



Un modèle de conception dédié à l'interaction collaborative colocalisée

THÈSE

présentée et soutenue publiquement le 7 octobre 2011

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille

par

Jérémy Ringard

Composition du jury

<i>Président :</i>	Luigi Lancieri	LIFL, Université de Lille 1
<i>Rapporteurs :</i>	Bruno Arnaldi Pierre Leclercq	IRISA, Rennes Université de Liège
<i>Examineurs :</i>	Indira Thouvenin Jean-Pierre Jessel	UTC, Compiègne IRIT, Toulouse
<i>Directeur :</i>	Christophe Chaillou	LIFL, Université de Lille 1
<i>Encadrants :</i>	Samuel Degrande Patricia Plenacoste	LIFL, Université de Lille 1 LIFL, Université de Lille 1

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

Laboratoire d'Informatique Fondamentale de Lille — UMR USTL/CNRS 8022

U.F.R. d'I.E.E.A. — Bât. M3 — 59655 VILLENEUVE D'ASCQ CEDEX

Tél. : +33 (0)3 28 77 85 41 — Télécopie : +33 (0)3 28 77 85 37 — email : direction@lifl.fr

Remerciements

je tiens à remercier Bruno Arnaldi et Pierre Leclercq pour m'avoir fait l'honneur de juger ce travail et d'en être les rapporteurs. Je remercie également Indira Thouvenin et Jean-Pierre Jessel pour avoir accepté de participer à mon jury, ainsi que Luigi Lancieri pour avoir accepté d'en être le président.

Merci à Christophe Chaillou pour m'avoir offert la possibilité de rejoindre l'équipe GRAPHIX et m'avoir permis de réaliser cette thèse.

J'aimerais également adresser toute ma reconnaissance à mes deux encadrants, Samuel De-grande et Patricia Plenacoste. Leur sympathie, leur soutien, et leur expérience m'ont été particulièrement précieux au cours de cette thèse et de mon master.

Je remercie également toute l'équipe du projet ANR Part@ge, dans le cadre duquel ce travail a été réalisé. Merci également aux membres de l'UMR de géographie de l'université de Lille 1, pour m'avoir accueilli en tant qu'observateur au sein de leur univers : Philippe Menerault, Collette Morice et Elodie Castex ainsi que leurs élèves.

Ce document n'aurait pas pu voir le jour sans le soutien de mes proches : Adeline, celle qui partage ma vie, qui a su se montrer compréhensive et qui m'a apporté le réconfort nécessaire. Elle a su trouver les mots justes pour me soutenir dans les moments de doute qui ont ponctué ce travail, et pour cela je lui en suis infiniment reconnaissant. Ma famille, qui a su accepter mes fréquentes absences physiques et mentales durant ces années, ainsi que ma belle-famille, en particulier Cécile qui constituait l'unique présence humaine qu'il était possible de trouver lors de mes nombreuses séances de travail nocturnes. Bien entendu, je remercie également mes amis proches, qui ont toujours été à l'écoute, et qui ont ce talent particulier pour dédramatiser toutes les situations.

Je remercie également les membres de l'Ornis Theatron et leurs oiseaux, avec qui les nombreux bons moments passés m'ont apporté une sérénité plus que bienvenue.

Ces années de travail au LIFL, au delà de l'aspect scientifique, ont été pour moi la source de rencontres humaines très importantes. Aussi, je souhaite remercier tous les membres et anciens membres des équipes GRAPHIX/MINT/SHAMAN pour leur amitié, leur humour, et la bonne ambiance qu'ils ont apporté dans les bureaux. Je pense entre autres à Jowan, Meukit, Max, Dr Peanuts, Christian, Hadrien, JP et Mario.

Je désire enfin adresser une pensée émue pour Michel Barbaut, qui n'aura jamais manqué de prendre des nouvelles de ma thèse. Je souhaite dédier ce document à sa mémoire.

Table des matières

Table des figures	7
Introduction	1
1 Interaction collaborative colocalisée : état de l'art	9
1.1 Le travail collaboratif assisté par ordinateur	10
1.1.1 Les systèmes collaboratifs	10
1.1.1.1 TCAO : définition	10
1.1.1.2 Taxonomie du TCAO	11
1.1.1.3 Collaboration synchrone distante	13
1.1.2 Technologies et concepts associés aux environnements collaboratifs	19
1.1.2.1 La réalité virtuelle et les environnements virtuels collaboratifs	20
1.1.2.2 Réalité Virtuelle immersive/non immersive	21
1.2 L'homme au centre d'une collaboration colocalisée	23
1.2.1 L'espace : une notion complexe	24
1.2.1.1 La notion d'espace virtuel dans une application interactive	24
1.2.1.2 De l'espace physique à l'espace virtuel : la particularité du travail colocalisé	25
1.2.1.3 "Space" vs "place" : une approche sémantique de l'espace	27
1.2.1.4 L'appropriation de l'espace en milieu colocalisé	29
1.2.2 La collaboration colocalisée : war room et communication de groupe	35
1.2.2.1 La communication de groupe en coprésence	36
1.2.2.2 War room : caractéristiques	38
1.2.2.3 Observations de l'utilisation des war room informatisées	41
1.2.2.4 La collaboration de groupe à groupe : notion de présence mixte	45
1.3 Exemples de plateformes prenant en charge la colocalisation	49

1.3.1	Affichage unique	49
1.3.2	Affichage partagé	50
1.3.3	Multi-dispositifs	51
1.3.4	Le multi-vues dans le cadre d'une collaboration distante	53
1.3.5	Conclusion	55
2	Interaction collaborative colocalisée : analyse et extraction des besoins	57
2.1	Analyse des travaux sur la collaboration et déduction des besoins	58
2.1.1	L'impact des comportements de groupe	58
2.1.2	L'impact de la notion d'espace	60
2.1.3	Les comportements individuels des membres d'une équipe	61
2.2	Conception de notre plateforme	64
2.2.1	La revue de projet : une étape qui met l'accent sur la collaboration	64
2.2.2	Notre proposition : la war room assistée par ordinateur	65
2.3	Reflexions sur le concept de canaux de visualisation et d'interaction	76
2.3.1	La structuration de l'activité dans la war room : le système de canaux	76
2.3.2	Illustration de l'utilisation des canaux	77
2.3.3	Les canaux à l'échelle des données : approche sémantique des interactions	80
2.3.4	Les canaux pour donner une portée concrète à des informations non représentables	81
2.3.5	Les canaux, notion distincte de l'interface	81
2.3.6	Les canaux : bilan	82
2.4	Conclusion de notre analyse	82
3	Approche logicielle de la collaboration : état de l'art	85
3.1	Introduction : distinction entre la définition et la représentation d'un objet virtuel	86
3.2	Caractéristiques logicielles des EVC	86
3.2.1	La notion d'objets partagés	86
3.2.2	Technologies : la distribution des données	87
3.2.3	Flux de communication et cohérence	90
3.2.4	Taxonomie et topologie	91
3.3	Les architectures logicielles	93
3.3.1	Les modèles de conception d'applications interactives	93
3.3.1.1	Les modèles monolithiques	94
3.3.1.2	Les modèles à agents	97
3.3.1.3	Les hybrides	100
3.3.2	Nouvelle approche de l'interaction : vers l'interaction instrumentale	102

3.3.3	SPIN 3D : une architecture logicielle singulière	103
3.4	Les systèmes interactifs complexes : les EVC	106
3.4.1	Virtools	107
3.4.2	OpenMask	108
3.4.3	SPIN 3D	109
3.4.4	Conclusion	110
4	Approche logicielle de la collaboration colocalisée : Adaptation d'une plate- forme EVC classique	113
4.1	Architecture logicielle : distribution libre de l'application	114
4.1.1	Les contraintes d'utilisation et leur impact sur l'architecture	114
4.1.2	Communication réseau : distribution des composantes	115
4.1.3	Architecture multi-agents : utilisation du modèle MVO de SPIN 3D	119
4.2	Support de canaux hétérogènes : la notion de modèle abstrait	124
4.2.1	Le MVO : les limites du modèle	125
4.2.1.1	Données concrètes et données abstraites	126
4.2.1.2	Cohabitation des données concrètes et abstraites	127
4.2.2	Les modèles existants	128
4.2.3	M_NV-O : Adaptation de la structure MVO	130
4.2.4	Transparence de l'adaptation et distribution distante	132
4.2.5	Multiplicité des niveaux d'abstraction	134
4.2.6	Insertion d'une application tierce dans le contexte multi niveaux d'abstrac- tion	136
4.2.7	Intégration dynamique d'un nouveau dispositif	136
4.2.8	Choix des données dans le cadre du multicanal	138
5	Mise en oeuvre de la plateforme en situation réelle	145
5.1	Mise en oeuvre de la distribution des composantes logicielles	146
5.1.1	Utilisation de la DAI de SPIN 3D	146
5.1.1.1	Communication inter-composantes : le système de proxy	147
5.1.2	Déploiement et configuration XML	157
5.2	Démonstrateur de distribution logicielle	158
5.2.1	Scénario et fonctionnalités	159
5.2.2	Implémentation : les composantes impliquées	164
5.2.3	Résultats	167
5.2.4	Perspectives	168
5.3	Mise en oeuvre de l'abstraction M_NV-O	169

5.4	Mise en application de la plateforme dans un projet d'urbanisme	173
5.4.1	L'urbanisme : contexte de notre étude	173
5.4.2	L'urbanisme : un domaine complexe, propice à l'utilisation de la war room	174
5.4.3	Observation d'un projet concret	176
5.4.3.1	Phase de production	176
5.4.3.2	La phase de maquettage informatique	179
5.4.4	L'installation matérielle mise en oeuvre	179
5.4.5	L'environnement virtuel partagé	180
5.4.5.1	Le canal 2D	181
5.4.5.2	Le canal 3D global	183
5.4.5.3	Le canal de modélisation 3D	186
5.4.6	Distribution de la plateforme	187
5.4.6.1	Caractéristiques générales	187
5.4.6.2	Le déploiement des composantes et la configuration XML	188
5.4.6.3	Les données et leur signification	192
5.4.7	Bilan	194
6	Conclusion	197
6.1	Résumé de la contribution	198
6.2	Questionnements et perspectives	200
	Bibliographie	203

Table des figures

1.1	Matrice de classification spatio temporelle.	11
1.2	Le modèle du trèfle des collecticiels	12
1.3	Le mur de téléprésence de Orange Labs.	14
1.4	Exemple d'utilisation d'un système VNC : le bureau de l'ordinateur distant apparaît dans une fenêtre (crédit photo : http://remote-desktop-control.com)	17
1.5	War room classique utilisant des supports papiers.	20
1.6	Exemple de cas d'utilisation de la réalité virtuelle pour la collaboration. De haut en bas : OpenMask, Dive, SPIN 3D	22
1.7	Environnement immersif de type CAVE	23
1.8	Echanges interactifs entre les utilisateurs et leur environnement. (tiré de [3])	26
1.9	Défilement des objets virtuels autour de la table tactile. Tiré de [HCS06]	30
1.10	le frisbee de Khan et al.	31
1.11	Le périphérique physique de localisation proposé par Terrell et al.	34
1.12	Système de flèche de Wigdor et al. pour l'affichage d'un document sur plusieurs dispositifs.	35
1.13	L'environnement de travail de la X-Team, dont les écrans affichent les différents sous-systèmes à gérer pour le projet.[Mar02]	43
1.14	L'environnement CoLaB de Xerox	51
1.15	Les dispositifs matériels de Roomware. De haut en bas : ConnecTables, Comm-Chairs, InteracTable et Dynawall	52
1.16	L'installation matérielle I-Room	53
1.17	Les deux modes de représentation de la plateforme d'interaction distante de Schaffer et al.	54
2.1	Illustration de la plateforme, présentée au salon professionnel Laval Virtual.	66
2.2	Schema de répartition des utilisateurs dans le cadre d'un environnement de collaboration distante type EVC (note : peu importe le moyen utilisé pour la synchronisation)	68
2.3	Evolution du schéma de répartition des utilisateurs au cours de la collaboration dans la war room. Les utilisateurs sont amenés à changer de dispositif en fonction de leur activité, éventuellement en formant des sous groupes autour du même dispositif. Légende : fleches rouges : interaction et visualisation fleches jaunes : visualisation seule fleches vertes : interaction seule	69

2.4	En haut : une approche traditionnelle, qui considère les stations de travail dans leur ensemble. En bas : notre approche, qui distingue nettement l'aspect "interaction" de l'aspect "visualisation".	71
2.5	Comparaison conceptuelle entre deux stratégies de collaboration à plusieurs dispositifs d'interaction	72
2.6	Trois représentations possibles d'un projet de conception d'un téléphone. Une première représentation propose l'objet 3D de la coque, une deuxième propose le schéma électronique et la troisième est une représentation combinée des deux précédentes, permettant de contrôler la cohérence globale du projet	79
3.1	Base de données centralisée	88
3.2	Base de données dupliquées à cohérence faible	88
3.3	Base de données dupliquées à cohérence forte	89
3.4	Base de données distribuée	89
3.5	Architecture centralisée	92
3.6	Architecture pair-à-pair	93
3.7	Le modèle Arch	94
3.8	Le modèle Zipper	95
3.9	Le métamodèle de Dewan	96
3.10	Le modèle MVC	97
3.11	Le modèle PAC mis en application pour la représentation d'un thermomètre	99
3.12	Le modèle ALV mis en application dans un contexte multi-utilisateurs	99
3.13	Le modèle clock	100
3.14	Le modèle PAC-amodeus	101
3.15	Le modèle coPAC	101
3.16	Trois objets distincts dans le même environnement. Selon le modèle MVC, les interactions provenant de l'utilisateur transitent par le contrôleur, propre à chaque objet	104
3.17	Utilisation d'un outil générique. Contrairement au contrôleur de MVC, cette composante est propre à une interaction et peut s'appliquer sur plusieurs objets distincts	104
3.18	L'outil select and manip. Il est constitué d'un automate à deux états.	105
3.19	La relation qui est établie entre le périphérique d'interaction et le modèle d'un objet passe par les opérateurs de l'outil.	105
4.1	En haut : distribution des composantes de 2 agents entre trois postes de travail. En bas : la relation client/serveur qui en résulte	120
4.2	Utilisation d'un terminal mobile comme dispositif d'interaction distante	121
4.3	Déploiement de l'application selon un modèle client-serveur	122
4.4	Utilisation d'un écran tiers de grande taille pour la représentation visuelle. Lorsque la présentation est terminée, l'utilisateur isolé peut reprendre son travail individuel simplement en ramenant la composante Vue sur son poste de travail local	123
4.5	Utilisation de deux composantes Vue concurrentes attachées au même modèle. Ces deux vues représentent simultanément le même objet selon des modes sensiblement différents	123
4.6	Combinaison sur le même poste de deux vues issues de deux objets distincts conçus sur des postes de travail individuels. L'écran central peut ainsi être utilisé pour représenter l'assemblage de deux pièces mécaniques	124

4.7	Utilisation d'une application tierce et transmission des données produites au sein de la plateforme via la DAI	125
4.8	Le modèle MVO de SPIN 3D.	126
4.9	Hierarchie d'agents PAC pour gérer plusieurs degrés d'abstraction	130
4.10	Déroulement des échanges inter-composantes : 1) l'outil modifie les données visuelles du modèle concret 2) le modèle concret notifie sa vue des changements et retranscrit l'interaction sous la forme d'un évènement abstrait transmis au modèle abstrait 3) suite à la reception d'un évènement, le modèle abstrait modifie ses champs et en notifie les modèles concrets qui lui sont attachés 4) le second modèle concret ayant reçu notification du modèle abstrait retranscrit cette interaction sur ses propres données et notifie sa vue.	132
4.11	Factorisation des données pour plusieurs vues appartenant au même canal, mais exécutées sur des postes différents.	133
4.12	Communication entre deux sites distants selon le modèle MVO classique : deux modèles similaires sont instanciés de part et d'autre.	133
4.13	Exemple de répartition asymétrique entre deux sites distants.	134
4.14	Exemple de hierarchie de plusieurs degrés d'abstraction. Le premier degré constitue le niveau "concret".	135
4.15	Utilisation d'un modèle concret comme une interface communiquant avec la DAI. Le modèle concret (cerclé de bleu sur le schéma) prend en charge les mécanismes de retranscription concret/abstrait, et est ainsi réutilisable par exemple en y ajoutant une composante vue.	137
4.16	Partie supérieure : poste de travail autonome fonctionnant en mode hors-ligne Partie inférieure : connexion du modèle concret du poste autonome au reste de la war room	138
4.17	Utilisation de deux données concrètes dans le même canal : le mécanisme de retranscription concret/abstrait peut prendre en compte les deux paramètres concrets.	139
4.18	Séparation des deux données concrètes en deux canaux distincts. En cas de modification du matériau, la masse de l'objet devrait être affectée, mais n'est pas calculable en raison de la méconnaissance du volume de l'objet. Ce cas de figure est invalide.	139
4.19	Encodage de la masse sous la forme de deux informations distinctes : Volume et Densité.	140
4.20	Factorisation des données concrètes au sein d'un unique modèle concret. Cette solution permet de structurer correctement les mécanisme de retranscription concret/abstrait.	141
5.1	Illustration des relations entre les composantes dans la plateforme SPIN 3D. Les liens entre ces composantes sont directs.	148
5.2	Illustration des deux types de composantes : à gauche, un modèle "proxy", et à droite un modèle "local".	150
5.3	Première proposition d'implémentation des composantes polymorphes.	150
5.4	Seconde proposition d'implémentation des composantes polymorphes.	151
5.5	Troisième proposition d'implémentation des composantes polymorphes.	152
5.6	Définition en IDL des interfaces partagées sur le bus CORBA	153
5.7	Diagramme de classes présentant l'implémentation du système de proxy intégré à la composante modèle	154

5.8	Diagramme de classes présentant l'implémentation du système de proxy intégré à la composante vue	155
5.9	Diagramme de sequence d'une interaction visant à opérer une rotation sur l'objet, par l'intermédiaire d'un item visuel	156
5.10	Diagramme de sequence d'une interaction réalisée grâce à un Outil.	156
5.11	Les trois postes de travail du démonstrateur de manipulation d'une maquette 3D.	160
5.12	L'interface 3D de la war room	161
5.13	L'interface 2D de la war room	162
5.14	Déplacement de l'affichage par clic sur un item visuel	163
5.15	distribution des composantes dans le premier prototype	165
5.16	distribution des composantes dans le premier prototype	166
5.17	Proposition de remaniement du prototype, exploitant différents degrés d'abstraction	169
5.18	Exemple de hierarchie de plusieurs degrés d'abstraction. Le premier degré constitue le niveau "concret".	170
5.19	Diagramme de classe illustrant l'implémentation du modèle concret dans l'architecture logicielle.	171
5.20	Composantes déployées dans un contexte multicanal	172
5.21	Diagramme de sequence dans le cadre d'une utilisation multicanal	173
5.22	Exemple de maquette réalisée dans le cadre d'un projet de réaménagement urbain	176
5.23	Photographie aérienne servant de support à la réalisation de la maquette	177
5.24	Tracé des surfaces significatives de la maquette	178
5.25	Construction de l'aspect volumétrique de la maquette	178
5.26	Le matériel utilisé pour le démonstrateur d'urbanisme.	180
5.27	Interface associée au canal 2D. (notez la présence du menu circulaire sur la droite)	182
5.28	Surligné de vert : le contrôle du nombre d'étages d'un bâtiment sur l'interface 2D.	184
5.29	Interface 3D du prototype d'urbanisme.	185
5.30	Déploiement des composantes pour le démonstrateur d'urbanisme. En bleu les objets éditables, en vert les objets standards.	189
5.31	Adaptation des données en fonction du canal : à gauche, la donnée abstraite. . .	193

Introduction

Cette thèse s'inscrit dans le domaine informatique et plus spécifiquement dans ceux de l'interaction homme/machine et de l'interaction collaborative. Avant de présenter en détail le sujet à l'origine de ces travaux et notre proposition, nous commencerons cet exposé par la définition du cadre de nos recherches vis-à-vis du domaine concerné, ainsi que le contexte scientifique.

Avant propos : d'un usage personnel vers des activités multi-utilisateurs

A leurs débuts, les outils informatiques proposés étaient de deux types : les systèmes qualifiés de "ouverts" (ou interactifs), par opposition aux systèmes dits "fermés" dont l'intégralité des actions est régie par des algorithmes, sans nécessiter d'interaction de la part de l'homme [Weg97]. Ces applications étaient par conséquent conçues dans une optique focalisée sur l'usage individuel : le dialogue qui s'instaurait lors de l'utilisation d'une application ouverte ne concernait que l'application elle-même et l'utilisateur. L'utilisateur avait alors un rôle d'interrogateur, tandis que la machine prenait en charge les traitements nécessaires à l'obtention de la réponse. Cette configuration pourtant simple a toutefois été à l'origine de nombreux questionnements relatifs aux moyens nécessaires à l'instauration de ce dialogue. En effet, les chercheurs et les utilisateurs ont très vite constaté que la qualité de l'utilisation d'une application était en majeure partie dépendante de la fluidité du transfert d'informations qui avait lieu entre l'utilisateur et l'ordinateur. Ce constat révèle que la démarche de conception et de mise en oeuvre d'une application interactive doit être centrée sur l'utilisateur [ND86].

La problématique de l'interaction homme/machine était alors définie selon deux axes :

- Les moyens matériels ou logiciels à mettre en oeuvre pour simplifier l'envoi de la requête de l'utilisateur vers la machine. Cet axe a donné lieu à la création de périphériques d'entrée tels que la souris ou les stylets, ainsi qu'aux interfaces graphiques permettant de s'affranchir du système de console texte.
- Les techniques mises en oeuvre par l'application pour communiquer ses informations à l'utilisateur. Elles ont été explorées, là encore à travers des travaux sur les interfaces graphiques, ou sur les dispositifs matériels dotés d'un retour sensoriel (visuel, tactile ou sonore).

Cette vision de l'usage individuel de l'outil informatique a toutefois été bouleversée par le développement des réseaux intranet ou extranet. La liaison constante entre plusieurs terminaux, et les horizons nouveaux qui se sont alors ouverts en terme d'usage ont ainsi progressivement poussé les applications à évoluer vers des configurations mettant en scène plusieurs utilisateurs interagissant avec un seul et même système. Aux problématiques d'interaction homme/machine déjà présentes se sont alors ajoutées des contraintes liées au partage des ressources. En effet, à cette époque, la notion de multi-utilisateurs était associée à la contrainte de gérer les accès concurrents aux ressources du système, autrement dit de "cacher et protéger les utilisateurs les uns des autres, leur laissant croire qu'ils sont seuls et que l'ensemble des ressources leur est réservé." [Rou00]. Mais contrairement à cette approche, dans la pratique, la notion de groupe d'utilisateurs implique essentiellement des activités collectives, que ce soit pour partager des ressources ou directement pour interagir conjointement sur les mêmes données. Il n'est en fait pas nécessaire de procéder à un isolement des utilisateurs, mais au contraire de les impliquer dans une activité de groupe. C'est à ce phénomène que l'on doit depuis plusieurs années l'arrivée d'applications de TCAO (Travail Collaboratif Assisté par Ordinateur), d'applications ludiques

(jeux en ligne massivement multi-joueurs tels que World of Warcraft de Blizzard Entertainment), et plus récemment l'émergence des réseaux sociaux virtuels tels que Facebook ou LinkedIn. Le point commun entre toutes ces technologies est qu'elles reposent sur le même besoin de transmettre et de partager des informations entre les utilisateurs.

Contexte : un besoin industriel

Les progrès de l'informatique et des systèmes de communication qui ont eu lieu depuis la démocratisation de l'ordinateur ont toujours eu un impact extrêmement fort sur le monde de l'entreprise. Les technologies relativement récentes telles que les téléphones portables ou les réseaux distants à haut débit ont été adoptées presque instantanément par les industries et les méthodes de travail ont très rapidement tiré profit de ce nouvel environnement technologique. Il n'est pas rare de voir des conférences téléphoniques ou même des vidéoconférences, et l'échange de courriers électroniques est devenu une pratique courante, même pour communiquer au sein de la même infrastructure.

Si les entreprises voient faciliter leur essor géographique en bénéficiant des systèmes de communication distante (avec leurs clients, leurs fournisseurs ou leurs partenaires commerciaux), l'organisation interne de l'entreprise s'est également modelée autour de l'utilisation des systèmes informatiques. Les outils de gestion logistique ont par exemple pris une place de tout premier ordre dans la vie de l'entreprise, permettant ainsi de centraliser toutes les données utiles à la coordination. L'intégralité des procédures qui ont lieu dans une société est désormais soumise à une distribution des données sur le réseau, qui offre un gain considérable en terme de productivité et d'automatisation des tâches. Si une entreprise peut être découpée et hiérarchisée en différents départements, chacun de ces éléments est étroitement lié aux autres à travers les outils informatiques. Cette structure aboutit alors à plusieurs systèmes d'information interconnectés, formant parfois une toile complexe.

Au delà du problème de gestion de données communes à l'entreprise, un besoin majeur s'est fait ressentir depuis plusieurs années : la gestion de projet, qui constitue une activité d'une importance primordiale. Ce type de travaux nécessite l'intervention d'un groupe d'utilisateurs, pour procéder à des tâches de conception, de discussion et de production vis-à-vis d'un objectif commun à tous. Le suivi d'un projet, quelle que soit son ampleur, requiert en effet l'intervention de plusieurs personnes afin de bénéficier de l'expertise et de la créativité propre à chacun. Il nécessite ainsi le concours de divers corps de métier, dont les spécialités respectives peuvent influencer positivement le cours de la conception. Si l'on observe de plus près le fonctionnement d'un groupe travaillant autour d'un projet, nous pouvons constater que leur structure est analogue à la façon dont fonctionne les réseaux d'informations interconnectés évoqués dans le paragraphe précédent : chaque utilisateur représente un ensemble de connaissances correspondant à sa propre représentation mentale du projet et ces ensembles s'avèrent étroitement liés les uns aux autres. De plus, en conséquence des technologies de communication distante, il est devenu envisageable pour les entreprises de mettre en oeuvre des projets de plus grande ampleur, faisant intervenir des spécialistes parfois éloignés géographiquement.

En parallèle à la notion de groupe et de collaboration, l'aspect matériel requiert également une attention particulière. Les projets développés par les industries relèvent d'un niveau d'expertise élevé, qui nécessite souvent une approche très technique des données. Nous pouvons aujourd'hui constater que la plupart des produits industriels sont mis au point par le biais d'outils numé-

riques de Conception Assistée par Ordinateur (CAO) tels que CATIA de Dassault Systèmes, ou Autodesk Inventor. Ces supports logiciels permettent de procéder à des activités de conception notamment à travers un environnement virtuel 3D.

Les paramètres à manipuler pouvant être d'une certaine complexité, il est alors apparu nécessaire de développer des systèmes répondant aux problématiques d'interaction homme/machine, c'est-à-dire les méthodes de représentation de ces données et les outils (tant matériels que logiciels) permettant de manipuler ces données. Le matériel s'est par conséquent adapté aux besoins des utilisateurs, proposant des dispositifs toujours plus pertinents et plus variés.

Les réponses apportées au niveau matériel

Pour répondre aux besoins de plus en plus précis des utilisateurs, l'évolution technologique matérielle a donné le jour à de nombreux dispositifs de visualisation et d'interaction. Au même titre que le débit des réseaux est en croissance constante, les machines disposent maintenant d'une puissance de calcul suffisante pour prendre en charge des affichages 3D complexes, des flux vidéos haute résolution, ou d'une manière plus générale des affichages associés à des données nombreuses et/ou de taille importante. Les dispositifs tendent ainsi à exploiter au mieux cette puissance, à la fois à travers les périphériques d'affichage et les périphériques d'interaction.

Il est devenu possible, par exemple, de disposer d'écrans de projection haute résolution, permettant de diffuser des informations de façon à être visible par le plus grand nombre. Cette accessibilité des dispositifs d'affichage amène d'ailleurs à la création d'infrastructures immersives, comme peuvent l'être les CAVE (Cave Automatic Virtual Environment) [CNSD⁺92a]. L'affichage 3D stéréoscopique, autrefois réservé aux professionnels de l'image, se démocratise de plus en plus, et atteint une accessibilité financière telle que le grand public peut désormais disposer de cette technologie dans son salon.

Outre l'affichage, la puissance des machines permet également le calcul d'algorithmes complexes de capture des gestes de l'utilisateur. Ces technologies tendent à rendre possible l'interaction avec un environnement 3D de façon intuitive avec le corps, la main ou même simplement avec les yeux, par simple détection vidéo grâce à une ou plusieurs cameras. Là encore ces systèmes bénéficient d'une certaine démocratisation, illustrée notamment par le périphérique Kinect de Microsoft. Les centrales inertielles ont également fait leur apparition dans les salons, à travers les contrôleurs de la Nintendo Wii, permettant ainsi de bénéficier d'interaction 3D à moindre coût.

Enfin, il est important de citer les périphériques multitouch, très populaires depuis quelques années, qui permettent une interaction à plusieurs mains/doigts par simple effleurement de l'écran. Ces systèmes d'interaction tactile sont, eux aussi, accessibles au plus grand nombre, à tel point que les grands constructeurs en téléphonie tendent tous à équiper leurs produits de ces technologies tactiles.

Un des exemples contemporains les plus frappant en regard de la démocratisation des équipements informatiques haut de gamme est l'adoption grandissante des salles de classe à écran interactif dans les établissements scolaires.

Les solutions matérielles sont, comme nous pouvons le constater, d'une grande diversité et offrent désormais la possibilité d'interagir avec un environnement donné de la façon la plus natu-

relle. Chacun peut maintenant choisir le dispositif le plus en adéquation avec le cas d'usage désiré.

Le support logiciel associé à ces dispositifs subit également une évolution constante, notamment à travers les améliorations des codecs vidéos et des réseaux à haut débit, qui permettent maintenant de transmettre et diffuser des flux vidéos haute définition en temps réel et bientôt des flux vidéos stéréoscopiques.

Les réponses apportées au niveau logiciel

Les solutions logicielles proposées par la recherche scientifique pour répondre au besoin industriel en matière de gestion de projet et de collaboration sont nombreuses : elles constituent, en effet, depuis de nombreuses années un domaine de recherche à part entière. Les solutions apportées ont d'abord exploré les contraintes de communication, à travers notamment les solutions de diffusion audiovisuelles, mais aussi les contraintes liées à la distribution des données (via des systèmes de documents partagés) et d'interaction. L'évolution majeure qui a impacté les outils de collaboration fût l'utilisation de la réalité virtuelle.

D'abord conçues dans une optique mono-utilisateur, les plateformes de réalité virtuelle ont permis d'aborder la manipulation de données complexes avec plus de fluidité et de simplicité. La recherche a exploré ce type d'environnements pour les faire évoluer vers des environnements collaboratifs, permettant la coopération de plusieurs personnes distantes, chacune travaillant sur son propre poste de travail et étant intégré à ces espaces virtuels immersifs.

Les utilisateurs ne perçoivent plus les moyens informatiques comme un simple moyen de communication ou de production mono-utilisateur, mais peuvent désormais se livrer à de réelles activités de groupe interactives, permettant de produire du contenu ensemble. Les environnements virtuels collaboratifs (EVC) distants ont ainsi apporté aux utilisateurs des capacités d'interaction complexe sur un projet virtuel, tout en s'affranchissant des distances.

Si ces environnements ont offert une gamme d'outils bien plus riches aux industriels, l'adoption de la réalité virtuelle a toujours été plus lente que pour les outils de communication ou de conception : en dehors des entreprises de grande ampleur disposant de moyens matériels importants et dont la méthodologie est très précise, l'utilisation de cette technologie est longtemps restée assez marginale. Si les coûts matériels ont par le passé constitué le plus gros frein à l'utilisation massive des EVC, les méthodes de travail n'ont toutefois pas toujours su trouver leur place dans le contexte en apparence très fermé des outils de réalité virtuelle collaboratifs. Ces outils restent malgré tout très spécialisés sur un type de tâche, ce qui a tendance à les rendre trop rigides vis-à-vis de la diversité potentielle des actions réalisées pendant une activité de groupe. Aujourd'hui ce phénomène a cependant tendance à s'amoinrir et les acteurs industriels s'ouvrent de plus en plus à cette technologie.

Problématique

En terme de collaboration distante, la communication constitue encore une contrainte majeure : à travers les EVC tels que nous les connaissons, les utilisateurs passent par une "communication médiatisée", c'est-à-dire passant par des moyens artificiels (audio, video, ou représentations symboliques telle que les avatars). Cette contrainte est à l'origine de nombreuses problématiques visant à conserver au maximum la notion de "conscience sociale" [DF07]. Si ces outils apportent

un avantage non négligeable lors d'une collaboration distante, les cas concrets de collaboration qui ont lieu à l'heure actuelle impliquent majoritairement un travail d'équipe dont les membres sont situés au même endroit géographique : c'est ce que nous appellerons la "coprésence", ou "colocalisation"¹. L'utilisation d'un système imposant cette communication médiatisée s'avère alors trop contraignante pour être exploitée de façon systématique, puisqu'elle est essentiellement tournée vers une communication de personne à personne, par opposition à une communication de groupe, qui implique au contraire une communication de sous-groupe à sous-groupe.

L'objet de cette thèse est de s'intéresser à cette notion de collaboration colocalisée dans le cadre de la réalisation d'une maquette numérique 3D. Ce type d'activité consiste à exploiter un support visuel tridimensionnel et à interagir sur celui-ci pour concevoir un projet en équipe. Ces activités peuvent être de la revue de projet, de la conception urbaine, ou tout autre projet reposant au moins en partie sur une représentation 3D. La maquette 3D constitue ici le référentiel principal autour duquel s'articule la collaboration. Il importe avant tout de comprendre quels paramètres freinent encore la généralisation des systèmes de réalité virtuelle dans les entreprises. La collaboration colocalisée constituant une part importante du travail fourni par les industriels, nous allons chercher à concevoir une plateforme logicielle et matérielle générique permettant à une équipe projet de coopérer dans les meilleures conditions dans un espace de travail commun.

Cadre de nos recherches

La réutilisabilité de la plateforme que nous voulons définir, de même que sa capacité à s'adapter à une large gamme de cas d'usage constitue un de nos principaux objectifs. Il est bien entendu impossible de concevoir une technologie "ultime", permettant de répondre à toutes les utilisations possibles. Nous devons donc nous positionner à un niveau d'abstraction plus élevé notamment en travaillant sur les moyens à mettre en oeuvre pour créer des applications de collaboration colocalisée. La question est alors de définir comment concevoir ce type de plateforme en conservant une certaine pertinence vis-à-vis des variables propres à chaque projet (type de projet, infrastructure matérielle, niveau d'expertise des utilisateurs).

Ce travail nécessite un questionnement sur l'aspect matériel de cette plateforme, c'est-à-dire de rechercher s'il existe une configuration matérielle suffisamment générique pour faire face à la plupart des cas de figure. S'il s'avère que les cas d'usage sont trop variés, faut-il concevoir l'application en prenant en compte les infrastructures matérielles destinées à l'accueillir, ou au contraire est-il possible de permettre une certaine liberté concernant cet aspect ? D'une manière générale, nous allons chercher à définir comment implanter l'activité librement sans être soumis aux problèmes issus des diverses contraintes matérielles. Ces questionnements donneront ainsi lieu à une réflexion sur les méthodes à mettre en oeuvre pour concevoir l'application d'un point de vue "génie logiciel". Dans quelle mesure peut-on exploiter les méthodes de conception existantes pour assurer la collaboration en situation de coprésence ? Ces modèles de conception sont-ils déjà prêts à accueillir de telles configurations ? Dans quelle mesure doivent-ils être adaptés ?

Notre objectif est la mise en place d'un outil répondant aux besoins relatifs à la collaboration colocalisée via un environnement virtuel non-immersif. Ces travaux vont ainsi chercher à mettre en avant le travail d'équipe en mettant le groupe d'utilisateurs au centre de nos considérations. En effet, l'utilisation d'un environnement immersif met l'accent en particulier sur le travail individuel,

1. le terme "colocalisation" est ici utilisé en tant que traduction du terme anglo-saxon "colocated" qui référence la proximité de plusieurs utilisateurs dans un même lieu

en adaptant l'univers virtuel à la perception d'une seule personne. Cette approche est selon nous opposable à la notion de groupe colocalisé. Dans notre cas, il nous apparaît plus pertinent de considérer l'équipe de collaborateurs dans son ensemble, comme une seule et même entité. La notion de collaboration distante constituant cependant un facteur important, nous devons évoquer également la possibilité de travailler exceptionnellement avec un ou plusieurs utilisateurs distants. Nous n'allons toutefois pas nous attarder sur l'aspect communication et conservation de la conscience sociale entre les collaborateurs distants. De nombreux travaux se focalisent déjà sur cette question, et le domaine a été largement étudié à travers les EVC traditionnels [PF04][IAHO10b][IAHO10a][PFD03].

Structure du document

Le présent document est divisé en cinq chapitres.

D'abord nous établirons un état de l'art concernant les différents sujets liés à notre problématique. Cette problématique étant concernée par une large gamme de domaines de recherche, l'état de l'art sera scindé en deux parties, chacune suivi d'une analyse.

Dans un premier chapitre, nous commencerons par étudier l'existant à travers les principaux systèmes collaboratifs disponibles, pour se focaliser ensuite sur les travaux traitant la coopération sous l'angle d'une équipe d'utilisateurs colocalisés. Cet état de l'art constituera une observation approfondie des différents facteurs humains qui se distinguent dans ce type de configuration et des conséquences de ces différents facteurs sur le déroulement de l'activité.

Le deuxième chapitre de ce document sera constitué de notre travail d'analyse, qui s'appuiera sur les données et observations de l'état de l'art. L'objectif de cette section est d'établir toutes les caractéristiques propres à la collaboration en situation de coprésence. Cela nous permettra alors de définir précisément en quoi ce type de collaboration est à distinguer des propositions existantes liées à la collaboration distante. Les observations et conclusions qui sont explicitées dans ce chapitre nous permettront alors d'identifier les caractéristiques à prendre en compte pour réaliser une plateforme spécialisée pour les utilisateurs colocalisés. Ceci constituera une première approche de la formulation de notre proposition

Le troisième chapitre reviendra plus en détail sur les solutions de collaboration distante, mais cette fois en se penchant sur la conception logicielle de ces plateformes. L'objectif de cette analyse est d'identifier dans quelle mesure nous pouvons utiliser et adapter les technologies existantes pour réaliser une plateforme qui illustre nos concepts en répondant efficacement aux besoins définis dans le chapitre précédent.

La quatrième partie de cet exposé constituera une analyse des travaux présentés dans le troisième chapitre, et nous permettra de formuler notre proposition sous l'angle de ses caractéristiques logicielles et de son architecture interne. Nous verrons ici dans quelle mesure nous pouvons nous réapproprier les outils existants dans l'état de l'art pour répondre à notre problématique.

La cinquième et dernière partie de ce document permettra enfin de valider tous les concepts évoqués dans cette introduction à travers le descriptif détaillé de deux applications prototypes que nous avons réalisées. Ces deux démonstrateurs placeront notre proposition dans un contexte

d'utilisation réel et donneront ainsi une dimension concrète aux propositions formulées dans les chapitres précédents.

Nous terminerons alors par une conclusion, qui constituera un tremplin pour exposer les idées pouvant naître à partir de nos travaux, et apportera quelques éléments de réponse pour chacune d'elles.

INTERACTION COLLABORATIVE COLOCALISÉE : ÉTAT DE L'ART

Sommaire

1.1	Le travail collaboratif assisté par ordinateur	10
1.1.1	Les systèmes collaboratifs	10
1.1.1.1	TCAO : définition	10
1.1.1.2	Taxonomie du TCAO	11
1.1.1.3	Collaboration synchrone distante	13
1.1.2	Technologies et concepts associés aux environnements collaboratifs . . .	19
1.1.2.1	La réalité virtuelle et les environnements virtuels collaboratifs	20
1.1.2.2	Réalité Virtuelle immersive/non immersive	21
1.2	L'homme au centre d'une collaboration colocalisée	23
1.2.1	L'espace : une notion complexe	24
1.2.1.1	La notion d'espace virtuel dans une application interactive . .	24
1.2.1.2	De l'espace physique à l'espace virtuel : la particularité du travail colocalisé	25
1.2.1.3	"Space" vs "place" : une approche sémantique de l'espace . . .	27
1.2.1.4	L'appropriation de l'espace en milieu colocalisé	29
1.2.2	La collaboration colocalisée : war room et communication de groupe . .	35
1.2.2.1	La communication de groupe en coprésence	36
1.2.2.2	War room : caractéristiques	38
1.2.2.3	Observations de l'utilisation des war room informatisées	41
1.2.2.4	La collaboration de groupe à groupe : notion de présence mixte	45
1.3	Exemples de plateformes prenant en charge la colocalisation	49
1.3.1	Affichage unique	49
1.3.2	Affichage partagé	50
1.3.3	Multi-dispositifs	51
1.3.4	Le multi-vues dans le cadre d'une collaboration distante	53
1.3.5	Conclusion	55

Les domaines de recherche concernés par notre problématique sont multiples. Nous devons nous pencher à la fois sur les systèmes collaboratifs existants, mais également sur les situations de collaboration qui ne font pas nécessairement appel à un quelconque support informatique. Nous constaterons que si les coopérations distantes et colocalisées sont relativement proches en apparence, elles possèdent aussi de nombreuses différences qui imposent de les considérer séparément. La collaboration distante représente toutefois l'axe qui a été le plus largement exploré par les dernières innovations scientifiques et technologiques ; elle constitue par conséquent une source importante d'informations qu'il convient d'étudier. Cette partie de l'état de l'art sera ainsi divisée selon trois axes.

Nous commencerons par faire le point sur les outils de support à la collaboration à travers leur historique, leur terminologie, et leurs concepts.

La deuxième partie de cet état de l'art traitera de l'homme au centre de la collaboration colocalisée. Cette section présentera plusieurs travaux permettant de mieux comprendre le comportement des utilisateurs et la manière dont ils perçoivent leur environnement de travail.

La troisième et dernière partie évoquera les différentes plateformes existantes traitant la collaboration sous l'angle d'une équipe colocalisée. Nous verrons dans quelle mesure les installations matérielles servant de support au travail collaboratif peuvent répondre aux problématiques évoquées dans la deuxième section.

Après avoir fait le point sur ces différents aspects, nous analyserons en détail dans le deuxième chapitre de ce document les différents phénomènes sociaux et cognitifs qui prennent place dans ce type de configuration. Nous extrairons ainsi les caractéristiques nécessaires à la réalisation d'une plateforme informatisée dédiée aux situations de coprésence.

1.1 Le travail collaboratif assisté par ordinateur

1.1.1 Les systèmes collaboratifs

Cette thèse s'inscrivant dans le domaine de l'interaction collaborative, il est nécessaire avant tout d'identifier les concepts fondamentaux de ce domaine. Cette section va donc rappeler les notions principales du travail collaboratif assisté par ordinateur et établir un historique des différentes solutions proposées dans ce domaine.

1.1.1.1 TCAO : définition

Le terme travail collaboratif assisté par ordinateur (TCAO), traduction de CSCW (computer supported collaborative work) est apparu dans la deuxième moitié des années 80 par Irene Greif [Gru94]. A cette époque, le TCAO est devenu un domaine de recherche à part entière. Il est difficile de définir avec précision cette notion, tant elle s'avère pluridisciplinaire. Il est d'ailleurs souvent fait l'amalgame entre le TCAO et la notion de collecticiel (groupware en anglais) [Lév90]. Les types d'application concernés par ce domaine de recherche couvrent des activités très variées, pouvant traiter de la communication, de la conception industrielle, de l'enseignement ou même de la maintenance. Cependant, malgré cette diversité, ces applications sont liées par des caractéristiques communes, tant en terme d'interaction que de communication. Il est donc possible de

	MEME LIEU	LIEU DIFFERENT
MEME INSTANT	-Salle de réunion	-Vidéoconférence -Audioconférence -Messagerie instantanée -Environnements virtuels collaboratifs
DIFFERE	-Tableau blanc -post-it	-E-mail / Memo -Forum de discussion -PLM (Product Life Management)

FIGURE 1.1 – Matrice de classification spatio temporelle.

les considérer dans leur ensemble à travers l'étude du TCAO.

