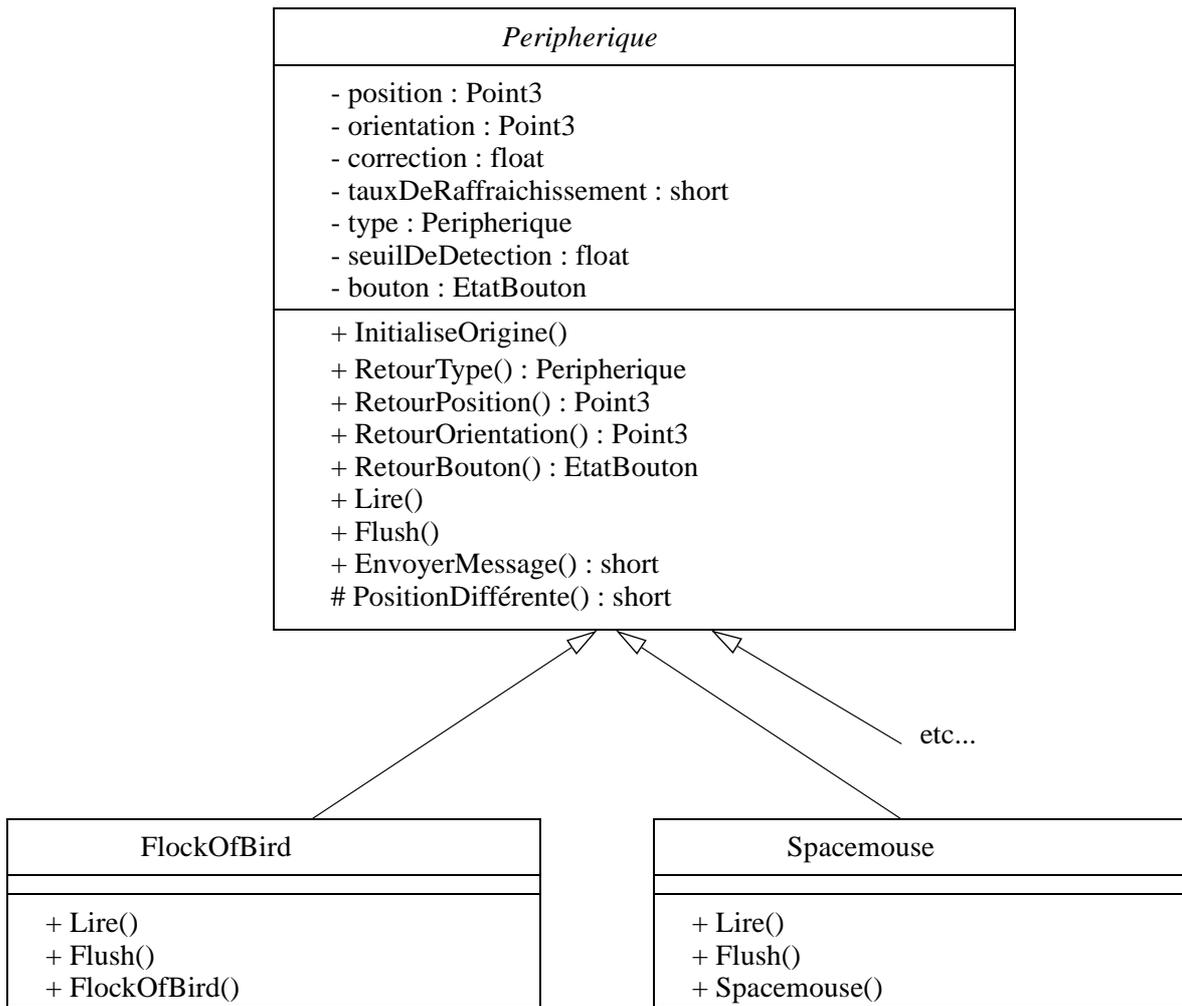


Figure 1.11 structure objet des périphériques

1.3.4 Communication réseau

SpIn utilise la diffusion Multicast de TCP/IP pour envoyer des messages entre les interfaces. Pour communiquer à travers Internet en dehors de MBone, on utilise un tunnel Unicast qui relie deux machines Unix, chacune présente sur un groupe de machines distantes.

L'aspect partagé peut être réalisé de différentes façons. Par exemple, chaque action qui modifie une donnée peut envoyer un message particulier aux autres interfaces pour réaliser la même modification à distance. Cela impose de créer des nouveaux messages pour chaque nouvelle action. On peut également utiliser des mécanismes d'objets partagés, séparés du noyau de SpIn, qui permettent de reporter automatiquement à distances les modifications.

Nous avons choisi le mécanisme le plus générique possible en choisissant d'envoyer directement à distance les mêmes messages qu'en local (en ciblant le message sur l'objet auquel il s'applique localement). Ainsi, les messages à destination d'un document partagé sont automatiquement renvoyés à distance en identifiant l'objet destinataire à l'intérieur du message (on précise le DEF de

l'objet à l'intérieur du message). A leur arrivée, ces messages sont directement renvoyés vers le document ciblé.

Cette méthode permet de gérer de façon transparente les mécanismes de partage de documents. Il n'y a rien de spécifique à réaliser dans une action lorsqu'elle s'applique à un document partagé, les interfaces distantes reproduisant exactement la même chose que ce qui se passe en local. Ce mécanisme utilise la diffusion Multicast de SpIn, il est donc sensible à la perte de paquets.