Wilson [Gru88] propose de définir le TCAO comme étant "un terme générique, qui combine la compréhension de la façon dont les gens travaillent en groupe à travers les technologies réseau informatiques, ainsi que les techniques matérielles et logicielles qui lui sont associées". Il s'agit donc de l'étude de la collaboration supportée par une solution logicielle, en l'abordant sous l'angle de ses effets psychologiques, sociaux, et organisationnels, ainsi que sous l'angle des moyens techniques matériels et logiciels qu'elle nécessite. Ellis et al. [EGR91] proposent une définition du terme collecticiel, qu'ils décrivent comme étant un "Système informatique qui assiste un groupe de personnes engagées dans une tâche commune (ou but commun) et qui fournit une interface à un environnement partagé" (traduction : Karsenty [Kar94]). Le terme collecticiel est donc employé pour définir les outils logiciels de support au TCAO.

1.1.1.2 Taxonomie du TCAO

• Matrice spatiotemporelle

Les collecticiels peuvent être classifiés de différentes manières. La méthode la plus courante est introduite par Johansen en 1988 [Joh88], reprise en 1995 par Baecker [BGBG95]. Cette classification consiste à placer les applications dans une matrice spatiotemporelle à deux dimensions (fig. 1.1).

Les deux colonnes traitent de la notion d'espace, c'est-à-dire qu'elles vont différencier les cas où les utilisateurs sont physiquement réunis au même endroit, et les cas où ces utilisateurs sont considérés comme éloignés, quelle que soit la distance (espace entre un bureau et un autre, ou même plusieurs milliers de kilomètres). La notion d'éloignement est ici assimilable à une distance

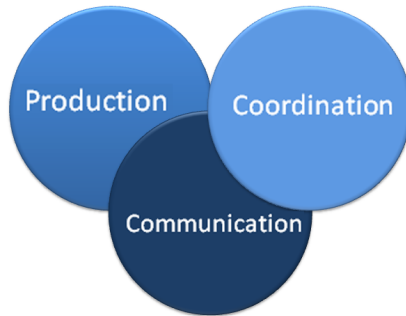


FIGURE 1.2 – Le modèle du trèfle des collecticiels

suffisante pour provoquer une perte du contact visuel direct. La seconde dimension correspond au temps, qui va définir la nature des échanges entre les utilisateurs. Ces lignes vont ainsi différencier la collaboration dite synchrone (tous les utilisateurs travaillent ensemble en même temps et les informations sont échangées instantanément) de la collaboration asynchrone (les actions sont différées, les étapes de la séquence de travail sont espacées dans le temps).

Bien qu'elle soit aujourd'hui encore la classification la plus courante, cette taxonomie sous forme de matrice a suscité quelques critiques. Elle a donc été le support de plusieurs propositions visant à la rendre plus complète. Tarpin-Bernard et al. [DB96, F97] ont cherché à définir plus clairement la notion de synchronisme. Ils définissent ainsi :

- **la coopération asynchrone** : les participants interagissent et échangent des données à des moments différents.
- **la coopération en session** : les participants travaillent simultanément, mais ne partagent pas leurs données. Chacun interagit sur ses documents de façon autonome.
- **la coopération en réunion** : les participants travaillent, communiquent et partagent leurs données de façon simultanée. Les interventions de chacun sont toutefois organisées selon un système de "tour de parole".
- **la coopération étroite** : elle est analogue à la notion de coopération en réunion, à la différence qu'il n'y a pas de notion de tour de parole. L'interaction, la communication et le partage des données se font en temps réel.

• Trèfle des collecticiels

Il s'agit d'un modèle permettant de représenter de manière conceptuelle les différentes facettes d'une application collaborative. Il fut en premier lieu proposé par Ellis et al. [EW94], puis complété par Salber et al. [SC95]. Comme son nom l'indique, ce modèle est composé de 3 parties (fig. 1.2) :

- **l'espace de communication**, qui est composé de la communication (formelle ou non) entre les utilisateurs du système. Cet espace ne comprend que la communication explicite, c'est-à-dire les échanges délibérés tels que la messagerie, la communication orale ou encore la visiophonie.
- **l'espace de production**, qui regroupe les données partagées sur lesquelles repose la collaboration, ainsi que les interactions qui ont lieu sur ces données (création, modification, accès en lecture). Cette facette correspond notamment à l'activité de conception.
- **l'espace de coordination**, quant à lui, correspond à toutes les contraintes de méthodologie de travail multi-utilisateur. C'est cet espace qui regroupe les droits des utilisateurs, leur hiérarchisation, leurs rôles respectifs, ainsi que l'ordonnancement des tâches. Dans une certaine mesure, l'espace de coordination peut être couvert par l'espace de communication (l'organisation du travail peut être définie directement par les utilisateurs entre eux). Néanmoins, l'existence d'outils permettant d'alléger cette tâche de coordination est utile à la collaboration.

1.1.1.3 Collaboration synchrone distante

La demande en terme de collaboration synchrone distante (même instant, lieu différent) se faisant de plus en plus forte (que ce soit pour des tâches de réunion à distance, d'assistance technique ou des activités pédagogique), plusieurs solutions ont été proposées. Ces applications de collaboration distante peuvent apparaître sous différentes formes. D'abord, certains outils se sont focalisés sur l'aspect "communication inter-utilisateurs". C'est notamment le cas des applications basées sur la vidéo. D'autres solutions s'appuient au contraire essentiellement sur la notion de partage de documents ou d'interaction avec ces documents. Enfin, il existe certaines solutions mixtes, qui visent à enrichir la communication vidéo par le biais de certaines capacités d'interaction avec les documents partagés (ou l'inverse).

• Collaboration audiovisuelle

La visiophonie est un mode de collaboration qui propose à plusieurs personnes distantes de communiquer oralement tout en bénéficiant d'un support visuel. Ce support visuel, qui est une simple représentation vidéo de l'interlocuteur, permet de favoriser la compréhension mutuelle grâce à la transmission d'éléments de communication non-verbale. Ces éléments (regard, expression faciale, gestuelle) sont d'une importance capitale pour assurer une bonne compréhension mutuelle. Cette catégorie de facteurs déterminants sont associés à la notion de téléprésence [BHI93].

Les premières tentatives en matière de communication audio/vidéo datent de 1927 : AT&T, la compagnie de téléphone américaine, met en place la première transmission audio et vidéo établissant un lien entre New York et Washington [TB94]. Les années qui suivirent ont alors permis à la société de développer cet outil jusqu'à obtenir en 1963 le picturephone, premier terminal expérimental capable d'émettre et de recevoir un flux vidéo. Si les contraintes techniques et financières de ce prototype ont empêché sa généralisation, cette innovation fût le premier pas vers la mise en place de systèmes de vidéoconférence.



FIGURE 1.3 – Le mur de téléprésence de Orange Labs.

A la fin des années 70 sont apparues les premières solutions commerciales de vidéoconférence. Destinées aux entreprises, ces salles de téléconférence permettaient de procéder à une communication de groupe à groupe, en diffusant dans des salles dédiées (et équipées à cet effet) un flux vidéo issue d'un groupe de collaborateurs distants.

Aujourd'hui, les outils permettant la communication vidéo entre plusieurs utilisateurs se sont généralisés et sont même accessibles au grand public. Par exemple, les logiciels de messagerie instantanée tels que Windows Live Messenger ou Skype, permettent de lancer une communication vidéo entre deux pairs. La transmission vidéo temps réel est également possible en utilisant des terminaux mobiles tels que les téléphones. Elle est, par exemple, illustrée par l'application FaceTime sur l'iphone de la société Apple. D'une manière générale, les principales recherches liées à la communication vidéo se sont portées sur l'aspect technique. Ces travaux se sont donc essentiellement focalisés sur les moyens de transmettre un flux important de données à travers des lignes de communication à débit limité.

Certaines études ont cependant cherché à résoudre des problèmes de perception causés par la direction du regard : l'utilisation de ces systèmes empêche de bénéficier pleinement de cet élément de communication non-verbale. Il est en effet impossible de détecter que le regard de l'interlocuteur se pose sur une personne ou un objet en particulier. Ceci est dû à des contraintes matérielles (pour capter le regard, la caméra doit être placée face à l'utilisateur, l'empêchant ainsi de voir correctement son écran), mais aussi à des problèmes logiciels : en affichant plusieurs flux vidéos sur le même écran, il est difficile de déterminer si l'utilisateur est en train de regarder l'un ou l'autre de ces interlocuteurs. Des travaux tels que les hydra units [Bux92] ou le mur de téléprésence de Orange Labs (fig. 1.3) ont donc développé des outils cherchant à résoudre ces contraintes, via l'utilisation d'un écran semi transparent ou celle de plusieurs caméras et écrans, chacune étant associée à l'affichage d'un flux vidéo.

L'évolution croissante des débits des réseaux a également permis la naissance des media spaces. Ce terme a vu le jour dans les années 80 à travers les travaux de R. Stults [Stu86]. Ces travaux ont dans un premier temps visé à établir une liaison vidéo entre deux sites distants de manière persistante. La communication entre les utilisateurs n'était ainsi plus mise en place délibérément par ces derniers, mais était disponible à tout instant dans des espaces dédiés, dans lesquels étaient installés des écrans permettant la liaison audiovisuelle entre deux sites. La nuance avec la vidéoconférence classique réside avant tout dans le fait qu'il n'était plus nécessaire de mettre en place une organisation préalable avec les autres collaborateurs pour démarrer une session de communication, comme c'était le cas avec les systèmes classiques. Progressivement, cette proposition a évolué vers une mise à disposition de cet outil vidéo un peu partout dans les locaux, que ce soit dans les couloirs et autres lieux de passage [BHI93] ou même directement sur les postes de travail des utilisateurs : les bureaux ont ainsi été équipés de caméras, de micros, et de dispositifs d'affichage diffusant en temps réel et de façon permanente les images, ainsi que de solutions logicielles permettant de contrôler l'affichage des flux vidéos provenant de tous les postes de travail équipés. Le bénéfice produit par l'utilisation des media spaces réside dans la possibilité de disposer d'un espace virtuel persistant et commun à tous les collaborateurs, indifféremment de leur éloignement géographique. Cette technologie propose ainsi de simuler une colocalisation entre plusieurs personnes, qu'ils soient voisins de bureau ou dans des pays étrangers. Plusieurs solutions visent d'ailleurs à reproduire les phénomènes qui ont lieu dans les conditions réelles. C'est notamment le cas du service appelé "le coup d'oeil" de Cool et al. [CFKL92], qui recrée une situation correspondant à une personne jetant un bref regard à travers les portes entrouvertes des autres bureaux lorsqu'elle passe dans un couloir. De même, le service "awareness view" permet de combiner simultanément les flux issus de plusieurs bureaux, comme pour recréer la vue que l'on aurait d'un bureau en open space [BT91][LGS97]. Cette colocalisation simulée autorise ainsi les échanges informels entre les collaborateurs, de même qu'elle améliore leur coordination.

Si ces technologies de vidéoconférence et de média space permettent de faciliter les échanges entre les utilisateurs, elles proposent un support logiciel qui ne s'applique qu'à l'aspect "télé-présence". Les flux vidéos étant difficile à manipuler, il n'est pas toujours aisé d'en extraire des informations relatives à la facette "production" de l'activité. Les collaborateurs distants travaillant sur leurs postes de travail respectifs ne peuvent pas partager d'information, ni interagir sur un document commun. Les technologies de communication audiovisuelle présentées précédemment ne portent par conséquent que sur les facettes de communication et de coordination du trèfle des collecticiels. Nous verrons cependant dans la suite du document qu'elles peuvent être couplées avec des systèmes de partage de documents numériques.

Dans la continuité de ces travaux, plusieurs recherches ont cherché à développer une notion d'interaction, en enrichissant les applications vidéo avec le partage d'un espace de travail. C'est notamment le cas de videoDraw [TM90], de clearboard [IKG93], ou encore de doubleDigitalDesk [Wel93]. Ces installations logicielles et matérielles exploitent la diffusion vidéo pour transmettre l'image d'une surface de travail réelle. Chaque utilisateur dispose alors d'un support physique sur lequel il peut procéder à des annotations avec un simple feutre. La vidéo de la surface de dessin de son interlocuteur est projetée sur le support, permettant ainsi de superposer les deux annotations. Cependant il ne s'agit pas à proprement parler d'un document partagé, mais de la fusion de deux documents distants. En effet, les utilisateurs ne peuvent pas agir sur le "calque" physique de leur collaborateur, par conséquent le document fusionné n'a qu'une existence visuelle et aucun document (physique ou virtuel) combinant le travail de l'équipe ne peut être produit à l'issue de la collaboration. Le prototype teamWorkstation [IM91] a au contraire cherché à palier

ces limitations en ajoutant un support logiciel, qui constitue un troisième calque virtuel. Ce calque supplémentaire est ainsi partagé entre les deux postes et il est annotable via les périphériques d'interaction de la station de travail. Là encore, l'utilisation de deux calques tangibles pose des problèmes de consistance du document final produit. Ces solutions reposant sur la vidéo souffrent donc d'une certaine insuffisance vis-à-vis de la production de données.

• Partage de documents : prise en charge de la facette "production" du trèfle

Pour répondre aux besoins impliqués par la facette "production" du trèfle des collecticiels, plusieurs solutions furent apportées. D'abord, tous les systèmes d'exploitation actuels permettent désormais de partager des fichiers à travers le réseau, avec une gestion approfondie des droits accordés à chaque utilisateur. Cependant cette solution ne s'avère pas suffisante à des fins de collaboration, essentiellement en raison de la gestion des conflits d'accès à ces fichiers. En effet, pour préserver la cohérence du système, le partage de fichiers place des verrous empêchant plusieurs utilisateurs de modifier simultanément les données du même document. L'utilisation de cette solution est ainsi assimilable à une activité de collaboration asynchrone, dans laquelle les interactions de chacun doivent être ordonnancées de façon séquentielle. De même, l'enregistrement d'un document modifié depuis un poste de travail consiste à remplacer le fichier partagé par la copie locale modifiée, effaçant ainsi les modifications apportées par les autres.

Dans cette optique, les solutions de gestion de version (CVS, SubVersion, GIT...) permettent d'archiver les modifications de chaque collaborateur et elles offrent des outils pour de fusionner deux versions concurrentes d'un même document texte. Là encore, ces solutions ne permettent pas de collaboration synchrone, puisqu'elles reposent sur le principe d'une copie du document stockée sur la machine locale, sans processus de synchronisation temps-réel. Elles trouvent donc plutôt leur place dans le cadre d'activités de groupe à forte modularité telles que les projets de développement logiciel.

Pour répondre aux besoins de simultanéité des interactions, il a alors été proposé de partager non plus les documents, mais de partager directement l'interface permettant d'interagir avec ces documents. C'est ainsi que sont nées les technologies de virtual network computing (VNC), proposées de nos jours sous la forme de nombreuses solutions commerciales ou libres. Ces outils lient virtuellement les périphériques d'interaction et de visualisation d'un poste de travail au système exécuté sur une machine distante. L'ordinateur client du service de VNC agit comme un terminal permettant d'exécuter des actions sur le serveur distant, tout en ayant le même retour visuel que ce poste serveur. L'utilisateur client du service a donc, au même titre que l'utilisateur de la machine serveur, la main sur l'intégralité des actions réalisées sur l'interface. L'inconvénient majeur de cette solution logicielle réside dans la difficulté de coordination entre les utilisateurs, provoquée par l'absence de système de communication interpersonnelle. De même, il est impossible pour les utilisateurs de connaître exactement l'état de l'activité des autres. Par exemple, lorsque le curseur de la souris se déplace, les utilisateurs ne peuvent savoir lequel de leurs collaborateurs est à l'origine de ce déplacement. Les applications de VNC sont alors difficilement utilisables dans le cadre d'une collaboration synchrone. Pour cette raison, elles sont principalement utilisées pour des tâches de téléopération et d'assistance à distance telles qu'on peut le voir dans les sociétés de help desk. Dans ce secteur le service VNC est d'ailleurs le plus souvent utilisé conjointement avec un système de communication téléphonique.

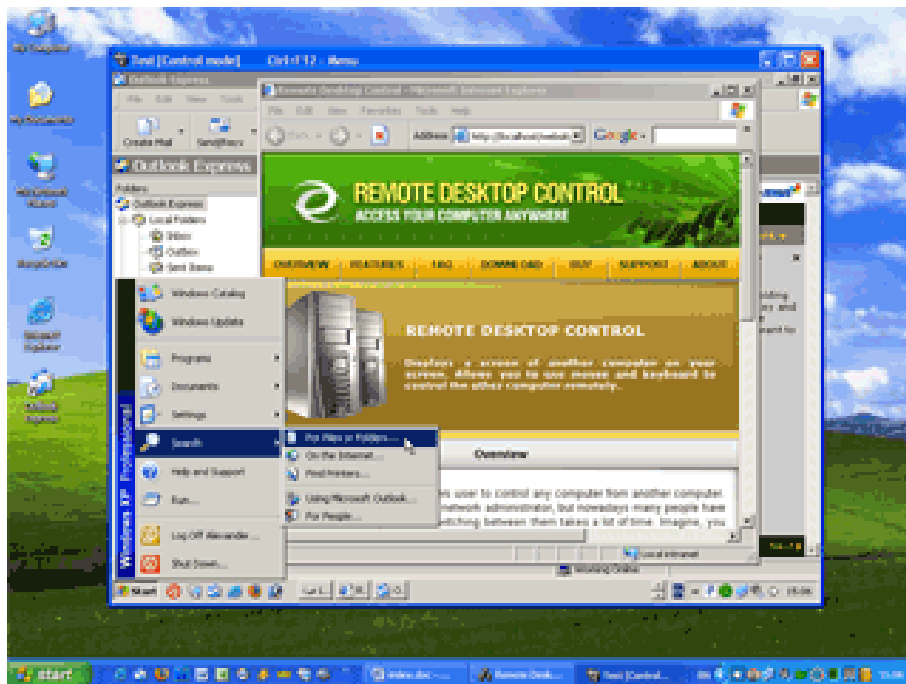


FIGURE 1.4 – Exemple d'utilisation d'un système VNC : le bureau de l'ordinateur distant apparaît dans une fenêtre (crédit photo : <http://remote-desktop-control.com>)

La solution proposée pour bénéficier à la fois de la souplesse d'utilisation des VNC et de la qualité de coordination issue de la vidéo consiste en l'utilisation d'une technologie appelée "desktop conferencing" [WSM⁺90]. Cette solution, illustrée entre autres par Microsoft Office Live Meeting, offre la possibilité de partager l'interface d'une application entre plusieurs utilisateurs distants, tout en maintenant le sentiment de coprésence grâce à un système de communication audiovisuel. Cet outil offre alors à plusieurs collaborateurs la possibilité de travailler sur le même document en se partageant les outils d'une application classique (logiciel de traitement de texte, tableur, outil de dessin, etc.). Selon cette configuration, chacun des utilisateurs est placé derrière son propre poste de travail, le logiciel d'édition du document n'étant exécuté que sur une seule machine. Cette unique instance de l'application voit alors son interface diffusée sur tous les postes distants, chacun pouvant prendre la main sur les outils de façon à interagir tout en étant vu de tout le monde.

De plus, les solutions de desktop conferencing mettent également à dispositions des outils supplémentaires utiles à la coordination, comme la capacité de travailler en privé sur une application, sans en partager l'interface. Les grands acteurs industriels ont largement adopté ces outils, comme nous pouvons le voir -entre autres- avec l'application smarteam de Dassault Systèmes, qui propose des outils de desktop conferencing partageant la plateforme de CAO "CATIA", du même éditeur.

Certains travaux proposent également un affichage de type WYSIWIS (acronyme de "What you see is what i see") "relâché" [SBF⁺87]. Ce type d'affichage, par opposition au WYSIWIS classique, permet aux utilisateurs de procéder à une réorganisation personnelle de l'espace de travail affiché, sans affecter la vue des autres utilisateurs. Cette possibilité n'est pas offerte avec un système de partage d'application classique.

Les applications partagées par le système de desktop conferencing n'étant que rarement conçues dans une optique multi-utilisateurs, plusieurs problèmes se posent vis-à-vis de la collaboration, et tout particulièrement de la coordination. Il devient alors nécessaire de séquencer les actions de chacun, au détriment de l'aspect synchrone de la coopération. Pour reprendre la taxonomie de Tarpin-Bernard [DB96, F97], ces outils se situent donc dans le cadre d'un synchronisme de type "coopération en réunion", les actions de chacun étant ordonnancées sous une forme proche de la notion de tour de parole. De plus, même lorsque l'application partagée supporte l'intervention simultanée de plusieurs utilisateurs, les solutions de ce type souffrent d'une contrainte technique majeure, liée d'une part à la latence d'une manipulation distante, et d'autre part à la surcharge de calcul du poste exécutant l'unique instance de l'application.

L'utilisation des systèmes de collaboration par partage montre ses limites même lorsqu'il est couplé à la vidéoconférence : Il est difficile pour les utilisateurs de percevoir naturellement l'activité des autres. S'il existe peu d'alternatives viables en terme de communication, il est envisageable d'enrichir les systèmes de desktop conferencing pour les rendre plus adaptés aux activités de groupe. C'est dans ce contexte qu'interviennent les environnements virtuels collaboratifs.

• La collaboration distante exploitant la réalité virtuelle

La dernière génération d'outils informatiques dédiés aux activités collaboratives d'un ensemble d'individus est représentée par les environnements virtuels collaboratifs (EVC, ou CVE, en anglais pour Collaborative Virtual Environment). Cette technologie consiste à établir virtuellement un espace de travail commun à tous les utilisateurs, le plus souvent sous la forme d'un environnement 3D.

Chaque participant de l'activité est invité à se connecter à cet environnement virtuel à partir d'un poste de travail individuel. L'environnement dans lequel l'utilisateur est immergé consiste en une représentation visuelle du projet sous la forme d'un ou plusieurs objets disposés dans l'espace artificiel. Pour procéder à l'accomplissement de leurs tâches, les participants disposent de différents outils d'interaction pour appliquer des modifications sur les objets virtuels grâce à leurs périphériques respectifs.

Ces espaces virtuels collaboratifs constituent un domaine de recherche ayant pris une ampleur considérable ces dernières années et de nombreuses plateformes ont vu le jour, chacune reposant sur des concepts et des objectifs distincts. Le troisième chapitre de ce document établira un état de l'art plus complet de ces technologies, et illustrera par des exemples les différentes approches proposées à ce jour.

• La collaboration colocalisée sous forme de war room

Dans notre environnement de travail quotidien, nous sommes régulièrement confrontés à des tâches nécessitant un travail d'équipe avec un ou plusieurs collaborateurs. La coopération qui en résulte est alors déterminante pour le bon déroulement du projet. Les outils disponibles pour collaborer à distance permettent déjà d'échanger des informations de façon simple, néanmoins le principe de réunion des personnes physiques reste, dans tous les secteurs industriels ou commerciaux, une étape indispensable à la réalisation d'un projet de grande ampleur. En effet, l'association de l'ensemble des collaborateurs dans une même pièce pour communiquer autour

du projet est reconnue comme un atout majeur pour coordonner les actions, procéder à une prise de décision, ou résoudre les problèmes [Gut98] [GG02] [BGG01]. Dans cette optique, nous assistons depuis quelques années à une généralisation du travail en *war room*.

La *war room*, comme son nom le laisse présager, tire ses origines du domaine militaire dans lequel elle constitue un espace physique et clos dédié à la planification et au commandement d'opérations stratégiques. Les officiers et opérateurs s'y réunissent pour décider des manoeuvres à effectuer en temps réel. La section policière de New York utilise par exemple ce type de méthodes pour les réunions de gestion de crise urbaines [nyp]. A l'heure actuelle, aucune traduction française n'a rencontré l'unanimité, même si le terme apparaît parfois sous la forme de "salle de commandement". Dans ce document, nous nous référerons par conséquent à ce concept à travers sa terminologie anglophone.

Selon la définition de Jason Leigh [RV02], dans le domaine industriel une *war room* (ou *project room*) est un espace de travail constitué d'une pièce unique équipée de différentes surfaces de travail, dans laquelle une équipe projet se réunit pour collaborer en temps réel. Les durées de l'activité y sont variables et elles sont souvent situées dans une fourchette allant de quelques jours à plusieurs mois. De plus, les surfaces de travail disponibles dans la salle sont diverses. Nous pouvons donc y voir apparaître des blocs-notes, feuilles volantes, tableaux blanc ou tableaux en liège (fig. 1.5). Ces différents supports offrent une grande persistance au travail produit, permettant ainsi à l'équipe de revenir à tout moment sur des éléments passés. A l'heure actuelle, les tâches de collaboration mises en oeuvre dans une *war room* se réfèrent le plus souvent à des activités de brainstorming, de management de cas d'urgence, de planification de lancement d'un nouveau produit ou de l'analyse globale d'importantes quantités de données. La collaboration en *war room* est à différencier de la simple réunion, d'abord par la durée des sessions de collaboration, mais aussi par le caractère complet des interactions de production qui y sont réalisées.

A la différence d'une situation de coopération classique, le travail en *war room* réunit les utilisateurs dans un même espace clos. Leur travail est par conséquent fortement centré autour de la communication entre les utilisateurs. Les activités de production et d'interaction ne sont alors pas indépendantes, mais découlent directement des échanges entre les participants. Dans le processus de travail traditionnel, les membres de l'équipe de conception organisent des meetings réguliers, à travers lesquels ils font le point et mettent en commun les parties du projet qu'ils ont développé et analysé individuellement. Au contraire, en situation de travail en *war room*, la notion d'individualité est réduite au minimum : les participants travaillent sur leurs tâches respectives de façon simultanée, en conservant un lien étroit avec le travail de leurs collaborateurs [GKEK04].

L'utilisation d'une *war room* représente par conséquent un degré extrême de collaboration colocalisée.

1.1.2 Technologies et concepts associés aux environnements collaboratifs

Cette section du document porte sur les environnement collaboratifs. L'objectif est ici d'identifier l'ensemble des concepts de base qui gravitent autour de la notion de réalité virtuelle.



FIGURE 1.5 – War room classique utilisant des supports papiers.

1.1.2.1 La réalité virtuelle et les environnements virtuels collaboratifs

L'usage du terme "réalité virtuelle" est attribuée à Jaron Lanier lors d'un salon professionnel dans la fin des années 80 (au Texpo'89 de San Francisco). Bien qu'il n'existe pas de définition formelle, nous pouvons le décrire comme un univers synthétique régi par des procédés informatiques. Philippe Fuchs propose par ailleurs de définir la réalité virtuelle ainsi : "La finalité de la réalité virtuelle est de permettre à une personne (ou à plusieurs) une activité sensori-motrice et cognitive dans un monde artificiel, créé numériquement, qui peut être imaginaire, symbolique ou une simulation de certains aspects du monde réel." [Fuc96]

Selon cette définition, un environnement virtuel ne fait donc pas nécessairement référence à un environnement tri-dimensionnel, mais peut également être représenté visuellement sous différentes formes (2D, ou même sous forme textuelle). La puissance croissante du matériel informatique permettant de calculer facilement des images 3D en temps réel, la tendance actuelle vise à généraliser ce mode de représentation, créant ainsi cette confusion. Nous prendrons donc soin dans cet exposé de préciser ce cas de figure par le terme "réalité virtuelle 3D".

Les environnements virtuels, quant à eux, peuvent être vus comme une mise en application pratique de la réalité virtuelle 3D. Ces environnements proposent d'immerger l'utilisateur dans cette réalité alternative, en lui permettant d'interagir avec cet univers. Cet outil ouvre de nombreuses possibilités que ce soit dans le but de reproduire notre réalité (comme c'est le cas pour les simulateurs) ou pour offrir une représentation artificielle assimilable à une construction mentale de données abstraites. Les environnements virtuels peuvent être conçus dans une optique mono-utilisateur, mais peuvent également permettre à plusieurs personnes de prendre place au

sein du même monde virtuel. C'est notamment le cas dans les jeux vidéos, qui proposent à plusieurs joueurs d'interagir dans le même environnement, que ce soit en situation de coprésence (via l'utilisation de consoles de salon par exemple) ou de manière distante (comme le proposent les jeux en ligne massivement multijoueurs). Il est intéressant de souligner que lorsque l'environnement de réalité virtuelle est accessible par plusieurs utilisateurs, cette situation de partage de l'environnement suppose implicitement que la perception visuelle de chaque participant est la même. Par exemple, un objet virtuel présenté sous forme d'un modèle 3D sera le plus souvent perçue par les autres à travers le même modèle 3D. Dans un tel contexte, la même scène virtuelle est donc partagée.

Les environnements virtuels collaboratifs (EVC) constituent une sous-catégorie qui connaît depuis plusieurs années un certain engouement de la part de la communauté scientifique. Cette technologie est, comme son nom l'indique, dédiée à des cas d'usages qui portent sur la notion de collaboration, c'est-à-dire impliquant un lien fort entre les interactions des différents utilisateurs. Il s'agit en fait d'une forme de réalité virtuelle exclusivement multi-utilisateurs, qui vise à permettre à plusieurs personnes de coopérer autour des mêmes objets virtuels. Les EVC marquent par conséquent une nuance importante avec les autres environnements, puisque pour optimiser la collaboration, ils doivent respecter certaines règles propres à ce domaine. Par exemple, il faut s'assurer que les utilisateurs aient à tout moment conscience de l'activité de leurs collaborateurs.

Les EVC possèdent donc une certaine complexité dans leur développement, dans la mesure où ils sont relatifs à plusieurs domaines de recherche. L'aspect interaction homme-machine (IHM) porte sur les méthodes de représentation de l'environnement ainsi que les techniques à mettre en oeuvre pour interagir avec celui-ci. L'aspect multi-utilisateurs, quant à lui, soulève des problématiques techniques (à travers les technologies de communication et de distribution des données), mais aussi des contraintes issues du TCAO telles que la communication entre les utilisateurs. De plus, la grande variété des cas d'usages (fig. 1.6) potentiels d'un environnement virtuel collaboratif (simulation d'entraînement aérospatial, applications pédagogiques, conception de systèmes mécaniques, visualisation de données complexes médicales ou physiques, revue de projet sur maquette virtuelle, etc) implique également de tenir compte des contraintes qui leur sont associées, ou de définir une plateforme suffisamment générique pour être facilement adaptable aux différentes situations. La conception d'un EVC tisse donc un lien étroit entre ces différents domaines.

1.1.2.2 Réalité Virtuelle immersive/non immersive

Les environnements virtuels collaboratifs peuvent être différenciés selon deux catégories : les systèmes immersifs et les systèmes non-immersifs.

Bien que la notion de réalité virtuelle implique de façon systématique cette notion d'immersion de l'utilisateur au sein de l'environnement artificiel, il faut en distinguer plusieurs degrés. Un environnement que nous qualifierons de pleinement immersif consiste à fournir à l'utilisateur un maximum de stimuli sensoriels (visuels, audio, et tactiles), de façon à lui procurer l'illusion qu'il est physiquement intégré dans l'environnement virtuel [Fuc06]. L'application gérant cet environnement doit alors être étroitement liée à l'utilisateur et à ses actions, de façon à lui envoyer en réponse des stimuli pertinents. Selon cette configuration, l'objectif est que l'utilisateur n'ait plus conscience de son environnement physique, qui peut constituer un frein à sa concentration.

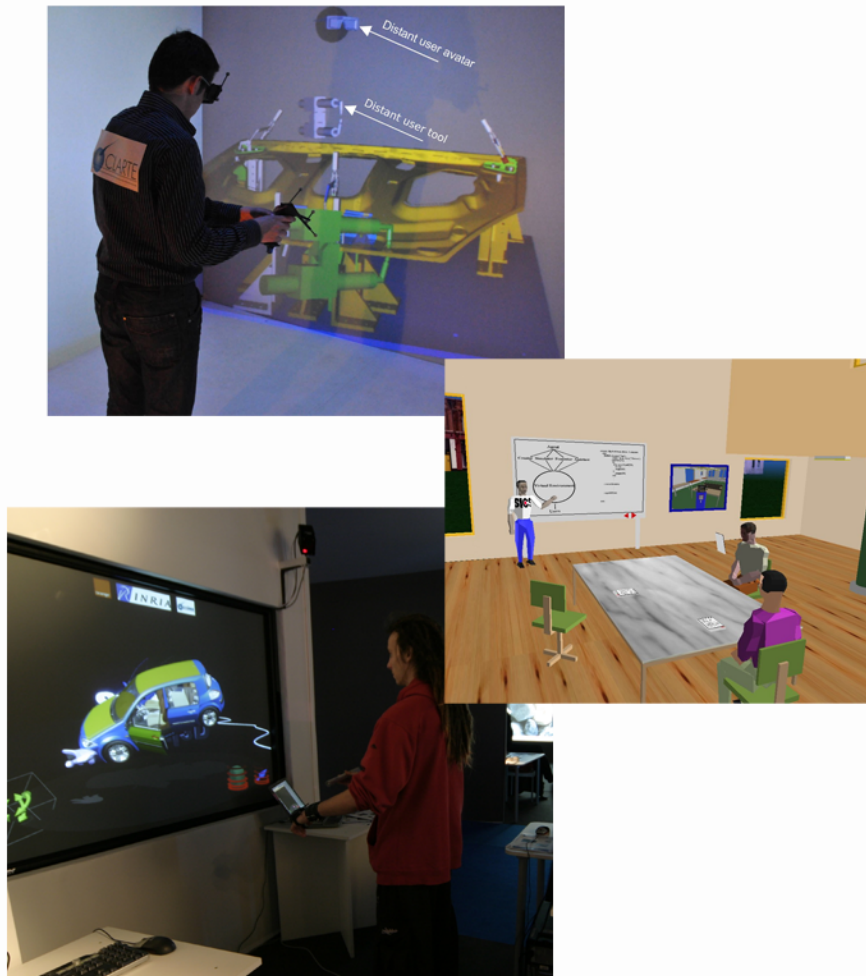


FIGURE 1.6 – Exemple de cas d'utilisation de la réalité virtuelle pour la collaboration. De haut en bas : OpenMask, Dive, SPIN|3D

Cette immersion totale nécessite des installations matérielles complexes, notamment sur le plan visuel, afin de minimiser la vision de l'environnement physique. Les casques de réalité virtuelle permettent à l'utilisateur de percevoir le monde virtuel selon une vue stéréoscopique qui occupe la totalité de son champ de vision. D'autres systèmes moins intrusifs ont également fait leur apparition : c'est le cas des installations type CAVE (Cave Automatic Virtual Environment) [CNSD⁺92b], qui consistent à placer l'utilisateur dans un cube complet ou partiel de plusieurs mètres, sur les faces duquel sont projetées les images de l'environnement virtuel (fig. 1.7). En terme de perception auditive, des systèmes de spatialisation sonore émettent des sons de telle sorte que la personne puisse percevoir son origine dans l'espace.

Outre les problèmes de coûts liés à ces installations matérielles, certains problèmes d'ordre physiologique peuvent survenir, tels que les troubles liés à la cinétose [DL92], ou "CyberSickness" (<http://www.cybersickness.org/>), c'est-à-dire une sensation de mal de mer provoquée par l'utilisation prolongée d'un dispositif mettant en conflit la perception visuelle et la perception sensorielle. En terme de collaboration, ce type de matériel s'avère plutôt dédié à un travail indi-



FIGURE 1.7 – Environnement immersif de type CAVE

viduel ou à une collaboration distante. En effet, les systèmes immersifs prennent difficilement en charge l'immersion de plusieurs utilisateurs simultanément : dans le cas d'un environnement de type CAVE par exemple, le positionnement spatial de l'utilisateur peut être détecté pour adapter la représentation visuelle, mais ce suivi n'est réalisé que sur un seul utilisateur à la fois.

Nous qualifierons, en opposition, de système "non immersif" une installation matérielle et logicielle permettant aux participants d'interagir avec l'environnement virtuel, tout en conservant une certaine conscience de leur environnement physique. Si cette conscience peut avoir un impact perturbateur dans le cadre d'une situation de collaboration exclusivement distante, elle peut cependant s'avérer indispensable lorsque plusieurs participants sont localisés dans le même environnement physique.

1.2 L'homme au centre d'une collaboration colocalisée

Dans le présent document, l'approche que nous avons du travail collaboratif repose essentiellement sur un positionnement selon lequel les différents intervenants sont situés (au moins en majeure partie) dans le même espace physique. Les tâches à accomplir par l'équipe d'utilisateurs restent les mêmes que dans les environnements d'interaction collaborative distante. Néanmoins, cette proximité physique entre les participants a un impact fort sur le déroulement de l'activité. Cette partie de l'état de l'art va nous aider à déterminer les différents facteurs humains qui interviennent et à distinguer précisément leurs influences sur le comportement du groupe. Cette mise en lumière des paramètres critiques d'une collaboration colocalisée nous permettra alors de définir les outils les plus pertinents pour rendre la communication la plus fluide et efficace possible.

La première partie de cette section fait le point sur la notion d'espace, qui révèle dans notre cas de figure une complexité supérieure par rapport aux environnements de collaboration distante.

1.2.1 L'espace : une notion complexe

1.2.1.1 La notion d'espace virtuel dans une application interactive

Avant d'aller plus loin dans l'état de l'art concernant la collaboration en situation de co-présence, il est nécessaire de faire le point sur la notion d'environnement virtuel. En effet, dans notre contexte d'utilisation, cet environnement est ici abordé de façon plus complète que pour une application classique.

Nous appelons environnement virtuel l'ensemble des objets numériques qui interviennent dans une activité disposant d'un support informatique. Il est donc assimilable à tout ce qui perceptible au travers de l'interface utilisateur d'une application, cette interface pouvant être distribuée sur plusieurs matériels. En d'autres termes, il s'agit de documents ou d'objets numériques, de représentations visuelles de ces objets, ainsi que d'outils permettant de modifier le contenu de ces documents virtuels. Ce qui est défini à travers le terme d'environnement virtuel représente par conséquent tout ce qui ne constitue pas un élément tangible de l'activité. Il peut donc être assimilé à un "espace" artificiel.

Les actions qui ont lieu dans cet espace, qu'elles soient de nature interactive (ie. la modification d'un document) ou de nature communicative (par exemple les échanges entre deux utilisateurs à travers cet environnement), peuvent alors être qualifiées de médiatisées, car passant impérativement par des moyens informatiques (par opposition au monde réel, dans lequel toutes ces actions sont directes).

Ainsi, tandis qu'une feuille de notes sur papier appartient à l'environnement de travail réel, une feuille similaire présentée via un procédé informatique sur l'écran d'un ordinateur appartient à l'environnement virtuel.

Selon une application classique (par exemple un logiciel tel qu'une suite bureautique de type Microsoft Office permettant de manipuler des documents textes et/ou des graphiques), l'environnement virtuel possède une signification sémantique très faible, puisqu'elle est assimilée à une simple surface de travail. Dans un tel contexte, l'ensemble de l'environnement perçu à travers l'interface utilisateur ne constitue qu'un support de visualisation et d'interaction, comme pourraient l'être des documents physiques sur papier.

Dés lors, si l'on se place dans le contexte de l'utilisation d'une telle application par un groupe d'utilisateurs en situation de collaboration, le support informatique proposé ne possède aucune caractéristique permettant de le distinguer d'un ensemble de documents papiers. Les utilisateurs n'exploitent cet espace virtuel qu'à des fins d'interaction, ou plus spécifiquement à des fins de production sur les documents, comme s'il s'agissait d'une surface d'écriture. Toute l'activité de collaboration liée aux échanges interpersonnels entre les participants est donc extérieure à l'espace virtuel. Le groupe d'utilisateur ne se projette pas au sein de cet environnement et une frontière nette est alors établie entre l'environnement réel, dans lequel se trouvent les participants, et l'environnement virtuel, dans lequel se trouvent les documents informatiques.

Dans le cadre d'une application reposant sur le principe de réalité virtuelle 3D (comme peuvent l'être les EVC), l'environnement virtuel prend une tout autre dimension. En effet, les utilisateurs impliqués dans l'utilisation d'un tel outil vont être amenés à se projeter mentalement dans cet espace artificiel. L'environnement virtuel prend alors une signification proche de celle de l'environnement réel. Il n'est ici plus assimilé à une simple surface de travail, ni à un simple outil, mais devient alors un espace complexe composé d'objets interactifs, qui est caractérisé par des

référentiels spatiaux et des capacités de navigation. Dans un tel contexte, l'utilisation que font les participants de l'environnement virtuel va au delà des activités d'interaction sur les objets. Les utilisateurs peuvent, en effet, se déplacer dans cet espace artificiel, échanger des informations avec leurs collaborateurs, modifier les objets, les déplacer, changer leur apparence, etc. D'une manière générale, nous pouvons dire que l'environnement virtuel tel qu'il est perçu ici fournit un support d'échange sémantique, qui ne repose pas uniquement sur le contenu des documents informatiques ou sur les facultés d'interaction sur ces documents.

Notre approche vise ainsi à définir la notion d'environnement virtuel non pas comme un simple support d'interaction, mais au contraire comme un espace de collaboration complet, au même titre qu'une salle de réunion dans une entreprise. La particularité de cette approche dans notre contexte, contrairement aux applications EVC classiques, réside dans le fait que cet espace abstrait entre dans une situation de cohabitation avec l'espace physique réel dans lequel se trouvent les utilisateurs. Il peut alors être considéré comme un espace complémentaire, c'est-à-dire une extension de l'espace physique de collaboration.

La notion d'espace étant prépondérante dans cette approche, nous allons maintenant détailler de façon plus concrète le rôle que joue l'environnement virtuel dans la collaboration, puis nous soulignerons la pluralité des environnements de travail dans le cadre d'une application de collaboration colocalisée.

1.2.1.2 De l'espace physique à l'espace virtuel : la particularité du travail colocalisé

L'espace est une notion qui a un impact déterminant sur les utilisateurs. Il détermine en effet la façon dont ces derniers peuvent s'appropriier les outils de travail ou le projet de la collaboration, mais il conditionne également la façon dont les différents participants vont communiquer ou interagir entre eux. Cet espace peut revêtir deux facettes :

- L'espace physique, dans lequel sont réunis les utilisateurs.
- L'espace virtuel, au sein duquel sont disposés les objets numériques.

L'influence de ces deux facettes de l'espace est très différente selon le contexte : lors de l'utilisation d'un environnement partagé sous forme de EVC distant, chaque utilisateur est placé derrière son propre poste de travail. La facette de l'espace qui monopolise l'attention de tous les utilisateurs est alors uniquement constituée de l'environnement virtuel. L'espace physique dans lequel se situe chaque participant devient par conséquent négligeable, puisqu'il ne fournit aucune fonction de communication interpersonnelle ni d'outils d'interaction. En effet, les liens qui unissent les participants entre eux (les systèmes de communication visuelle ou verbale médiatisés), de même que les liens qui relient les utilisateurs aux objets interactifs (les outils d'interaction) sont pris en charge par l'environnement virtuel. C'est pour cette raison que des artefacts visuels destinés au maintien de la conscience sociale sont disposés dans cet environnement (avatars, télépointeurs, etc).

L'approche de la notion d'espace à travers les EVC traditionnels se rapporte donc exclusivement à l'espace virtuel.

Au contraire, dans le cadre d'une plateforme de collaboration colocalisée bénéficiant d'un environnement virtuel, l'espace physique joue un rôle beaucoup plus important. Comme le souligne Michel Beaudouin-Lafon [BL04], dans le monde réel les utilisateurs interagissent avec leur ordinateur, mais aussi avec des objets physiques, leur environnement, et d'autres collaborateurs.

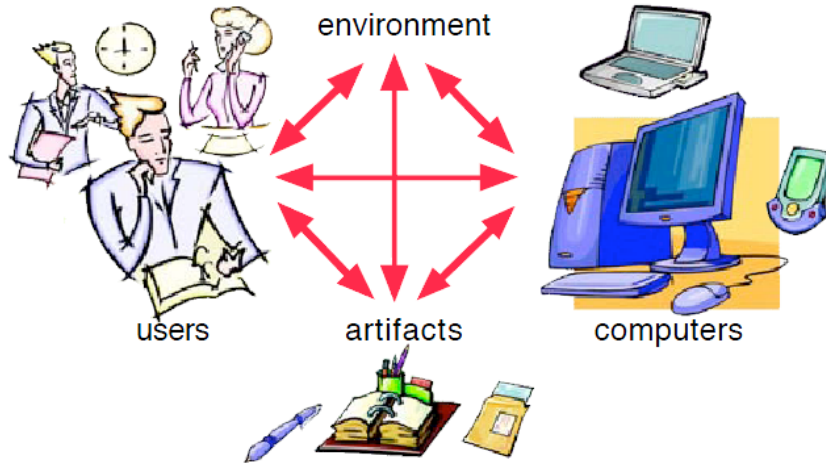


FIGURE 1.8 – Echanges interactifs entre les utilisateurs et leur environnement. (tiré de [3])

Cette diversité des interactions implique plusieurs caractéristiques physiques, sociales, organisationnelles et structurelles qui possèdent un impact fort sur l'activité d'un utilisateur (fig. 1.8).

Harrison et Dourish évoquent ainsi la notion d'espace hybride, qui est constitué d'un espace physique et d'un espace représenté par des moyens informatiques [HD96]. Les media spaces décrits plus haut (p. 14) dans ce manuscrit appartiennent à la catégorie des espaces hybrides, dans la mesure où l'environnement physique distant transmis à travers le flux vidéo est représenté par des moyens informatiques. Cette représentation et l'impossibilité d'interagir directement sur cet espace lui donne une dimension artificielle, lui conférant ainsi la qualification d'espace virtuel. Les rôles attribués aux deux facettes de l'espace sont donc différents dans le cadre d'une collaboration colocalisée. Les utilisateurs situés dans le même espace physique voient leurs relations apparaître sous deux formes :

- Les aspects relatifs à la communication interpersonnelle et à la coordination de l'équipe sont pris en charge à travers l'espace physique. Les participants peuvent se voir, se parler, exprimer leurs opinions ou montrer leurs émotions à travers leur gestuelle ou leurs expressions faciales. Cette communication de type face-à-face s'avère d'une qualité bien meilleure que la communication médiatisée, qui s'efface alors au profit d'échanges plus naturels.
- La seconde forme sous laquelle apparaissent les relations se situe au niveau de l'interaction. Les activités et les différentes tâches relatives à la production du travail de l'équipe passent par les outils fournis par l'environnement virtuel.

Cette délimitation des relations qu'ont les utilisateurs vis-à-vis de leur environnement de travail (technologique ou humain) n'est toutefois pas aussi nette. En effet, si les utilisateurs colocalisés passent majoritairement par le monde réel pour communiquer entre eux, l'application de réalité virtuelle joue malgré tout un rôle important. Au même titre que peut l'être un tableau blanc sur lequel les utilisateurs tracent des schémas, cet espace virtuel sert également de support à la communication, et les outils peuvent être utilisés à des fins de communication. Par exemple, le pointeur d'une souris peut être utilisé pour montrer une zone de l'environnement virtuel, de la même façon qu'un mouvement déictique (mouvement de pointage appartenant à la

communication non-verbale) aurait lieu dans le monde réel. De même, les activités de production, c'est-à-dire d'interaction avec le projet, ne sont pas exclusivement réalisables à travers l'application logicielle. Il est envisageable de voir les participants utiliser des artefacts physiques, en plus de leur équipement informatique. Par exemple, l'utilisation d'un tableau blanc peut permettre à l'équipe de réaliser des schémas, de prendre des notes ou de discuter d'un aspect du projet.

Certaines recherches ont abordé l'espace d'une façon singulière. C'est le cas de Harisson et Dourish [HD96], qui ont proposé une approche basée sur la perception sémantique de l'espace, qu'ils distinguent alors en "space" ou en "place".

1.2.1.3 "Space" vs "place" : une approche sémantique de l'espace

Si l'on observe l'espace physique de notre environnement de travail classique, nous constatons que cet espace est fortement conditionné par nos activités. Ainsi, les objets nécessaires au travail courant sont situés proches de nous (c'est le cas des claviers/souris d'ordinateur, ou des documents papiers relatifs au projet actuel), tandis que d'autres artefacts moins importants sont entreposés à une distance plus importante (par exemple, les livres sont disposés dans une armoire ou dans une bibliothèque séparée).

De même, l'espace physique à plus grande échelle est structuré par nos interactions avec nos collaborateurs : les bureaux de deux personnes ayant l'habitude de collaborer sont souvent proches et le degré d'ouverture de la porte peut servir d'outil de communication (ouverte ou fermée, en fonction du besoin d'indépendance dans la tâche)[HD96]. Nos méthodes de travail ont donc modelé l'espace physique qui nous entoure pour le rendre plus pertinent vis-à-vis de nos activités.

Harisson et Dourish ont procédé à une analyse en profondeur de cette notion d'espace [HD96]. Ils définissent ainsi une nuance qui divise l'espace en deux catégories : le "space" et le "place". Les environnements virtuels collaboratifs reposent sur le concept de "space" : comme nous l'avons évoqué plus haut (section 1.2.1.1), la réalité virtuelle consiste le plus souvent en la reconstruction d'un espace assimilable à notre monde réel. Cet espace est ainsi composé d'objets et/ou d'avatars disposés à différents endroits, et sont des représentations spatiales de la réalité. Cette approche vise à exploiter la perception innée que nous avons de l'espace tridimensionnel tel que nous le connaissons dans notre monde réel. En effet, plusieurs facteurs de l'espace physique sont réutilisés dans le "space" :

- **l'orientation et sa réciprocité entre les utilisateurs.** Même lorsqu'ils ont des points de vue différents sur l'environnement virtuel, certaines notions restent communes à tous les utilisateurs, en particulier le référentiel spatial. C'est par exemple le cas de la notion de haut et de bas. Ce référentiel spatial est alors vu comme un point de repère permettant de coordonner les utilisateurs et leurs actions.
- **la notion de proximité.** Comme dans la réalité, les utilisateurs sont plus à même d'interagir avec les objets qui leur sont proches dans l'espace virtuel. Ce constat permet aux utilisateurs d'analyser plus facilement les activités des autres participants. Observer un groupe de personnes autour d'une table permet par exemple de déduire de cette organisation spatiale qu'ils sont tous impliqués dans la réalisation d'une tâche commune. Si un utilisateur est à plus grande distance, cet indice constitue une preuve que son implication

dans cette tâche est moindre.

- **le partitionnement.** Comme pour la notion de proximité, le partitionnement de l'espace virtuel en différentes zones peut être associé au partitionnement du travail collaboratif en plusieurs sous-tâches.
- **la présence et la conscience sociale.** La représentation virtuelle des utilisateurs (à travers des avatars 3D par exemple) leur permet d'adapter leurs activités en fonction des indices visuels qu'ils perçoivent de l'avatar des autres.

Cette notion de "space" ne repose toutefois pas uniquement sur la perception visuelle, puisque les applications de type MUD (Multi-Users Dungeon) expriment aussi cette notion de spatialité à travers une interface en mode texte. Ces applications constituent ainsi leur environnement sur la manière dont le "space" structure notre perception et nos activités d'interaction. Harisson et Dourish marquent ainsi une différence entre ce "space", et une notion qu'ils ont baptisé le "place" : "space is the opportunity, place is the understood reality".

Le terme de "place" est alors défini non pas par l'espace perceptible dans l'environnement virtuel, mais par la perception sémantique qu'en ont les utilisateurs. Cette approche fait par exemple état que deux lieux ayant des propriétés spatiales similaires (une salle de concert et un amphithéâtre universitaire par exemple) n'émettent cependant pas les mêmes codes et reflètent donc une perception sémantique différente. Bien que les deux notions soient étroitement liées, le "place" est plutôt vu comme un phénomène culturel et social, alors que le "space" est une approche plus concrète basée sur la perception. Le "place" repose sur une compréhension de l'espace qui va au delà de l'agencement tri-dimensionnel des objets virtuels. Le phénomène d'appropriation d'un espace par un utilisateur repose ainsi sur sa capacité à percevoir le "space" sous la forme d'un "place". L'utilisation des systèmes de media spaces tels que celui de Bellcore [FKC90] est ici critiquable dans la mesure où les utilisateurs ne peuvent interagir librement sur l'espace affiché par la vidéo. Cette limitation dans l'interactivité de l'espace entraîne alors un échec de cette appropriation [HD96].

Selon Harisson et Dourish, la conception d'un environnement virtuel ne doit donc pas reposer systématiquement sur une représentation artificielle du monde réel, mais doit plutôt se focaliser sur une signification sémantique à donner à l'espace virtuel.

Cette analyse tire par ailleurs une conclusion intéressante vis-à-vis de notre approche : lorsqu'un environnement de travail fait cohabiter un espace physique et un espace virtuel, comme c'est le cas dans le cadre d'un environnement collaboratif colocalisé, il n'est pas nécessaire de reconstruire un environnement virtuel 3D basé sur des métaphores spatiales du monde réel. En effet, l'utilisation de ces métaphores vise le plus souvent à recréer une notion de "place", mais cet espace sémantique pourrait entrer en conflit avec le "place" émis par l'espace physique. Les solutions de collaboration mettant en oeuvre à la fois un espace physique réel et un univers virtuel peuvent donc proposer des modes de représentation virtuels plus abstraits. L'univers artificiel que nous pouvons proposer peut ainsi se permettre plus de liberté quant à la représentation des données, puisqu'il n'est pas nécessaire de s'appuyer sur une métaphore spatiale du monde réel.

1.2.1.4 L'appropriation de l'espace en milieu colocalisé

La complexité de la configuration spatiale d'une plateforme collaborative colocalisée soulève de nombreuses questions. La conception d'un environnement de type war room repose, en effet, sur des problématiques relatives à la configuration spatiale des équipements matériels, et sur la manière dont cette configuration influencera la collaboration. Les questions soulevées portent donc sur l'agencement spatial de l'ensemble : les équipements matériels servant de support au travail d'équipe doivent-ils respecter une configuration précise ? Dans quelles mesures les métaphores d'interactions et de visualisation, de même que la structuration spatiale de ces équipements impactent la collaboration ? Quels facteurs spatiaux de l'environnement virtuels sont à prendre en compte lorsque les utilisateurs exploitent le même dispositif simultanément ? D'une manière plus générale, quelles sont les lignes de conduite à adopter pour optimiser les facultés d'interactions des utilisateurs, sans pour autant opposer d'obstacle à leur communication ?

• Positionnement des surfaces de travail

Différentes études ont cherché à apporter des réponses vis-à-vis de la contrainte spatiale virtuelle et physique. C'est notamment le cas de Kiyokawa et al. [KBH⁺02], qui aborde la question en observant la collaboration de deux utilisateurs confrontés à une tâche à réaliser simultanément sur le même dispositif. La problématique de ces travaux repose sur le constat que même lors d'une collaboration en situation de coprésence, des utilisateurs placés devant un écran partagé voient leur attention focalisée sur cet affichage, ce qui impacte négativement la communication inter-utilisateurs en raison d'une rupture du contact visuel. Cette étude démontre ainsi que le placement de l'objet qui requiert l'attention de l'équipe (en l'occurrence, le dispositif d'affichage), entre les deux collaborateurs s'avère être une proposition pertinente. Le type de représentation adopté par le logiciel constitue un paramètre à prendre en compte : lorsque les données sont représentées sous forme d'une interface en 2D, les utilisateurs éprouvent un confort supérieur lorsqu'ils sont côte à côte devant une surface d'affichage verticale. La raison de ce constat est la similarité de leurs points de vue respectifs, qui rend leur perception plus uniforme. Cette affirmation n'est toutefois pas aussi catégorique lorsque le projet est affiché en 3D. En effet, la perception visuelle de l'environnement 3D que les individus ont dans le monde réel les a rendu coutumier du fait de ne pas avoir le même point de vue sur les objets. Un environnement virtuel 3D reprenant cette disparité des points de vue n'est par conséquent pas problématique.

Les travaux étudiant la collaboration sur une table tactile démontrent eux aussi que les participants sont sensibles à l'orientation des données lorsqu'ils travaillent dans le même espace physique [KCSG03]. Ainsi, les utilisateurs éprouvent un grand besoin de mobilité dans cet espace, notamment pour pouvoir se déplacer librement autour de la table et ainsi changer leur point de vue sur les données [HCS06][SI03]. Certaines propositions visent alors à rendre ce point de vue mobile par un procédé logiciel pour minimiser les déplacements physiques des utilisateurs, par exemple via un système de défilement circulaire des items virtuels autour de la table (fig. 1.9) [HCS06].

Le positionnement des utilisateurs par rapport à leur surface de travail est ainsi un élément important : la proximité des utilisateurs par rapport à un écran de grande taille, par exemple, influence des aspects tels que le confort d'utilisation, la facilité de mouvement de l'utilisateur, et la capacité à percevoir toutes les données affichées [HKR⁺05]. Cette proximité est d'ailleurs liée au mode d'interaction utilisé : une interaction directe telle que celle disponible sur un écran



FIGURE 1.9 – Défilement des objets virtuels autour de la table tactile. Tiré de [HCS06]

tactile impose à l'utilisateur d'être à une distance réduite du dispositif matériel, tandis qu'un procédé offrant une interaction à distance peut faciliter les mouvements. Les travaux de Huang et al. [HM03] révèlent par ailleurs que le positionnement et la taille des surfaces d'affichage utilisées par un groupe d'individus sont à l'origine de formes d'interactions et d'usages différents comparativement aux espaces de travail classiques.

Plusieurs études [Ken90][Tan91][Mar02] soulignent d'ailleurs qu'une activité est toujours associée à un lieu. La localisation des différents participants dans le lieu accueillant l'équipe fournit de façon implicite de nombreuses informations concernant le "qui fait quoi?". La position d'un utilisateur donne en effet une indication concernant la tâche qu'il est en train de réaliser. Ce facteur est reconnu comme étant un des points critiques pour le maintien d'une bonne coordination lors d'une tâche collaborative. Ce point a d'ailleurs longtemps été étudié dans le domaine de la collaboration distante, avec des fonctionnalités telles que les avatars ou les télépointeurs [GP02][LLO08][ARH⁺05].

Néanmoins, malgré le couplage fort entre positionnement et activité, il peut apparaître nécessaire de minimiser le besoin de mobilité des utilisateurs : s'il est pertinent de leur donner la liberté de mouvement nécessaire à la fluidité de la collaboration, il est aussi approprié de leur offrir la possibilité d'interagir librement sans être prisonniers de la localisation des dispositifs. Plusieurs recherches explorent donc la possibilité d'interagir à distance sur un ordinateur, notamment à travers des systèmes de télécommande virtuelle comme le frisbee de Khan et al. (fig. 1.10) [KFA⁺04].

- **Positionnement des utilisateurs entre eux**



FIGURE 1.10 – le frisbee de Khan et al.

Le positionnement des utilisateurs entre eux a également un impact important sur leur confort de communication et sur l'établissement d'une compréhension mutuelle [HKR⁺05][Tan91].

Les participants perçoivent leur espace physique comme étant partitionné en plusieurs zones, qui définissent le type de leurs interactions interpersonnelles [Hal66] : l'espace considéré comme intime est ainsi peu utilisé pour une interaction prolongée (Allen [All84] définit d'ailleurs que l'écartement nécessaire pour qualifier deux personnes de "distantes" est d'environ 30 mètres). Les utilisateurs préfèrent préserver la distance d'un bras, qui leur permet une communication aisée, tout en préservant leur espace personnel. Plusieurs travaux ont cependant révélé la nécessité d'une étroite proximité physique entre deux utilisateurs qui interagissent en commun sur la surface de travail [SK95] [Som69]. Cette configuration leur offre un point de vue similaire par rapport à l'interface, ce qui leur permet d'améliorer la coordination de leurs actions, ainsi que leur conscience de l'espace de travail.

D'une manière générale, la dynamique de communication repose en partie sur l'agencement spatial des utilisateurs dans la pièce. Les actions que chaque individu entreprend sont décidées et coordonnées en fonction de ce qu'il perçoit des positions de ses collaborateurs [Rog97]. Hutchins et al. [HP97] qualifient ce phénomène de cognition distribuée, qui repose sur le fait que les participants assimilent leur environnement social et physique comme une ressource nécessaire à l'accomplissement d'une tâche. Dans ce processus, des individus différents développent une approche générale commune, à travers l'ensemble des activités de la session collaborative [M.90][OW90].

Ces conclusions relatives au positionnement géographique des utilisateurs dénotent une grande diversité dans les configurations spatiales optimales vis-à-vis de l'espace physique. Le contexte de l'activité étant susceptible de changer au cours de la session de collaboration, de nouvelles

configurations spatiales peuvent devenir nécessaires. Il nous apparaît donc pertinent de prendre en compte le caractère évolutif de l'agencement spatial, en offrant aux utilisateurs une liberté de mouvements la plus importante possible dans notre espace de travail collaboratif, sans les contraindre à des positions fixes pour réaliser leurs interactions. Un état de l'art plus poussé concernant les comportements sociaux cognitifs des groupes d'individus colocalisés sera présenté dans la partie suivante (section 1.2.2).

• Positionnement des données dans l'espace virtuel

De même que pour l'espace physique, le phénomène d'appropriation de l'espace virtuel est un facteur critique à prendre en compte. Comme nous pouvons le voir dans les applications EVC purement distantes, l'agencement des éléments graphiques d'un environnement virtuel doit conserver une certaine cohérence, tout en permettant à chaque utilisateur de le personnaliser en fonction de ses besoins, ses préférences ou ses habitudes de travail. La mise en place d'environnements visuels de type WYSIWIS relâché (voir p.18) est une mesure visant à répondre à ce besoin. Si l'on étudie l'utilisation d'une même interface de travail par un groupe d'individus colocalisés, Scott et al. constatent un certain besoin de partitionnement de l'espace virtuel pour chaque utilisateur [SGM03].

Scott et al. préconisent ainsi de segmenter l'espace de travail en différentes zones, via l'utilisation simultanée de plusieurs dispositifs matériels, ou via la mise en place d'un partitionnement logiciel de l'interface de travail. Ce partitionnement logiciel peut par exemple intervenir pour distinguer les objets partagés et les objets propres à la tâche d'un utilisateur, introduisant une notion de travail individuel comme dans les travaux sur table tactile de Shen et al. [SLV⁺02], qui proposent de placer les objets partagés au centre de la table et les objets "privés" sur les cotés. D'une manière générale, nous pouvons constater une importante territorialité de la part des utilisateurs vis-à-vis de leur surface de travail [Sco05]. La conscience de ces territoires de la part des autres participants facilite la coordination et la collaboration. L'utilisation de ces espaces individuels s'avère dotée d'une certaine dynamique au cours de l'activité : les utilisateurs confrontés à un espace partagé de grande taille et un espace personnel sous forme de mini-interface ont tendance à se focaliser d'abord sur l'espace commun à l'équipe, pour se tourner vers leur interface individuelle au cours du processus de travail [HCS06]. Selon les termes de Thimbleby et al. [TAW90], la taxonomie des applications de travail de groupe doit marquer une nuance entre le travail personnel et le travail collaboratif.

Il est toutefois nécessaire d'établir un juste milieu dans cette individualité, dans la mesure où il est nécessaire de maintenir une certaine cohésion du groupe pour garantir des échanges efficaces entre les participants. En effet, les utilisateurs trop focalisés sur leur espace individuel souffrent d'une rupture du contact visuel (voire même oral) avec leurs collaborateurs. Cette rupture peut porter préjudice à la communication interpersonnelle, verbale ou non. Les individus confrontés à certains mouvements déictiques par exemple (le mouvements de pointage pour montrer un élément du projet) peuvent souffrir d'une surcharge cognitive due à la présence d'interfaces dupliquées au sein de l'environnement : les utilisateurs n'ayant pas assimilé la spatialisation des données selon le même repère, ils ne distinguent pas naturellement à quel élément de l'interface le mouvement de pointage est dédié [BOO95][Bly88][Tan91][Gut98].

Outre un partitionnement séparant les espaces de travail individuels et collectifs, les travaux

de Grudin [Gru01] évaluant l'utilisation de plusieurs surfaces d'affichage montrent que la structuration spatiale des informations en fonction de leur type est nécessaire au confort d'utilisation ressenti par les individus. Cette constatation est d'ailleurs également vérifiée lors de l'analyse du processus de travail individuel : lorsque l'on observe l'arrangement spatial des données produites par un utilisateur seul travaillant sur un tableau blanc, il apparaît que l'ensemble de la tâche à réaliser est segmentée en plusieurs types de données (notes de rappel, idées, listes de sous-tâches). L'individu répartit ces ensembles de données sur la surface de travail de façon ordonnée, dans le but d'organiser son travail [Myn99]. Ce comportement illustre le rôle de la spatialisation des informations en terme de construction mentale du problème à résoudre.

• **Positionnement des données virtuelles dans l'espace physique : le cas du multi-dispositifs**

Il existe par ailleurs un lien étroit entre l'espace physique et l'espace de travail (en l'occurrence l'espace virtuel). Terrell et al. [TM06] met par exemple l'accent sur l'utilisation conjointe de l'environnement virtuel et de l'espace physique, en proposant un artefact tangible pour localiser géographiquement les collaborateurs (fig. 1.11). De nombreux travaux étudiant la collaboration colocalisée s'intéressent d'ailleurs à la spatialisation physique des différents éléments de l'environnement virtuel. En effet, lors de sessions collaboratives impliquant plusieurs surfaces de travail (physiques ou informatiques), la question du positionnement spatial des données du projet est prépondérante. Conformément aux instructions données par Scott et al. [SGM03] visant à partitionner l'espace de travail grâce à plusieurs installations matérielles, il s'avère judicieux de diviser cet espace de la même façon que pour le planning d'un projet, c'est-à-dire en l'organisant sous forme de sous-tâches suffisamment indépendantes pour pouvoir être considérées individuellement. La nécessité de segmenter l'ensemble de la tâche à réaliser en plusieurs sous-tâches étant démontrée, la question est alors de savoir comment distribuer les données relatives à ces segments dans l'espace physique [TLGF09]. Les travaux de Jonathan Grudin relatifs à l'utilisation de plusieurs moniteurs sur un ordinateur mettent en avant le fait que l'utilisation de plusieurs supports d'affichage distincts permet une meilleure structuration spatiale des données, aidant au traitement de tâches multiples [Gru01].

La construction mentale d'un problème à partir de la répartition des données sur un écran se prête par conséquent tout aussi bien à la répartition sur plusieurs dispositifs. En effet, Jason Leigh [RV02] souligne que les participants mis en situation de collaboration dans une même pièce impliquant plusieurs surfaces telles que des tableaux blancs, des blocs-notes ou des tableaux en liège sont plus à même de faire appel à leur mémoire visuelle. Les utilisateurs procèdent ainsi à une mémorisation spatiale des données, leur permettant d'associer chaque donnée du projet à une localisation dans la pièce. Gloria Mark [Mar02] appuie ce résultat en évoquant la spatialité des références : lorsqu'une information a été affichée sur l'un des supports disponibles et qu'elle a été assimilée par les participants, cette information prend une consistance plus significative. Les utilisateurs vont alors instinctivement se référer à cette emplacement par des gestes déictiques ou des regards lorsqu'ils tiendront une discussion en rapport avec cette information. Ce lien entre positionnement géographique et donnée est tellement étroit que le référencement de l'endroit s'avère indépendant de la persistance de l'information. En effet, même si l'information n'est plus visible sur le support (si elle a été effacée ou remplacée), les utilisateurs continueront à référencer ce même emplacement lors de leurs discussions.



FIGURE 1.11 – Le périphérique physique de localisation proposé par Terrell et al.

Les recherches réalisées par Teasley et al. illustrent toutefois la nécessité de disposer d'une certaine dynamique dans la répartition spatiale des données [TCKO00]. Dans le cadre d'une activité de collaboration, les utilisateurs ont éprouvé le besoin de pouvoir basculer l'affichage de certaines informations sur un autre support de plus grande taille pour pouvoir en débattre plus confortablement avec le reste du groupe. La flexibilité de la distribution est ainsi un paramètre à prendre en compte pour s'adapter aux besoins en constante évolution des utilisateurs et de la tâche. Tang et al. démontrent d'ailleurs que la localisation spatiale des données et leur contexte d'utilisation sont extrêmement liés. Ce lien s'applique non seulement aux données, mais aussi aux modes de représentation ou aux modalités d'interaction [TLGF09][Sco05]. En effet, le contexte d'utilisation des informations est directement dépendant de l'individu, de ses compétences, de ses besoins, et de ses objectifs.

Dans la mesure où cette répartition spatiale possède une influence importante, les utilisateurs sont amenés à adapter cette organisation en fonction de leurs besoins : le travail d'Isenberg relatif à l'analyse visuelle de l'information en environnement fermé met en évidence que les participants ont à plusieurs reprises procédé à une réorganisation des différents artefacts dans la pièce [ITC08]. L'espace de travail et l'organisation des données dans cet espace ont ainsi été modélisés en fonction de la tâche à accomplir à cet instant, mais aussi en fonction des besoins des utilisateurs en terme d'espace.

Les travaux traitant de l'interaction de type "table centric" (littéralement : centré sur la table) consistent en la mise en place d'un espace collaboratif colocalisé proposant plusieurs dispositifs d'affichage, tous contrôlés par la table tactile centrale [ESRF06][WSFB06]. Ces travaux ont mis en évidence la configuration dynamique de l'espace [ESRF06]. L'une des raisons évoquées pour expliquer ce phénomène réside dans l'habitude qu'ont les utilisateurs de travailler sur des supports papiers mobiles (feuilles volantes ou bloc-notes) dans leur environnement de travail quotidien.

Certains travaux d'interaction "table centric" ont conservé la notion de référence spatiale des

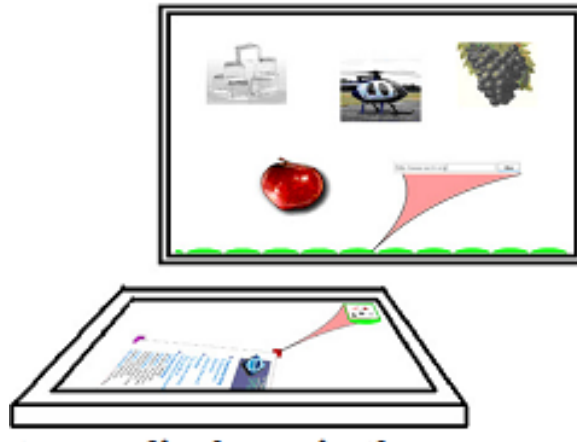


FIGURE 1.12 – Système de flèche de Wigdor et al. pour l’affichage d’un document sur plusieurs dispositifs.

données, en implémentant un système visuel ayant pour fonction de localiser les données de façon indépendante de leur surface d’affichage : ainsi, Wigdor et al. [WSFB06] proposent d’afficher une flèche reliant un item original situé sur la table à sa représentation distante située sur l’un des écrans verticaux de la pièce (fig. 1.12).

Les travaux présentés dans cette section ont mis en évidence la complexité de l’espace, tant au niveau de l’appropriation qu’en font les utilisateurs, qu’au niveau de la structuration des informations. Les recherches que nous venons d’observer illustrent l’importance capitale de la distribution de l’activité dans l’espace : la compréhension sémantique des informations relatives à la collaboration repose sur la distribution spatiale de celles-ci au sein de l’environnement, ainsi que sur la dynamique de cette distribution.

L’activité gagne à avoir l’ensemble des données être distribué dans l’espace, quelle que soit leur nature (données produites par l’interaction, et données servant de support à la discussion). La question de la duplication des données est alors soulevée : certains cas d’utilisations précautionneront de limiter l’affichage de chaque information à une seule instance pour simplifier son assimilation spatiale, tandis que l’existence de plusieurs représentations simultanées de la même donnée peut faciliter l’interaction multi-utilisateurs.

1.2.2 La collaboration colocalisée : war room et communication de groupe

Cette section fait le point sur les comportements d’un ensemble de collaborateurs en situation de coprésence. La question soulevée ici est de déterminer les indices comportementaux relatifs à la mise en place d’une dynamique de groupe lors d’une activité collaborative. D’autres travaux viendront ensuite illustrer la cohabitation de cette dynamique colocalisée et d’une collaboration distante, sous la forme d’activités de travail selon une configuration de type "groupe à groupe".

Pour proposer une plateforme collaborative colocalisée qui répond aux besoins des utilisateurs de façon pertinente, il est indispensable de connaître leur comportement en tant que groupe. Comme pour la notion d’espace, la notion de groupe revêt une forme complexe : en effet, le comportement cognitif et social des utilisateurs doit être analysé à la fois sous l’angle d’un ensemble d’individus (le groupe), mais aussi sous l’angle du comportement individuel de chacun.

De plus, ces deux notions impliquent l'existence d'une forme hybride, qui considère la totalité du groupe sous la forme d'un ensemble de sous-groupes. Les travaux que nous allons présenter ici se focalisent essentiellement sur ce dernier aspect, à travers l'étude de l'activité collaborative d'un groupe d'utilisateurs colocalisés. Nous verrons qu'au delà du simple fait de réunir physiquement les participants, la coopération en coprésence doit s'appuyer sur des méthodes de travail adaptées.

1.2.2.1 La communication de groupe en coprésence

Comme le souligne Gloria Mark [MA04] et comme nous l'avons extrait dans l'ensemble de la section 1.2.1, le partage d'un même environnement physique par plusieurs membres d'une équipe a pour effet d'engendrer des conditions favorables à l'inter-subjectivité (également appelé "représentation commune"). Ce terme, issu d'une théorie de la communication, définit le fait pour les individus d'un groupe de percevoir les choses de façon mutuelle et réciproque. Il illustre par conséquent la capacité de prendre en considération la perception des collaborateurs [Sch70]. L'inter-subjectivité constitue donc une forme de sens-commun, qui sert de référentiel permettant aux utilisateurs d'interagir et de communiquer naturellement selon les mêmes perspectives. La mise en relation de plusieurs collaborateurs dans un même espace physique a donc été le support à de nombreuses études relatives à la communication de groupe. A travers leurs travaux, Mark et al. [MA04][MAN03][Mar02] ont ainsi mis en évidence cette inter-subjectivité en constatant que les collaborateurs réunis dans la même pièce ont tendance à adopter massivement le même vocabulaire pour communiquer autour de leur projet. La terminologie commune engendre ainsi une meilleure compréhension mutuelle. Ces mêmes travaux ont également illustré que l'adoption de ce vocabulaire commun est plus délicate lorsque les utilisateurs sont mis en relation à travers une communication médiatisée, en raison de la rupture dans la proximité géographique. Baltes [BDS⁺02] illustre d'ailleurs que les groupes en situation de coprésence s'avèrent plus efficaces et plus rapides dans leurs prises de décision comparativement aux groupes constitués d'utilisateurs distants. Kraut et al. [Krrb90] distingue 4 catégories d'interactions entre les utilisateurs colocalisés :

- **les rencontres planifiées**, telles que les réunions arrangées à l'avance
- **les interactions intentionnelles**, qui sont déclenchées par le désir d'un seul interlocuteur
- **les interactions opportunistes**, qui sont, comme les interactions intentionnelles, voulues par un seul interlocuteur. La différence ici est que les interactions opportunistes ne sont pas instantanées, mais n'interviennent que lorsque l'occasion se présente (rencontre dans un couloir par exemple)
- **les interactions spontanées**, qui ne reposent sur aucune planification et ne sont décidées à l'avance par aucun des interlocuteurs.

Pour appuyer l'importance des interactions entre collaborateurs lors d'un travail en équipe, Whittaker [WFDJ94] souligne par ailleurs que 31% du temps d'un travailleur est passé en interactions informelles avec ses collaborateurs. Isenberg et al. [ITC08] ont analysé la coopération d'un petit groupe de participants dans une configuration de coprésence. Ces travaux dénotent une certaine dualité entre l'aspect individuel et le groupe. Ils reposent essentiellement sur la constatation que bien qu'ils soient intégrés au sein d'un groupe, les participants gardent une autonomie forte, et possèdent des méthodes de travail, des approches et des besoins qui leurs

sont propres. Cet aspect potentiellement conflictuel entre les besoins et objectifs des participants est également appuyé par C. Gutwin [GG98].

L'analyse révèle par conséquent qu'une plateforme de collaboration doit supporter plusieurs facteurs antagonistes en terme de temporalité des séquences de travail : les individus confrontés à un projet ont tous des méthodes d'analyse qui sont construites à partir de leurs propres expériences, succès et échecs. De même, les différentes spécialités et domaines d'expertises propres à chaque participant impliquent des méthodes de travail uniques. Cette approche unique à chacun entraîne notamment des disparités dans l'ordonnancement et la durée des sessions de travail individuel. Isenberg [ITC08] suggère ainsi qu'une plateforme collaborative doit être non-restrictive vis-à-vis de la disparité des méthodes de travail des différents participants. L'application doit pouvoir permettre de s'engager dans une tâche individuelle ou revenir dans une activité collective à tout moment sans perturber le travail des autres utilisateurs. L'ensemble de la méthodologie de l'équipe est flexible et la plateforme doit supporter naturellement cette flexibilité. Les travaux d'Isenberg soulignent un second point critique de l'individualité des membres d'un groupe colocalisé : les changements de stratégie qui ont lieu durant la session de travail -passage dynamique d'une activité de groupe à une activité individuelle- nécessitent d'être supportés de façon fluide au sein de l'application. Ainsi, les outils de visualisation de l'information doivent être conçus pour supporter à la fois les approches individuelles des différents utilisateurs, et les approches collectives de l'ensemble de l'équipe. Il est à noter que malgré cette disparité des systèmes de visualisation et d'interaction, la diffusion des conséquences des actions de chacun doit être gérée de façon subtile. En effet, les interactions réalisées lors d'un travail sur une vue individuelle doivent avoir une portée suffisamment importante pour conserver la cohérence de l'ensemble du projet, mais ne doivent pas pour autant perturber l'activité des autres utilisateurs.

Tang et al. [Tan91] ont procédé à une analyse détaillée de la collaboration de groupes en situation de coprésence. Leurs travaux sont issus de l'observation de huit groupes de trois ou quatre personnes partageant une même surface de dessin (un chevalet ou un tableau). Les participants devaient travailler sur une tâche de conception d'une télécommande universelle. Plusieurs résultats ont été tirés de ces observations :

- la gestuelle des utilisateurs est une ressource importante pour communiquer des informations.
- lorsqu'un utilisateur dessine une forme sur la surface de travail, le processus de création du dessin constitue une source d'information plus importante que le dessin final. En effet, le tracé d'un dessin dans ce contexte collaboratif est accompagné d'explications orales et de gestes. Le dessin final issu de ce processus a d'ailleurs peu de sens lorsqu'il est sorti de son contexte de communication. Les dessins réalisés sur le tableau ne servent donc que rarement d'espace de stockage d'information.
- la surface de dessin possède un rôle central dans la médiation de la collaboration, notamment pour la distribution des tours de parole des utilisateurs. Par exemple, un utilisateur qui écrit sur le tableau peut orienter ce dernier vers l'un de ses collaborateurs pour exprimer son désir de le voir intervenir.
- la surface de travail tient en fait un rôle double. D'une part, elle est un espace de production et de conception. D'autre part, elle est le point de convergence de l'attention des

participants. Elle sert ainsi de support visuel à la discussion de l'équipe.

Ces travaux mettent toutefois en évidence que dans un environnement comprenant un seul support d'affichage, la spatialité des données n'est pas toujours significative. En effet, certaines interactions sont référencées comme étant productrices de données non spatialisées. Par exemple des notes et des memos peuvent être placés n'importe où dans l'espace sans pour autant que leur position n'ait une signification particulière.

En terme de communication non-verbale, l'aspect gestuel détient un rôle majeur dans les échanges inter-utilisateurs. En effet, il apparaît au cours de la collaboration une certaine tendance au mimétisme. Les participants reproduisent des gestes déjà réalisés par leurs collaborateurs pour référencer des informations similaires. Cette adoption d'une sémantique gestuelle commune est à rapprocher de l'adoption d'une terminologie comme évoquée par Mark [MA04].

Les travaux de Tang référencent également une catégorie d'actions qu'il appelle "express idea". Ces actions peuvent être verbales, mais peuvent aussi être des mouvements gestuels ou des interactions avec la surface de travail (tracer un schéma par exemple). Le but de ces actions est de rendre tangible des concepts abstraits pour les rendre plus facilement perceptibles par les collaborateurs. Elles constituent donc un appel à réagir, à montrer son approbation ou au contraire à exprimer un certain désaccord.

D'un point de vue spatial, l'utilisation de l'expression gestuelle est directement concernée par l'organisation des éléments autour de l'utilisateur (support de travail et collaborateurs)[Tan91]. Tang met également en avant une certaine difficulté liée à la dynamique de collaboration colocalisée : les participants doivent se montrer capables d'interagir, de percevoir les interactions des autres et de rester à tout moment à l'écoute des canaux de communication (verbale ou gestuelle). Dans la plupart des applications courantes, les interactions sont d'ailleurs très séquentialisées. En effet, pour dessiner ou pour écrire du texte, il est nécessaire de changer le mode d'interaction. Dans la collaboration sur un tableau blanc, au contraire, ces interactions sont liées au même mode d'utilisation (un tracé au crayon) et ne constituent pas une séquence d'actions.

De même, un participant peut remplir plusieurs rôles simultanément : lorsqu'il a la main sur la surface d'affichage, il a un rôle d'interacteur avec les données, mais il remplit également la tâche d'exprimer une idée, ainsi que de capturer l'attention générale. Une plateforme collaborative, pour conserver une bonne fluidité dans l'activité, doit répondre à ce besoin de simultanéité des actions.

1.2.2.2 War room : caractéristiques

L'efficacité d'une équipe travaillant dans une war room bénéficie d'un gain important par rapport aux conditions de collaboration plus courantes : de nombreux travaux analysant les configurations de type war room mettent ainsi en évidence une productivité doublée [JOR⁺98][OCR⁺98][TCKO00].

Pour expliquer cet important gain de performances, Leigh [RV02] avance quatre caractéristiques majeures dans les war rooms :

- **la persistance des informations** : les données du projet apparaissent sur les différents supports et sont par conséquent disponibles durant toute la session de travail. Les participants peuvent donc facilement les consulter, mais aussi les éditer ou même les déplacer

dans la pièce.

- **les références spatiales et déictiques** : le placement des données dans l'espace influe sur la mémorisation des informations et des espaces par les participants. Ils savent donc à tout moment où trouver l'information dont ils ont besoin. L'intérêt du déploiement de l'information sur les supports de travail est aussi appuyé par Garcia et al. [GKEK04].
- **la conscience du travail des autres** : les participants sont dans une situation de proximité physique propice au maintien du contact visuel. Ils ont donc à tout moment la capacité de savoir qui fait quoi.
- **l'accès immédiat aux connaissances de chacun** : de la même manière que pour la conscience de l'activité des autres, la situation de coprésence permet aux participants d'interroger leur collaborateur instantanément. La formation de sous-groupes peut avoir lieu, permettant de regrouper les différents domaines d'expertise nécessaires à la résolution d'un problème précis.

Une constatation générale réalisée dans l'ensemble des travaux analytiques des war rooms porte sur la communication inter-utilisateurs. Il s'avère que la collaboration d'utilisateurs réunis dans un tel environnement de travail repose essentiellement sur des activités de brainstorming. La particularité du brainstorming en théorie de la communication est qu'il s'agit d'une tâche basée sur la rapidité des échanges. Stefik et al. soulignent d'ailleurs que dans un environnement collaboratif colocalisé, les activités de brainstorming ne doivent pas être entravées par les notions de turn taking (i.e. la gestion des tours de parole) telles qu'on peut les observer dans une collaboration médiatisée [Par03]. L'utilisation d'un espace de type war room se prête également particulièrement bien aux activités de codesign, qui sont caractérisées par un problème nécessitant un important travail de négociation et d'articulation entre les participants qui sont impliqués dans la réalisation du projet [Car00]. En effet, la collaboration sur un travail de codesign couvre de vastes domaines d'expertise : Herbsleb et al. [HMFG00] corrobore d'ailleurs l'importance de la diversité des compétences dans un travail collaboratif, et la nécessité de faciliter la transmission de ce savoir dans les collectifs. La résolution de ce type d'activité ne suit pas une progression ordonnée entre l'objectif de départ et le design final. Par exemple, les utilisateurs sont susceptibles de revenir sur des problèmes ayant déjà été source de discussion, de modifier leur proposition ou de l'invalidier. La résolution du projet suit cependant un processus itératif, basé sur la résolution de problèmes simultanés [MAN03]. La mise en place d'une collaboration en situation de war room répond efficacement à ces exigences puisqu'elle fournit aux utilisateurs une conscience permanente de l'état courant du projet, tout en fournissant un accès instantané aux autres collaborateurs pour négocier, analyser le projet ou discuter des potentielles divergences d'opinion [Rob97][SD02].

Garcia et al. [GKEK04] indiquent dans leurs travaux que la dynamique de communication des environnements de collaboration extrême (terme souvent employé pour qualifier le travail qui a lieu dans une configuration de type war room) reposent sur deux caractéristiques :

- Premièrement, les membres de l'équipe peuvent improviser des mini-réunions sous la forme de sous-groupes pour répondre de manière instantanée à la survenue d'un problème imprévu. Les sous-groupes impliqués dans ce type de réunions exceptionnelles sont variables en fonction de la situation à traiter. Par conséquent, l'organisation interne des individus

au sein de l'équipe se modifie dynamiquement au fur et à mesure de l'avancée du projet. Cette création interactive de sous-groupes au sein de l'équipe est constatée dans tous les travaux d'analyse du travail en war room.

- La conscience de l'activité des autres collaborateurs évoquée plus haut (p. 39) implique une forme de connexion sociale entre les utilisateurs, assimilable à une tâche de fond. Par conséquent, les participants peuvent s'impliquer spontanément dans le travail des collaborateurs pour intervenir lors de la détection d'un risque de mauvaise prise de décision. De même, cette connexion fonctionne de manière réciproque : un participant peut à un instant donné adapter son activité pour inclure un résultat intermédiaire provenant de l'interaction de ses collaborateurs. La multiplicité des supports d'interaction impacte l'activité selon une dynamique similaire : le partage permanent d'informations relatives au projet peut induire une analyse à la volée des résultats intermédiaires, ou peut amener les participants à procéder à une simulation pour étudier précisément dans quelle mesure ce résultat intermédiaire impacte les tâches qui lui sont associées.

L'étude analytique des war rooms utilisées dans l'U.S. NAVY révèle plusieurs aspects significatifs du processus de travail [CHSH00]. L'activité qui a lieu dans les NCW (Network Centric Warfare) constitue un effort intense et très focalisé, permettant d'améliorer la compréhension d'un problème complexe, en facilitant la visualisation et l'assimilation des données, ainsi que la compréhension des liens qui relient ces données. Trois points cruciaux sont énoncés pour caractériser une war room :

- la vitesse de prise de décision.
- la vitesse de communication entre les participants.
- la vitesse d'implémentation des décisions prises.

Cette méthode de travail permet à une équipe de résoudre des problèmes complexes sur des masses importantes d'informations, en les scindant en plusieurs parties compréhensibles. Cette segmentation favorise ainsi la structuration du dialogue et le brainstorming. La résolution du problème est ici organisée selon 5 phases itératives :

- la formation d'une équipe dont les membres présentent des domaines d'expertise variés, et la formulation de la question centrale du projet.
- la collecte des données : rapports, livres, articles ou toute autre forme de document physique ou électronique
- l'analyse des données collectées et le début de la négociation, via la formulation des opinions de chaque participant. C'est dans cette phase que sont définies les sous-tâches et les problèmes à résoudre au cours de l'investigation. Ces sous-tâches sont alors répertoriées sur un tableau, qui sera édité progressivement pour ajouter les solutions prises pour répondre à chaque problème.
- la revue des solutions, pendant laquelle les participants sont amenés à débattre de la pertinence des solutions proposées. Durant cette étape, des utilisateurs extérieurs à l'équipe sont susceptibles d'être consultés.
- la constitution des révisions qui doivent être réalisées. Ces modifications sont directement

retranscrites sur le tableau des tâches, via la suppression, la modification ou l'ajout d'un point clé. Il peut arriver que la discussion remette en question le bien fondé de certains thèmes définis au début. L'activité est alors reconstruite à partir d'un nouveau tableau pour relancer une nouvelle session de collaboration.

Aujourd'hui, l'usage des war rooms tend à se généraliser dans les grandes structures (General Motors, NASA, etc.).

L'utilisation d'une war room doit toutefois rester dans un contexte pertinent : la war room doit être conçue pour les projets dans lesquels les différentes tâches à réaliser sont hautement interdépendantes, mais aussi lorsque les relations entre les participants sont dynamiques [Mar02]. En effet, si les tâches à réaliser sont suffisamment indépendantes, la présence des autres utilisateurs dans la pièce et l'agitation que cela entraîne risquent d'impacter négativement l'activité d'un individu.