Un mécanisme de verrouillage d'objets empêche une action de se lancer sur un document ou l'une de ses parties lorsque celui-ci est en cours d'utilisation. Lorsqu'un utilisateur veut accéder à un document partagé, SpIn diffuse une demande de verrou à tous les participants. Si une autre instance de SpIn possédait déjà un verrou sur ce document, elle le cède et le nouvel utilisateur prend le verrou. Si personne ne le possédait, SpIn attend que chaque utilisateur ait répondu qu'il ne le possède pas. Ce système impose de connaître tout le monde. Ce qui ne pose pas un problème dans SpIn où le système gère tous les participants connectés. Il faut cependant un gestionnaire qui gère les pannes, comme la déconnexion involontaire d'un utilisateur.

1.4 Informations interprétées

Un certain nombre d'informations sont interprétées pour être envoyées à distance. Elles sont alors traitées sur les machines distantes pour délivrer une information à l'utilisateur. On trouve en particulier la gestion du télépointeur et des clones.

1.4.1 Télépointeur

Le télépointeur d'un clone suit le mouvement de son pointeur local. Pour cela, on calcule dans SpIn en permanence la distance du pointeur au document pointé le plus proche. Si la distance est inférieure à un seuil, et que le document est partagé, on envoie les coordonnées du pointeur à distance sous la forme :

- l'identificateur du document désigné (chaque document a un identificateur unique),
- coordonnées du pointeur par rapport au centre du document, en corrigeant la position en fonction du point de vue de l'utilisateur.

A distance, on recalcule les coordonnées du télépointeur dans le repère global de la scène à partir des deux éléments ci-dessus, de la position du document désigné dans l'interface locale, et de son orientation.

1.4.2 Le regard des clones

Pour diriger le regard du clone, nous utilisons l'information de focus (voir chapitre 6). Si l'utilisateur a pour focus le document au centre du bandeau, ou le document placé sur la table, on envoie l'identificateur de ce dernier aux autres instances de SpIn. Dans les autres cas, comme lorsque l'utilisateur regarde un document privé, on envoie une valeur nulle.

A la réception, SpIn identifie l'origine du message, retrouve le clone de l'utilisateur qui l'a envoyé, et modifie la direction de son regard. Pour cela, on calcule l'angle entre la position du clone et le document qu'il regarde dans l'interface. On divise alors la valeur de l'angle en deux pour faire tourner la tête et les yeux de la moitié de la valeur.

Lorsque SpIn reçoit une valeur de focus nulle, le regard du clone concerné est dirigé vers la table.

Annexe

Éléments de l'évaluation



Titre : Un modèle d'interaction 3D : Interaction Homme-Machine et Homme-Homme dans les interfaces 3D pour le TCAO Sychrone.

Le projet SpIn définit un modèle d'interface tridimensionnelle pour le travail coopératif synchrone. Il s'agit de recréer virtuellement les conditions d'une réunion, en augmentant (par rapport au monde réel) les possibilités des utilisateurs.

Dans les réunions de petits groupes médiatisées, nous avons constaté les limites des interfaces 2D habituelles, et des solutions utilisant la vidéo pour établir une coopération. Nos travaux portent sur une nouvelle organisation de l'espace de travail, en trois dimensions. Il a fallu pour cela développer un système d'interaction, des couches les plus basses aux primitives d'interaction de haut niveau, afin de créer un modèle cohérent et complet, permettant d'interagir sur les documents partagés de la réunion, et avec ses interlocuteurs. Ce modèle est composé de règles de conception qui concernent aussi bien les dispositifs d'entrée que de sortie, il est basé sur des évaluations et sur une analyse des problèmes de perception de l'utilisateur face à un environnement 3D virtuel.

A partir de la définition de ces méthodes d'interaction de base du modèle, nous construisons des primitives d'interaction de haut niveau comme les menus 3D. Nous décrivons également nos choix en matière de communication, en tenant compte des paramètres essentiels de la coopération dans une réunion. La représentation des acteurs, sous forme de clones 3D, et des documents se fait dans le même espace, permettant de créer une cohérence visuelle forte de la scène, qui aide les utilisateurs dans la compréhension et la réalisation de leur tâche. Les clones étant entièrement paramétrables, on les anime pour informer un utilisateur sur l'activité de ses interlocuteurs.

Cet ensemble, basé sur la synthèse d'image temps réel et une démarche d'évaluations, forme un outil permettant d'organiser des réunions entre personnes distantes, comme des réunions de bureau d'étude, de la télévente, de l'enseignement à distance, etc.

Mots Clés : interface 3D, interaction 3D, bi-manuel, évaluation, indices visuels, TCAO synchrone, clone, Environnement Virtuel Coopératif.

Title: A 3D interaction model : Human Computer Interaction and Human Computer Human Interaction in 3D Interface for synchronous CSCW.

We present a three-dimensional user interface for synchronous co-operative work, Spin; which has been designed for multi-user synchronous real-time applications to be used in, for example, meetings and learning situations. Spin is based on a new metaphor of virtual workspace.

We have designed an interface, for an office environment, which recreates the three-dimensional elements needed during a meeting and increases the user's scope of interaction. In order to accomplish these objectives, animation and three-dimensional interaction in real time are used to enhance the feeling of collaboration within the three-dimensional workspace. Spin is designed to maintain a maximum amount of information visible. The workspace is created using artificial geometry -as opposed to true three-dimensional geometry- and spatial distortion, a technique which allows all documents and information to be displayed simultaneously while centering the user's focus of attention. Users interact with each other via their respective clones, which are three-dimensional representations displayed in each user's interface, and are animated with user action on shared documents. An appropriate object manipulation system (direct manipulation, 3D devices and specific interaction metaphors) is used to point out and manipulate 3D documents. All these propositions make a set of rules, some guidelines for 3D interface design, to work in a single or especially in a multi-user environment.

Keywords: Synchronous CSCW, CVE, clone, three-dimensional interface, 3D interaction, bimanual, evaluation, visual cues.