1.2.2.3 Observations de l'utilisation des war room informatisées

Cette partie étudie les recherches qui se sont penchées sur la collaboration dans des environnements de type war room "informatisées". S'il existe encore peu de plateformes logicielles conçues pour ce type de configuration, de nombreux travaux analytiques ont toutefois eu lieu en regard du comportement socio-cognitif des utilisateurs en situation de collaboration extrême. Les travaux que nous étudions ici ne bénéficient pas d'une plateforme dédiée assimilable aux EVC, mais disposent cependant d'un support informatique. Le support informatique est alors utilisé différemment d'un exemple à l'autre, illustrant les différentes stratégies envisageables.

• L'approche multi-dispositifs pour une utilisation séquentielle

Garcia et al. [GKEK04] ont étudié les méthodes de travail à mettre en place sur une tâche de conception d'un projet. Outre leur proposition d'ontologie d'un projet qui vise à structurer l'activité sous la forme de plusieurs sous-tâches incrémentales, l'expérience qu'ils ont réalisée met également en application l'utilisation d'un espace de type war room disposant d'un support informatique.

Dans cette mise en application, une équipe pluridisciplinaire de six participants est amenée à travailler durant une journée sur une activité de conception d'un bâtiment. La configuration matérielle de leur espace de travail repose sur la mise à disposition de plusieurs postes de travail et d'écrans de grande taille permettant de diffuser de façon continue les différentes données relatives au projet. Les outils logiciels offrent alors la possibilité de partager des informations telles que des feuilles de notes, des diaporamas, ou des fichiers de conception de type CAO (le package logiciel est constitué d'un ensemble de logiciel propriétaires : Word, Excel, Autodesk ATD, etc.). La plateforme logicielle ne présente donc pas de particularité spécifique à la colocalisation en terme d'interaction, mais confronte les utilisateurs à un environnement informatisé multi-dispositifs pour réaliser un travail dont les sous-tâches ont été séquencées.

Ces travaux présentent dans leurs résultats une analyse de l'impact de la situation de colocalisation sur l'activité globale du groupe de collaborateurs. L'expérience met ainsi en évidence une grande différence dans le contenu des échanges inter-utilisateurs : selon une méthodologie de travail traditionnelle (travail individuel et réunions), les événements de prise de décision représentent moins de 5% des échanges. La raison en est que, lors des réunions, les utilisateurs

passent un temps important à définir le contexte des données qui sont évoquées pour articuler la communication. En outre, la configuration des salles de réunion classiques est peu propice aux initiatives individuelles, et les participants laissent souvent leurs collaborateurs décider pour eux. Dans l'évaluation réalisée en war room, les événements de prise de décision ont constitué 21% du temps total. En effet, le support visuel interactif disponible sur les surfaces de travail permet de réduire au minimum les échanges destinés à situer le contexte de la conversation. De la même manière, les interactions de négociation entre les participants sont passées de 15% à 5%, grâce au phénomène d'inter-subjectivité.

Nous déduisons donc de ces résultats qu'une activité de collaboration colocalisée reposant sur un support informatique n'a pas d'influence néfaste sur la fluidité de la communication propre à la war room. En effet, les postes de travail proposés ici n'ont pas provoqué d'isolement des utilisateurs dans leur activité respective et ont rempli correctement leur rôle de support à la collaboration.

• La X-team

Dans ses travaux sur les war rooms, Gloria Mark [Mar02] (cf. p. 36) appuie son argumentation sur la disponibilité permanente de l'information. Cette disponibilité est permise non-seulement grâce aux supports d'affichage disposés dans la pièce, mais aussi par la facilité d'accès aux autres utilisateurs. En effet, la diversité des compétences disponibles auprès de chaque participant constitue une importante ressource d'informations sur le projet. Par exemple, Williams et al. [WK00] démontrent que deux développeurs travaillant côte à côte sont plus productifs et font moins d'erreur que s'ils sont dans des environnements distincts.

Gloria Mark a donc observé une équipe de collaborateurs habitués au travail en war room, la X-team. Cette équipe est constituée d'un ensemble de 16 consultants de la NASA qui procèdent à la conception de tous les aspects d'une mission spatiale. Chaque membre de l'équipe possède une spécialité qui lui est propre et l'ensemble des compétences est distribué entre chacun.

L'environnement de travail de la X-team est constitué de plusieurs grands dispositifs d'affichage fournissant chacun une information différente (fig. 1.13) : gestion des coûts, de la température, affichage de la structure mécanique, prévisualisation orbitale, archives des anciennes missions, ou tout autre paramètre susceptible d'influencer la conception de la mission. La présence de ces différents supports spécialisés permet aux utilisateurs d'assurer la surveillance de l'ensemble du projet à travers toutes ses facettes simultanément. Ils disposent également de postes de travail individuels pour consulter les documents et produire des feuilles de notes.

En terme de structure logicielle, les postes sont tous reliés en réseau et un système de souscription/publication des feuilles de notes est disponible, assurant la communication des informations entre les postes de chaque utilisateur.

Les résultats de cette expérience mettent en avant le fait que l'efficacité de cette équipe en terme de planification est importante (5% de marge d'erreur sur les estimations de coût d'une mission) et ils se sont avérés capables de concevoir l'intégralité d'une mission d'exploration spatiale en 3 sessions de 3 heures, ce qui est évidemment très peu.

L'analyse d'une session de collaboration de la X-team montre une très importante mobilité



FIGURE 1.13 – L'environnement de travail de la X-Team, dont les écrans affichent les différents sous-systèmes à gérer pour le projet.[Mar02]

des participants : au cours de la session, les utilisateurs se déplacent continuellement dans la pièce, pour aller discuter face-à-face avec leurs collaborateurs pour leur fournir ou leur demander des informations. Chaque participant est à tout moment sur le qui-vive, prêt à intervenir sur les problèmes potentiels.

Lorsqu'une contrainte ou une incohérence imprévue survient, les utilisateurs concernés se déplacent immédiatement vers le lieu où est détecté le problème, pour former un sous-groupe autour de l'écran affichant le problème. Lorsqu'un tel sous-groupe se forme, les autres participants peuvent décider de s'y joindre, ou de rester sur place en échangeant des informations verbalement à distance. L'ensemble de la collaboration est ainsi orchestré par les déplacements des membres de l'équipe.

L'observation de l'activité met également en évidence une importante multiplicité des ressources d'informations. Ces ressources disponibles pour chaque individu sont ainsi référencées selon plusieurs types :

- le leader de l'équipe.
- les sous-groupes d'utilisateurs formés pendant la session.
- les conversations publiques impliquant la totalité du groupe.
- les surfaces d'affichage publiques (en l'occurrence, les grands écrans).
- les documents issus du travail de chaque individu (ici, sous la forme des feuilles de notes partagées).
- l'écran personnel regroupant les ensembles de feuilles de notes auxquelles l'utilisateur a souscrit.
- l'écran personnel du voisin, qui fournit d'importantes informations sur son activité.

Les travaux de Mark mettent également en évidence une forme d'écoute propre aux war room. Il s'avère en effet que tous les membres de l'équipe ont développé la faculté de conserver une partie de leur attention auditive sur l'ensemble des conversations qui ont lieu dans la pièce. Chaque participant peut donc se livrer à son activité personnelle (travail individuel, ou discussion avec un ensemble de collaborateurs) tout en écoutant les dialogues environnants, pour maintenir la capacité d'intervenir à tout moment. Ce comportement singulier porte le nom de "phénomène de cocktail party" [Che53] : dans un environnement bruyant, les individus focalisent leur écoute sur la recherche de certains mots clés. Dans la war room de la X-team, les participants restaient donc attentif à la perception de mots associés à leur domaine de compétence. La spécialisation des utilisateurs conditionne ainsi leur comportement et leur approche de la discussion.

Dans cette optique, la dynamique de groupe repose essentiellement sur les domaines d'expertise respectifs des membres de l'équipe. Ainsi, les participants procèdent à la construction mentale de ce qu'ils qualifient de "carte des interdépendances". Cette carte est assimilable à un graphe dans lequel chaque noeud est représenté par une compétence (et implicitement, par l'individu possédant ces compétences). Les noeuds de ce graphe sont alors liés par la dépendance des domaines entre eux. Grâce à cette construction mentale, les participants sont en mesure d'identifier immédiatement les collaborateurs concernés par un point du projet susceptibles d'apporter une solution aux problèmes. Le réseau d'informations ainsi construit n'est cependant pas statique ; il peut évoluer en fonction de la nature des problèmes rencontrés.

La construction de ces cartes d'interdépendance permet ainsi de réduire la charge cognitive en limitant pour chaque cas la densité du réseau d'intervenants potentiels.

L'étude de la X-team mettent en évidence deux points cruciaux pour notre problématique : il apparaît que la plateforme de collaboration doit mettre l'accent sur la diversité des compétences des utilisateurs. Cette diversité constitue une forme d'intelligence distribuée à assimiler dans son ensemble, mais la plateforme doit également permettre à chacun d'exploiter au mieux ses aptitudes individuelles et son approche personnelle du projet. Le second point que l'on peut extraire est que, dans le cas de la X-team, les compétences sont symbolisées dans l'application à travers l'utilisation de plusieurs supports distincts. Cependant, pour conserver la cohérence de l'ensemble des domaines d'expertise, les participants doivent à tout moment suivre l'évolution des autres tâches. Le nombre de paramètres étant très important, le support logiciel leur permet d'élaguer la quantité d'information suivies (à travers le système de souscription/publication des feuilles de notes). Néanmoins, la plateforme logicielle présentée ici n'offre pas d'outil permettant d'assurer automatiquement la cohérence de l'ensemble. Pour une équipe de collaborateurs moins expérimentés que la X-team, il peut s'avérer pertinent d'alléger la charge cognitive des utilisateurs en fournissant des outils de gestion automatique de la consistance du projet.

• War room dans le cadre du développement logiciel

Teasley et al. [TCKO00] ont mené des travaux analytiques similaires, mais cette fois sur des équipes de 6 à 8 personnes aux compétences variées travaillant sur un projet de développement logiciel. En effet, il s'avère que sur le temps de travail d'un développeur, seuls 50% sont occupés aux tâches de développement. Cette constatation est d'ailleurs généralisée à l'ensemble des domaines d'activités, dans lesquels des employés de bureau passent 30% à 50% de leur temps de travail en réunion [Pan64][SB92]. La seconde moitié du temps est ainsi allouée aux activités de résolution de problème, de réunions, et d'activités de communication [Bar87]. La salle dédiée à

la collaboration proposée dans ces travaux était équipée d'ordinateurs individuels, ainsi que de tableaux blancs et de chevalets.

Dans cette expérience, les équipes avaient, en plus de la war room, accès à des salles dédiées au travail individuel. Cette particularité vise à mettre en évidence la nécessité ou non de fournir aux utilisateurs un espace physique composé de plusieurs pièces. Il s'avère que lors de la collaboration, les participants n'ont que très peu cherché à s'isoler dans ces espaces physiques privés. Cette constatation peut être rapprochée du phénomène de cocktail party évoqué dans les travaux de G. Mark [Mar02].

Outre un important gain de productivité grâce à la war room, l'analyse du travail révèle que les utilisateurs ont exprimé le besoin de disposer d'un support informatique plus complet. En effet, les participants possédant des domaines de compétences très divers, les informations affichées sur leur poste de travail sont différentes les unes des autres. Cependant, lorsque certains participants veulent réunir un sous-groupe autour d'un problème particulier, l'utilisation de leur écran individuel pour illustrer le sujet de la discussion n'est pas adaptée. Ils expriment ainsi le désir de pouvoir diffuser leur approche du problème sur un écran de plus grande taille, plus propice à la collaboration d'un ensemble d'utilisateurs. Pour reprendre le travail de façon plus individuelle en revanche, l'utilisateur éprouve un confort plus important à travailler sur un écran personnel de taille plus réduite.

De même, lorsqu'un groupe de collaborateurs interagit simultanément sur une grande surface partagée pour la résolution d'un problème, il leur est nécessaire de pouvoir transférer le fruit de cette collaboration directement sur les autres postes de travail, sans avoir à reproduire le travail réalisé en groupe pour l'intégrer au système.

Ces constatations sont symptomatiques d'un besoin de liberté dans le partage de l'information au sein des équipements matériels de la war room. Outre la mise en relation de tous les postes de travail via un réseau local, il est nécessaire de fournir des outils de visualisation qui s'adaptent à la dynamique de groupe. A cette adaptation dynamique, nous pouvons également ajouter le besoin de disposer d'une certaine hétérogénéité des dispositifs d'affichage, permettant de basculer entre une activité individuelle et une activité de groupe.

1.2.2.4 La collaboration de groupe à groupe : notion de présence mixte

En parallèle à l'utilisation d'une plateforme logicielle de type war room, un certain nombre de travaux ont cherché à coupler des outils de collaboration distante avec des environnements de travail colocalisés. Cette approche repose sur le constat que les projets à grande échelle nécessitent un large spectre de compétences et qu'il est parfois difficile de réunir l'intégralité des participants en un même lieu. Contrairement aux environnements de collaboration exclusivement distante comme peuvent l'être les EVC classiques, il est fréquent dans les cas d'usages réels qu'un certain nombre de collaborateurs soient géographiquement proches. Cette vision de la collaboration a donné naissance aux environnements de présence mixte [MRM07]. Ce terme est utilisé pour qualifier le fait que sur l'ensemble d'un groupe de travail, seuls certains participants sont colocalisés et sont amenés à coopérer avec des individus distants. La communication est alors scindée en deux flux, le premier étant le flux de communication face-à-face entre utilisateurs colocalisés, tandis que le second porte sur la communication médiatisée nécessaire aux échanges avec les utilisateurs distants.

• La présence mixte et la disparité de l'espace

Les plateformes de présence mixte visent donc à réunir virtuellement un ensemble d'utilisateurs sous forme de plusieurs groupes de taille variable. La notion de mixité est ici occasionnée par le fait que les utilisateurs doivent coopérer à la fois avec des participants distants, et donc à travers des outils de communication médiatisée, mais également avec des utilisateurs situés dans le même espace physique. Tang et al. [TBG04] soulignent par ailleurs qu'une telle configuration remet en question la taxonomie des environnements collaboratifs sous forme de matrice (cf 1.1.1.2) : les systèmes de communication de groupe à groupe appartiennent à la fois aux deux catégories de distribution spatiale (locale et distante). De même, la dimension temporelle est remise en question : si les utilisateurs possèdent la faculté d'interagir simultanément, ils peuvent aussi laisser certains objets dans la pièce pour les participants absents, entraînant ainsi une forme de collaboration asynchrone [SUM⁺98].

Cette configuration hétérogène s'avère cependant hautement problématique. En effet, la communication interpersonnelle se déroule différemment selon que l'on soit en situation de face-à-face, ou devant un système de communication électronique. Cette configuration est parfois appelée "mixed media" [BSO⁺04] : dans la vie courante, lorsqu'un media est utilisé, il est inégalement distribué. Par exemple, lors d'une réunion, il est fréquent que plusieurs participants soient réunis dans la même pièce et communiquent avec une ou plusieurs personnes par téléphone. Les participants distants souffrent souvent d'un phénomène d'isolement dans la collaboration.

• La notion "d'affective awareness"

Hofer et al. [RE01] ont notamment constaté qu'il existait moins de "affective awareness" [McA95] -complicité- entre des individus distants qu'entre deux personnes colocalisées. Ce phénomène peut inhiber la collaboration lorsque le projet rencontre des incertitudes, des crises ou des conflits personnels. De plus, lorsqu'une incompréhension survient, cela provoque une rupture dans le processus de résolution de la tâche, créant des sentiments négatifs qu'il est plus difficile d'apaiser à distance [BSO⁺04]. La manière dont sont répartis les utilisateurs entre les sites a par ailleurs un impact important sur l'organisation des échanges.

• Disparité des comportements des utilisateurs

Plusieurs travaux ont illustré la tendance qu'ont les utilisateurs à se créer des identités de groupes et à exprimer des comportements particuliers [BM08][BSO⁺04][BON⁺06][Bro00][LS92] : lorsque la distribution des ressources est inégale, les participants ont tendance à favoriser les membres de leur groupe, et à négliger les personnes extérieures [DC01]. Cette tendance à adopter une identité de groupe possède un impact positif sur la faculté à collaborer, tant que ce phénomène n'aboutit pas à la création de sous-groupes au sein d'une équipe [McD95]. L'existence de plusieurs sous-ensembles dans l'équipe a en effet tendance à créer une dynamique d'opposition entre les membres et les non-membres, qui tend d'ailleurs à être exagérée par les participants, désireux de marquer leur identité en tant que groupe [Taj78]. Dans le cas d'un environnement de présence mixte, la formation des groupes est essentiellement influencée par leur distribution géographique. En effet, la proximité physique entraîne beaucoup plus d'interactions et de partage d'informations avec les partenaires proches qu'avec les participants distants [Cra02]. Ce phénomène porte d'ailleurs le nom de "colocation blindness" : les utilisateurs distants deviennent en quelque sorte invisibles lorsque plusieurs collaborateurs sont situés dans la même pièce. L'utili-

sation du pronom "nous" lorsque les utilisateurs s'adressent à leurs collaborateurs distants est d'ailleurs symptomatique de la formation d'une identité de groupe entre les membres locaux [AC02].

• La présence mixte et l'interaction

Les travaux de Tang et al. [TBG04] ont alors abordé la collaboration en présence mixte sous l'angle des interactions. La plateforme logicielle repose ici sur un espace de travail partagé, en l'occurrence un tableau blanc virtuel.

Ces recherches mettent l'accent sur la constatation d'une double disparité au sein des environnements de groupe à groupe :

- la disparité de présence, qui suggère des problèmes de communication en raison d'une rupture dans la consistance du groupe (rupture à la fois visuelle et auditive).
- la disparité des affichages : les dispositifs matériels disponibles au sein de chaque site sont susceptibles d'être différents. Cette hétérogénéité matérielle peut être source d'incompatibilités entre les interactions de chacun. Par exemple, si deux utilisateurs dessinent face à face sur une table tactile et qu'un troisième utilisateur distant travaille lui sur un poste de travail classique, l'orientation des données sur l'écran constituera un point critique : le participant isolé devra adopter l'un ou l'autre des points de vue de ses collaborateurs, sans pouvoir bénéficier des deux simultanément.

La question de la disparité des affichages est ici traitée sous l'angle de l'orientation des informations. Outre la réorientation des télé-pointeurs sur l'espace de travail affiché du côté de l'utilisateur isolé, d'autres solutions ont été explorées, par exemple, en définissant une notion artificielle de haut et de bas sur l'interface horizontale de la table. Cependant, cette proposition a pour conséquence de contraindre l'utilisation du dispositif tactile en lui ôtant sa capacité à être utilisée indifféremment de l'orientation, provoquant ainsi une surpopulation sur le côté associé au bas artificiel. La solution adoptée est alors d'illustrer la position des utilisateurs de la table sur l'interface du participant isolé. Le constat selon lequel la disparité matérielle représente une source d'incompréhension est également appuyée par [MRM07]. L'observation de deux couples d'utilisateurs travaillant respectivement sur une table tactile et sur un poste de travail standard illustre le fait que les participants n'avaient pas conscience du matériel utilisé par leurs collaborateurs. Outre la difficulté à communiquer autour des objets virtuels (il leur arrive de discuter de deux objets différents sans s'en rendre compte), les utilisateurs ont procédé à des interactions dont le résultat était inadapté au dispositif d'affichage utilisé sur le site distant.

• La présence mixte et la communication

Outre ses travaux sur la X-team, qui était purement colocalisée, Gloria Mark a également étudié le comportement de plusieurs groupes en situation de présence mixte lors d'une activité de conception d'une mission spatiale [MAN03][MA04]. Ces travaux ont abordé l'étude de la communication entre les utilisateurs.

La particularité de leur plateforme est que les participants sont nombreux et sont répartis également entre les sites. L'équipe est donc constituée de plusieurs groupes de taille variable, communiquant grâce à des dispositifs de vidéoconférence de type mediaspace. Ces travaux mettent

en évidence la complexité du réseau de communication interpersonnelle : en effet, l'échange d'information doit transiter efficacement entre les sites, entre les utilisateurs colocalisés, mais également entre les sous-groupes formés au sein d'un site. En outre, la mise en relation d'un important nombre d'utilisateurs implique une complication supplémentaire liée au fait qu'un individu éprouvera plus de difficulté à prendre la parole pour intervenir.

La complexité de ce réseau de communication implique notamment des pertes potentielles dans le suivi du projet : lorsqu'un sous-groupe parvient à trouver une solution à un problème intermédiaire, la solution en question n'est pas systématiquement communiquée aux groupes distants. Les travaux analytiques de G. Mark ont ainsi adopté une approche relationnelle qui vise à étudier les relations entre un individu et le reste du groupe. Cette approche, par opposition aux études qui se focalisent essentiellement sur les relations entre individus, permet de mettre en évidence la notion de "space between" [BL00]. Ce concept (littéralement l'*espace entre*) traduit les connexions, interdépendances et espaces qui existent entre les différentes entités (individu ou groupe d'individus). Il constitue une forme de distance abstraite qui détermine la faculté qu'ont les utilisateurs à se comprendre et communiquer efficacement. Le "space between" est donc lié à la distance géographique entre les interlocuteurs, mais surtout à leur capacité à communiquer et à se mettre d'accord efficacement.

La plateforme de collaboration était constituée d'une communication audiovisuelle via l'application netmeeting de Microsoft, ainsi que d'outils de production bureautique partagés (MS Powerpoint et Excel). L'expérience a mis en relation une équipe hautement pluridisciplinaire. En effet, les nombreux domaines d'expertise des participants, et la vision du projet qui en découle, permet d'envisager beaucoup plus de scénarii possibles pour la conception du projet. Chaque scénario est issu des considérations propres à chaque spécialité de l'équipe. Le choix du projet (conception de mission spatiale) est justifié par l'importante activité de résolution de problèmes qu'elle implique. Ces problèmes sont nombreux et reposent sur un nombre important de paramètres, ils sont donc sujets à une forte interdépendance. Les prises de décision nécessaires engendrent ainsi une importante interaction entre les participants.

Les travaux de Mark et al. ont relevé trois inconvénients majeurs à la collaboration de groupe à groupe :

- **les sous-groupes** : les discussions privées en petits groupes d'individus ont été constatées au sein des équipes colocalisées. Le groupe de plus grande taille a même vu plusieurs sous-groupes simultanés se former. Ces discussions étaient auto-organisées (aucun individu ne s'est illustré comme étant le leader de la conversation), et pouvaient avoir lieu rapidement, pour de simples questions de clarification. En revanche, ce phénomène est resté très marginal entre sites distants. Les sous-groupes mixtes (ie. impliquant à la fois des utilisateurs colocalisés et des utilisateurs distants) se sont avérés peu autonomes, et peu spontanés ; contrairement aux sous-groupes colocalisés, la discussion mettait ici plusieurs minutes à s'organiser et n'avait lieu que pour des problèmes importants.
- **les méthodologies et hypothèses incompatibles** : les sites distants n'ont pas su s'accorder sur leurs méthodologies. De même, la terminologie employée est restée problématique, malgré plusieurs discussions visant à faire en sorte que toute l'équipe adopte un vocabulaire commun. Les utilisateurs n'ont en effet pas fait l'effort d'adopter cette terminologie au sein de leur groupe.

- **la confiance excessive dans la technologie** : les utilisateurs n'ont pas su adapter leur communication aux contraintes technologiques. En effet, ils n'ont pas montré de signe prouvant leur conscience du fait que la diffusion des documents produits n'apportait aux utilisateurs distants aucune information relative à la méthodologie qui a entraîné ce résultat. De même, lors de l'expression orale d'une idée, les utilisateurs n'ont pas pris soin de s'assurer que leurs collaborateurs distants comprenaient correctement leurs propos : la présence de plusieurs participants colocalisés fournissait un retour visuel illustrant la bonne intelligence de ce discours, donnant la sensation que cette compréhension était généralisée à toute l'équipe distribuée.

1.3 Exemples de plateformes prenant en charge la colocalisation

Après avoir étudié les nombreux phénomènes comportementaux relatifs à la collaboration en situation de coprésence, nous allons étudier les plateformes proposées dans ce domaine. En dépit du nombre important de travaux analytiques en regard des groupes d'utilisateurs colocalisés, peu de plateformes dédiées ont été développées. L'une des raisons que l'on peut évoquer pour expliquer ce constat réside dans le fait que les besoins majeurs associés à la collaboration dans un même espace physique peuvent sembler au premier abord assez proches des contraintes à prendre en compte pour le développement d'application coopératives distantes. Cependant, plusieurs propositions ont été formulées, tant dans le domaine matériel que dans le domaine logiciel.

En terme de proposition matérielle, les dispositifs actuels tendent à se tourner vers une utilisation multi-utilisateurs. Ainsi, nous assistons à l'important développement des tables tactiles multitouch, permettant de détecter plusieurs points de contact simultanément. Ce type de support a d'ailleurs fait l'objet de travaux qui ont conclu que sa disposition horizontale permet de distribuer plus efficacement les interactions entre les participants (par opposition aux tableaux blancs, dont l'utilisateur le plus proche est associé à un rôle de leader)[RL04]. Les applications développées pour ce type de support matériel sont par conséquent implicitement adaptées à l'interaction simultanée de plusieurs utilisateurs. La sortie du Diamond Touch [DL01] a d'ailleurs été plus loin dans cette optique multi-utilisateurs en proposant une table tactile capable d'associer un point de contact à un utilisateur, permettant à l'application de les identifier automatiquement. De même, le développement d'installations matérielles visant à proposer des périphériques d'affichage de grande taille peut également être associé à l'idée de fournir un support visuel confortable pour plusieurs participants à la fois.

1.3.1 Affichage unique

Un certain nombre de solutions proposées pour répondre aux besoins d'un groupe colocalisé consistent en l'utilisation d'un seul poste de travail ayant des facultés d'interaction et d'affichage adaptées à l'intervention de plus d'un utilisateur à la fois. Cette approche est associée au terme de SDG, pour Single Display Groupware (ie. un groupware à dispositif d'affichage unique), terme établi par Stewart et al. [SBD99].

La plateforme TIVOLI [PMMH93] issue du Laboratoire Xerox Parc, propose une installation constituée d'un tableau électronique sur lequel les utilisateurs peuvent écrire et dessiner avec un

stylet. Cet outil, dédié aux réunions de travail, supporte ainsi jusqu'à trois interacteurs simultanés. Dans le même laboratoire, Bier et al. [BF91] ont proposé une application permettant aux utilisateurs d'interagir sur le même ordinateur en utilisant trois souris simultanément. Cette idée d'utiliser plusieurs périphériques de pointage à la fois a également été exploitée par de nombreux autres travaux [HB99][TG04][Paw06][MIEL99]. Le travail d'analyse de Zanella et al. [ZG00] sur les situation de collaboration reposant sur l'utilisation d'une surface d'affichage unique soulèvent la question du nombre de dispositifs d'interaction. Cette contribution confirme qu'une activité collaborative, de part la multiplicité de ses participants, nécessite de proposer plusieurs périphériques permettant d'interagir simultanément sur l'environnement. Cette particularité permet en effet de minimiser les conflits et incompréhensions inhérents à la gestion des tours de parole et d'interaction. Plus récemment, la plateforme Dynamo [IBR⁺03] a été développée dans le but de permettre à plusieurs utilisateurs d'interagir côte-à-côte sur une même interface. Néanmoins, cette installation matérielle n'est pas limitée à l'utilisation d'un seul dispositif d'affichage. L'interface en question peut ici être affichée via l'utilisation de plusieurs écrans connectés ensemble : l'interface de travail est ainsi étendue sur plusieurs écrans, à la manière des configurations multi-moniteurs disponibles sur les postes de travail standards.

1.3.2 Affichage partagé

Dans l'optique de la collaboration colocalisée, le projet CoLaB du Xerox Parc (fig. 1.14) a également abordé la nécessité d'utiliser plusieurs postes de travail distincts simultanément [SFB⁺87]. Ces travaux ont ainsi entraîné une proposition consistant en la mise à disposition de plusieurs ordinateurs individuels dans une même salle, dont le support logiciel permet de prendre le contrôle d'un écran de grande taille situé dans la même pièce. L'écran géant constitue alors un espace visuel partagé entre les participants. Plusieurs recherches ont également exploré la possibilité de fournir un outil de collaboration sous la forme d'une plateforme logicielle grâce à laquelle les utilisateurs peuvent contrôler un grand écran à partir de leur poste de travail personnel [EE68][CEG⁺87].

L'innovation du Xerox Parc permettant de distribuer l'activité entre plusieurs ordinateurs a d'ailleurs été rapprochée de l'apparition des tables tactiles pour donner naissance à d'autres propositions, telles que les installations de type table centric. Il s'agit d'une catégorie de plateforme qui, comme son nom l'indique, prend le parti d'articuler l'ensemble de l'activité collaborative autour d'une unique table multitouch. Ce dispositif tactile constitue alors le centre de contrôle de l'application, à partir duquel les espaces de travail sont distribués sur d'autres écrans de la salle.

Wigdor et al. [WSFB06] ont ainsi développé un espace de travail interactif constitué de plusieurs écrans disposés autour de la table. Ce dispositif central permet alors de basculer l'affichage des différentes fenêtres sur les ordinateurs voisins, tout en conservant une miniature de ces fenêtres sur la table. Ce système de WIM (world in miniature) propose ainsi aux participants d'interagir sur les fenêtres via leur aperçu sur la table multitouch. Les dispositifs d'affichage satellites n'ont alors vocation qu'à être utilisés pour la visualisation et ne disposent pas d'outils d'interaction locaux. Cette approche repose par conséquent sur le désir de maintenir les utilisateurs à une position statique.

Le système Multispace [ESRF06], au contraire, est un système table centric permettant de



FIGURE 1.14 – L'environnement CoLaB de Xerox

basculer l'affichage des documents vers une surface tactile verticale sur laquelle les utilisateurs peuvent interagir. De la même manière, la plateforme ARIS (Application Relocation in Interactive Space) constitue une surcouche logicielle permettant de modifier la distribution des fenêtres d'applications du système d'exploitation pour les afficher sur l'un ou l'autre des écrans mis à disposition dans la salle [BB04].

1.3.3 Multi-dispositifs

D'autres approches ont proposé des plateformes multi-dispositifs, mais cette fois en conservant une certaine mobilité des utilisateurs dans l'espace physique. Ces travaux tendent donc plus à se rapprocher des méthodes de travail caractéristiques des war rooms. C'est par exemple le cas de I-Land (pour Interactive Landscape) [SGH⁺99], qui évoluera plus tard sous le nom de Roomware [PST04]. Cette plateforme à la fois matérielle et logicielle propose d'installer dans la pièce des dispositifs tactiles, connectés via une plateforme logicielle (fig. 1.15) :

- **les ConnecTables** : il s'agit de postes tactiles individuels et mobiles, installés sur un chariot. Ces dispositifs peuvent être déplacés dans la pièce, à la manière d'un bloc-notes. La connexion logicielle offre alors aux utilisateurs la possibilité de joindre deux ConnecTables par leur arrête supérieure, formant un espace partagé représenté sur les deux écrans. L'utilisation conjointe de deux ConnecTables permet ainsi aux deux utilisateurs d'interagir sur le même espace virtuel.
- **les CommChairs** : ce sont, comme les ConnecTables, des postes individuels autonomes intégrés à une chaise de bureau.



FIGURE 1.15 – Les dispositifs matériels de Roomware. De haut en bas : ConnecTables, CommChairs, InteracTable et Dynawall

- **l'InteracTable** : il s'agit d'une table interactive grâce à laquelle plusieurs utilisateurs peuvent interagir simultanément au moyen de stylets.
- **le DynaWall** : il s'agit d'un affichage mural large constitué de trois segments. L'espace virtuel représenté par ces trois segments est unique, à la manière d'un affichage multi-moniteurs classique.

La plateforme logicielle associée à cette installation, BEACH [Tan04], permet en outre d'utiliser un mode de partage appelé le Passage. Ce système offre aux participants la possibilité de déplacer des documents virtuels entre l'InteracTable et le DynaWall via l'utilisation d'un objet tangible quelconque. Le support logiciel permet ainsi de distribuer un espace de travail 2D entre



FIGURE 1.16 – L'installation matérielle I-Room

différents dispositifs tactiles, en le segmentant en plusieurs espaces ou en l'étendant sur plusieurs affichages. L'application collaborative offre plusieurs éléments déplaçables (les magnets), sur lesquels les participants peuvent écrire ou dessiner. Les utilisateurs peuvent donc utiliser la plateforme Roomware pour écrire leurs idées, les annoter, les structurer, les hiérarchiser ou en discuter entre eux.

Le système I-Room [JFW02][JFW10] de l'université de Stanford propose également une salle équipée de plusieurs surfaces d'affichage (une table rétro-projetée et plusieurs écrans verticaux de grande taille) permettant de distribuer la visualisation des documents dans l'espace physique. Pour permettre une interaction à distance, un ensemble clavier-souris situé au centre de la pièce permet d'interagir sur les écrans satellites (fig. 1.16). Cette solution repose sur le développement d'une interface logicielle qui a pour fonction de rediriger les événements systèmes tels que les clics de la souris de façon à les transmettre à un autre poste de travail. Cette stratégie visant à implémenter des communications réseau au niveau de la gestion de l'interaction ou de l'affichage d'un système est également employée dans la contribution de Gjerlufsen et al. [GKE⁺11].

Si ces solutions offrent une approche intéressante de la dynamique à mettre en oeuvre pour supporter la colocalisation des utilisateurs, elles s'avèrent toutefois limitées dans les outils d'interaction et de visualisation qu'elles proposent. En effet, la vision du projet est abordée selon l'angle d'une vue globale, commune à tous les participants de la collaboration. Les utilisateurs disposent tous du même panel d'outils et ils communiquent autour des mêmes artefacts visuels. Cette vision nous semble trop restrictive pour prendre en compte la grande diversité des compétences impliquées dans un travail d'équipe.

1.3.4 Le multi-vues dans le cadre d'une collaboration distante

Les travaux de Schafer et al. [SB05], bien qu'ils ne soient pas basés sur une collaboration localisée mais sur une activité distante type EVC, abordent une certaine diversité dans l'ap-

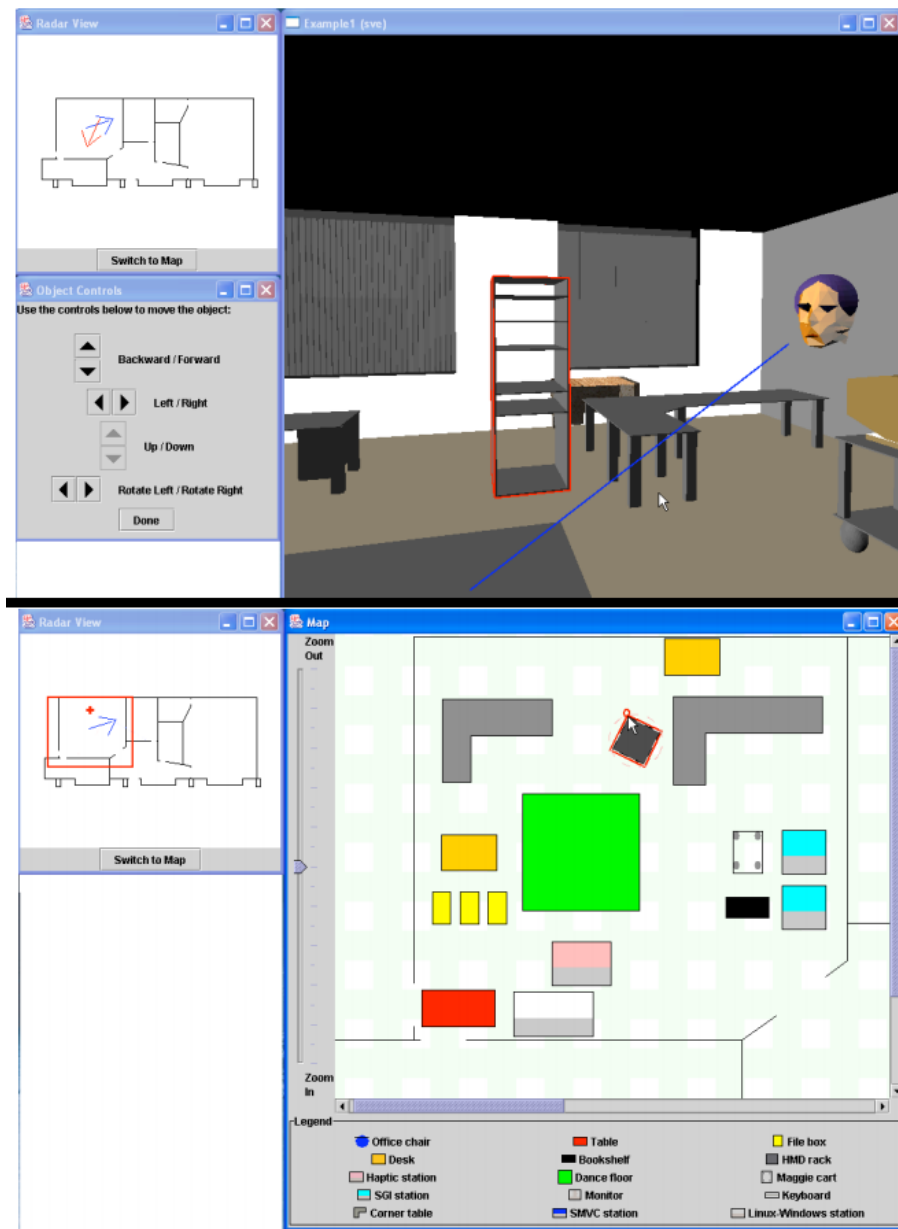


FIGURE 1.17 – Les deux modes de représentation de la plateforme d'interaction distante de Schafer et al.

proche des objets. Leurs travaux, dédiés à la collaboration spatiale² proposent une plateforme d'aménagement d'intérieur d'un bureau. Cette tâche de collaboration spatiale révèle un besoin d'analyser l'espace selon plusieurs points de vue, pour discuter sur des concepts tels que la localisation des objets, les mesures de distances ou même la planification des chemins des usagers. La particularité des tâches d'aménagement est qu'elles nécessitent une importante part d'interactions liées à la navigation : déplacement, rotation, zoom.

2. Une activité collaborative dans laquelle les utilisateurs sont amenés à aménager un espace en choisissant le positionnement et l'orientation d'éléments divers

La plateforme proposée pour réaliser cette tâche à distance offre une vue double sur le projet, sous la forme d'un affichage plan en 2D ou d'un affichage immersif en 3D (fig. 1.17). Les utilisateurs ont ainsi la capacité de choisir le mode selon lequel ils désirent percevoir l'espace. Le support matériel est un équipement standard (ensemble clavier/souris pour chaque participant) et le système de communication offre les fonctionnalités de préservation de conscience sociale nécessaires (affichage de télépointeurs et avatars virtuels dans chaque vue). Il s'est avéré que le fait de disposer sur demande d'une vue 2D ou 3D simplifie l'interaction et met en évidence l'importance du mode de représentation sur l'activité. Les utilisateurs préfèrent ainsi utiliser l'un ou l'autre des modes de représentation en fonction de la tâche qu'ils désirent réaliser. La vue 3D permet de procéder à une organisation massive des objets dans l'espace virtuel, tandis que la vue 2D dotée de facultés de zoom s'avère plus pertinente pour les interactions nécessitant plus de précision. L'utilisation de l'affichage tridimensionnel permet également d'observer l'espace virtuel selon une vue subjective, permettant de détecter certains mauvais choix de conception qui s'avéraient indétectables selon la vue 2D. La possibilité de modifier dynamiquement le mode de représentation d'un utilisateur est également un avantage majeur. L'expérimentation en conditions réelles a mis en évidence le fait que les participants ont changé plusieurs fois ce mode au cours de la collaboration.

L'évaluation de cette proposition émet toutefois quelques réserves vis-à-vis de la faculté à coordonner les actions des utilisateurs : le fait d'utiliser des représentations différentes réduit le sentiment de conscience sociale, qui est un facteur critique pour une application de collaboration distante. Ainsi, les participants ont éprouvé des difficultés à identifier précisément l'activité de leurs collaborateurs et lorsque deux utilisateurs employaient deux modes de vue différents, ils ont été victime de plusieurs incompréhensions dues à leurs perceptions différentes. Les mouvements déictiques utilisant le télépointeur se sont notamment avérés difficiles à mettre en oeuvre. Pour les mêmes raisons, la constitution de sous-groupes pour discuter de sous-problèmes s'est avérée difficile, voire impossible. Cette approche hybride du projet constitue par conséquent un avantage indéniable concernant la collaboration d'une équipe, mais souffre de ses facultés d'hétérogénéité lorsqu'elle est appliquée à une collaboration distante.

1.3.5 Conclusion

Les contributions que nous venons de voir apportent des éléments particulièrement intéressants vis-à-vis de l'environnement à mettre en place pour assurer la collaboration colocalisée.

Les plateformes basées sur l'utilisation d'un unique dispositif sont trop restrictives dans la mesure où elles ne fournissent qu'un unique affichage. Cette solution, outre le fait qu'elle ne permet pas de tirer réellement parti de la pluridisciplinarité des équipes, n'exploite pas non plus l'espace physique : les utilisateurs sont focalisés sur le même dispositif et ne peuvent exploiter la dynamique de répartition spatiale qui possède un impact sémantique important sur la perception de l'activité.

Les solutions basées sur le principe d'un écran partagé permettent au contraire de répartir les participants dans l'espace, offrant aux utilisateurs un espace physique plus privé qui favorise le confort d'utilisation. Cependant, l'écran partagé reste là encore l'unique centre d'attention, et l'activité n'est pas réellement distribuée dans la pièce. De même, les collaborateurs sont disposés à une position fixe dans l'espace, restreignant là encore les avantages du partage d'un même espace physique.

Les approches multi-dispositifs, quant à elles, offrent la possibilité de déployer l'activité libre-

ment dans l'espace physique. Ce déploiement a d'ailleurs lieu à la fois sur les utilisateurs, mais également sur les documents qui servent de support à la collaboration. Cette approche libère tout le potentiel de la dynamique de distribution spatiale puisque les utilisateurs peuvent se déplacer, former des sous-groupes, analyser l'activité des autres simplement en observant leur positionnement dans la pièce. De même, les données prennent une dimension plus concrète, puisqu'elles sont désormais mobiles et peuvent être agencées dans l'espace de façon à apporter au projet une structuration implicite supplémentaire. Les plateformes proposées n'abordent cependant pas la pluridisciplinarité des équipes, dans la mesure où les données numériques relatives au projet ne sont pas affichées selon des représentations graphiques hétérogènes : un objet virtuel (en l'occurrence des documents numériques) conserve le même mode de représentation quels que soient sa position dans l'espace ou le dispositif matériel qui l'affiche. Dans cette optique, l'approche multi-vues de Schafer offre une fonctionnalité particulièrement intéressante : la représentation visuelle du même objet virtuel sous différentes formes simultanément. Leur évaluation a mis en avant la difficulté qu'éprouvent les utilisateurs distants à ressentir une conscience sociale lors de leur activité. L'application d'une telle approche dans le cadre d'une collaboration colocalisée ne souffrirait cependant pas de ce phénomène puisque les utilisateurs se trouvent dans la même pièce, et leur proximité physique suffit à fournir ce sentiment de conscience sociale.

INTERACTION COLLABORATIVE COLOCALISÉE : ANALYSE ET EXTRACTION DES BESOINS

Sommaire

2.1	Analyse des travaux sur la collaboration et déduction des besoins .	58
2.1.1	L'impact des comportements de groupe	58
2.1.2	L'impact de la notion d'espace	60
2.1.3	Les comportements individuels des membres d'une équipe	61
2.2	Conception de notre plateforme	64
2.2.1	La revue de projet : une étape qui met l'accent sur la collaboration . . .	64
2.2.2	Notre proposition : la war room assistée par ordinateur	65
2.3	Reflexions sur le concept de canaux de visualisation et d'interaction	76
2.3.1	La structuration de l'activité dans la war room : le système de canaux .	76
2.3.2	Illustration de l'utilisation des canaux	77
2.3.3	Les canaux à l'échelle des données : approche sémantique des interactions	80
2.3.4	Les canaux pour donner une portée concrète à des informations non représentables	81
2.3.5	Les canaux, notion distincte de l'interface	81
2.3.6	Les canaux : bilan	82
2.4	Conclusion de notre analyse	82

Les différents travaux répertoriés dans la partie précédente nous fournissent de nombreuses informations, abordant la problématique selon des approches différentes. Cette diversité des points de vue nous permet d'aborder le problème de façon globale, et de répondre au mieux aux multiples cas de figure potentiels. Nous construisons par conséquent notre proposition selon une méthodologie hybride, prenant en considération l'aspect technique et logiciel d'une part, et la place centrale que prendront les utilisateurs au sein de l'activité de coopération d'autre part. Notre travail va ainsi consister à extraire de l'état de l'art les différents éléments critiques dont il faut tenir compte pour optimiser les facultés de collaboration d'une équipe d'utilisateurs colocalisés, à la fois en terme de groupe et en terme d'individu. A partir de ces informations, nous construisons ainsi un ensemble de conclusions relatives à la plateforme logicielle et matérielle à mettre en place afin de concevoir une architecture logicielle capable de s'adapter à de multiples scénarios d'utilisation.

Nous verrons dans cette partie que les conclusions que nous pouvons tirer des travaux analysés dans l'état de l'art peuvent s'avérer divergentes, voire même conflictuelles. Notre objectif consiste alors à proposer un compromis acceptable qui s'avèrera cohérent dans la majorité des cas d'utilisation de la plateforme. La solution que nous proposons ici ne consiste par conséquent pas en la conception d'une application clé en main, mais vise plutôt à proposer un outil plus polyvalent, qui fournira à un concepteur possédant des connaissances minimales les fonctionnalités nécessaires à la configuration de la plateforme adaptée à ses besoins.

2.1 Analyse des travaux sur la collaboration et déduction des besoins

2.1.1 L'impact des comportements de groupe

L'observation des groupes d'individus collaborant dans un même espace physique est à l'origine de diverses conclusions relatives aux fonctionnalités nécessaires et optimales pour une collaboration efficace. Nous avons vu que lorsque les participants disposent de plusieurs pièces distinctes pour travailler sur leur projet, le groupe révèle une forte tendance à rester ensemble dans le même espace [TCKO00]. Il s'avère par conséquent nécessaire dans notre proposition d'unifier l'espace physique en réunissant les participants dans un unique espace clos, qui sera plus propice aux échanges, à l'écoute de l'ensemble de collaborateurs et au suivi de l'évolution du projet. Notre plateforme doit ainsi prendre place dans un environnement unique de type war room.

L'installation de plusieurs dispositifs d'interaction et de visualisation constitue également une proposition permettant la prise en charge de certains comportements de groupe. En effet, lors de la réalisation d'un projet, l'ensemble des collaborateurs présente une tendance à constituer des sous-groupes pour résoudre certains problèmes (section 1.2.2.2). La disposition spatiale des utilisateurs concernés n'est pas anodine, puisqu'elle est définie par la position du support d'affichage illustrant la partie critique du problème. De même, le fait que les utilisateurs analysent inconsciemment la position de leurs collaborateurs dans la pièce pour en déduire la nature de leur activité appuie l'importance de disposer dans la pièce de différentes zones associables à certaines tâches. Ce comportement induit un partitionnement du projet. L'utilisation de plusieurs postes de travail relatant chacun l'une de ces parties fournit un référentiel spatial pour la constitution des sous-groupes et pour l'analyse visuelle des autres collaborateurs. Nous prenons par consé-

quent le parti de répartir l'activité sur plusieurs dispositifs distribués dans la war room.

Bien que les participants soient amenés à procéder à une mémorisation spatiale des données, lorsqu'ils sont confrontés à une installation matérielle disposant de plusieurs affichages simultanés, ils ne les exploitent pas de la même façon au fur et à mesure de la session de collaboration. Ils ont ainsi tendance à se focaliser sur l'espace public au début, puis ils aspirent progressivement à disposer d'un support visuel plus personnel (section 1.2.1.4). L'affichage des données sur les différents dispositifs de la war room doit par conséquent supporter cette utilisation fluctuante tout en proposant une répartition dynamique des données représentées sur les différentes surfaces d'affichage de la war room. Cette répartition dynamique répond d'ailleurs dans une certaine mesure au besoin de fluidité dans le passage d'une tâche à une autre.

La proximité des utilisateurs entre eux, tout particulièrement lorsqu'ils travaillent conjointement sur une sous-tâche précise du projet, implique en outre la nécessité de partager au maximum leur espace de travail physique. Ce partage total de l'espace d'interaction peut ainsi prendre place à travers l'utilisation d'un dispositif matériel adapté à une collaboration multi-utilisateurs. Les récents systèmes de table tactile multi-points font partie des solutions répondant au mieux à cette nécessité. Le choix des installations matérielles disponible dans notre war room peut par conséquent revêtir une importance majeure dans la qualité de la collaboration.

Il n'existe cependant pas de solution universelle répondant aux besoins matériels d'un projet : l'utilisation de dispositifs mono-utilisateur peut répondre au besoin d'indépendance des participants. Ainsi, pour procéder à un travail individuel, il peut parfois s'avérer pertinent de proposer aux participants des postes de travail personnels. Cette diversité potentielle des installations matérielles ne doit pas être un frein à l'application de notre système. La plateforme logicielle que nous proposons doit par conséquent disposer d'une certaine autonomie vis-à-vis du parc matériel mis en oeuvre dans la war room. La salle pourra ainsi être équipée de terminaux individuels, mobiles, ou d'installations plus importantes, telles que des tables tactiles, des écrans de grande taille ou des systèmes de vision stéréoscopiques. Nous décidons ainsi de fournir un support logiciel capable d'être adapté facilement par le concepteur pour prendre en charge la diversité matérielle nécessaire à la collaboration sur le projet visé.

Les caractéristiques que nous en déduisons peuvent être résumées ainsi :

1. L'utilisation d'un unique espace physique pour centraliser l'activité dans une pièce.
2. Une installation matérielle multi-dispositifs.
3. Un choix de dispositifs suffisamment varié pour adapter l'activité en fonction de la tâche à réaliser (dispositifs tactiles multitouch pour accueillir un groupe d'utilisateurs, stations de travail personnelles pour les tâches plus individuelles, etc.).
4. La répartition dynamique de l'activité entre les postes de travail disposés dans la pièce. Les objets virtuels doivent pouvoir être affichés simultanément sur plusieurs postes, ou au contraire n'apparaître que sous la forme d'une unique instance pour bénéficier d'une consistance visuelle similaire à celle d'un document physique. De même, il doit être possible de "déplacer" ces objets virtuels dans l'espace physique. Cette dynamique étant propre à chaque activité, notre proposition doit également pouvoir s'adapter facilement aux besoins

de chaque cas d'utilisation.

2.1.2 L'impact de la notion d'espace

A travers l'observation de l'utilisation que font les utilisateurs de leur espace physique, nous pouvons constater que les configurations idéales sont complexes et variées, car extrêmement dépendantes des facteurs humains, organisationnels et des scénarios d'utilisation. En effet, le fait que les participants éprouvent le besoin d'adapter leur environnement physique en fonction de l'activité nous porte à considérer que la configuration spatiale des éléments d'interaction ne peut être définie de façon précise et universelle. Par conséquent, la plateforme collaborative que nous voulons proposer doit fournir une certaine liberté dans le positionnement spatial des données, mais aussi des outils d'interaction, des objets physiques et même des utilisateurs. Notre proposition doit alors permettre au concepteur de déployer chacun de ces éléments de la façon la plus adaptée au cas d'utilisation. De même, les installations matérielles ne posséderont pas un positionnement prédéterminé, comme ce peut être le cas pour les plateformes de type table centric, mais seront au contraire placées en fonction des choix des personnes responsables du déploiement du support matériel. Selon la conception de Harisson et Dourish, cela consiste à donner aux utilisateurs la capacité de transformer leur espace physique pour en faire un "place".

La conclusion des travaux d'Harisson et Dourish [HD96] nous apporte également une ligne de conduite vis-à-vis de la représentation des données : notre environnement virtuel ne représente qu'une partie de l'espace de collaboration ; contrairement aux EVC classiques, la colocalisation des participants joue un rôle prépondérant dans leurs échanges interpersonnels, et l'espace physique constitue ainsi une part importante de l'espace de collaboration. Par conséquent, il n'est pas indispensable de procéder à une reconstruction artificielle basée sur une métaphore spatiale du monde réel. Les données affichées par le support logiciel sont donc libres en terme de représentation : il est envisageable d'exploiter des modes de visualisation abstraits, conceptuels, ou sous forme de fiches techniques sans risque que cette représentation n'altère la capacité des utilisateurs à assimiler l'organisation spatiale de ces données.

Nous avons également pu constater une importante mobilité de la part des différents membres d'une équipe lorsqu'ils collaborent dans la même pièce (section 1.2.1.4). Parmi les facteurs à l'origine de cette mobilité, nous pouvons citer notamment la nécessité pour les utilisateurs de modifier leur point de vue sur le projet. Ils se déplacent ainsi pour observer les données sous un autre angle. Cette tendance au déplacement est également influencée par le mode de représentation des données. Un petit groupe d'utilisateurs va naturellement chercher à avoir le même point de vue quand il est question d'étudier des données sous forme de document 2D. Nous en déduisons plusieurs propositions pour notre plateforme :

- La liberté de mouvement pour les utilisateurs. Cela signifie qu'il faut éviter d'assigner à chaque participant une place fixe, comme ce pourrait être le cas si chacun dispose de son propre poste de travail ou s'ils sont disposés sur leurs sièges devant un unique écran de grande taille.
- La seconde proposition consiste quant à elle à satisfaire cette mobilité au sein même de l'application, à travers la mise en place d'un système de visualisation mobile : pour aborder les informations et les documents sous un autre angle de vue, les participants peuvent

éventuellement interagir directement sur le positionnement et l'orientation des données du projet, comme ils le feraient avec une feuille de note dans le monde réel.

- Lors de l'utilisation d'un environnement disposant de plusieurs supports matériels, les déplacements des utilisateurs peuvent toutefois s'avérer fastidieux. Pour conserver un bon confort d'utilisation, les participants ne doivent pas être contraints à se déplacer systématiquement pour chaque interaction. En ce sens, une disposition de type table centric s'avère pertinente puisqu'elle maintient les utilisateurs dans un espace propice à la communication (autour de la table), tout en leur proposant sur la table tactile les outils permettant d'interagir sur un dispositif à distance. Cependant cette considération n'est selon nous pas opposable à la liberté de mouvement des collaborateurs. En effet, il nous apparaît naturel de proposer une configuration mixte, permettant à la fois de se déplacer pour prendre le contrôle d'un poste de travail précis, mais aussi de contrôler ce même poste de travail à distance via les périphériques d'interaction branchés à un autre dispositif de la pièce. Nous considérons que cette possibilité sera naturellement exploitée par les utilisateurs en fonction de l'ampleur de la tâche qu'ils ont à réaliser. Par conséquent, les participants doivent pouvoir se déplacer pour former un groupe de discussion autour d'un problème, engageant ainsi une session de collaboration et d'interaction significative. Dans le même temps, dans le cas d'une interaction courte (ajuster une donnée, modifier l'orientation d'un document, etc.), il doit être possible de procéder à cette action sans avoir à bouger.

Pour résumer les caractéristiques déduites de ce constat, nous pouvons les synthétiser sous la forme de trois fonctionnalités nécessaires :

1. La capacité de se déplacer librement dans l'espace physique, de façon à utiliser directement l'un des dispositifs ou d'une manière plus générale l'une des surfaces de travail.
2. La capacité d'inclure des dispositifs mobiles permettant l'interaction et/ou la visualisation du projet sans être contraint de respecter une position dans l'espace.
3. La capacité d'interagir à distance sur une surface de travail en utilisant les périphériques d'interaction d'un autre poste.

2.1.3 Les comportements individuels des membres d'une équipe

Au delà de la prise en charge d'un groupe, la réalisation d'une plateforme efficace doit également passer par la résolution de plusieurs contraintes propres aux individus. Nous devons ainsi nous assurer que la proposition que nous formulons réponde correctement aux attentes de chaque utilisateur individuellement, tout en maintenant les solutions apportées pour satisfaire l'équipe dans son ensemble.

Le phénomène principal que l'on peut extraire de notre observation de cette partie de l'état de l'art réside dans l'importante part d'individualité qui subsiste en dépit de l'intégration de l'utilisateur au sein d'un groupe. En effet, il apparaît que chaque participant possède des besoins et des préférences qui lui sont propres. Ces besoins peuvent porter sur :

La structuration de l'interface. Les participants éprouvent le besoin d'adapter l'organisation des éléments selon leurs propres habitudes de travail.

La territorialité. Les membres d'une équipe travaillant sur un espace virtuel partagé montrent des signes distinctifs impliquant une nécessité de conserver une partie de cet espace pour leurs besoins propres.

Le support d'affichage des informations. Les utilisateurs montrent un certain désir de disposer d'une surface d'affichage de grande taille pour servir de support visuel à une discussion de groupe, mais ils éprouvent également le besoin de disposer d'un support d'affichage plus restreint pour se livrer à une tâche individuelle.

La navigation. De la même manière que lors de l'utilisation d'un environnement de type EVC, les utilisateurs se déplacent dans l'espace virtuel de façon autonome. Ils peuvent décider de se grouper pour observer le même endroit de cet espace, mais peuvent également décider de gérer leur propre navigation pour observer un élément sous l'angle qui correspond le mieux à leur tâche. Les utilisateurs montrent ainsi une certaine autonomie dans la visualisation de l'espace virtuel.

Les méthodes de travail. Chaque membre d'une équipe possède des méthodes de travail qui lui sont propres. Ils peuvent aborder un même projet de façon radicalement différente en fonction de leurs compétences individuelles et ne procèdent par conséquent pas aux mêmes interactions sur celui-ci. Cette diversité des approches constitue d'ailleurs la qualité principale des équipes pluridisciplinaires, dans la mesure où elle permet d'anticiper plus efficacement les problèmes potentiels relatifs à chaque domaine.

Le mode de représentation du projet. Ce point est la conséquence directe de la diversité des méthodes de travail : les raisonnements et les interactions réalisées par un utilisateur reposent sur un ensemble de données différent des autres participants et sont dépendants de leurs compétences propres. Leurs perceptions visuelles d'un même projet sont donc extrêmement différentes et ils n'ont par conséquent pas les mêmes besoins en terme de mode de représentation des données.

Il est important de souligner que l'individualité de l'activité doit être gérée de façon à ne pas entrer en conflit avec la cohésion de groupe. La nécessité de considérer les utilisateurs individuellement est soumise à une dimension temporelle : par exemple, le besoin de constituer un espace privé pour travailler seul sur une tâche donnée n'apparaît que ponctuellement au cours de l'activité collaborative. D'une manière générale, nous pouvons dire que notre proposition doit tenir compte à la fois de l'aspect individuel et de l'aspect "groupe d'utilisateurs", mais qu'en cas de besoins conflictuels les intérêts de l'équipe dans sa globalité doivent passer en priorité.

Ces constatations sur les besoins individuels nous amènent à aborder la relation unissant les participants à leur projet sous un angle différent. La diversité des compétences dans une équipe travaillant dans notre war room ne doit pas impacter uniquement leur communication orale, mais doit aussi se refléter dans les outils de production dont ils disposent. La war room que nous proposons doit alors fournir à chacun des méthodes de visualisation et d'interaction

diverses, qui s'avèrent chacune les plus adaptées aux diverses approches du projet. Cette diversité ne doit cependant pas constituer une contrainte à la cohérence de l'ensemble des données et elle ne doit pas provoquer de rupture dans la cohésion du groupe. Il importe donc de s'assurer que ces différentes approches seront liées aux mêmes contraintes globales et que les actes que chaque utilisateur réalise sur le projet commun auront une répercussion directe sur la perception qu'en ont les autres.

A l'inverse, certaines données propres à une approche du projet peuvent n'avoir d'utilité que sur cette approche, sans avoir d'impact sur le travail des autres participants. Le système logiciel de notre war room doit alors être capable de prendre en charge ces données autonomes sans perturber les autres niveaux de perception.

La diversité des approches du projet constitue par conséquent la pierre angulaire de notre proposition de support à la collaboration colocalisée. Nous devons conserver la notion de groupe indispensable à la collaboration, mais nous prenons également le parti de fournir à chaque participant un environnement de travail le plus proche possible de celui auquel il est habitué lorsqu'il est amené à travailler de façon individuelle. L'hétérogénéité importante qui apparaît dans les cas d'utilisation intervient ainsi à plusieurs niveaux :

L'hétérogénéité matérielle : outre la nécessité de prendre en charge des dispositifs multi-utilisateurs ou individuels, nous devons être capable de gérer les méthodes de travail propres à chaque corps de métier impliqué dans l'activité. Par conséquent, les dispositifs d'interaction se doivent d'être suffisamment variés pour proposer à chaque participant la modalité qui coïncide le plus avec les outils qu'il utilise habituellement.

L'hétérogénéité humaine : le nombre de participants est variable d'un projet à l'autre. Les compétences représentées par les membres de l'équipe le sont aussi. Les sous-groupes susceptibles de se créer au cours de la collaboration dépendent également des différents corps de métiers, ainsi que des points critiques sur lesquels ils seront amenés à collaborer au cours de la réalisation du projet. Cette diversité des configurations humaines entraîne ainsi des besoins différents vis-à-vis du nombre et du type des dispositifs à intégrer, de même que vis-à-vis des outils à implémenter au niveau logiciel.

L'hétérogénéité logicielle : au même titre que les supports matériels, les méthodes de représentation des données doivent s'adapter aux compétences diverses des utilisateurs et aux dispositifs matériels. Il s'agit donc de proposer plusieurs façons d'aborder une même donnée du problème, sous forme de plusieurs interfaces différentes. Chacune d'entre elles permet ainsi de visualiser l'objet interactif d'une façon plus intelligible pour chaque participant.

La non-uniformité de la plateforme constitue par conséquent une contrainte importante, puisqu'elle doit malgré tout conserver une cohérence maximale sur le projet commun à l'équipe.

Pour résumer ces caractéristiques, nous pouvons citer plusieurs points :

1. La nécessité de s'adapter à des équipes d'utilisateurs dont l'effectif est variable. Pour une adaptation naturelle, le support logiciel de notre proposition ne doit pas considérer les utilisateurs individuellement comme c'est le cas pour un EVC classique, mais au contraire

aborder l'équipe comme une unique entité.

2. La nécessité de prendre en compte la diversité des domaines d'expertise des utilisateurs à travers l'interaction. Chaque dispositif matériel doit fournir une modalité d'interaction adaptée aux tâches à effectuer et le parc matériel mis en place dans la pièce doit être suffisamment varié pour répondre aux besoins de chacun.
3. La nécessité de prendre en compte la diversité des domaines d'expertise des utilisateurs à travers la visualisation. Chaque utilisateur doit pouvoir percevoir l'activité selon le mode le plus adapté à ses compétences. Ce besoin se traduit par une certaine diversité des modes de représentation des objets virtuels permettant de fournir à chaque spécialiste l'environnement visuel le plus proche de ses habitudes de travail.

2.2 Conception de notre plateforme

2.2.1 La revue de projet : une étape qui met l'accent sur la collaboration

Notre proposition s'articule avant tout autour du domaine de la revue de projet. En effet, ce type d'activité présente un cas d'utilisation qui se prête particulièrement bien à l'utilisation d'un système d'aide à la collaboration colocalisée : la revue de projet consiste à observer l'avancement du travail de plusieurs équipes sur une tâche commune. En effet, dans le contexte de la réalisation d'un projet de grande ampleur, les tâches à réaliser sont variées et souvent dépendantes de plusieurs domaines d'expertise, sous forme d'équipes distinctes ou de différents bureaux d'études. Si cette séparation est nécessaire pour la qualité du travail de production, elle peut s'avérer être la source de conflits : les différents intervenants sont très spécialisés et n'ont par conséquent pas toujours les compétences suffisantes pour prendre en compte les contraintes liées aux autres domaines d'expertise. Ceci entraîne certains problèmes, puisque tous les choix réalisés par l'une des équipes possèdent potentiellement un impact fort sur les tâches à réaliser par les autres. Or, le projet bâti à partir du travail de chacun doit être libre de toute incohérence. Il arrive donc fréquemment que deux éléments constitutifs d'un projet repose sur des considérations conflictuelles, remettant alors en question l'intégralité du projet.

Chaque domaine va donc être amené à produire une partie du projet de façon relativement indépendante et la problématique de gestion de projet consiste alors à s'assurer que les différents éléments s'imbriquent correctement, sans générer de conflit et en conservant les objectifs de base. Les différents domaines d'expertise travaillent ici individuellement et l'activité de collaboration a alors lieu lors de ces réunions de revue durant lesquelles les représentants de chaque domaine font le point sur l'avancée de chacun et discutent de la tournure à prendre pour satisfaire au mieux les prérogatives propres à chaque équipe. C'est lors de ces réunions que vont se prendre les principales décisions et que les lignes de conduites nécessaire à la bonne marche du projet vont être définies.

Dans notre contexte, un des intérêts de cette activité réside dans le fait qu'elle est constituée d'une importante part de communication. Les participants vont aborder tous les problèmes susceptibles d'influencer leur travail et le groupe de décision procèdera à plusieurs manoeuvres de négociation pour définir le compromis idéal. La conception de tous les aspects concrets du

projet va donc y être réalisée. Ces réunions de revue de projet prennent ainsi place en amont des tâches de production et sont déterminantes sur les choix que devront prendre chacune des équipes.

Dans la plupart des systèmes d'interaction collaborative colocalisée comparables aux war rooms, la tâche à réaliser consiste essentiellement en l'accomplissement d'une tâche de conception ou une activité de brainstorming, qui prend place au début du "workflow" de gestion de projet. L'aspect "production" de l'activité est ainsi souvent délaissé au profit d'outils majoritairement basés sur le tracé à main levée de schémas, de notes ou l'édition de documents textuels. En effet, le projet à ce stade de développement possède une dimension concrète quasi-nulle et les ressources sémantiques servant de support à la discussion sont le plus souvent constitués de différentes feuilles de données indépendantes spécifiant les contraintes du projet (ce que l'on peut appeler les "choix externes"). L'attention à porter dans ce contexte de collaboration porte avant tout sur la communication interpersonnelle. Par conséquent, les activités d'interaction prenant place dans ce contexte n'exige de la part du support logiciel que peu de gestion automatique de la cohérence, excepté pour la prise en charge de plusieurs affichages simultanés d'une même interface.

Au contraire, la notion de maquette que nous évoquons dans notre vision de la revue de projet sur maquette numérique constitue l'objectif central de l'application. Dans le domaine industriel par exemple, l'activité de revue de projet peut porter sur la conception d'une pièce mécanique complexe, définie par plusieurs données portant sur sa géométrie ou sur des paramètres intrinsèques tels que les coefficients de rigidité, la matière ou l'élasticité. A titre plus général, la maquette virtuelle telle que nous la définissons dans notre plateforme est comparable à un objet technique complexe, éventuellement constitué de plusieurs objets. A ce titre, elle peut être considérée comme un assemblage de pièces. Cette maquette est ainsi définie par plusieurs données et elle est dotée de fonctionnalités liées à sa réactivité. Les utilisateurs qui interagissent sur le projet peuvent ainsi manipuler ses éléments constitutifs, ou modifier leurs paramètres intrinsèques. La présence d'objets et de documents multiples est bien sûr possible, permettant ainsi de consulter des ressources d'information externes à l'objet. Cependant, l'objet interactif sur lequel chacun interagit va constituer le référentiel principal de la communication et sa capacité à réagir aux interactions lui permettra de servir de modèle destiné à tester les différents scénarios possibles et à simuler les propositions des participants.

2.2.2 Notre proposition : la war room assistée par ordinateur

Cette partie va s'appuyer sur les propositions théoriques formulées dans les sections précédentes pour réaliser un descriptif exhaustif des fonctionnalités techniques que la plateforme de collaboration colocalisée doit implémenter. Nous avons vu dans la section précédente qu'en fonction du cas d'utilisation, des domaines de compétences des participants, de leur nombre, ainsi que de l'ensemble des dispositifs matériels disponibles, la configuration idéale à mettre en oeuvre pouvait différer. Par conséquent, nous prenons le parti de développer une plateforme polymorphe, permettant à un concepteur de configurer l'environnement de collaboration pour l'adapter à un scénario d'utilisation précis (fig. 2.1).

Notre plateforme doit être assimilable à une boîte à outils offrant la possibilité de développer une application collaborative colocalisée, sans nécessiter d'importantes connaissances en programmation. La configuration logicielle doit ainsi se faire sous la forme de simples fichiers de configuration facilement éditables. Si la majorité des besoins sera implémentée dans la so-



FIGURE 2.1 – Illustration de la plateforme, présentée au salon professionnel Laval Virtual.

lution logicielle, dans le cadre de besoins nécessitant des modes d'interactions très particuliers, la structure de notre application doit également permettre de développer différents modules qui pourront se greffer naturellement aux composants pré-existants.

Nous pouvons classer les caractéristiques de notre plateforme en plusieurs points :

- **Adaptabilité au matériel**

Le choix des installations matérielles est un facteur qu'il est difficile d'imposer. En effet, les dispositifs spécialisés peuvent s'avérer couteux et les utilisateurs ne possèdent pas toujours les moyens financiers ni les infrastructures nécessaires pour accueillir des dispositifs imposants. De

plus, le choix des dispositifs est conditionné par les tâches à réaliser. En effet, la collaboration peut impliquer la nécessité de disposer d'une table interactive multitouch pour les interactions bidimensionnelles à réaliser en groupe, de stations de travail standards pour le travail individuel, de terminaux mobiles tels que les PDA ou les smartphones pour satisfaire les utilisateurs soumis à une importante mobilité dans la pièce, d'écrans à affichage stéréoscopique pour la visualisation d'objets 3D ou encore de divers périphériques d'interaction à retour d'effort pour les interactions liées à une simulation physique. Notre plateforme doit donc être massivement orientée vers une installation multi-dispositifs, mais également être conçue pour être adaptée efficacement à cette hétérogénéité matérielle et prendre en charge les multiples modalités d'interaction/visualisation nécessaires.

Le concepteur en charge de la configuration de la war room doit donc être libre de choisir les installations en fonction de leur pertinence et de leur disponibilité. Pour assurer le bon fonctionnement de la plateforme logicielle, une seule contrainte est toutefois indispensable : tous les dispositifs impliqués dans l'activité doivent être connectés aux autres à travers un réseau local (LAN), pour des raisons de performances (latence, débit). Les chapitres suivants traitant de l'architecture logicielle décriront plus en détail les mécanismes logiciels qui permettent cette adaptation multi-dispositifs.

• Schéma de répartition des utilisateurs : une approche différente des EVC

Dans le cas d'une configuration EVC classique, chaque utilisateur est disposé devant son propre poste de travail. Le nombre de dispositifs (en l'occurrence des ordinateurs individuels) est donc égal au nombre d'utilisateurs. Chaque participant est alors installé à un emplacement fixe et n'est pas amené à passer sur un autre poste pour procéder à une interaction, ni même à aller assister l'un de ses collaborateurs en s'installant au même poste. Il existe de cette façon un lien symbolique qui unit l'utilisateur à son dispositif personnel. Ce lien est créé au début de la collaboration et reste le même jusqu'à la fin de la session. Selon cette approche, le couple utilisateur-poste de travail constitue une seule et unique entité composée (fig. 2.2). La critique que l'on peut formuler vis-à-vis de cette configuration est que l'échange qui a lieu en terme de collaboration n'a pas à proprement parler lieu entre les utilisateurs, mais entre les couples utilisateur-PC.

Dans notre war room, le schéma de répartition des utilisateurs doit être différent. En effet, notre objectif est de s'affranchir au maximum de toute médiation dans la communication, mais aussi dans l'interaction. Les échanges collaboratifs n'ont plus lieu entre des entités composées, mais directement entre les utilisateurs. Le support informatique redevient alors un outil d'assistance et n'est plus assimilable à une contrainte définissant un "space between" entre les participants. Pour ce faire notre proposition implique que le nombre de postes de travail nécessaires au bon déroulement de l'activité collaborative est indépendant du nombre de participants prenant part à cette activité. Le placement des utilisateurs est donc libre et peut même aboutir à la formation d'un groupe dans le groupe, par exemple en disposant plusieurs utilisateurs autour d'un unique poste de travail.

Par ailleurs, la distribution des participants dans la war room évolue au cours de la session. Par conséquent, même s'il existe toujours un lien unissant un utilisateur à un poste de travail, ce lien perd sa dimension statique. En effet, au fur et à mesure du déroulement de l'activité, les liens peuvent se voir rompus, d'autres peuvent se créer et être modifiés à tout moment en fonction des exigences des différents participants. La figure 2.3 illustre plusieurs configurations

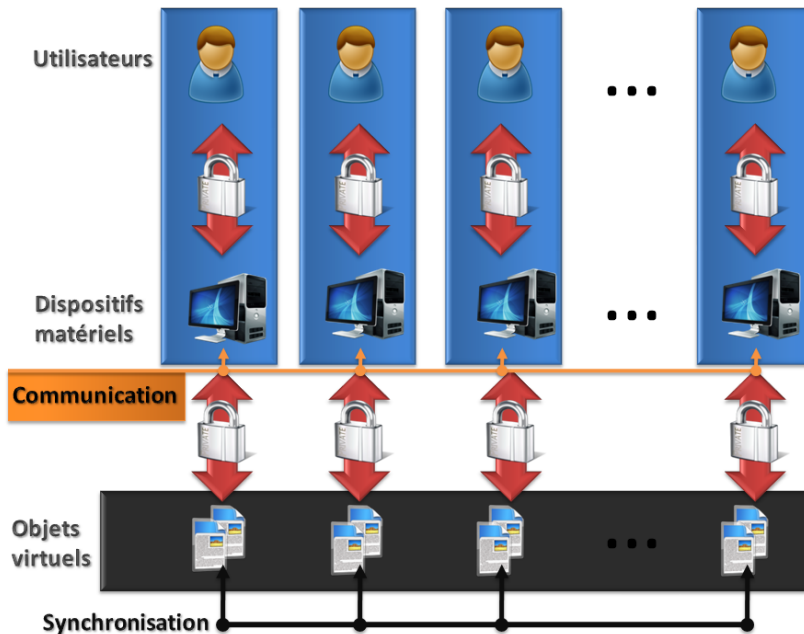


FIGURE 2.2 – Schema de répartition des utilisateurs dans le cadre d'un environnement de collaboration distante type EVC (note : peu importe le moyen utilisé pour la synchronisation)

imaginables dans la war room au cours d'une même session de travail collaboratif.

• Duplication des vues ou basculement d'un écran vers un autre

Les représentations visuelles qui prennent place dans le cadre d'une war room multi-dispositifs peuvent être multiples. Elles doivent en outre faire l'objet d'une dynamique de distribution singulière. Parmi les modes d'affichage qui doivent prendre place dans notre plateforme, nous pouvons définir trois configurations en particulier :

- Le premier mode de distribution des représentations est un système d'affichage unique pour un objet donné. Ce mode vise à résoudre les éventuels problèmes de mémorisation spatiale. En effet, l'affichage unique propose d'afficher la représentation d'un objet sur un seul poste de travail. Les utilisateurs peuvent ainsi associer une certaine tangibilité à l'objet virtuel en assimilant son emplacement dans l'espace. Cependant, ce mode de représentation n'offre que peu de dynamique au support logiciel, c'est pourquoi nous lui préférons la deuxième possibilité.
- Le deuxième mode consiste à conserver le caractère unique de la représentation de l'objet, mais en lui offrant une certaine dynamique. Les participants confrontés à une configuration de la plateforme proposant ce mode possèdent la même faculté de mémorisation spatiale, mais peuvent également déplacer virtuellement l'objet, c'est-à-dire déplacer sa représentation visuelle. L'affichage de l'objet concerné peut par conséquent être basculé d'un poste à

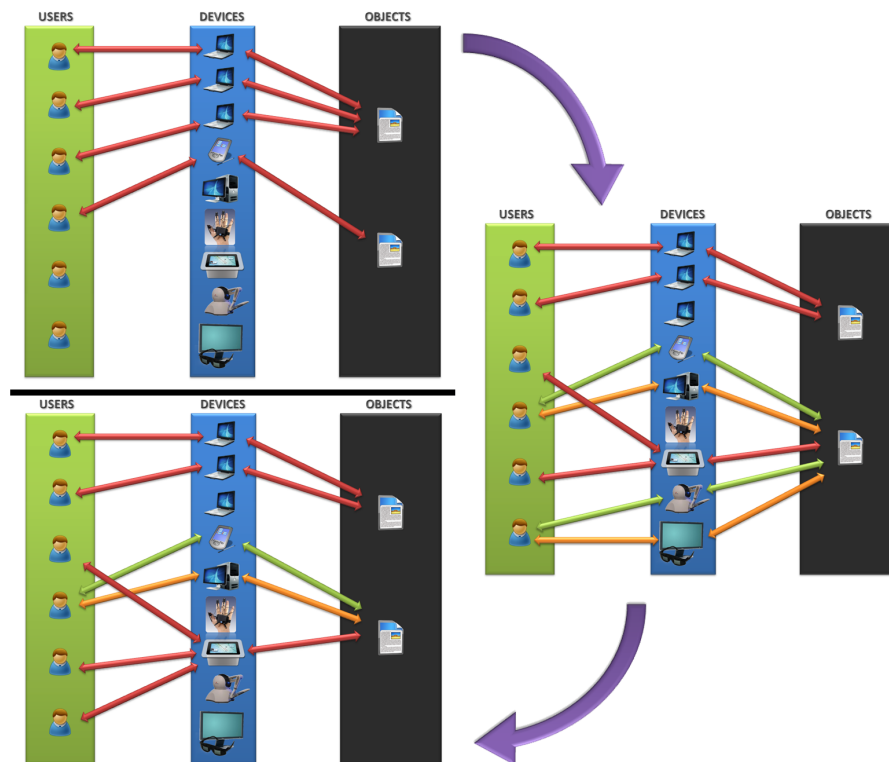


FIGURE 2.3 – Evolution du schéma de répartition des utilisateurs au cours de la collaboration dans la war room. Les utilisateurs sont amenés à changer de dispositif en fonction de leur activité, éventuellement en formant des sous groupes autour du même dispositif.

Légende :

feches rouges : interaction et visualisation

feches jaunes : visualisation seule

feches vertes : interaction seule

un autre, disparaissant de l'écran du poste d'origine pour apparaître sur le poste de destination. Cette métaphore tend à se rapprocher des war rooms classiques, dans lesquelles l'utilisation de documents papiers permet un déplacement spatial des informations dans la pièce (par exemple lorsque les utilisateurs prennent une feuille de notes sur la table pour l'épingler sur un tableau). Ce geste de déplacement de l'affichage ne vise par conséquent pas uniquement à changer de poste de travail pour bénéficier des périphériques d'interaction dudit poste, mais également à offrir une dynamique de déplacement spatial similaire à celle d'un document physique sur papier.

- Le troisième mode consiste à permettre l'affichage simultané du même objet virtuel sur plusieurs surfaces. Ainsi, deux participants travaillant sur deux postes de travail différents dans la pièce peuvent observer un même objet virtuel et interagir sur lui de façon simultanée, tout en conservant un haut degré de cohérence entre les représentations. Si l'un des utilisateurs procède à une interaction, ses conséquences seront alors automatiquement répercutées sur l'affichage du second collaborateur. Le mécanisme logiciel de préservation de la cohérence visuelle sera expliqué dans le chapitre 4.

Il est important de souligner que les trois modes de représentations présentés ici ne constituent pas un unique choix conditionnant la totalité de la plateforme, ces modes de représentations étant propres à un objet virtuel. Par conséquent, il est envisageable de proposer un environnement virtuel impliquant deux objets, dont l'un est affiché simultanément sur plusieurs postes, tandis que le second est représenté par une unique vue dynamique. De plus, un même objet virtuel doit pouvoir être soumis simultanément à plusieurs modes différents au sein de la war room. Il peut par exemple être représenté de façon persistante sur l'un des postes de travail et disposer d'une autre représentation simultanée qui elle, peut être déplacée dynamiquement entre les autres postes. La distribution des représentations visuelles des objets virtuels doit ainsi être complètement libre et les règles de diffusions de celles-ci seront soumises aux choix de conception de l'utilisateur.

• Séparation de l'interaction et de la visualisation

Le concept de la war room que nous proposons tend à la voir équipée de plusieurs périphériques d'interaction différents. Cette approche permet de multiplier les modalités d'interactions disponibles et d'obtenir des méthodes plus pertinentes en fonction de la tâche à réaliser. Cependant, il est nécessaire que cette diversité des outils matériels puisse être prise en charge efficacement par le support logiciel. La dynamique de collaboration offerte par la répartition des représentations visuelles vues dans le paragraphe précédent ne doit pas être rompue par une utilisation trop statique des périphériques d'interaction : sur une application traditionnelle, les stratégies de gestion des entrées utilisateurs ont souvent tendance à lier de façon étroite tous les échanges entre l'utilisateur et l'ordinateur à travers le concept d'interface utilisateur. Ainsi, classiquement, le poste de travail chargé d'assurer l'affichage de l'environnement virtuel est également la machine prenant en charge les actions de l'utilisateur sur cet environnement. Selon cette optique, si un utilisateur de la war room désire utiliser un périphérique d'interaction en particulier, il devra basculer l'affichage de l'objet virtuel sur l'ordinateur concerné. Cela illustre plusieurs contraintes relatives à la dynamique de la collaboration :

- Premièrement, l'interaction simultanée issue de plusieurs utilisateurs peut s'avérer nécessaire. Or, si le poste de travail utilisé ne dispose pas d'un jeu de périphériques d'interaction multi-utilisateurs, cette configuration s'avèrera impossible et obligera les participants à passer sur un poste de travail adapté aux entrées multiples. La modalité d'interaction proposée par les périphériques de ce nouveau poste de travail peut toutefois s'avérer mal adaptée à l'interaction désirée. La réponse envisageable pour contrer cette problématique peut consister à utiliser conjointement deux postes de travail affichant simultanément la même interface dupliquée. Cependant, cette solution n'est pas toujours acceptable. En effet, comme nous l'avons souligné dans l'état de l'art (cf. section 1.2.1.4), deux utilisateurs agissant conjointement sur le même document sont plus à même de collaborer efficacement s'ils utilisent le même support d'affichage. L'utilisation de deux postes distincts reviendrait ainsi à une situation proche de celle qui a lieu dans le cadre d'une collaboration distante via un EVC.
- De même, dans le cas d'une installation matérielle fortement hétérogène, l'utilisateur peut être amené à procéder à une utilisation reposant sur un assemblage interaction/visualisation différent de celui qui est proposé par chacun des dispositifs. Il est possible qu'un utilisateur soit désireux d'observer un objet virtuel via un écran en particulier. Dans de telles conditions, selon une application traditionnelle, cet utilisateur se verrait contraint d'interagir via les périphériques d'entrée liés au même poste de travail que l'écran. Cepen-

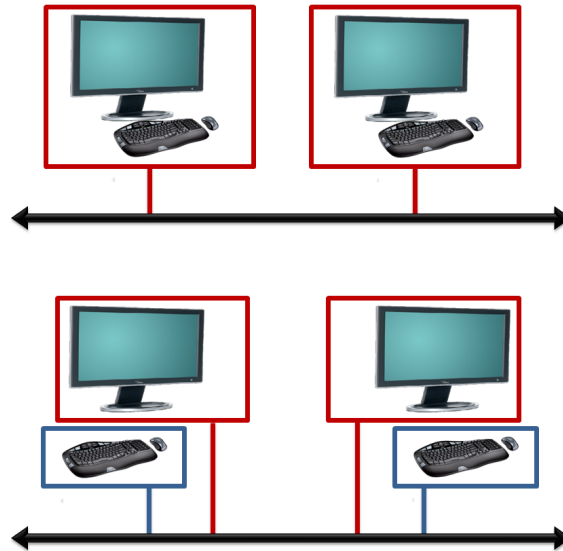


FIGURE 2.4 – En haut : une approche traditionnelle, qui considère les stations de travail dans leur ensemble. En bas : notre approche, qui distingue nettement l'aspect "interaction" de l'aspect "visualisation".

dant, ce même utilisateur pourrait éprouver le besoin d'utiliser un dispositif d'interaction différent, situé sur un autre poste de travail de la pièce. Nous considérons par conséquent qu'il est nécessaire d'éviter toute contrainte matérielle de ce type en permettant aux participants d'utiliser les périphériques d'interaction qu'ils désirent, tout en bénéficiant du retour visuel sur l'écran qu'ils désirent (sous réserve de compatibilité).

Pour éviter aux utilisateurs d'être confrontés à ces limitations, notre plateforme logicielle doit offrir d'autres possibilités de distribution de l'activité (fig. 2.4). Chaque poste de travail impliqué dans la war room peut être perçu selon deux facettes distinctes : la visualisation, qui est assimilée au dispositif d'affichage (écran ou vidéoprojecteur), et l'interaction, qui est représentée par les périphériques d'entrée connectés à cette station (clavier, souris, ou périphériques plus complexes). Un objet virtuel doit pouvoir être représenté simultanément sur plusieurs écrans, ou offrir aux utilisateurs la possibilité de déplacer cette représentation visuelle d'un écran à l'autre. Nous pouvons traduire ce besoin par une totale indépendance des représentations visuelles vis-à-vis de la configuration matérielle multi-dispositifs. La gestion de l'interaction doit elle aussi être soumise à une dynamique similaire en rendant l'interaction sur les objets indépendante des installations matérielles. Les utilisateurs doivent en effet être capables d'interagir sur un objet indifféremment de la localisation de sa représentation visuelle. Ils pourront ainsi exploiter les périphériques d'entrée d'un poste de travail pour interagir sur un objet affiché sur l'écran d'une autre station.

De plus, l'écoute des outils d'interaction par la plateforme ne doit pas être exclusive. Il sera ainsi envisageable de voir une même représentation visuelle soumise aux modifications issues de deux postes de travail à la fois. Deux utilisateurs pourront ainsi interagir simultanément sur l'objet affiché sur l'un des écrans de la pièce, même si seul l'un de ces utilisateurs utilise les périphériques d'entrée du poste d'affichage. D'un point de vue conceptuel, cet exemple est assimilable à l'utilisation d'un EVC exécuté sur un unique poste et doté de plusieurs périphériques d'entrée (fig. 2.5).

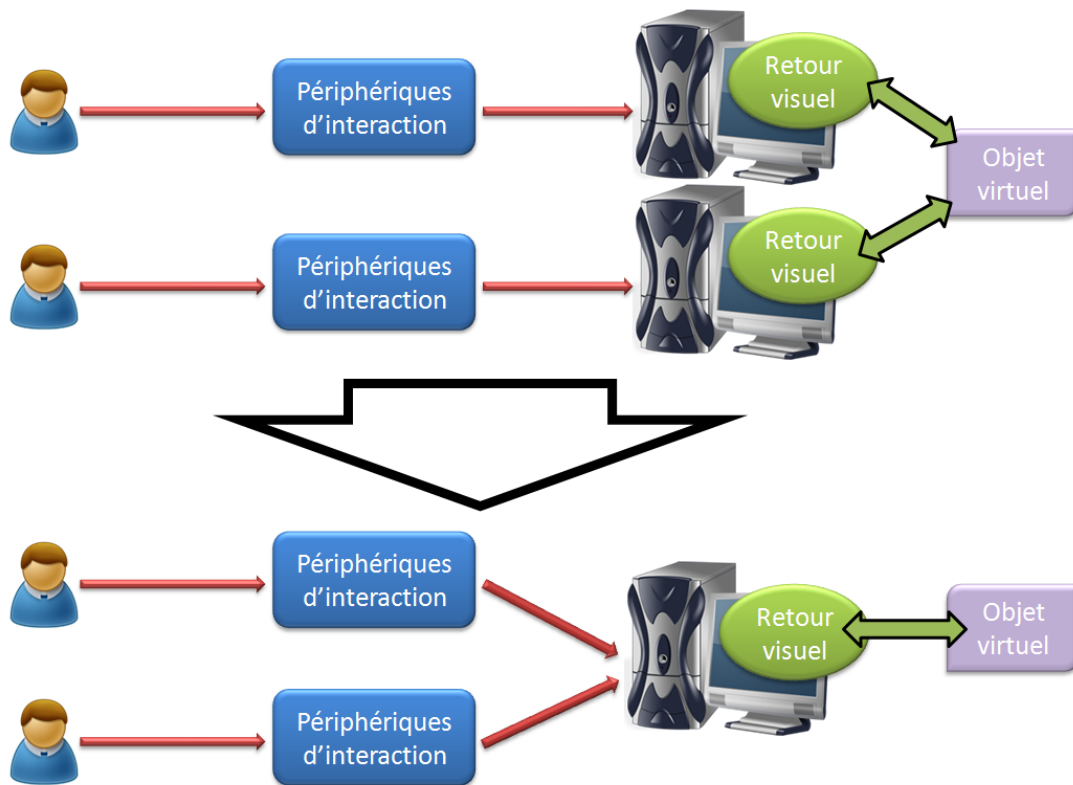


FIGURE 2.5 – Comparaison conceptuelle entre deux stratégies de collaboration à plusieurs dispositifs d'interaction

• Multiplicité des approches

Le système de représentations multiples que nous proposons permet d'intégrer plusieurs utilisateurs dans la session collaborative. La dynamique de distribution à mettre en place concernant ces vues permettra de servir la dynamique collaborative des utilisateurs. Cependant, ce système montre ses limites dès lors qu'il est question de prendre en charge le facteur d'individualité dans l'activité des participants. En effet, les représentations dupliquées d'un objet sont les mêmes. La différence perceptible entre les tâches d'interaction réalisées par deux utilisateurs sur le même objet apparaît dans la proposition actuelle sous deux formes :

La modalité d'interaction : grâce au procédé de distribution des outils d'interaction, les utilisateurs sont libres d'utiliser n'importe quel périphérique d'interaction disponible dans la pièce. Deux utilisateurs possédant des compétences (et par conséquent des méthodes de travail) différentes pourront alors interagir sur l'objet en utilisant des modalités d'interaction différentes.

Le wysiwis relâché : les représentations employées pour afficher l'objet sur deux surfaces simultanément sont les mêmes, puisqu'elles reposent sur la même définition des données relatives à cet objet. Cependant, si l'orientation et la position de l'objet dans l'interface graphique ne sont pas définis comme faisant partie des paramètres intrinsèques du projet,

leurs valeurs sont libres d'être indépendantes du reste de l'application. Dans ce cas de figure, l'objet représenté dans l'une des interfaces peut éventuellement se trouver positionné et orienté différemment dans les autres interfaces de la war room, sans pour autant rompre la cohérence du projet partagé.

Ces deux solutions permettant de gérer les besoins individuels des participants sont toutefois trop restrictives en terme de possibilités. La diversité des compétences des membres d'une équipe nécessite une hétérogénéité plus importante que la simple distinction de points de vue apportée par le wysiwis relâché. De même, les dispositifs matériels pour l'interaction, même lorsqu'ils sont hétérogènes, doivent être utilisés conjointement à des outils logiciels adaptés. Nous devons donc orienter notre proposition sur une plateforme logicielle qui va plus loin en terme d'hétérogénéité des approches des objets virtuels partagés. Cette nécessité nous pousse à proposer sur les dispositifs d'affichage de la war room plusieurs représentations concurrentes de chaque objet. Ces représentations, même si elles portent sur le même objet virtuel, doivent pouvoir être très différentes les unes des autres.

La duplication d'un objet sémantique sous la forme de plusieurs objets différents ne constitue toutefois pas une solution efficace : il faut en effet assurer une cohérence constante entre les approches de chaque utilisateur. De plus, chaque objet sémantique impliqué dans une activité étant unique, il est pertinent de conserver pour chaque objet virtuel associé son caractère unique. Pour apporter de la diversité dans les différentes approches d'un objet, la solution que nous proposons consiste à le doter de plusieurs facettes proposant chacune une représentation visuelle et des modalités d'interactions particulières. De cette manière, toute modification réalisée par un utilisateur sur l'objet virtuel sera appliquée directement sur l'unique instance de cet objet, se répercutant donc automatiquement sur toutes les autres facettes. La cohérence entre les différentes facettes sera par conséquent préservée de manière naturelle.

Notre approche va donc consister, pour chaque objet sémantique sur lequel les utilisateurs sont amenés à travailler, à proposer plusieurs modes d'interaction et de visualisation indépendants les uns des autres. Un objet virtuel de l'environnement de travail pourra donc être abordé par les utilisateurs de plusieurs façons différentes à travers des représentations visuelles hétérogènes et les modalités d'interaction associées. Nous introduisons ainsi la notion de "canal d'interaction/visualisation", que nous abrègerons par "canal" dans la suite de ce document. Un canal constitue un moyen de visualiser et d'interagir sur un objet selon un mode particulier, indépendamment de ses autres canaux. Il s'agit en quelque sorte d'associer à chaque objet virtuel un ensemble d'interfaces différentes, chacune étant dédiée à une approche particulière de l'objet sémantique. Par exemple, une activité impliquant de travailler sur des données numériques dans un contexte multicanal pourra offrir aux utilisateurs la possibilité de travailler à travers une feuille de calcul via un premier canal ou à travers un graphique via un autre canal. Ce procédé offre donc plusieurs approches simultanées différentes, sans pour autant remettre en question le caractère unique du jeu de données numériques qui pilote ces deux représentations.

L'utilisation de canaux hétérogènes permet donc de fournir un environnement de travail offrant des approches similaires aux habitudes de travail des différents corps de métiers impliqués dans l'activité. La cohésion globale de l'activité sera maintenue à tout moment puisque comme pour un environnement mono-utilisateur, les objets virtuels sont uniques.

L'architecture logicielle à mettre en place pour supporter cette non-uniformité de l'environnement virtuel diffusé dans la war room doit permettre en outre de gérer des interactions réalisées sur deux canaux différents tout en conservant un haut degré de cohérence dans les données définissant le projet. Les différents canaux représentés simultanément sur les postes de travail

doivent être mis à jour instantanément dès qu'une interaction est détectée, quel que soit le canal à travers lequel cette interaction a été réalisée.

Le système de canaux hétérogènes doit pouvoir être exploité de deux manières différentes :

- Apporter une vision différente du projet de façon à ce que chaque interaction impacte directement l'objet général. Les interactions réalisées sur l'objet à travers deux canaux distincts peuvent alors appliquer des modifications sur les mêmes données.
- Permettre à l'équipe de procéder à des interactions influant sur des données propres à un canal. Ce mode ne met pas l'accent sur la cohérence globale du projet, mais peut permettre aux utilisateurs d'ajuster leur approche de l'objet de façon à disposer d'un environnement plus adéquat, voire même de conditionner certains de leurs choix de conception. Le canal constitue dans ce cas une forme d'enrichissement de l'objet général : les données manipulées n'ont alors pas nécessairement d'influence sur la représentation des autres canaux et peuvent porter uniquement sur la représentation visuelle du canal courant.

Il faut préciser que l'existence de canaux simultanés ne constitue pas une contrainte à la mémorisation spatiale par les utilisateurs. En effet, bien que cette approche équivaut à représenter un même objet à des emplacements différents dans la pièce, les canaux utilisés dans la war room proposent des représentations suffisamment différentes pour être appréhendées distinctement par les participants. Dans le cas d'un projet consistant en l'assemblage de plusieurs pièces par exemple, la configuration des canaux offre suffisamment de liberté pour les utiliser de façon à afficher sur chaque canal une pièce différente. Cette manoeuvre maximiserait les possibilités d'assimilation de chaque pièce à un endroit de la war room.

De plus, l'utilisation de ces canaux peut fournir une information plus complète relative à la perception de l'activité des autres : les utilisateurs voyant leurs collaborateurs travailler sur l'un des canaux peuvent identifier de façon plus précise l'activité à laquelle ils sont en train de se livrer. En effet, si dans le cadre d'un projet de conception d'une télécommande par exemple, l'un des membres de l'équipe travaille à travers le canal représentant la télécommande sous l'angle de son schéma électronique plutôt que sous l'angle de son design 3D, ses collaborateurs l'observant identifieront immédiatement la nature de la tâche qu'il réalise.

• Collaboration distante : support de la présence mixte

Notre proposition s'articule autour de situations de collaborations dans lesquelles les utilisateurs sont tous en situation de coprésence. Cependant, pour couvrir le plus grand nombre de cas d'usages possible, il peut s'avérer pertinent de prendre en charge dans une certaine mesure la possibilité de collaborer à distance. En effet, notre problématique s'appuie sur le constat que dans la majorité des cas d'usage d'un outil de collaboration assistée par ordinateur, l'équipe de participants est constituée au moins en grande partie d'utilisateurs colocalisés. Toutefois, nous voulons également assister au mieux la réalisation d'un projet lorsque l'équipe est massivement pluridisciplinaire. Or, lorsqu'un travail nécessite l'intervention de personnes aux domaines d'expertise très précis, il peut s'avérer difficile de réunir simultanément la totalité des participants en un même lieu. Cette contrainte aboutit alors à une situation de présence mixte. Par ailleurs, la réalisation de projets complexes impliquant de lourdes tâches de prise de décision, il peut s'avérer nécessaire de faire intervenir ponctuellement un spécialiste distant. Ce participant spécialisé n'a

pas vocation à être intégré à l'équipe durant la totalité de l'activité de conception, mais peut être appelé pour être consulté sur un problème critique du projet.

Pour répondre à ce besoin, il semble judicieux de prendre en compte dans notre proposition la notion de communication distante. Sans pour autant constituer une situation standard d'utilisation de la plateforme que nous voulons mettre en place, la situation de présence mixte peut apporter aux collaborateurs la possibilité de faire intervenir de façon ponctuelle un expert distant pour apporter sa contribution à la résolution d'un problème critique. Dans cette optique, la représentation visuelle de l'environnement du côté de l'utilisateur isolé doit pouvoir bénéficier du concept de canaux de façon à tirer parti d'un environnement de travail adapté à la tâche à réaliser par cet expert distant. Pour aller plus loin dans ce concept, nous pouvons aller jusqu'à envisager la possibilité de connecter deux war rooms situées sur deux sites distincts. Cette approche nécessite que notre proposition soit complètement indépendante des installations matérielles, permettant ainsi de gérer des installations matérielles asymétriques. De cette manière, les war rooms qui entrent en communication ne seront pas soumises à l'obligation de disposer de la même configuration matérielle. De même, un utilisateur isolé pourra intégrer l'activité à partir d'un simple ordinateur personnel comme s'il était physiquement présent sur le site.

Lorsque l'on observe cette particularité de distribution de l'environnement virtuel, nous pouvons constater que son fonctionnement s'avère proche d'une application de type EVC distant classique mais à un niveau plus global. En effet, ce contexte d'utilisation reprend les concepts et les contraintes techniques propres aux situations de collaboration distante. La différence conceptuelle avec notre approche réside ici dans le fait que contrairement aux EVC, dans lesquels chaque entité collaborative est constituée d'un unique utilisateur utilisant un ordinateur personnel, la war room que nous proposons considère ces entités collaboratives sous une forme plus complexe. Une entité peut ici être constituée d'un nombre variable de dispositifs matériels différents, ainsi que d'un ensemble -également variable- d'utilisateurs.

D'un point de vue logiciel, cela signifie qu'un utilisateur isolé utilisant son poste de travail personnel ne constitue pas la seule forme que peut prendre une entité collaborative. Chaque entité peut au contraire être composée d'une plateforme war room complète, dotée de plusieurs interfaces, représentations et canaux d'interaction/visualisation.

• Utilisation de logiciels externes

En dépit du fait que notre war room sera conçue pour répondre à des besoins d'utilisation très divers, il est envisageable que certaines approches très spécifiques d'un projet impliquent l'utilisation d'outils logiciels particuliers. De plus, lorsque certains corps de métier impliqués dans la collaboration sont accoutumés à l'utilisation d'un logiciel particulier, la configuration d'un canal équivalent peut s'avérer fastidieuse et consommer un temps de développement superflu. Le système logiciel prenant en charge la war room doit par conséquent offrir la possibilité d'intégrer logiciel dans le panel d'outils de l'équipe.

Une interface de communication externe dédiée doit donc être incluse dans la plateforme pour permettre à une application tierce de se greffer au reste de la plateforme, comme s'il s'agissait d'un canal supplémentaire. Ainsi, l'application en question se verra capable d'interagir directement sur les données du projet, en utilisant les outils d'interaction et les représentations visuelles internes à cette application. La plateforme de la war room entretiendra alors avec l'application externe une relation assimilable à un système client-serveur.

La solution permettant cette connexion inter-application pose en revanche certaines contraintes. L'application externe que l'utilisateur désire utiliser doit être suffisamment modifiable pour supporter la connexion au flux de communication, que ce soit par le biais d'un plugin ou directement à partir de son code source.

2.3 Reflexions sur le concept de canaux de visualisation et d'interaction

En observant les différentes contraintes que notre proposition doit prendre en compte, nous pouvons constater que la notion de diversité doit être au centre de nos préoccupations. Cette diversité doit apparaître à la fois au niveau de la visualisation des objets de l'environnement virtuel, mais aussi au niveau de l'interaction sur ces objets. Les travaux de Schafer et al. [SB05] que nous avons évoqués dans la section 1.3.4 constituent une approche intéressante de la diversité des approches des utilisateurs. Cette contribution, en proposant aux participants de l'activité de percevoir l'environnement simultanément selon des modes différents (en l'occurrence, sous la forme d'environnements 2D ou 3D) fournit une hétérogénéité visuelle qui s'avère pertinente dans le cadre de notre proposition. Cette diversité s'avère propice à la prise en charge de la pluridisciplinarité des groupes d'utilisateurs telle que nous la considérons dans notre approche. L'environnement de travail que nous voulons fournir dans notre proposition doit donc reposer sur une certaine hétérogénéité en terme d'interaction et de visualisation, que nous conceptualisons ici à travers une notion que nous avons appelé les "canaux". Dans cette section, nous allons approfondir cette notion de "canaux", et illustrer les différentes possibilités qu'ils offrent en terme d'activité collaborative.

2.3.1 La structuration de l'activité dans la war room : le système de canaux

Le support logiciel qui prendra place dans notre proposition consiste à structurer l'activité collaborative selon une méthode particulière. Notre proposition met l'accent sur deux points :

- Premièrement, la simultanéité des tâches réalisées par chaque collaborateur. Cette interaction en parallèle de chaque corps de métier tend à optimiser la dynamique de collaboration, de façon à détecter en temps réel les incompatibilités dans les choix de conceptions réalisés par chacun. La principale contrainte liée au désir d'optimiser cette simultanéité réside dans la diversité des approches du projet : chaque participant de l'activité, de part son domaine de compétence, possède une forte tendance à aborder le même projet sous un angle différent de ses collaborateurs.
- Le second point réside dans la conservation de la cohérence et la communication étroite qui est établie entre les différents processus parallèles. En effet, selon notre approche, le projet sur lequel doivent travailler les utilisateurs est associé à un seul et même concept sémantique. Il n'est ici pas nécessaire de diviser le projet en plusieurs segments, puisque notre proposition opère en amont de la gestion des données définissant le projet. Le projet sur lequel l'équipe travaillera sera donc un objet sémantique unique, sur lequel chacun interagira en passant par un canal personnalisé.

Cette approche n'est bien sûr pas opposable à une structuration sous la forme de documents multiples : en effet, il est tout à fait possible qu'un projet soit constitué de plusieurs sous objets indépendants, ne nécessitant pas de mécanisme assurant la cohérence entre ces objets. Dans ce contexte, notre approche veille cependant à fournir pour chaque objet des représentations différentes, qui ne sont pas définies comme des sous-objets : les canaux.

2.3.2 Illustration de l'utilisation des canaux

Pour expliciter au mieux le concept des canaux, prenons un exemple simple.

Dans le cadre d'une activité de conception d'un téléphone portable, l'équipe pourra être composée de multiples participants aux compétences diverses :

- Spécialistes en électronique, qui assureront la conception des schémas des circuits, ou définiront les besoins en terme de composants.
- Designers, qui prendront en charge l'aspect esthétique du téléphone, à travers sa forme ou ses matériaux.
- Spécialistes en ergonomie, qui contrôleront l'utilisabilité de l'objet, notamment à travers l'analyse de sa forme et de la pertinence des choix vis-à-vis des interactions nécessaires avec l'appareil.
- Directeurs techniques, qui détermineront les décisions à prendre concernant les technologies à intégrer (écran tactile ou pavé numérique, présence ou non d'équipement de géolocalisation, autonomie nécessaire de la batterie, etc.) en assurant le respect du cahier des charges.
- Responsables fabrication, qui se verront impliqués dans les décisions à prendre concernant les matériaux et les modèles de composants, en fonction des coûts de fabrication.

Tout en étant extrêmement liés, le travail de chacun de ces participants compte une importante part d'individualité. Par exemple, le designer n'aura d'influence directe que sur l'aspect extérieur de l'objet. De même, les responsables de la partie électronique ne travailleront directement que sur la partie interne du téléphone. Cependant, ces deux domaines d'expertise doivent travailler en étroite collaboration l'un avec l'autre : en effet, le nombre, les dimensions et la forme des circuits et composants à intégrer au niveau électronique représentent des contraintes importantes vis-à-vis de la forme de la coque qui les contiendra. La réciproque est d'ailleurs également valide : si le designer prévoit une forme particulière reflétant l'image qu'il veut véhiculer du produit, le système électronique devra, dans la mesure du possible, voir son agencement interne adapté à la forme du contenant.

Le point clé de notre approche réside dans le fait que chacun de ces spécialistes aura une vision différente du téléphone : tandis que le responsable en électronique étudiera la conception à travers la réalisation de schémas techniques éditables, le designer l'abordera sous l'angle d'un modèle 3D qu'il pourra manipuler et sculpter à sa guise.

Dans notre application, le système des canaux vise à reproduire cette situation en adaptant l'environnement virtuel en fonction des besoins de l'utilisateur. Dans le cas de notre exemple,

l'objet représentant le projet (en l'occurrence, le téléphone) disposera de plusieurs canaux :

- Un premier, offrant une représentation du téléphone au sens électronique du terme, c'est-à-dire sous la forme de schémas 2D permettant d'éditer la structure interne de l'objet.
- Un deuxième, présentant l'objet sous la forme un modèle 3D et doté d'outils qui illustre l'environnement de travail du designer.
- Un troisième, illustrant le téléphone sous la forme d'une fiche technique listant les différentes pièces, leur modèle et leur prix. Ce document constituera ainsi une version plus synthétique et plus proche des besoins du responsable fabrication.

Ces trois canaux représentent visuellement le même objet abstrait (ils sont d'ailleurs tous nécessaires pour faire une description exhaustive de l'objet conçu par l'équipe). Ils ne peuvent donc pas à proprement parler être considérés comme indépendants et sont pourtant très différents les uns des autres en terme de représentation.

Le fait que plusieurs canaux soient liés au même objet sémantique permet à notre application d'assurer de façon plus naturelle le maintien de la cohérence de l'ensemble. En effet, la mise en commun des informations relatives au design et à l'électronique au sein d'un même objet sémantique fournit une structure de données capable de contrôler la pertinence de chaque choix, par exemple en construisant un modèle général dont la représentation permettra de considérer le projet en bénéficiant d'une reconstruction simultanée des deux domaines.

Concrètement, dans notre exemple, la représentation généraliste de cet objet sémantique peut consister en une construction 3D du téléphone comprenant à la fois sa coque et son système électronique. Cette construction visuelle hybride ne propose pas d'approche particulièrement pertinente vis-à-vis d'un spécialiste en particulier, mais peut s'avérer utile pour s'assurer visuellement qu'aucune collision n'est visible entre le circuit électronique et le boîtier. L'installation multi-dispositifs de notre war room est par conséquent toute indiquée pour exploiter cette configuration, puisqu'elle permet d'afficher dans la même pièce les canaux spécialisés et la représentation généraliste (fig. 2.6).

La diversité des canaux n'opère pas uniquement au niveau de la représentation, mais aussi de l'interaction : chaque représentation implique des modalités d'interaction qui lui sont propres. S'il est possible de factoriser certaines méthodes d'interaction de façon à les rendre similaires sur plusieurs canaux simultanément, la majorité des outils de production sont dépendants du mode de représentation utilisé. Par exemple, le canal dédié au designer doit offrir des outils de déformation pour modeler la coque du téléphone, tandis que le canal dédié à la partie électronique doit fournir des outils permettant d'éditer les tracés de circuits entre les composants.

La notion de canal consiste ainsi en une entité composée permettant de convertir les échanges entre un participant et le projet, pour leur donner des significations :

- Plus concrètes pour les représentations visuelles : l'objet abstrait "téléphone" est convertit en une représentation compréhensible par l'utilisateur.
- Plus abstraite pour les interactions : les actions concrètes réalisées par l'utilisateur sur une donnée visuelle sont répercutées de façon cohérente sur la description abstraite du projet.

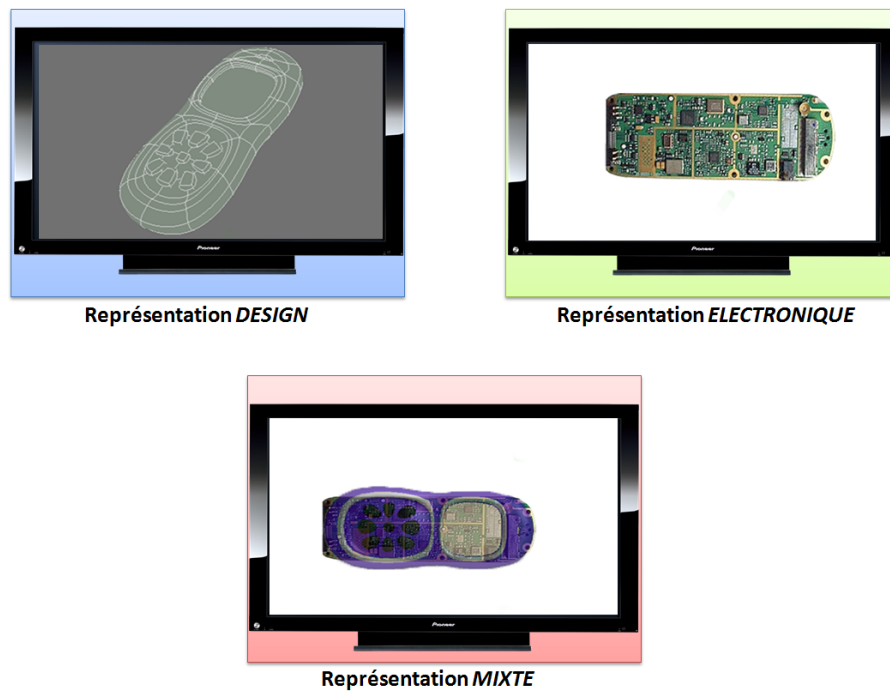


FIGURE 2.6 – Trois représentations possibles d'un projet de conception d'un téléphone. Une première représentation propose l'objet 3D de la coque, une deuxième propose le schéma électronique et la troisième est une représentation combinée des deux précédentes, permettant de contrôler la cohérence globale du projet

Dans un cas idéal, chaque canal peut être utilisé pour reconstruire un clone des applications que les participants ont l'habitude d'utiliser dans leur environnement de travail habituel. Cependant, il peut parfois s'avérer plus abordable de simplifier ces environnements logiciels pour les doter uniquement d'outils principaux et s'affranchir de ceux qui ne présentent pas d'intérêt dans le cadre de la collaboration. En effet, notre application n'a pas pour objectif de combler la totalité des besoins du workflow, puisqu'elle ne se situe pas dans le cadre de la production, mais plutôt de la conception. Néanmoins, l'intérêt de ces multiples canaux au sein de la plateforme logicielle de la war room possède un certain nombre d'avantages pour les utilisateurs :

- Chaque spécialiste de l'équipe confronté à un canal proche de son environnement logiciel habituel reprendra plus facilement ses habitudes de travail, et réalisera des actions plus précises qu'à travers une représentation généraliste. Le niveau de spécialisation du canal aura donc un impact potentiel sur la collaboration : si tous les utilisateurs travaillent sur une même approche généraliste, le manque de diversité des outils peut les pousser à approximer certaines de leurs actions. Cette approximation peut par conséquent provoquer une marge d'erreur dans le contrôle de la compatibilité des actions de chacun.
- La connaissance de l'environnement logiciel leur permettra d'identifier plus facilement les problèmes, puisque la représentation visuelle leur sera connue.
- La proximité conceptuelle entre le canal et l'environnement logiciel habituel offre aux uti-

lisateurs une meilleure capacité d'anticipation des problèmes. En effet, ils seront capables de mapper leurs méthodes de travail habituelles sur le canal et planifieront mieux la suite des actions qu'ils seront amenés à réaliser au cours du projet.

- Les participants pourront simuler instantanément leurs propositions comme s'ils étaient devant leur poste de travail habituel. Ils pourront alors envisager certaines solutions et contrôler immédiatement si ces solutions sont compatibles avec le travail réalisé par les autres.

2.3.3 Les canaux à l'échelle des données : approche sémantique des interactions

L'exemple donné ci-dessus illustre le fait de définir un objet abstrait comme étant un important jeu de données. Cependant, les informations bas niveau (i.e. les données d'ordre visuel) ne possèdent qu'un impact indirect sur les autres données. En effet, lorsque le designer modifie la forme du boîtier du téléphone, il ne modifie directement aucune information relative à l'électronique. Le lien qui unit les données de la forme du boîtier et celles définissant l'aspect électronique peut alors être qualifié de lien sémantique. Le système de canaux présenté jusqu'à maintenant opérait donc à l'échelle du projet.

Cependant, les cas d'utilisation réels se limitent rarement à ce type d'impact. En effet, lors de la réalisation d'un projet complexe, deux participants, même s'ils sont issus de deux domaines d'expertise différents, peuvent être amenés à travailler sur la même information, quel que soit son mode de représentation. Prenons pour exemple simple une situation de collaboration selon laquelle plusieurs utilisateurs travaillent conjointement sur la conception de la carrosserie d'une automobile. La war room accueillant cette équipe peut proposer deux canaux simultanés pour aborder cette voiture. Ainsi, tandis que le premier canal fournit une interface de modélisation surfacique 3D, le second canal permettra d'observer le projet sous la forme d'une fiche technique représentant ses caractéristiques aérodynamiques et de "tenue de route". Lorsque l'un des utilisateurs utilise la représentation 3D pour modifier par exemple le degré d'inclinaison d'un aileron, il procède en fait à une simple modification d'une information graphique, en l'occurrence une valeur d'angle. Cependant, son action possède une portée sémantique plus importante : à travers ce geste, l'utilisateur désire modifier la force appliquée par cet aileron, c'est-à-dire modifier l'une des caractéristiques abstraites relatives à la voiture. Il est donc légitime de voir l'impact de cette interaction reporté sur le canal présentant une fiche technique. Cependant, sans préservation de la cohérence à l'échelle de la donnée représentant cette force, la modification qui découle de la manipulation de l'aileron ne sera pas perceptible sur la fiche technique. Le système de gestion des canaux de notre application doit donc permettre de maintenir le lien direct qui existe entre ces deux informations. Ainsi, lorsqu'un canal est utilisé pour modifier une information d'ordre géométrique, tous les autres canaux seront potentiellement mis à jour en conséquence.

Grâce à la structuration de l'activité sous forme de canaux hétérogènes, les interactions réalisées dans notre plateforme par n'importe quel spécialiste pourront alors avoir des répercussions directes et immédiates sur l'interface de tous les utilisateurs, indifféremment du canal employé, permettant ainsi à l'ensemble des participants de détecter immédiatement les conflits.

2.3.4 Les canaux pour donner une portée concrète à des informations non représentables

Ce lien entre l'approche visuelle et l'approche abstraite d'un projet peut également être abordé dans l'autre sens. En effet, le système de canaux peut également être utilisé pour répondre à une contrainte de modalité d'interaction : si un projet est défini par des informations conceptuelles, ces données ne sont pas toujours utilisables directement pour offrir un retour visuel au sein de la war room. L'interaction sur cette donnée sera par conséquent impossible. Une approche traditionnelle viserait alors à fournir l'information sous sa forme brute, par exemple en affichant une donnée numérique dans l'interface. Dans ce contexte, les canaux peuvent au contraire permettre de retranscrire cette information abstraite de façon à la rendre perceptible plus naturellement pour les utilisateurs. Ainsi, la donnée abstraite constituée d'une simple donnée numérique peut prendre une signification plus visuelle, par exemple à travers l'affichage d'artefacts visuels représentant métaphoriquement cette donnée. Elle peut être également représentée selon un mode moins métaphorique, par exemple en l'associant à une information visuelle concrète (comme c'est le cas lorsque l'on représente les caractéristiques aérodynamiques sous la forme d'un aileron).

2.3.5 Les canaux, notion distincte de l'interface

Nous avons défini les canaux comme étant un mode de représentation et d'interaction sur un objet. Dans les exemples cités précédemment, cette définition semble proche de la notion d'interface utilisateur d'une application. Dans l'application mixte proposée par Schafer [SB05] par exemple, l'environnement commun est représenté sous la forme d'une interface utilisateur 2D et d'un interface 3D. Cette approche est cependant différente de la notion de canal que nous avons introduite :

la plateforme de Schafer affiche dans chaque interface utilisateur l'intégralité de l'environnement virtuel de la même façon. Ainsi, deux objets virtuels présentés dans l'interface 2D seront obligatoirement affichés en 2D. Dans ce cas de figure, la plateforme est hétérogène, mais chaque terminal (et donc chaque interface) est uniforme puisqu'il ne propose qu'un seul mode de représentation.

L'hétérogénéité permise par le système de canaux, au contraire, ne s'applique pas obligatoirement à l'ensemble de l'interface utilisateur d'un terminal. Elle peut en effet s'appliquer sur les objets virtuels de façon indépendante. Il est ainsi possible, grâce à cette approche, de proposer une représentation non-uniforme au sein d'une même interface utilisateur. Concrètement, cela peut se traduire par la mise en place sur l'un des dispositifs d'une interface unique reposant simultanément sur plusieurs canaux de visualisation/interaction.

Par exemple, une interface utilisateur mixte multi-canal peut proposer d'afficher simultanément deux objets, l'un sous la forme d'un modèle 3D (ie. à travers un canal 3D), et l'autre sous la forme d'un plan technique 2D. Dans ce cas de figure, l'interface est unique, mais est constituée de plusieurs canaux différents (3D et 2D). Cet exemple met en évidence le fait que la notion de canal n'est pas à proprement parler assimilable à l'interface utilisateur du terminal, mais bien au mode de représentation de l'objet au sein de cette interface. Bien entendu, il est nécessaire que l'interface proposée soit à même de supporter les deux modes de représentation.

Pour aller plus loin dans cette idée, il est également possible d'envisager la représentation simultanée d'un même objet virtuel selon deux canaux différents au sein de la même interface.

Par exemple, il est possible de mettre en oeuvre dans la war room un terminal proposant de manipuler un téléphone sous la forme de deux objets visuels distincts tels qu'un modèle 3D et une fiche technique. Dans ce cas de figure, il est impossible de qualifier l'interface comme étant "dédiée au canal X", puisqu'elle permet d'en manipuler deux à la fois. Elle peut alors être qualifiée d'interface "multi-canaux".

La segmentation conceptuelle de la plateforme que nous souhaitons mettre en place dans la war room peut donc être considérée selon plusieurs schémas :

- Une segmentation selon les différents canaux disponibles pour les objets
- Une segmentation selon les interfaces-utilisateur déployées
- Une segmentation selon les postes de travail disposés dans l'environnement physique

Il est intéressant de souligner que cette segmentation n'est jamais réalisée en fonction des utilisateurs. Cette constatation met en évidence le fait que selon notre approche, le groupe d'utilisateurs est considéré comme une unique entité pluridisciplinaire.

2.3.6 Les canaux : bilan

Comme nous pouvons le constater, la structuration du projet sous la forme d'un unique paradigme composé de plusieurs représentations concurrentes constitue une approche particulièrement intéressante dans le cadre d'un environnement colocalisé tel qu'une war room. En effet, la facilité, la dynamique et la nature directe des échanges interpersonnels qui ont lieu dans un tel contexte nous ouvre des perspectives nouvelles en matière de manipulation simultanée des éléments virtuels. Il nous apparaît ainsi pertinent de bâtir notre plateforme sur cette diversité des approches d'un projet.

Pour prendre en charge l'hétérogénéité des compétences impliquées dans l'activité collaborative, de même que la dynamique spatiale possible dans une war room, nous prenons par conséquent le parti d'établir la principale spécificité de notre approche sur cette notion de canaux, les plaçant ainsi au rang de caractéristique majeure dans la prise en charge de notre contexte d'utilisation.

2.4 Conclusion de notre analyse

Comme nous avons pu le constater au travers de cette analyse, outre plusieurs signes distinctifs propres à notre approche, les besoins relatifs à une plateforme dédiée à la collaboration colocalisée possèdent d'importantes similitudes avec les caractéristiques des environnements virtuels collaboratifs classiques. En effet, selon notre approche, le projet sur lequel les utilisateurs sont amenés à interagir est constitué d'un environnement virtuel constituant un espace de travail à part entière. De même, dans nos cas d'utilisation, ce projet est représenté au moins en partie par des procédés d'affichage 3D. Le déroulement de ces interactions sur les objets virtuels fonctionne également grâce à des mécanismes similaires aux EVC existants. De plus, de part la diversité des plateformes EVC actuelles, de nombreux concepts de la collaboration et de la communication sont d'ores et déjà disponibles.

La question que l'on est alors amené à se poser consiste à s'interroger sur les possibilités de réappropriation des EVC pour les adapter aux situations de coprésence des utilisateurs. Est-il possible de modifier une plateforme EVC existante pour l'enrichir avec des caractéristiques et

des fonctionnalités lui permettant de prendre en charge la totalité des besoins potentiels d'un contexte colocalisé de type "war room" ? Les fondements logiciels sur lesquels nous avons bâti les environnements collaboratifs distants jusqu'à aujourd'hui restent-ils pertinents lorsqu'il est question de développer une application spécialisée sur le travail d'une équipe localisée dans le même espace physique ?

APPROCHE LOGICIELLE DE LA COLLABORATION : ÉTAT DE L'ART

Sommaire

3.1	Introduction : distinction entre la définition et la représentation d'un objet virtuel	86
3.2	Caractéristiques logicielles des EVC	86
3.2.1	La notion d'objets partagés	86
3.2.2	Technologies : la distribution des données	87
3.2.3	Flux de communication et cohérence	90
3.2.4	Taxonomie et topologie	91
3.3	Les architectures logicielles	93
3.3.1	Les modèles de conception d'applications interactives	93
3.3.1.1	Les modèles monolithiques	94
3.3.1.2	Les modèles à agents	97
3.3.1.3	Les hybrides	100
3.3.2	Nouvelle approche de l'interaction : vers l'interaction instrumentale . . .	102
3.3.3	SPIN 3D : une architecture logicielle singulière	103
3.4	Les systèmes interactifs complexes : les EVC	106
3.4.1	Virtools	107
3.4.2	OpenMask	108
3.4.3	SPIN 3D	109
3.4.4	Conclusion	110

3.1 Introduction : distinction entre la définition et la représentation d'un objet virtuel

Dans une application interactive quelle qu'elle soit, l'utilisateur est amené à interagir sur des objets. Ces objets sont les éléments à travers lesquels la tâche sera accomplie, soit par manipulation directe, soit par modification via des outils d'interaction, ou bien par un assemblage permettant d'aboutir à un résultat. Pour permettre l'utilisation d'une interface graphique, ces objets doivent bien entendu disposer d'une représentation visuelle, symbolique ou non. Cependant, si la structuration des processus et des traitements internes est trop dépendante de l'aspect visuel, ceci peut constituer un frein important à la diversité des actions que cette application sera en mesure de réaliser.

La conception de la structure d'une application doit reposer sur les traitements à effectuer plutôt que sur l'agencement des éléments graphiques. Il faut par conséquent établir une distinction entre les objets (au sens sémantique du terme) et leur représentation visuelle. Dans cette optique, le domaine de l'architecture logicielle a développé de nombreuses approches permettant de construire une application à partir des traitements qu'elle doit effectuer, tout en gardant la possibilité de greffer à cette structure le système de représentation visuelle nécessaire à la constitution de l'interface graphique. Cette problématique se positionne donc au coeur de nos travaux, puisque notre proposition d'application collaborative distingue nettement les objets virtuels visibles de la partie fonctionnelle de l'application interactive.

Ce chapitre constitue un état de l'art concernant les architectures logicielles dédiées aux interfaces pour les environnements interactifs et plus spécifiquement les environnements virtuels. Ceci nous permettra d'explorer les possibilités qui s'offrent à nous et de déterminer comment il nous est possible de tirer parti de l'une de ces architectures pour répondre au mieux aux besoins de notre application.

Dans un premier temps, il importe de faire le point sur les concepts principaux des applications collaboratives. Ces applications constituent en effet des solutions reposant sur des problématiques proches de nos besoins : comme pour notre proposition, il est en effet question dans ces applications de mettre en relation plusieurs ordinateurs pour partager et interagir sur des données.

3.2 Caractéristiques logicielles des EVC

3.2.1 La notion d'objets partagés

L'un des concepts fondamentaux sur lequel repose un EVC est la notion d'objet partagé. A l'instar d'un objet du monde réel, un objet partagé est un élément de l'environnement virtuel qui est visible et manipulable par les différents utilisateurs participant à la collaboration. Cet objet virtuel est caractérisé par un certain nombre de propriétés qui le définissent. Les propriétés d'un objet partagé sont ainsi exposées à tous les participants, qui peuvent les modifier via leurs interactions. La question technique qui découle de ce concept réside dans la conservation de la cohérence de ces propriétés : si la plupart des systèmes de coopération distantes prennent le parti de dupliquer les données sur tous les postes de travail (essentiellement pour des raisons de performances), il est nécessaire de mettre en place des mécanismes assurant que les propriétés des objets aient la même valeur pour tous les utilisateurs, et ce, à chaque instant.

Pour maintenir la cohérence entre les différents ordinateurs impliqués dans l'activité collaborative (dans un tel contexte, ces ordinateurs sont appelés des "noeuds"), il est nécessaire d'assurer

la consistance des données entre les noeuds. Selon Delaney et al. [DWM06], cette consistance est caractérisée par trois critères :

- **La synchronisation** : elle apparaît sous deux formes. La synchronisation temporelle vise à assurer la simultanéité des événements déclenchés sur chaque noeud. La synchronisation spatiale consiste à assurer qu'à tout instant un objet sera positionné au même endroit dans l'environnement virtuel affiché par chaque noeud.
- **La causalité** : l'ordre d'exécution des événements doit être le même sur tous les noeuds
- **La concurrence** : ce critère porte sur les interactions conflictuelles. Il faut s'assurer qu'une même donnée partagée ne soit pas modifiée simultanément par deux utilisateurs distincts, ou mettre en place un mécanisme visant à moyenniser les deux interactions de manière à produire un unique "événement".

De plus, selon les applications, la gestion de la consistance de l'environnement peut dépendre des choix des utilisateurs : il est possible qu'un utilisateur décide de travailler en mode privé, c'est-à-dire de telle sorte que ses interactions et leurs conséquences sur les objets ne soient pas visibles par les autres participants. Dans ce mode, les mécanismes de conservation de cohérence doivent prendre en compte ce paramètre et stopper la synchronisation entre les propriétés dupliquées.

3.2.2 Technologies : la distribution des données

La distribution des données consiste à définir de quelle façon vont être organisées les informations relatives à l'environnement virtuel et surtout comment elles vont être mises à disposition des utilisateurs. La façon dont sont distribuées les données est étroitement liée à la structure des communications réseau qui seront décrites en section 3.2.4.

S. Bangay propose de distinguer 4 types de distribution de données [S.96] :

- **La base de données centralisée** : elle consiste à stocker l'intégralité des données au même endroit, sur un serveur qui fournit les informations à tous ses clients (fig. 3.1). Ce type de distribution garantit une cohérence constante puisque les données n'existent que sous la forme d'une seule instance. Les clients connectés à ce serveur sont liés à ces données distantes pour afficher l'environnement et lorsqu'une interaction a lieu depuis un poste client, les modifications sont opérées par le serveur directement sur les données qu'il prend en charge. Les données sont alors diffusées à tous les clients. Cette architecture facilitant le mécanisme de conservation de la cohérence s'avère toutefois problématique dans le cadre d'un environnement virtuel complexe ou incluant un grand nombre de participants : les environnements 3D reposent sur un nombre important de données, qui doivent être consultées à une fréquence élevée pour que le client procède à l'affichage [MF98]. La limite basse acceptable pour garantir un certain confort d'utilisation est évaluée à 25 images par secondes. Delaney et al. [DWM06] estiment qu'une latence comprise entre 40 et 300ms permet d'assurer une collaboration temps-réel.
- **La base de données dupliquée à cohérence faible** : ce mode consiste à fournir à chaque participant une copie de la base de données, cette copie pouvant être complète ou

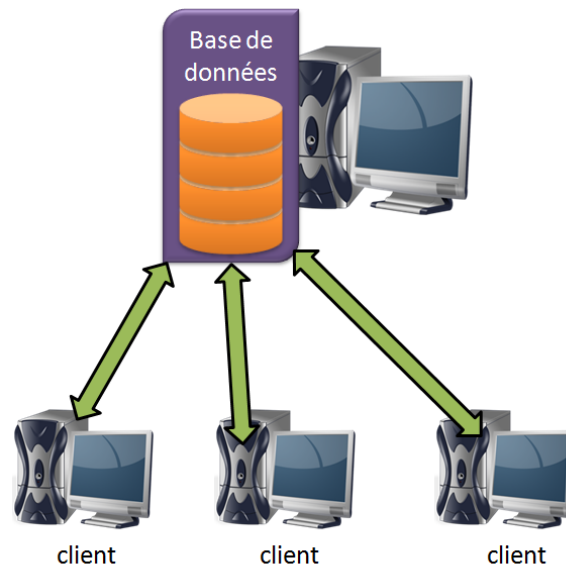


FIGURE 3.1 – Base de données centralisée

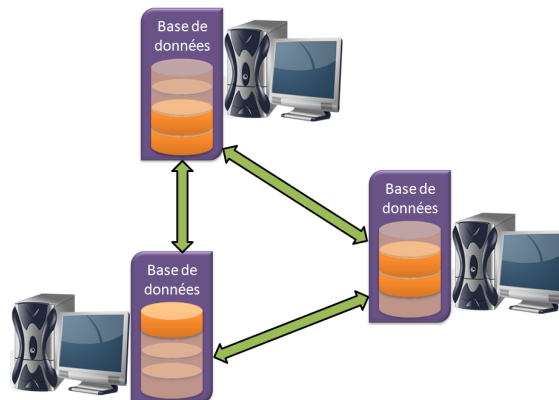


FIGURE 3.2 – Base de données dupliquées à cohérence faible

partielle (fig. 3.2). La cohérence est maintenue par l'envoi régulier de messages de mise à jour entre les clients. La duplication des données offre l'avantage de sa robustesse vis à vis des défaillances potentielles des dispositifs matériels. Elle est d'ailleurs plus performante qu'une base centralisée puisqu'elle nécessite moins de communications réseaux. Elle souffre cependant d'une certaine latence dans la conservation de la cohérence et n'est pas pertinente pour toutes les données.

- **La base de données dupliquée à cohérence forte** : dans ce mode, les bases sont complètes et les données stockées par chaque poste client sont synchronisées avec les autres à chaque instant (fig. 3.3). La base dupliquée à cohérence forte possède donc les mêmes caractéristiques que celle à cohérence faible, à la différence que les données homologues sont liées par un haut degré de synchronisation. Cette synchronisation a cependant un

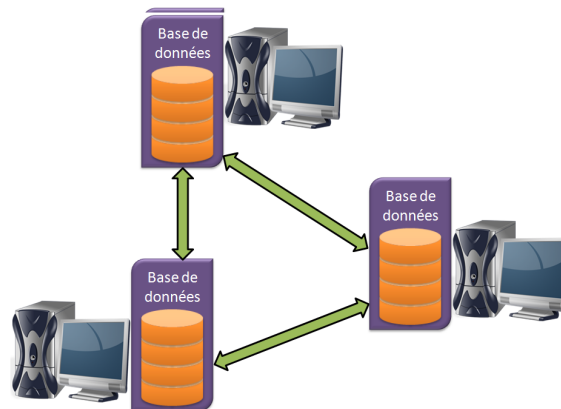


FIGURE 3.3 – Base de données dupliquées à cohérence forte

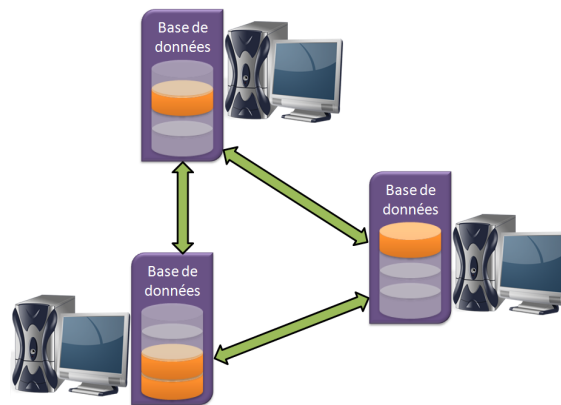


FIGURE 3.4 – Base de données distribuée

coût en terme de bande passante ; un compromis doit être fait entre la cohérence et les performances.

- **la base de données distribuée** : elle est, comme la base centralisée, constituée de données dont il n'existe qu'une seule instance. La nuance réside ici dans le fait que la base distribuée est partitionnée entre plusieurs serveurs (fig. 3.4). Au même titre que la base de données centralisée, cette approche est susceptible d'occasionner une latence dans l'interaction en raison du nombre de communications réseau nécessaires. Les performances sont toutefois meilleures sur ce modèle, puisqu'il n'existe pas qu'un unique serveur. Ceci limite donc le phénomène de goulot d'étranglement.

L'utilisation d'une base de données dupliquée rend l'ensemble résistant aux pannes : la disparition inopinée d'une machine du réseau n'aura pas d'influence sur la cohérence des données, puisqu'elles existent sur tous les postes. La mise à l'échelle d'un tel système s'avère toutefois compliquée, dans la mesure où la modification d'une donnée implique la mise à jour de tous les postes, provoquant ainsi un coût réseau important.

Ce problème n'apparaît pas lors de l'utilisation d'une base de données distribuée, puisqu'une modification de donnée ne concerne que deux postes au maximum (le poste à l'origine de cette modification, et celui qui effectue la mise à jour). Cependant, ce système est, au même titre que la base centralisée, sensible aux pannes. Pour résoudre ce problème, il est possible d'utiliser une base de données redondante, c'est-à-dire une base distribuée dont les données sont stockées sur plusieurs postes à la fois.

3.2.3 Flux de communication et cohérence

Dans le cadre d'une application collaborative reposant essentiellement sur l'échange de données entre les postes, la notion de flux de communication revêt une importance capitale dans la conception de la plateforme d'interaction. D'une part, le système doit se révéler suffisamment réactif pour offrir à l'utilisateur une interaction fluide lorsqu'il manipule un objet partagé. D'autre part, l'ensemble des informations relatives aux objets partagés nécessitent une constante cohérence entre les différents postes de travail impliqués, de façon à éviter toute incompréhension de la part des collaborateurs. Il est alors possible de distinguer trois types de données, qui constituent trois flux de communication différents [PIC03] :

- les données ponctuelles qui sont liées au contrôle de l'interface ou à toute action ponctuelle de la part de l'utilisateur. C'est, par exemple, le cas d'un clic sur un bouton. Ce type de données nécessite de maintenir une cohérence forte. C'est-à-dire qu'elles doivent être transmises instantanément de façon fiable entre les différents postes de travail.
- les flux d'informations qui sont continus et illimités dans le temps. Les flux de données issues d'une communication audio ou vidéo appartiennent à cette catégorie. Ces données sont transmises en continu et ne reposent pas sur une valeur incrémentale. La perte d'une partie du flux d'informations n'a pas d'incidence sur les données suivantes à transmettre. La transmission des flux de ce type peut ainsi reposer sur une cohérence relâchée. Ce type de cohérence permet une certaine tolérance vis-à-vis des pertes potentielles lors de la communication : les données transmises après une perte permettent alors de rétablir la consistance de l'information.
- les flux d'informations qui correspondent aux évolutions continues et limitées dans le temps, telles que les animations des objets partagés de l'environnement virtuel. Ces informations peuvent reposer sur une cohérence hybride entre les deux types précédents : lorsqu'un utilisateur manipule un objet partagé, la cohérence de l'ensemble du système repose souvent principalement sur la fiabilité de la dernière information, en l'occurrence la position finale de l'objet. Cette valeur nécessite ainsi une cohérence forte. Les valeurs intermédiaires de positionnement de cet objet, quant à elles, ne servent qu'à améliorer la compréhension de l'activité entre les utilisateurs. La transmission de ces données intermédiaires peut tolérer une cohérence relâchée.

Il est cependant nécessaire d'émettre une certaine réserve vis-à-vis de la tolérance relâchée. En effet, les cas présentant des flux d'information continus peuvent nécessiter une cohérence plus stricte dans certaines situations. Par exemple, dans le cas de la manipulation d'un objet virtuel, les positions intermédiaires de cet objet au cours de l'interaction peuvent avoir une importance majeure. C'est notamment le cas lorsque l'environnement est soumis à une simulation physique :

ces positions intermédiaires peuvent correspondre à une situation de collision entre l'objet et son environnement, qui devra entraîner certains événements de simulation pour lesquels la position exacte des objets constitue une information capitale. Pour contourner cette contrainte, certaines solutions permettent d'approximer les valeurs intermédiaires par des procédés d'extrapolation du mouvement (dead reckoning).

De même, dans le cadre d'un environnement permettant l'interaction simultanée de plusieurs utilisateurs sur le même objet virtuel, la cohérence relâchée s'avère insuffisante. Il est en effet nécessaire, dans ce cas de figure, de maintenir une consistance constante entre les environnements perçus par chaque utilisateur.

Ces nuances dans les types d'informations à échanger imposent au concepteur d'un environnement virtuel collaboratif de faire des choix pertinents concernant le degré de cohérence des données à intégrer dans l'application.

3.2.4 Taxonomie et topologie

La topologie réseau d'une application de type EVC définit le mode de communication qui sera établi entre les différents noeuds du réseau. Dans ce domaine, trois modes de transmissions peuvent être identifiés :

- **L'unicast** : selon ce mode, chaque noeud communique avec un seul de ses homologues.
- **Le broadcast** : chaque noeud transmet ses informations à tous ses homologues en même temps.
- **Le multicast** : chaque noeud transmet ses données à un ensemble de noeuds distants (le "groupe multicast").

De la même façon que pour la distribution des données, la répartition des différentes briques constituant une application collaborative peut se faire de plusieurs manières. Le spectre de structures est assez large, mais nous pouvons extraire trois grands types d'architectures [PIC03] : l'architecture centralisée (aussi appelée architecture client-serveur), l'architecture pair à pair (peer to peer) et les architectures hybrides.

L'architecture centralisée (fig. 3.5 dans les cas des EVC) consiste à diviser une application monolithique classique en deux processus : le front end, qui correspond à l'interface utilisateur et le back end qui est le moteur de l'application. Cette architecture est le plus souvent associée à une distribution centralisée des données. Dans cette configuration, l'application est gérée en totalité par un unique poste de travail faisant office de serveur, auquel se connectent les machines clientes.

Ce serveur prend alors en charge la collecte des données, les calculs à effectuer sur ces données, ainsi que la diffusion des résultats aux clients. Les postes clients, quant à eux, ne prennent en charge que la partie interface utilisateur, c'est-à-dire l'affichage des données du système et la lecture des périphériques d'interaction de l'utilisateur. Cette architecture s'avère relativement simple à mettre en oeuvre, mais souffre de plusieurs inconvénients : le serveur étant unique, le système déployé est vulnérable aux pannes de celui-ci, puisque les applications clientes ne possèdent aucune autonomie. De plus, un trop grand nombre de clients peut être à l'origine d'une surcharge processeur du serveur, ralentissant ainsi l'exécution sur tous les postes. Le serveur

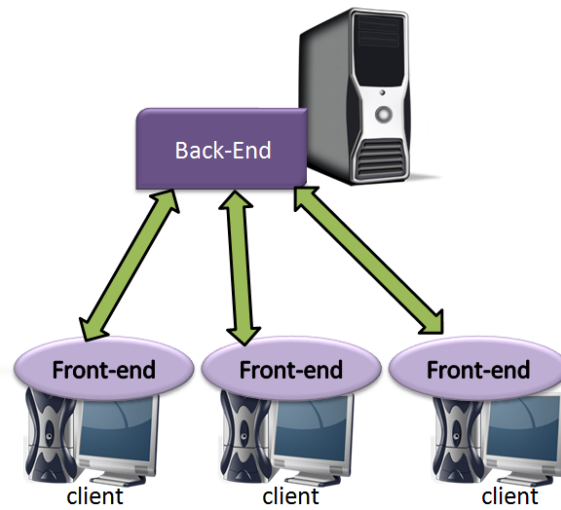


FIGURE 3.5 – Architecture centralisée

constitue ainsi un goulot d'étranglement en terme de puissance de calcul. Enfin, la connexion d'un nouveau poste client durant l'activité peut s'avérer pénalisante pour tous les participants.

L'architecture pair à pair (fig. 3.6) consiste à structurer l'application sous forme de plusieurs pairs, tous identiques, qui communiquent arbitrairement sans passer par l'intermédiaire d'une structure centrale. Cette architecture possède l'avantage d'être robuste vis-à-vis des pannes et de ne pas souffrir de l'existence d'un goulot d'étranglement. Cependant, elle est à l'origine d'un important trafic réseau et peut s'avérer complexe à mettre en oeuvre, notamment en ce qui concerne les mécanismes de préservation de la cohérence des données. Par exemple, si le mode de communication est de type unicast, pour transmettre un évènement à N pairs, un noeud doit réaliser N transmissions. Cette situation met en péril la synchronisation temporelle des données et donc la consistance globale de l'environnement.

Les architectures hybrides, quant à elles, consistent comme leur nom l'indique à mélanger les architectures centralisées et pair à pair. Selon cette configuration, une partie de l'application est centralisée, tandis qu'une autre partie est propre à chaque poste de travail. Cette architecture cumule les avantages des deux stratégies précédentes, mais nécessite un important travail de conception pour assurer la fiabilité de l'ensemble.

Ces trois grandes familles d'architectures présentent trois approches différentes pour concevoir une application collaborative. Aucune solution ne permet de répondre de façon optimale à tous les cas d'utilisation. Il convient donc de faire un choix en fonction des besoins de l'application en prenant en compte les avantages et les inconvénients de chaque stratégie.

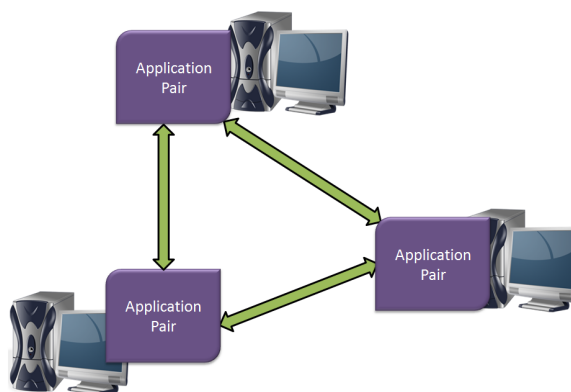


FIGURE 3.6 – Architecture pair-à-pair

3.3 Les architectures logicielles

Cette section est dédiée à l'observation des différents modèles de programmation des applications interactive. Avant d'entrer dans la description de ces modèles, il est important de clarifier notre approche vis-à-vis de cet état de l'art.

L'objectif de nos travaux est de voir dans quelle mesure il est possible, à partir d'une application de collaboration distante classique, de développer une plateforme répondant efficacement aux besoins d'une situation de collaboration colocalisée. L'objet de notre contribution n'est donc pas de définir un nouveau modèle de programmation, ni même d'en choisir un dans l'état de l'art. En revanche, l'observation de l'existant dans ce domaine peut constituer un guide pour aborder correctement l'étape de l'implémentation. Nous allons donc chercher dans ces modèles une source d'inspiration pour déterminer par quels moyens nous pouvons enrichir l'architecture d'une plateforme collaborative de façon à offrir des fonctionnalités adaptées au travail colocalisé.

3.3.1 Les modèles de conception d'applications interactives

Après l'analyse des besoins, la mise au point d'un cahier des charges et le développement conceptuel d'une plateforme de collaboration, la question de son implémentation logicielle revêt une importance capitale. En effet, le code de l'application doit être structuré de façon à répondre au mieux aux spécificités établies dans les phases réalisées en amont.

L'architecture logicielle d'une application interactive consiste à décomposer l'ensemble du système sous forme de différents composants interactifs. Ces composants vont interagir entre eux soit par le biais de liaisons directes, soit par celui de certaines entités dédiées. La structuration de ces différentes briques fonctionnelles va alors permettre de concrétiser les mécanismes internes de l'application, que ce soit en terme de définition des processus ou en terme d'allocation des ressources, puis d'identifier ses capacités d'évolution et d'adaptation. Dans ce chapitre, nous allons nous intéresser tout particulièrement à la partie "haute" de l'architecture, c'est-à-dire aux concepts liés à la représentation des objets interactifs au sein de l'application. Cet état de l'art ne traite donc pas des architectures à granularité plus fine comme les mécanismes de communication réseau ou la gestion du déploiement.

Il existe trois grands types d'architectures logicielles pour les systèmes interactifs : les modèles monolithiques, les modèles multi-agents, et les modèles hybrides. Ces trois modèles, essentielle-

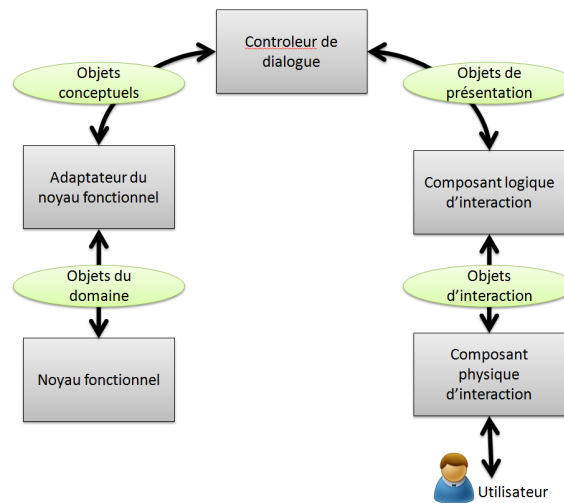


FIGURE 3.7 – Le modèle Arch

ment définis et utilisés pour les IHM 2D, ont pour objectif de procéder à une décomposition fonctionnelle de l'application, en particulier sous la forme d'une séparation de la partie noyau fonctionnel et de la partie interface homme-machine. Le noyau fonctionnel constitue le noyau central d'une application, c'est-à-dire les processus internes qui permettent de faire les calculs demandés au logiciel. La partie interface, quant à elle, constitue la partie du programme qui établit la communication entre le noyau fonctionnel et l'utilisateur. Cette partie est, dans la plupart des cas, associée à l'affichage graphique de l'application.

3.3.1.1 Les modèles monolithiques

Les modèles de type monolithiques ont la particularité de procéder à une décomposition fonctionnelle sous forme d'une superposition de plusieurs niveaux. L'application est ainsi composée de ces niveaux communiquant avec leurs voisins les plus proches.

• **Le modèle ARCH.** Le modèle ARCH [mra92] (fig. 3.7) constitue un modèle proposant de décomposer l'application interactive en cinq niveaux distincts, disposés sous forme d'arche. Le noyau fonctionnel est ici très éloigné de l'interface et n'est pas dépendant de la forme sous laquelle elle apparaît dans l'implémentation. L'objectif est de pouvoir recréer une nouvelle interface, sans nécessiter une refonte de la totalité du code de l'application, nous distinguons alors :

- Le composant Physique d'interaction : ce composant est situé en amont de l'interaction. Il constitue la partie permettant de communiquer avec l'utilisateur, en lisant les informations de ses périphériques physiques et en les appliquant aux outils virtuels.
- Le composant Logique d'interaction : cette couche constitue une forme plus abstraite de la partie interaction. En effet, le composant logique d'interaction reçoit les objets d'interaction issus du composant physique et prend en charge leur conversion et leur transmission vers le contrôleur de dialogue. Il est composé de boîtes à outils abstraites, qui possèdent une plus grande indépendance que le composant physique vis-à-vis du contexte matériel et du système d'exploitation.

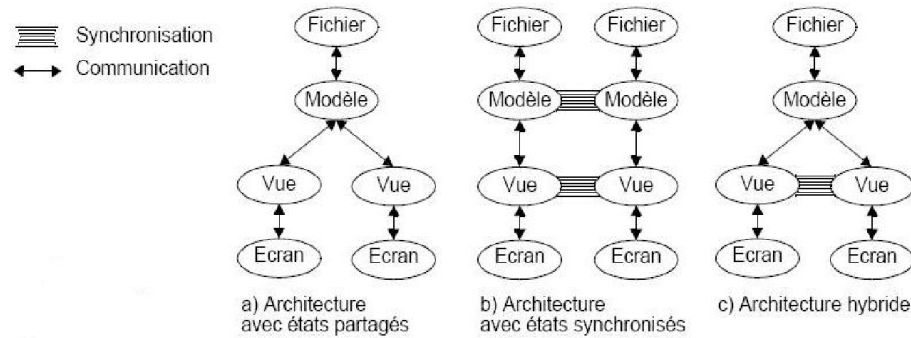


FIGURE 3.8 – Le modèle Zipper

- Le contrôleur de dialogue : il s'agit du composant principal de l'architecture ARCH. Il établit le lien entre l'interaction perçue de l'utilisateur et le noyau de l'application. Le contrôleur de dialogue prend ainsi en charge la retranscription conceptuelle des interactions et l'ordonnancement des instructions.
- L'adaptateur du noyau fonctionnel : ce composant est utilisé pour établir une communication entre le noyau fonctionnel et le contrôleur de dialogue. Il a donc pour rôle de convertir les objets conceptuels issus du contrôleur de dialogue en objets du domaine, à transmettre au noyau fonctionnel. Les objets du domaine constituent alors la forme la plus concrète des structures de données qui transitent au sein de l'application.
- Le noyau fonctionnel : il est constitué des processus de traitement les plus concrets, sans aucune dépendance vis-à-vis de l'interface.

Le modèle ARCH propose une décomposition fonctionnelle permettant de séparer efficacement l'interface du noyau fonctionnel. Néanmoins, la spécification de ce modèle n'apporte pas d'informations détaillée sur la structure interne du contrôleur de dialogue. De plus, cette approche monolithique n'aborde pas la possibilité de prendre en charge l'interaction de plusieurs utilisateurs simultanés.

• **Le modèle Zipper.** Le modèle Zipper (traduit par fermeture-éclair) (fig. 3.8) est un modèle monolithique orienté vers la conception d'applications multi-utilisateurs [Pat95]. L'application décomposée selon ce modèle se voit alors divisée en quatre niveaux, associés à un mode de partage.

- L'écran : il représente la partie la plus concrète, c'est-à-dire les périphériques d'interaction et de visualisation.
- La vue : ce niveau est constitué de la représentation des données, c'est-à-dire l'interface utilisateur.
- Le modèle : il est l'équivalent du noyau fonctionnel de l'architecture ARCH.

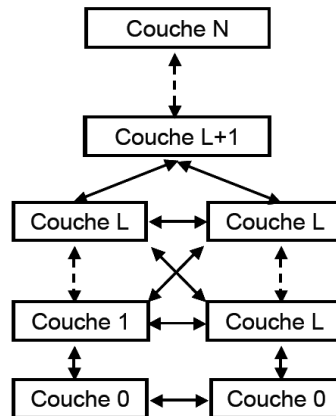


FIGURE 3.9 – Le métamodèle de Dewan

- Le fichier : le fichier représente la version persistante des données du modèle.

Ces quatre états sont également définis selon leur mode de partage, qui peut prendre trois formes : partagé, synchronisé ou répliqué. Le mode partagé consiste en l'instanciation unique du composant. Le mode synchronisé autorise plusieurs instances simultanées, mais doit assurer que ces instances sont à tout moment identiques, pour conserver la cohérence des données. Le mode répliqué, quant-à lui, permet la création d'instances multiples et indépendantes les unes des autres.

• **Le métamodèle de Dewan.** Le métamodèle de Dewan propose, comme le modèle Zipper, une décomposition fonctionnelle dédiée aux systèmes multi-utilisateurs [Dew99] (fig. 3.9). Il est d'ailleurs considéré comme une généralisation du modèle Zipper. La philosophie de ce métamodèle consiste à considérer une application comme étant constituée d'un nombre variable de couches. Ces couches sont ici associées aux différents niveaux d'abstraction de l'application. Ainsi, la couche la plus basse est associée à l'aspect matériel, tandis que la couche de plus haut niveau constitue l'aspect sémantique de l'application. Cette architecture porte le nom de métamodèle en raison de l'absence de spécification exacte des processus et des données prises en charge par chaque couche.

Comme pour le modèle Zipper, l'architecture selon Dewan est constituée d'une racine commune à tous les utilisateurs et de plusieurs branches constituées des couches bas-niveau associées aux interfaces personnelles des différents participants.

La gestion de l'interaction selon ce modèle apparaît sous deux formes : premièrement, les interactions réalisées par un utilisateur remontent la pile de façon séquentielle, partant du niveau correspondant à l'interface jusqu'à impacter la couche la plus élevée du métamodèle. La seconde forme est associée à la communication entre les interfaces des utilisateurs : les différentes couches communiquent avec leurs homologues distantes.

Cette approche mixte permet ainsi à l'application de différencier les actions réalisées par un utilisateur ou par ses collaborateurs. Les deux types d'événements transmis au sein du système

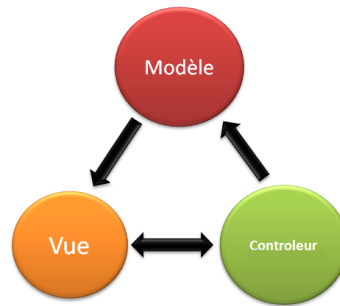


FIGURE 3.10 – Le modèle MVC

sont différenciés sous la forme d'évènements d'interaction ou d'évènement de collaboration.

Selon la terminologie de ce modèle, la couche de plus haut niveau échangeant des évènements de collaboration est appelée le degré de collaboration. L'intérêt de l'identification de ce degré de collaboration est utilisé pour identifier les couches qui nécessitent d'inclure des mécanismes de gestion des accès concurrents aux ressources.

3.3.1.2 Les modèles à agents

Les modèles à agents se distinguent des modèles monolithiques dans la mesure où ils ne s'appuient pas sur le regroupement des données du noyau fonctionnel sous la forme d'un bloc unique. En effet, ces modèles préconisent le découpage fonctionnel de l'application sous la forme de multiples entités réactives appelées "agents", possédant leurs propres données et leurs propres représentations dans l'interface-utilisateur.

• **MVC.** Le modèle MVC (Modèle, Vue, Contrôleur) (fig. 3.10) est issu du langage Small-talk [KP88]. Ce modèle propose une décomposition de l'application sous forme d'agents, dont la structure permet de distinguer l'aspect interface utilisateur du noyau central de l'application. Un agent MVC est ainsi constitué de trois composantes :

- **Le modèle :** cette composante prend en charge la gestion des structures de données définissant l'agent. Le modèle peut être associé à la base de données de l'agent, ainsi qu'à son noyau fonctionnel.
- **La vue :** elle a pour rôle de constituer une représentation des données du modèle qui soit perceptible par l'utilisateur. Elle est par conséquent majoritairement composée d'une représentation visuelle, mais peut aussi prendre la forme d'un retour sonore ou haptique. La vue est donc associée à l'interface utilisateur, mais ne prend en charge que la sortie, c'est-à-dire la communication dans le sens application-utilisateur.
- **Le contrôleur :** il constitue la composante complémentaire à la vue. En effet, au même titre que la vue, ce composant est assimilé à l'interface. Cependant, le contrôleur prend en charge les entrées issues de l'utilisateur. Il procède à une lecture des informations envoyées

par les périphériques d'interaction, pour en répercuter les conséquences sur la structure de données prise en charge par le modèle, par l'intermédiaire d'appels à son "noyau fonctionnel".

La communication établie entre le modèle et les deux autres composants repose sur un système d'événements. Lors de l'interaction (par exemple la détection d'un mouvement de la souris), le contrôleur transmet au modèle l'information correspondant à l'entrée du pointeur sur l'objet concerné. Le modèle passe alors en mode "is over" et cet état est répercuté graphiquement sur la Vue. Les composants vue et contrôleur étant tous les deux liés exclusivement à la représentation visuelle, il peut arriver que ces deux facettes soient fusionnées pour prendre la forme d'un unique composant, comme c'est le cas pour la boîte à outils graphique Java Swing [ELW98].

Lorsqu'une application est conçue selon une architecture de type MVC, les agents sont hiérarchisés sous forme de graphe de scène. Ainsi, une fenêtre de dialogue d'une application peut être composée de plusieurs agents hiérarchisés, chacun représentant un élément de cette fenêtre (bouton, slider, champ textuel). La racine de cette hiérarchie est alors un agent MVC représentant la fenêtre et affichant ses agents fils à leur emplacement.

• **PAC.** Le modèle PAC (présentation, abstraction, contrôle) est, comme MVC, un modèle à agent constitué de trois composantes [CH84].

- **La présentation :** elle constitue toute la partie interface-utilisateur. Elle prend en charge l'intégralité des échanges, qu'il s'agisse d'entrées ou de sorties. La présentation peut ainsi être perçue comme l'équivalent du couple contrôleur-vue de MVC.
- **L'abstraction :** ce composant prend en charge les informations relatives à l'agent, indépendamment de sa représentation dans l'interface. Elle est l'équivalent du modèle d'un agent MVC.
- **Le contrôle :** la facette contrôle d'un agent PAC possède un rôle double. D'abord, elle assure dans les deux sens la transmission et la transcription des informations qui transitent entre la présentation et l'abstraction. Ensuite, le composant contrôle prend en charge la communication entre plusieurs agents PAC lorsque ces derniers sont organisés selon une structure hiérarchique.

Le point fort de l'architecture PAC réside dans l'universalité du composant contrôle. En effet, ce dernier permet d'assurer des échanges complexes entre l'abstraction et la présentation. Ceci offre par exemple la possibilité d'avoir un agent PAC dont la facette abstraction prend en charge des données d'un plus haut niveau d'abstraction que les données nécessaires à la présentation. Par exemple, pour la représentation d'un thermomètre, l'abstraction peut manipuler une donnée abstraite représentant la température, tandis que la présentation visuelle de ce thermomètre reposera sur la retranscription de cette température sous forme d'une hauteur (fig. 3.11). La retranscription de l'information température est ainsi assurée par le contrôle.

• **ALV.** Le modèle ALV, pour "Abstraction", "Link" et "View" (fig. 3.12) est un modèle à agents spécifique aux applications multi-utilisateurs [HBR⁺94]. Ce modèle propose en effet d'articuler les multiples interfaces autour d'une facette partagée qui est commune à tous.

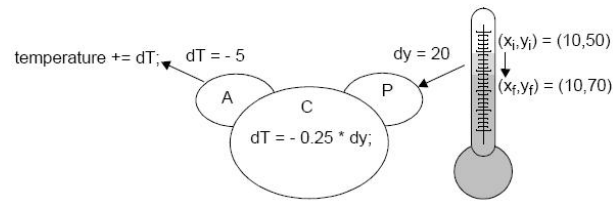


FIGURE 3.11 – Le modèle PAC mis en application pour la représentation d'un thermomètre

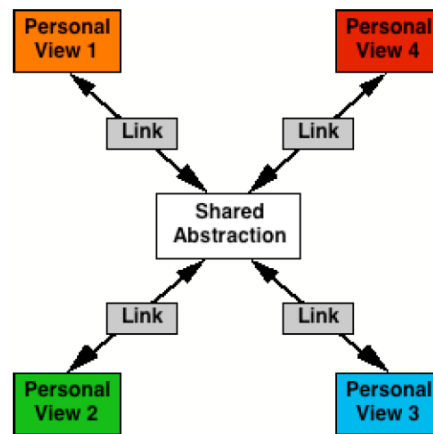


FIGURE 3.12 – Le modèle ALV mis en application dans un contexte multi-utilisateurs

Un agent ALV est constitué des trois composantes suivantes :

- **L'abstraction** : elle est l'équivalent du modèle de MVC ou de l'Abstraction de PAC. Elle prend en charge les données partagées.
- **La vue (view)** : comme pour la présentation du modèle PAC, la vue est la composante qui assure la communication avec l'utilisateur, à la fois pour la réception des entrées (les périphériques d'interaction) et pour l'envoi des sorties (représentation de l'interface utilisateur).
- **Le lien (Link)** : cette composante remplit l'un des deux rôles attribués au contrôle du modèle PAC. En effet, le lien prend en charge la transmission des informations entre une abstraction et ses vues. Il a pour tâche d'assurer la cohérence entre les données et leur représentation.

L'utilisation du modèle ALV pour les applications multi-utilisateurs consiste à associer plusieurs vues à une unique abstraction. La communication entre ces composantes est alors assurée par l'utilisation d'une composante "lien" pour chaque vue. La conservation de la cohérence entre les vues est par conséquent dépendante des informations manipulées par l'abstraction, puisqu'aucune communication entre composantes homologues n'est proposée.

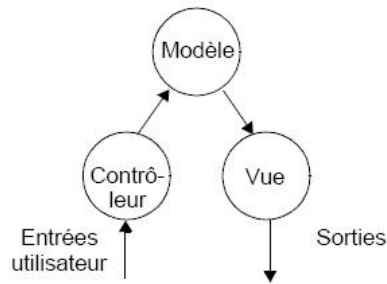


FIGURE 3.13 – Le modèle clock

• **Clock** Le modèle Clock [GNU97] est un modèle dérivé de MVC qui apporte des modifications mineures (fig. 3.13). Il reprend la construction des agents selon les trois composantes Modèle, Vue, et Contrôleur.

Le modèle reprend ici le même rôle que dans le MVC classique, à savoir la prise en charge des données. De même, la vue tient le rôle de la représentation de ces données dans l'interface, tandis que le contrôleur interprète les données reçues lors de l'écoute des périphériques d'interaction.

La différence notable avec le MVC réside dans le fait qu'il n'existe plus de lien entre le contrôleur et la vue. De même, les liens établis entre les trois composantes sont tous unidirectionnels. Dans le cadre d'une application partagée multi-utilisateurs, l'architecture est organisée selon une hiérarchie d'agents clock. Les branches de l'arbre ainsi formé, correspondant à la construction visuelle de l'interface, sont uniques pour chaque utilisateur.

3.3.1.3 Les hybrides

• **PAC-Amodeus** PAC-Amodeus est un modèle développé par L. Nigay [Nig94] (fig. 3.14). Il s'agit d'un modèle hybride, dans la mesure où il est constitué à la fois d'une approche multi-agent et d'une approche monolithique. En effet, PAC-Amodeus reprend la structure générale du modèle ARCH, à la différence que le composant contrôleur de dialogue est ici constitué d'une hiérarchie d'agents PAC. La nuance avec le modèle PAC classique est qu'avec PAC-Amodeus l'intégralité de l'application n'est pas constituée uniquement d'une hiérarchie d'agents PAC, mais possède les composantes de noyau fonctionnel, d'adaptateur de noyau fonctionnel, de composant physique d'interaction et de composant logique d'interaction.

Dans ce modèle, contrairement à PAC, l'implémentation des agents n'implique pas nécessairement d'implémenter toutes leurs composantes. Ainsi, un agent peut être constitué uniquement d'une abstraction et d'un contrôle. Il est alors appelé agent ciment, constituant le dénominateur commun entre ses agents fils. De même, un agent peut disposer de plusieurs composantes présentations différentes.

• **CoPAC** Le modèle CoPAC [Sal95] est un modèle dérivé de PAC-Amodeus (fig. 3.15). La particularité de cette architecture réside dans le fait que les agents PAC peuvent être dotés d'un composant supplémentaire : la facette communication. Cette nouvelle facette offre ainsi aux agents la possibilité de communiquer entre eux sans passer par un quelconque intermédiaire. Comme pour la présentation et l'abstraction, la facette communication est liée au composant

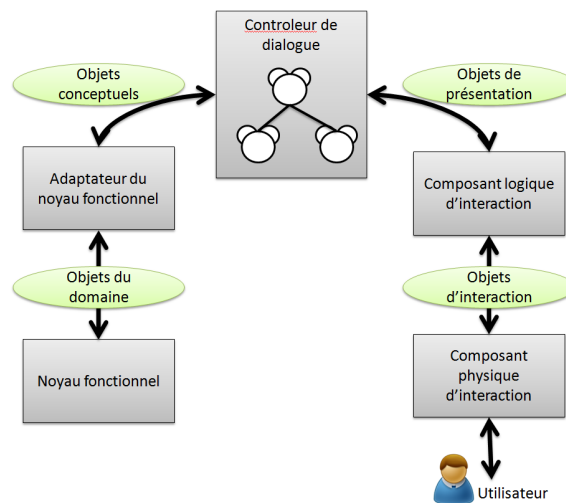


FIGURE 3.14 – Le modèle PAC-amodeus

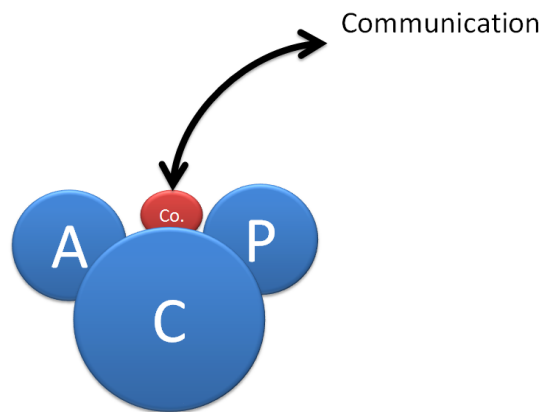


FIGURE 3.15 – Le modèle coPAC

contrôle de l'agent.

Grâce à ce modèle, les agents PAC peuvent échanger des événements avec leur analogue distant sous réserve que ce dernier soit placé au même endroit de la hiérarchie d'agents. Le modèle CoPAC permet ainsi trois modes de partage entre les composants PAC :

- Commun : le composant commun est unique et disponible pour tous les postes
- Couplé : le composant est dupliqué sur tous les postes, mais ses instances sont synchronisées pour en conserver la cohérence.
- Découplé : le composant est autonome, et n'est pas soumis à une liaison distante.

De part cette composante dédiée à la communication, le modèle CoPAC est tourné vers les applications multi-utilisateurs. La spécification du modèle définit entre autres les communications distantes qui prennent place entre les composants.

Conclusion Les modèles à agents proposent un découpage de l'application qui s'avère intéressant dans le cadre d'une activité collaborative en environnement virtuel. Leur approche permet en effet d'aborder l'application sous l'angle d'un assemblage d'éléments interactifs qui peut être mis en parallèle avec l'ensemble d'objets interactifs d'un environnement virtuel. Ce mode de structuration logicielle offre au concepteur de développer l'architecture de façon plus concrète, puisque proche de la perception qu'auront les utilisateurs de l'application. De plus, cet assemblage de composants permet une importante réutilisabilité du code puisqu'il est possible de développer plusieurs interfaces utilisateur distinctes à partir des mêmes composants. Enfin, l'application peut être distribuée de façon plus libre que pour les architectures monolithiques pour lesquelles la répartition est contrainte par les différentes couches logicielles.

3.3.2 Nouvelle approche de l'interaction : vers l'interaction instrumentale

Les modèles de conception logicielle en IHM ont longtemps gravité autour d'une unique approche de l'interaction. En effet, une majorité d'applications sont encore développées selon une approche de type WIMP (Window, Icon, Menu, Pointer). Cela signifie que pour être en mesure d'interagir sur un document électronique, l'utilisateur doit passer par des menus, boutons ou d'autres éléments visuels tiers. Par exemple, dans une application de traitement de texte classique, pour modifier la taille de la police de caractère d'un document, il est nécessaire de passer par un menu composé d'une liste déroulante proposant les différentes tailles applicables au texte. Ces éléments réactifs se positionnent entre l'utilisateur et l'objet numérique sur lequel il désire interagir. Cependant, ils sont constitués d'éléments complètement externes à l'objet numérique concerné. Ils ne disposent donc pas d'une réelle cohérence visuelle permettant de l'assimiler à cet objet.

Dans l'optique de rétablir une interaction plus naturelle et plus cohérente d'un point de vue visuel, une nouvelle contribution a vu le jour sous la forme de la "manipulation directe" de Ben Shneiderman [Shn83]. Ce concept vise comme son nom l'indique à proposer à l'utilisateur d'interagir directement sur les objets virtuels, sans lui imposer la contrainte de passer par la manipulation d'un élément tiers tel qu'un formulaire. Cette vision, plus proche de l'interaction telle qu'elle apparaît dans le monde réel, a rapidement été associée à l'appellation "post-WIMP".

Dans la continuité de ces travaux, un nouveau concept, ainsi que des modèles de conception associés sont apparus, cette fois sous le nom "d'interaction instrumentale" [BL97]. Cette approche fournit une mise en situation concrète de la manipulation directe, en introduisant la notion d'instrument. En effet, tout comme dans la réalité, l'interaction sur un objet ne peut se résumer à une manipulation "manuelle" (par exemple, enfoncer un clou dans une planche nécessite l'utilisation d'un marteau). Il est donc nécessaire de fournir des outils, qui permettront de définir le type et la modalité des interactions qui seront opérées sur l'objet numérique. La notion d'instrument introduite ici se positionne donc entre les périphériques d'interaction de l'utilisateur et l'objet numérique, mais cette fois de façon plus transparente que sur un système WIMP : contrairement à un formulaire, l'instrument n'est pas systématiquement associé à une

entité visuelle au sein de l'environnement.

L'une des caractéristiques de l'interaction instrumentale qui présente un intérêt particulier dans notre cas d'usage consiste en la généralité de l'instrument. En effet, comme dans la réalité, l'instrument n'est pas exclusivement dédié à un objet particulier. Il est ainsi utilisable sur plusieurs objets différents (cette caractéristique est appelée "polymorphisme"), le comportement de l'instrument s'adaptant dynamiquement à l'objet concerné. Dans un environnement virtuel présentant plusieurs objets, cette approche permet ainsi de bâtir une architecture plus facilement adaptable aux différents cas d'utilisation : la création d'un nouvel objet n'implique alors pas nécessairement le développement de composantes d'interaction propres à cet objet.

3.3.3 SPIN|3D : une architecture logicielle singulière

Dans le cadre des architectures logicielles basées sur un ensemble d'agents, la plateforme SPIN|3D [CSGCP99] développée au LIFL en collaboration avec Orange Labs repose sur une approche singulière. En effet, la conception logicielle de SPIN|3D apparaît sous la forme d'un assemblage de composantes logicielles proches du paradigme Modèle Vue Contrôleur, possédant toutefois des spécificités reprenant certains concepts de l'interaction instrumentale. Cette particularité nécessite donc de le considérer séparément. Cette architecture dérivée de MVC est présentée sous le nom de MVO, pour "Modèle", "Vue", "Outil".

Tout comme pour le MVC, cette architecture repose sur le principe de définir chaque objet de l'environnement sous la forme d'un ensemble de données et de leur représentation graphique, à travers le couple de composantes vue et modèle. La différence réside alors dans le fonctionnement interne de la composante chargée de lire les entrées, c'est-à-dire les interactions provenant de l'utilisateur. Selon le modèle de conception MVC, cette composante est appelée le "contrôleur". Cependant, cette composante " contrôleur " possède pour caractéristique majeure d'être propre à chaque objet : chaque objet virtuel comprend dans son découpage fonctionnel ses propres contrôleurs, qui ne peuvent alors pas être utilisés pour gérer l'interaction sur d'autres objets. Les périphériques d'interaction utilisés sont par conséquent écoutés par de nombreux contrôleurs simultanément. Ainsi, selon une architecture MVC classique, si plusieurs objets similaires sont disponibles au sein de l'environnement virtuel, les contrôleurs seront instanciés autant de fois qu'il y a d'objets, même si ils contiennent exactement les mêmes mécanismes de gestion de l'interaction (fig. 3.16).

Contrairement à cette approche, la prise en charge de l'interaction par SPIN|3D passe par un procédé sensiblement différent reprenant les concepts d'interaction instrumentale de Beaudouin-Lafon [BL04], et selon lequel les interactions de l'utilisateur avec les objets virtuels se font par l'intermédiaire "d'outils d'interaction". Cette notion d'outil se distingue du contrôleur classique dans la mesure où cette composante possède un certain caractère générique et indépendant. En effet, dans la plateforme SPIN|3D, un outil est une composante logicielle indépendante capable d'interagir indifféremment sur plusieurs objets virtuels de l'environnement. D'un point de vue conceptuel, l'outil ne peut donc plus être considéré comme l'un des éléments constitutifs d'un objet. L'outil du MVO est ainsi chargé de modifier les données stockées par les composantes modèle des différents objets, à travers la génération d'événements d'interaction haut-niveau (fig. 3.17).

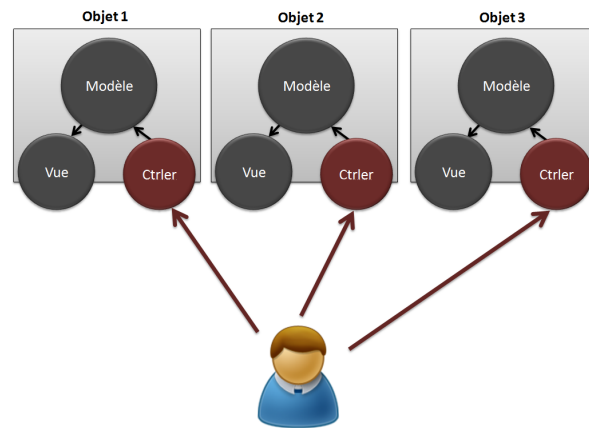


FIGURE 3.16 – Trois objets distincts dans le même environnement. Selon le modèle MVC, les interactions provenant de l'utilisateur transitent par le contrôleur, propre à chaque objet

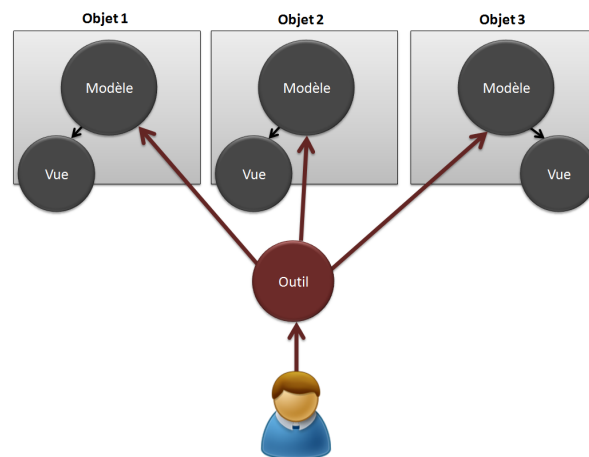


FIGURE 3.17 – Utilisation d'un outil générique. Contrairement au contrôleur de MVC, cette composante est propre à une interaction et peut s'appliquer sur plusieurs objets distincts

Pour décrire plus en détail le fonctionnement d'un outil, précisons sa constitution interne (fig. 3.18) : un outil est une machine à états, composée d'un automate contenant plusieurs opérateurs. Son état courant est défini par l'opérateur actif à cet instant. Prenons l'exemple de l'outil de manipulation appelé "Select and Manip", constitué de l'assemblage de deux opérateurs. L'utilisation de cet outil se déroule en deux temps. D'abord, l'utilisateur va sélectionner l'objet de l'environnement qu'il désire manipuler. Pour ce faire, il utilisera un périphérique de pointage tel qu'une souris classique. Dans ce mode, l'état de l'outil d'interaction est défini par le fait que l'automate est arrêté sur l'opérateur "select". Lorsque l'utilisateur validera sa sélection, en l'occurrence par un clic de la souris, l'automate avancera d'un état, et l'opérateur courant deviendra donc l'opérateur "manip". Dans cet état, l'outil va écouter les informations du périphérique de manipulation (clavier, spacemouse ou un autre périphérique doté de plusieurs degrés de liberté) (fig. 3.19).

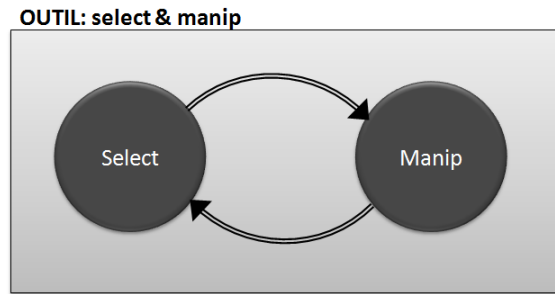


FIGURE 3.18 – L'outil select and manip. Il est constitué d'un automate à deux états.

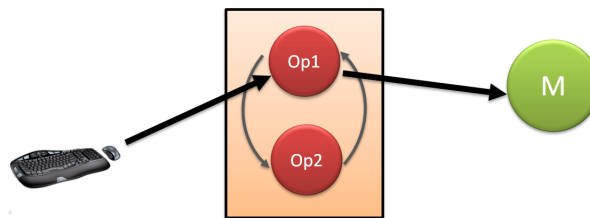


FIGURE 3.19 – La relation qui est établie entre le périphérique d'interaction et le modèle d'un objet passe par les opérateurs de l'outil.

Cette construction des outils sous la forme de plusieurs opérateurs élémentaires offre un avantage en terme de factorisation du code. En effet, cette approche permet au concepteur de l'application de créer plusieurs outils différents en réutilisant des opérateurs déjà implémentés. De plus, à chaque opérateur déclaré dans l'application, le concepteur associe un ensemble de périphériques d'interaction. L'outil peut ainsi permettre la gestion d'interactions "à deux mains", à travers l'écoute simultanée de plusieurs dispositifs d'interaction. Les données brutes issues des périphériques seront alors converties en évènements d'interaction de plus haut niveau tels que les valeurs incrémentales des angles de rotation de l'objet sélectionné. Ces évènements sont enfin transmis au modèle de l'objet, entraînant ainsi la modification des attributs de cet objet, qui se répercutera alors naturellement sur les vues.

L'opérateur est une sous-partie de l'outil qui est pensée pour se "connecter" à une interface donnée. De cette manière, il permet d'interagir sur tous les objets disposant de l'interface adaptée.

Pour permettre de considérer l'outil indépendamment des objets d'interactions disposés dans l'environnement virtuel, il est nécessaire d'adapter son comportement au contexte de l'action. En effet, dans un environnement quel qu'il soit, les objets virtuels sont susceptibles d'avoir des comportements différents vis-à-vis d'un même outil. Ainsi, certains objets peuvent par exemple être définis comme non-sélectionnables et donc non-manipulables, ou nécessiter certaines modifications avant de valider leur état sélectionné, telles que la pose d'un verrou empêchant les autres utilisateurs de réaliser des interactions conflictuelles sur le même objet. Cependant, la nature " haut-niveau " des évènements d'interaction générés par l'outil ne permet pas aux objets d'adapter leurs réactions face à ces évènements. L'outil d'interaction possède par conséquent la

capacité d'analyser le contexte de l'interaction (en l'occurrence, l'objet virtuel pointé par l'utilisateur et son environnement) et de contraindre son comportement au contexte détecté.

Dans la plateforme SPIN|3D, l'adaptation au contexte est définie à travers un ensemble de règles permettant de détecter ces cas particuliers. En effet, si certains objets possèdent des comportements particuliers vis-à-vis d'un outil, la plupart d'entre eux possèdent cependant des caractéristiques similaires. Il ne s'avèrerait pas pertinent de définir les actions réalisées par l'outil pour chacun des objets en particulier. L'outil est alors défini par son comportement par défaut, et l'ensemble des règles d'interaction ne sont présentes que pour enrichir ce comportement en l'adaptant aux objets spécifiques.

Ces règles prennent place lors du déclenchement d'une action significative qui entraîne le changement d'état de l'outil. Ce mécanisme est susceptible d'intervenir sous plusieurs formes :

- **le déni** Dans le cas, par exemple, d'un objet dont la sélection n'est pas autorisée, l'outil doit adapter son comportement de façon à ne pas procéder à la sélection. Dans ce contexte, une règle est définie entre l'opérateur "select" et l'objet en question. Cette règle a pour rôle de bloquer l'enchaînement de l'automate de l'outil, empêchant la génération de l'évènement de sélection et interdisant le passage de l'état de l'outil vers l'opérateur de manipulation. L'outil est par conséquent neutralisé jusqu'à ce que son contexte d'utilisation change.
- **l'augmentation par un processus spécifique** Dans les environnements impliquant plusieurs objets de nature différente, l'objet d'interaction est susceptible de ne pas réagir correctement à un outil. Il peut par exemple nécessiter l'exécution de certains mécanismes non prévus dans le fonctionnement par défaut de l'outil. Ce dernier doit alors voir son fonctionnement adapté à l'objet visé. En d'autres termes il peut s'avérer nécessaire, dans certains contextes d'interaction, d'ordonner à l'outil d'exécuter certaines instructions supplémentaires.

Dans ce cas de figure, le système de règles apporte une solution en enrichissant les processus exécutés par les opérateurs. De cette manière, la règle intervient lors du passage de l'automate entre deux états. Elle soumet alors à l'outil l'exécution d'un ensemble d'instructions avant de valider l'activation de l'opérateur élémentaire suivant.

3.4 Les systèmes interactifs complexes : les EVC

Les différentes solutions dédiées à l'interaction collaborative distante sont nombreuses. Les cas d'utilisation étant particulièrement hétérogènes, ils expriment des besoins très divers. Pour cette raison, il apparaît plus naturel de s'appuyer sur des boîtes à outils pour développer un environnement virtuel adapté au cas désiré. Cette solution s'avère plus souple que de concevoir directement une application dédiée à un cas d'usage et offre ainsi une grande réutilisabilité. Par conséquent, l'évocation d'un EVC implique souvent l'utilisation d'un outil de conception pour le créer.

Dans cette optique, certaines technologies proposent de passer par un système purement descriptif grâce auquel il est possible de concevoir un environnement virtuel interactif en allégeant le plus possible les tâches relatives à l'implémentation logicielle. Ces systèmes proposent donc de

décrire l'environnement à travers une configuration souvent en mode texte qui est par conséquent plus facile d'accès. C'est par exemple le cas des technologies VRML/X3D ou MPEG-4.

Les outils-Auteur, quant à eux, reposent également sur une simplification du développement à travers une interface de conception. Il s'agit de solutions logicielles permettant au concepteur de créer son environnement virtuel en assemblant différents composants qui couvrent une grande partie des besoins les plus courants. Contrairement aux systèmes descriptifs qui sont relativement limités en terme de fonctionnalités, les outils auteurs proposent aux concepteurs avancés de développer leurs propres composants, permettant ainsi de répondre aux cas d'usages plus spécifiques. Les outils tels que VIRTOOLS ou DIVE [YJsZ09][eOH93] sont à classer dans la catégorie des outils auteurs.

La troisième catégorie que nous pouvons distinguer regroupe les solutions de type boîte à outils : ces technologies sont plus volontiers dédiées aux concepteurs possédant de bonnes connaissances en développement, puisqu'elles sont constituées d'un ensemble de bibliothèques logicielles offrant les fonctions élémentaires nécessaires à la création d'un environnement virtuel complexe. L'avantage des boîtes à outils est avant tout qu'elles ne sont théoriquement pas limitées et s'adaptent à n'importe quel cas d'usage. Si elles ne nécessitent que peu de développement logiciel pour les cas les plus simples, elles imposent toutefois une certaine maîtrise dès lors qu'il est question de fournir des fonctionnalités plus complexes.

Dans cet partie nous allons décrire quelques EVC à travers trois plateformes, à savoir Virtools, OpenMask et SPIN|3D.

3.4.1 Virtools

Virtools est une plateforme de développement permettant de créer des environnements virtuels interactifs basés sur un système de visualisation comprenant 3D temps réel et vidéo. Il est constitué d'une interface de conception assimilable à un environnement de développement. Cette interface permet de concevoir l'intégralité de l'application interactive.

Pour ce faire, le concepteur doit importer dans l'application "virtools dev" les modèles 3D qui seront les éléments constitutifs de l'environnement. Pour rendre l'ensemble interactif, l'application repose sur un système de "building blocks" (BB), qui sont des briques paramétrables permettant de définir des comportements. Ainsi, le concepteur peut spécifier le comportement dynamique des objets 3D.

Cet assemblage de briques est assimilable à un graphe dans lequel on place des noeuds. Un building block peut recevoir des informations issues d'un autre building block et lui répondre, à lui ou à un autre. Les BB proposés par défaut couvrent un large panel de possibilités. Néanmoins, pour répondre à certains besoins précis, le concepteur a la possibilité de développer ses propres BB.

Une fois l'application conçue via l'éditeur "virtools dev", le code de l'application interactive est généré automatiquement. La visualisation de l'application est alors possible directement dans un navigateur web équipé du plugin virtools ou en tant qu'application native. De même plusieurs modules complémentaires sont proposés, permettant de prendre en charge des mécanismes d'intelligence artificielle, de simulation physique, ou de gestion de périphériques spécifiques. Pour concevoir des applications multi-utilisateurs, virtools possède ainsi un module complémentaire appelé le multi-user pack.

La définition d'un objet partagé passe par la définition d'un Distributed Object. Cet objet distribué est une propriété accessible par tous les utilisateurs dont la valeur est synchronisée à distance. Un objet partagé de l'environnement est donc un objet auquel sont attachées des propriétés sous forme de "distributed object".

L'architecture réseau utilisée pour la synchronisation des différents postes de travail est une architecture centralisée. Tous les échanges distants passent ainsi par un serveur central, qui assure la cohérence de l'ensemble et permet de poser des verrous. Ces verrous sont utilisés par l'application pour empêcher les conflits d'accès à une donnée partagée. Ainsi, deux utilisateurs ne peuvent pas agir simultanément sur la même propriété d'un objet.

La mise en place d'une application collaborative multi-utilisateurs via virttools implique donc une représentation de l'environnement 3D qui est similaire pour tous les participants. Les objets partagés sont les mêmes et la perception qu'en ont les utilisateurs est homogène. Les données relatives à ces objets sont copiées sur chaque poste de travail pour des raisons d'optimisation réseau. La préservation de la cohérence bénéficie toutefois de la présence d'un serveur central, qui prend en charge les données de référence (ainsi que les verrous), sur lesquelles chaque poste client se synchronise.

3.4.2 OpenMask

OpenMask [Mar01], est une boîte à outils C++ permettant la création d'environnements virtuels dédiés à la simulation et à l'animation d'objets. Cette boîte à outil fournit au concepteur les méthodes nécessaires à la mise en place de cette simulation à travers des outils de multithreading ou de distribution réseau. OpenMask offre également différents services nécessaires à la gestion de cette simulation : service de nommage des objets, communication entre objets simulés, gestion du temps, etc. Pour permettre une distribution des calculs liés à la simulation, OpenMask permet d'assigner les différents objets à des machines différentes. La communication entre ces objets étant toutefois nécessaire au bon déroulement de la simulation, un système de proxy permet les échanges d'informations de façon transparente [eTD02]. Les objets peuvent alors être simulés sur des machines différentes, occasionnant un important gain de performances. Les proxy des objets sont, dans ce cas de figure, synchronisés à leur référentiel pour le maintien de la cohérence de l'ensemble. Dans le contexte de la simulation temps-réel impliquant d'importantes exigences en terme de synchronisation, des algorithmes d'extrapolation des valeurs sont également présents. Outre l'aspect simulation, la plateforme OpenMask permet d'afficher l'environnement virtuel en temps réel et de le rendre interactif. Le système de visualisation est défini comme un objet de simulation communiquant avec les autres et adaptable au dispositif d'affichage désiré. Pour permettre une activité de collaboration sur plusieurs postes de travail simultanément, le concepteur peut alors instancier cet objet de visualisation sur chacune des machines.

Il existe une distinction entre les objets visuels interactifs et les objets de simulation. La partie logicielle prenant en charge les échanges avec l'utilisateur (représentation visuelle d'une part et lecture des périphériques d'interaction d'autre part) constitue par conséquent une couche supplémentaire qui se greffe au reste de la plateforme. Chaque objet perceptible dans l'environnement 3D présenté aux utilisateur n'est donc, d'un point de vue logiciel, pas le même objet que celui constituant son comportement fonctionnel.

Concernant l'interaction sur les objets, là encore, le système de proxy permet de distribuer l'application : les objets sont instanciés sur leurs postes de travail respectifs, et lorsqu'un utilisa-

teur décide d'interagir avec un objet distant, la plateforme relaie les messages d'interaction sur le réseau à travers son proxy local.

Dans l'environnement OpenMask, les objets sont conçus selon un modèle de type producteur/consommateur. Cela signifie qu'un objet peut consommer des informations en entrée (un événement de manipulation par exemple) et produit en sortie la valeur des propriétés qui lui sont définies comme publiques (sa position dans l'environnement).

Pour répondre aux contraintes de latence réseau ou de disparité des fréquences de la simulation, la plateforme peut conserver la cohérence des données par le biais d'algorithmes de dead reckoning. Ces algorithmes vont extrapoler les mouvements des objets, permettant ainsi de produire une valeur approchée de la valeur réelle, en fonction des dernières valeurs sûres connues. OpenMask a été utilisé pour la création d'environnements dans des cas d'usage très divers, tels que la navigation et la prise de décision autour de données scientifiques complexes [DFNA08], ou des applications d'entraînement au maniement d'équipements industriels [GMA07].

3.4.3 SPIN|3D

SPIN|3D [CSGCP99] est une plateforme de création d'environnements virtuels collaboratifs 3D développée conjointement par le LIFL (Laboratoire d'Informatique Fondamentale de Lille) et Orange Labs et qui est dédiée plus particulièrement au TCAO. L'espace de travail de SPIN|3D est constitué d'un environnement virtuel 3D interactif. Cet environnement peut parfois reprendre le concept de bureau virtuel, qui se présente alors sous la forme d'une table sur laquelle sont disposés les objets d'interaction. Les utilisateurs sont représentés par des avatars 3D disposés autour de cette table. Les participants de la session de collaboration disposent d'outils divers permettant d'interagir avec les objets. Ces outils sont implémentés sous forme de machines à état permettant de séquencer l'interaction d'un utilisateur et d'adapter le comportement de l'outil en fonction de l'objet et du contexte d'interaction (cf. 3.3.3). Les actions réalisables par les utilisateurs sont donc variées, allant de la simple tâche de manipulation à des actions plus complexes telles que des tâches d'assemblage.

L'environnement partagé de la plateforme SPIN|3D repose sur une visualisation de type WY-SIWIS relâché : les participants immergés dans cet environnement commun peuvent donc choisir de modifier leur point de vue et l'organisation de l'univers virtuel (ie. le positionnement des objets) sans influencer la perception visuelle des autres utilisateurs.

Pour améliorer la compréhension globale de l'activité de la part des utilisateurs, chaque participant manipule un télé-pointeur. Ce télé-pointeur est similaire au curseur de souris affiché dans les environnements graphiques 2D, à la différence qu'il est visible par tous les autres utilisateurs connectés à l'environnement virtuel. Il permet de visualiser à tout moment l'état de l'activité, ainsi que les centres d'intérêts des autres. La gestion des conflits d'interaction de SPIN|3D repose sur un système de verrouillage appliqué aux objets (ou aux parties d'objets), empêchant deux utilisateurs de passer tous les deux en mode "manipulation" sur le même objet. D'un point de vue "architecture réseau", les objets d'interaction et les propriétés qui leurs sont associées sont dupliqués sur chaque poste de travail impliqué dans la collaboration. La cohérence est maintenue par un système de synchronisation multicast.

La conception d'un environnement virtuel via la plateforme SPIN|3D vise à être la plus accessible possible. C'est pourquoi la création d'un environnement standard ne nécessite que la

création de fichiers de configuration au format XML (eXtended Markup Language). Ces fichiers de configuration vont permettre de définir les objets virtuels, leurs propriétés (modèle 3D au format VRML, propriétés géométriques, visuelles ou d'interaction), mais également les outils (via la description des automates et de leurs états potentiels). La plateforme permet en outre de développer de nouveaux outils ou divers plugins spécifiques au cas d'usage désiré. De même, les canaux de communications offrent également la possibilité de "connecter" une application tierce à l'ensemble, de façon à interagir avec l'environnement par le biais de cette application.

3.4.4 Conclusion

Après avoir observé les différents types de plateforme d'interaction collaborative, nous pouvons formuler une critique de ces applications : dans la plupart des cas, ces plateformes sont conçues de manière à fournir aux utilisateurs un support de travail commun. Cette approche vise avant tout à partager des objets virtuels et à assurer la cohérence dans les interactions de chacun. Dans un tel contexte, le nombre, la localisation et la diversité des habitudes de travail individuel des différents intervenants ne constitue pas un facteur déterminant pour l'organisation de l'environnement déployé. Nous pensons au contraire que ces facteurs doivent être déterminant dans les choix de conception.

En effet, les EVC actuels sont conçus de façon à constituer un environnement visuel interactif homogène, sans que celui-ci ne soit déterminé par la constitution de l'équipe de collaborateurs. La critique que nous formulons à l'égard de ces plateformes est qu'elles ne prennent pas en compte les caractéristiques intrinsèques de l'équipe, en particulier la diversité des compétences de chaque collaborateur. Ces environnements reposent sur une homogénéité complète, à la fois en terme de représentation visuelle mais aussi au niveau de l'interaction. Cette vision est selon nous trop restrictive pour notre étude, puisque pour une équipe pluridisciplinaire, cette homogénéité constitue un frein à la liberté d'action de chacun.

Dans cette optique, les EVC sont assimilables à des environnements interactifs généralistes. L'aspect collaboratif n'apparaît alors que par l'utilisation qui est faite de ces environnements. Ces solutions n'ont par conséquent pas vu leur structure logicielle orientée vers la diversité des sources d'interaction qui apparaissent lors d'une activité collaborative. Notre approche vise au contraire à baser l'essentiel de l'architecture logicielle sur la collaboration colocalisée et ses effets sur le déroulement de l'activité.

Les environnements virtuels collaboratifs existants, s'ils ne sont pas parfaitement adaptés à une collaboration en situation de coprésence, peuvent nous fournir une base relativement complète, à partir de laquelle nous pouvons développer une solution dédiée à la colocalisation des utilisateurs. En effet, ils proposent un environnement interactif 3D qui permet une certaine diversité dans les outils d'interaction. Ils disposent également de systèmes de communication réseau qui peuvent être exploités pour supporter l'aspect "multi-dispositifs" de notre war room. Le chapitre suivant va ainsi développer la méthodologie à suivre pour modifier une plateforme EVC existante afin de réaliser une plateforme supportant les contraintes de la war room.

Pour notre étude, nous avons fait le choix de nous tourner vers l'enrichissement logiciel de la plateforme SPIN|3D. En effet, son architecture basée sur un ensemble d'agents logiciels, sa distinction nette entre interaction et visualisation, la généricité des outils d'interaction et la diversité des interfaces de communication réseau fait de SPIN|3D un support pertinent pour

implémenter efficacement les fonctionnalités relatives à notre besoin. Néanmoins la plupart des propositions que nous ferons dans le chapitre 4 sont applicables à d'autres plateformes, sous réserve que celles-ci reposent sur un concept proche de celui des "agents logiciels".

APPROCHE LOGICIELLE DE LA COLLABORATION COLOCALISÉE : ADAPTATION D'UNE PLATEFORME EVC CLASSIQUE

Sommaire

4.1	Architecture logicielle : distribution libre de l'application	114
4.1.1	Les contraintes d'utilisation et leur impact sur l'architecture	114
4.1.2	Communication réseau : distribution des composantes	115
4.1.3	Architecture multi-agents : utilisation du modèle MVO de SPIN 3D . . .	119
4.2	Support de canaux hétérogènes : la notion de modèle abstrait . . .	124
4.2.1	Le MVO : les limites du modèle	125
4.2.1.1	Données concrètes et données abstraites	126
4.2.1.2	Cohabitation des données concrètes et abstraites	127
4.2.2	Les modèles existants	128
4.2.3	M_N V-O : Adaptation de la structure MVO	130
4.2.4	Transparence de l'adaptation et distribution distante	132
4.2.5	Multiplicité des niveaux d'abstraction	134
4.2.6	Insertion d'une application tierce dans le contexte multi niveaux d'abs- traction	136
4.2.7	Intégration dynamique d'un nouveau dispositif	136
4.2.8	Choix des données dans le cadre du multicanal	138

Pour implémenter correctement les différentes caractéristiques désirées dans notre war room, il est primordial de construire l'application sur une architecture logicielle pertinente. Ce chapitre décrit la structure interne de notre proposition logicielle, et justifie nos choix par rapport à nos besoins. Nous allons voir que le choix d'une architecture multi-agents permet une importante modularité de l'application globale. Cette structuration, lorsqu'elle est enrichie par une interface de communication réseau suffisamment flexible, offre une distribution de l'activité facilement adaptable au contexte d'une activité collaborative colocalisée multi-dispositifs.

4.1 Architecture logicielle : distribution libre de l'application

4.1.1 Les contraintes d'utilisation et leur impact sur l'architecture

Les premières contraintes concernent la réutilisabilité du code. En effet, pour permettre à un concepteur de créer facilement une application correspondant à son cas d'utilisation, il est nécessaire de lui fournir le maximum de briques constitutives. L'agencement de ces briques à travers la mise en place d'un fichier de configuration permettra de développer une application précise. Par conséquent, il est indispensable de maximiser la réutilisabilité de chacune de ces briques constitutives, de façon à ce qu'elles puissent intervenir dans la construction de plusieurs applications ayant des fonctionnalités similaires. L'architecture logicielle mise en oeuvre doit ainsi permettre de factoriser au maximum le code, tout en veillant à ce que les composantes logicielles générées soient suffisamment indépendantes les unes des autres pour pouvoir être réagencées dans la war room. Le déroulement de l'activité dans notre plateforme dépendra par conséquent moins du contenu des composantes logicielles que de leur organisation interne au sein de l'application.

Pour que notre plateforme puisse être utilisée dans un maximum de cas d'utilisation, elle doit être structurée selon un ensemble de composantes. Les architectures telles que les modèles Seeheim ou ARCH [mra92] et leurs dérivés proposent de définir une architecture sous la forme de différentes couches empilées, représentant les différents niveaux allant du noyau fonctionnel de l'application à la présentation. Ces architectures offrent l'avantage de distinguer la partie interface utilisateur du reste de l'application. Cependant, la structuration de l'application autour d'un noyau central nécessite de la part de celui-ci qu'il prenne en charge la totalité des fonctionnalités du système. Cette approche s'avère assez peu compatible avec le désir de disposer d'une plateforme générique, puisqu'elle rend difficile la tâche d'adaptation de la plateforme et de ses processus internes à plusieurs applications interactives différentes.

La structuration sous la forme d'un ensemble d'agents interactifs (comme le proposent, entre autres, les modèles de conception MVC [KP88] ou PAC [CH84]) se révèle par conséquent plus pertinente, puisqu'elle permet de diviser notre application en plusieurs éléments homologues dont l'assemblage définit le fonctionnement de l'application. La modification d'une application n'implique que le remplacement de certains de ces agents, sans nécessiter de reprendre le développement à partir de la racine de l'application. De même, la création d'une application nouvelle consistera en l'assemblage de ces agents, et peut éventuellement ne nécessiter aucun travail d'implémentation si tous les agents interactifs nécessaires sont déjà existants dans d'autres assemblages.

Parmi les contraintes de notre application, notre architecture doit être capable de dissocier naturellement la gestion de l'interaction et la représentation visuelle de l'interface. En effet, l'activité qui prend place dans notre environnement collaboratif implique la possibilité pour les

utilisateurs d'interagir sur les objets de l'environnement sans être complètement dépendants de l'interface d'interaction. Par exemple, il est possible pour les participants d'interagir à distance sur un poste de travail en utilisant les périphériques connectés à une autre station.

De plus, il est nécessaire de distinguer la prise en charge de l'affichage des objets et la gestion des données définissant cet objet. En effet, comme nous l'avons vu dans les chapitres précédents, il doit être possible de disposer simultanément de plusieurs représentations visuelles d'un même objet. Si ces représentations sont multiples, elles sont toutefois similaires. Il n'est donc pas nécessaire de dupliquer les données sur lesquelles ces représentations reposent. Les affichages concurrents peuvent dans ces conditions reposer sur la même composante de gestion de données.

Comme nous l'avons souligné, l'installation matérielle qui accueille notre plateforme logicielle doit pouvoir être constituée de plusieurs postes de travail différents. Cette configuration implique naturellement de mettre en place des mécanismes de communication réseau. Les différentes stratégies décrites dans le chapitre 3 proposent plusieurs possibilités pour structurer les échanges réseau. La section suivante va donc décrire la stratégie de communication réseau que nous avons choisie.

4.1.2 Communication réseau : distribution des composantes

Dans le cadre d'une application de type EVC classique, l'environnement virtuel est commun à tous les utilisateurs. Chaque participant posté sur son ordinateur personnel visualise et manipule donc les mêmes objets, qui sont affichés via des représentations similaires. L'intégralité de cet environnement est alors disponible sur tous les postes de travail. Ces postes vont communiquer entre eux pour assurer la cohérence globale des interactions et leurs effets sur les objets virtuels. En terme de déploiement logiciel, cette situation signifie que tous les ordinateurs impliqués dans la collaboration exécutent la même application, synchronisée entre ces postes. Dans un tel contexte, tous les agents logiciels constituant les objets de l'environnement virtuel sont donc dupliqués pour être disponibles localement. Nous pouvons qualifier cette structuration d'environnement symétrique.

Notre proposition consiste au contraire à ne pas reproduire cette symétrie : notre objectif principal est de prendre en charge l'aspect hétérogène de l'activité collaborative en la répartissant au maximum entre les dispositifs. L'environnement virtuel perceptible à travers chacun des postes de travail peut potentiellement être différent de celui présenté sur les autres ordinateurs de la war room. La compréhension globale de l'activité par les collaborateurs n'en souffrira pas puisque dans un contexte colocalisé, chacun des participants conserve un accès immédiat à toutes les stations de travail disposées dans l'espace physique.

Pour mettre en place un environnement asymétrique tel que celui que nous proposons, la présence de tous les agents logiciels sur tous les dispositifs matériels n'est plus nécessaire : si un ordinateur ne présente dans son environnement qu'une partie de l'ensemble des objets virtuels de l'activité, il est en effet inutile que les agents logiciels relatifs aux autres objets soient disponibles localement. Cette constatation nous amène à proposer une architecture distribuée, selon laquelle les agents ne sont instanciés que sur les ordinateurs pour lesquels l'objet virtuel concerné est perceptible dans l'environnement.

Cette approche est toutefois encore trop restrictive : prenons l'exemple d'une activité impli-

quant -entre autres- deux postes de travail permettant de visualiser le même objet virtuel. Selon notre proposition, en terme de déploiement logiciel, cette situation impliquerait d'instancier le même agent sur chacun des deux postes de travail. Ceci entraîne également la nécessité de mettre en place un mécanisme de synchronisation réseau entre les deux jeux de données dupliqués de cet objet virtuel et donc entre les deux ordinateurs. Bien que cette configuration soit viable, de part le contexte matériel de la war room (qui implique un réseau local à faible latence), nous pouvons aller plus loin dans la distribution logicielle de l'environnement. Pour reprendre notre exemple, les deux postes de travail sont utilisés pour la visualisation de l'objet virtuel. Or, un agent logiciel est constitué d'un ensemble de composantes dont une seule est dédiée à la représentation visuelle de cet agent : c'est le cas de la Vue dans le modèle MVC, ou de la Présentation dans le modèle PAC. Dans une telle situation, il devient envisageable de procéder à une distribution non plus à l'échelle des agents, mais à l'échelle des composantes. Concrètement, cela signifie que sur les deux postes de travail de notre exemple, la seule entité logicielle qu'il est nécessaire d'instancier localement est celle dédiée à la représentation visuelle de l'objet. Les autres composantes peuvent alors être instanciées arbitrairement sur n'importe laquelle des stations de travail de la war room.

La particularité d'une telle approche est que les composantes logicielles qui constituent les agents n'apparaissent que sur les ordinateurs pour lesquels elles sont utiles. Bien entendu, la composante en charge de la gestion des données de l'objet (le modèle dans le cadre de MVC ou l'abstraction pour PAC) doit exister en un exemplaire dans la war room et être disponible pour toutes les composantes qui lui sont attachées, qu'elles soient locales ou distantes. Cette composante ne doit d'ailleurs être instanciée qu'une seule fois dans toute la war room, permettant ainsi de s'affranchir des procédés de synchronisation nécessaire lorsque des données sont dupliquées.

Nous pouvons dire que cette stratégie vise à distribuer l'environnement de telle sorte que chaque objet soit "conceptuellement" unique dans la war room. Seules ses représentations visuelles peuvent éventuellement être multiples. Cette idée constitue un parallèle avec l'environnement physique : lors d'une session de travail dans la war room, un environnement virtuel est distribué dans un espace physique comme s'il s'agissait de documents papiers qui étaient disposés sur des bureaux. Les deux environnements, physique et virtuel, sont donc en quelque sorte fusionnés sous la forme d'un unique espace mixte.

Cette proposition de distribution de l'environnement demande de développer un système de liaison réseau permettant aux composantes de communiquer entre elles. La constitution des liens réseaux dans notre proposition est singulière dans la mesure où la relation qui est établie entre un poste de travail et ses homologues est potentiellement multiple : l'entité communicante est la composante logicielle d'un agent. Les connexions réseau qui sont établies sont donc dépendantes de la distribution des composantes entre les ordinateurs.

La communication réseau mise en place n'opère donc pas à l'échelle de l'application, mais à l'échelle des composantes logicielles qui la constituent.

Pour résumer les raisons du choix de cette architecture distribuée au niveau des composantes, nous pouvons citer plusieurs points :

- **un "mapping" naturel des composantes sur les ordinateurs.** Le choix de distribuer les composantes logicielles peut naturellement être associé à la diversité matérielle que nous désirons mettre en avant. En effet, chaque poste de travail impliqué dans l'activité peut être assimilé à l'une des facettes logiques de l'application. Par exemple, un poste de travail dédié à l'affichage 3D se verra confier l'unique tâche de calculer ce mode d'affichage. Il apparaît donc naturel de ne lui assigner que les composantes logicielles associées à cette tâche.

- **un gain en terme de complexité.** Comme nous l'avons souligné dans le point précédent, chaque poste de travail ne prend en charge qu'un sous ensemble des différentes fonctionnalités de la war room. L'utilisation d'une architecture distribuée permet ainsi de ne déployer les composantes logicielles que sur les postes qui en ont réellement besoin. De plus, la complexité en terme d'encodage des objets interactifs se voit réduite de la même manière : chaque objet de l'environnement virtuel est unique même s'il est associé à plusieurs représentations visuelles simultanées. Ce caractère unique se reflète donc aussi au niveau de sa structure logicielle, puisque la définition d'un objet n'est jamais dupliquée.
- **maintien de la cohérence.** Le fait que chaque objet n'est instancié qu'une seule fois dans toute la plateforme facilite le maintien de la cohérence entre tous les postes de travail. Cette solution distribuée propose en effet un avantage comparable à une architecture client-serveur dans la mesure où il n'est pas nécessaire d'assurer une synchronisation cohérente entre plusieurs instances d'un même objet.
- **tirer parti du réseau local.** La configuration matérielle de la war room dispose d'une connexion réseau de type LAN (Local Area Network), disposant par conséquent d'une importante bande passante et d'une latence très faible. Notre approche tire parti de cette stabilité pour s'affranchir de la nécessité de dupliquer les composantes en vue du maintien de la cohérence. Dans le cadre d'une communication distante, les connexions internet sont en effet moins réactives, et nécessitent la mise en place de mécanismes de synchronisation complexes qui n'ont plus lieu d'être dans le contexte de l'utilisation d'un réseau local.

Pour être viable et éviter de contraindre le concepteur à respecter une implémentation complexe, cette solution nécessite cependant que les couches de connexion réseau soient transparentes pour l'application. Ainsi, il est possible de définir la structure interne de l'application indépendamment des contraintes de communication. Cette approche nous offre plusieurs avantages relatifs à nos besoins :

- **L'indépendance vis-à-vis des installations matérielles.** L'aspect logiciel du développement d'une application collaborative à l'aide de notre plateforme est complètement indépendant des installations matérielles. Le nombre de postes de travail mis en place dans la war room ne présente aucune contrainte vis-à-vis de la conception architecturale de l'application. De plus, l'aspect indépendant de chaque poste, c'est-à-dire ses dispositifs d'affichage ou d'interaction, ne constitue pas non plus un point critique dans la conception. L'application étant décrite comme s'il s'agissait d'un unique bloc hiérarchisé, le déploiement consiste en l'attribution des différents éléments constitutifs de ce bloc entre les postes de travail.
- **La complexité de l'application.** Lorsque la distribution des fonctionnalités entre différents ordinateurs est soumise à une architecture de type pair-à-pair ou centralisée, la définition globale de l'activité peut s'avérer d'une importante complexité. En effet, la conception des mécanismes internes de l'environnement virtuel devra prendre en charge à la fois la diversité des modes d'interaction ou de visualisation, mais sera également soumise à la complexité due à la planification de la distribution. Dans notre cas, la structuration des composantes n'est pas tributaire des mécanismes de communication réseau. Ce degré de complexité en moins, il est possible au concepteur de définir des applications contenant

des mécanismes de collaboration complexes sans se soucier du déploiement réseau de la plateforme.

- **La généricité du support logiciel.** La plateforme peut être structurée de façon entièrement libre, permettant de développer des applications correspondant à des domaines d'utilisation très divers. Le concepteur de l'application collaborative pourra accéder aux composantes déjà disponibles ou en développer de nouvelles, et les ré-agencer pour définir un nouveau cas d'utilisation.
- **La factorisation des composantes.** La communication réseau pouvant prendre place à n'importe quel endroit entre deux composantes, il est possible au concepteur de les répartir de façon pertinente entre les postes de travail. Cette transparence dans la communication permet par conséquent de n'instancier chaque composante qu'une fois, et de la connecter à ses proches à travers le réseau. Contrairement à une architecture pair-à-pair, aucune donnée ne nécessite d'être dupliquée entre les différentes instances de l'application. Toutes les informations et les mécanismes manipulés au sein de la plateforme peuvent donc être factorisés à volonté.
- **La prise en compte des possibilités matérielles.** En dépit de la forte indépendance entre la structure interne de la plateforme et les dispositifs matériels sur lesquels elle s'exécute, il est toutefois possible au concepteur de répartir les éléments constitutifs de son application de façon à optimiser l'utilisation des stations de travail. En effet, les composantes impliquées dans l'architecture de notre proposition ont toutes une fonctionnalité dédiée (affichage graphique, gestion de l'interaction par écoute des périphériques, gestion des données). Chacune de ces composantes pouvant être distribuée librement entre les postes de travail, il devient possible de gérer cette distribution selon deux considérations : la construction interne du logiciel peut se faire non seulement en fonction des besoins liés à l'activité (à travers la mise à disposition de représentations déplaçables, d'outils d'interaction distants, etc.), mais également en fonction des performances de chaque machine. Par exemple, un poste de travail doté de périphériques de stockage de l'information à hautes performances, comme peut l'être un serveur de base de données, est plus à même de stocker et de manipuler d'importantes quantités de données. Le concepteur de l'application pourra lui attribuer les composantes dédiées au management des données. Cette attribution n'aura pas d'impact direct sur la perception visuelle des utilisateurs, mais permettra en interne de bénéficier de performances supérieures à moindre coût. De même, si parmi les représentations visuelles d'un objet, l'une d'entre elles possède un degré de complexité graphique important (nombre de polygones d'un modèle 3D, résolution, ou algorithmes de rendu coûteux), il peut s'avérer nécessaire de dédier ce mode de représentation à une machine adaptée, par exemple un poste de travail équipé d'une carte graphique à haute performance. Pour ce faire, le concepteur de l'application aura donc la capacité de déterminer la distribution de façon à assigner les affichages graphiques aux postes les plus pertinents.

De même, la distribution telle qu'elle est abordée dans notre proposition peut également permettre de prendre en charge des dispositifs matériels aux capacités plus restreintes. Par exemple, l'utilisation d'un terminal mobile de type PDA ou smartphone peut parfois s'avérer problématique en raison des performances limitées de ces dispositifs. Ces dispositifs ne sont en effet pas toujours capables d'exécuter une application mettant en oeuvre un système de réalité virtuelle. Une architecture pair-à-pair ou centralisée impliquerait le développe-

ment d'une application cliente dédiée, plus légère en terme de besoin matériels, entraînant ainsi un surcoût en terme de développement. L'architecture que nous utilisons dans notre proposition permet au contraire de distribuer les composantes de façon à n'attribuer à un terminal que les éléments nécessaires à son bon fonctionnement. Ainsi, un smartphone utilisé uniquement pour interagir à distance ne devra pas prendre en charge l'affichage des objets, et ne se verra attribué que les composantes utiles à la lecture et l'interprétation de l'interaction.

4.1.3 Architecture multi-agents : utilisation du modèle MVO de SPIN|3D

Dans le contexte des architectures basées sur un ensemble d'agents interactifs, la plupart des modèles proposent des solutions qui consistent à grouper l'interaction et la visualisation à travers le développement d'une même entité. Ainsi, la lecture des périphériques d'interaction est réalisée par la même entité logicielle que celle prenant en charge l'affichage des informations. En effet, dans le cadre du développement d'une application classique, la distinction entre les flux entrants et sortants ne constitue pas une nécessité majeure, puisque le ou les utilisateurs sont amenés à interagir directement sur le poste de travail fournissant un retour visuel.

Cependant, cette approche ne répond pas aux besoins relatifs à la dynamique de la collaboration en war room. En effet, durant l'activité, les utilisateurs doivent être en mesure d'interagir sur les objets virtuels de façon indépendante de leur représentation graphique. La plateforme SPIN|3D offre justement ce type de comportement. La composante logicielle en charge de l'interaction, c'est-à-dire de l'écoute de périphériques d'interaction et de la modification des données relatives à l'objet virtuel, dispose en effet d'une certaine généralité. Dans le contexte d'une application distribuée telle que nous l'avons décrite dans le paragraphe 4.1.2, cette composante d'interaction autonome peut offrir la possibilité d'interagir sur n'importe quel objet virtuel à partir d'un dispositif quelconque, et ce, indifféremment de la localisation des composantes d'affichage. Notre choix s'est par conséquent porté sur l'architecture issue de SPIN|3D, qui est une dérivée du modèle MVC appelée MVO (pour Modèle, Vue, Outil).

Dans l'environnement virtuel prenant place dans notre proposition, un objet virtuel est constitué d'un modèle et d'une ou plusieurs vues qui lui sont liées. Pour assurer la cohérence visuelle de l'ensemble de l'application, chaque modèle aura pour rôle, à chaque modification de l'un de ses champs, de notifier la totalité des vues qui lui sont associées. De part la nature distribuée de cette architecture, les relations intervenant entre les composantes sont assimilables à une relation client/serveur. Cependant, contrairement à une application de type client/serveur classique, cette relation peut intervenir dans notre war room en de multiples endroits de l'ensemble d'agents. La plateforme logicielle de la war room est par conséquent interprétable comme un ensemble de clients et de serveurs répartis en de multiples points du réseau (fig. 4.1).

Dans notre proposition, la composante outil d'interaction ne constitue pas la seule entité logicielle capable d'impacter les données stockées par les modèles des agents interactifs :

- la distribution réseau permise par l'architecture que nous proposons dispose également d'un procédé de communication externe inclus dans SPIN|3D, appelé la DAI (pour Distributed Authoriting Interface). Il s'agit d'un bus de communication exécuté en parallèle du reste de l'application. Cette interface permet à une application tierce de se connecter à un objet de l'environnement, plus précisément à sa composante modèle, afin de consulter les données

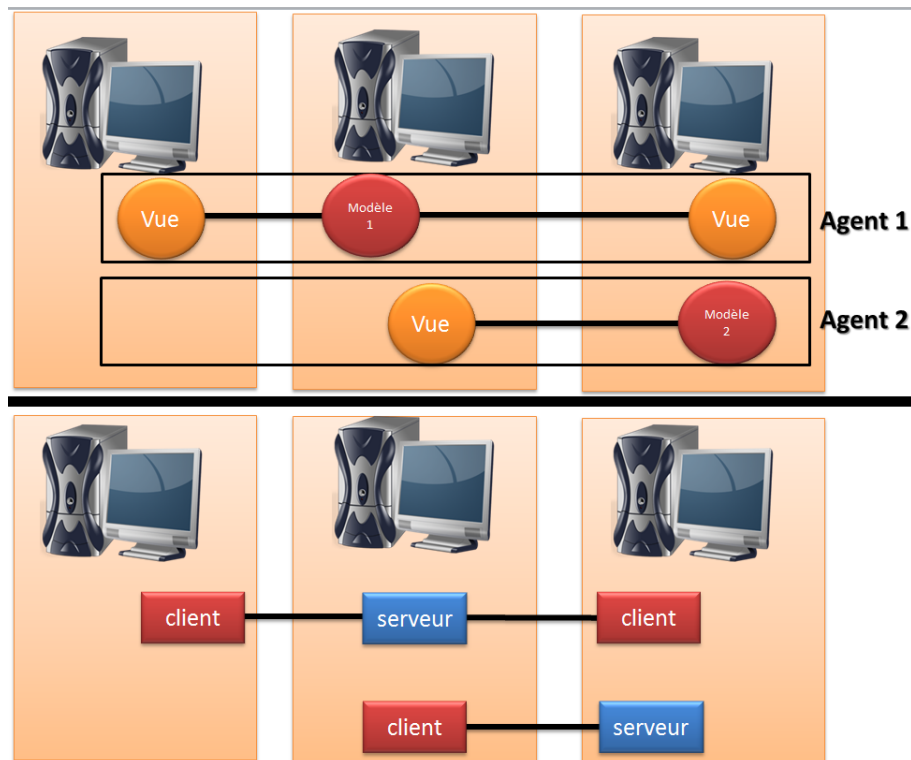


FIGURE 4.1 – En haut : distribution des composantes de 2 agents entre trois postes de travail. En bas : la relation client/serveur qui en résulte

qu'elle prend en charge, et de lui transmettre des événements d'interaction. Ce procédé permet à un développeur de créer de toute pièce une application interactive et grâce à la DAI, de la faire communiquer librement avec la plateforme logicielle de la war room comme si elle en faisait partie intégrante. L'intérêt de cette interface vis-à-vis de la collaboration de type war room réside dans la capacité qu'obtient le concepteur à intégrer dans les outils fournis aux utilisateurs une application particulière, adaptée aux besoins de l'un des corps de métier impliqués dans l'activité.

- comme nous l'avons précisé plus haut dans ce document, la possibilité de partager l'environnement virtuel avec un ou plusieurs utilisateurs distants constitue également un besoin potentiel pouvant intervenir dans nos cas d'utilisation. Pour ce faire, la plateforme logicielle est dotée d'un dispositif de communication dédié à ce type de situation. Il est alors possible d'exécuter d'autres applications conçues à travers la plateforme sur des postes de travail géographiquement distants, tout en maintenant la cohérence globale de l'environnement partagé. En terme de conception logicielle, ce procédé ne peut s'appuyer sur une distribution des composantes telle qu'elle apparaît dans une situation de collaboration exclusivement colocalisée. En effet, le temps de latence du réseau, ainsi que les pertes possibles dans une communication distante ne permettent pas d'assurer la stabilité de la plateforme. La collaboration distante dans notre proposition s'appuie donc sur le mode de fonctionnement de base de SPIN|3D, qui consiste à procéder à une duplication des données de façon à les rendre locales sur chaque site. Un processus de synchronisation des modèles homologues distants est ainsi exécuté en parallèle, assurant ainsi la cohérence de l'activité

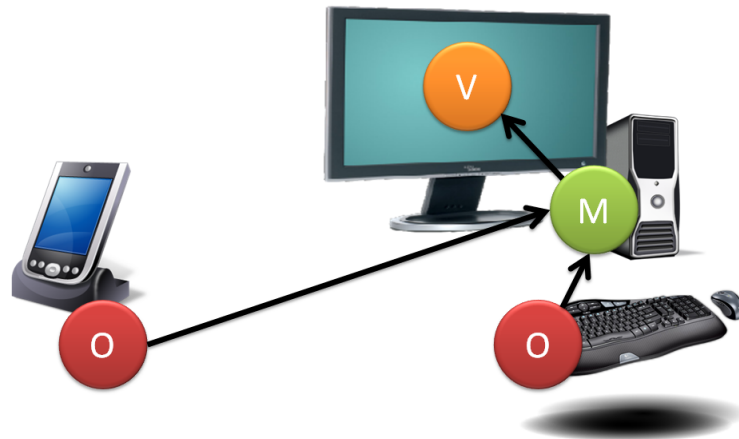


FIGURE 4.2 – Utilisation d'un terminal mobile comme dispositif d'interaction distante

entre les différents sites.

La distribution des multiples facettes de l'application collaborative entre les différents postes de travail intégrés à la war room consiste alors à répartir ces composantes MVO en fonction des cas de figure désirés. Pour illustrer au mieux la façon dont un concepteur peut choisir de distribuer les composantes, observons quelques exemples significatifs :

- **L'interaction distante.** L'ensemble des dispositifs matériels installés dans la war room est susceptible d'accueillir des terminaux mobiles connectés à la plateforme à travers une connexion réseau sans fil (fig. 4.2). Ces terminaux peuvent être utilisés par les participants de façon à disposer à tout moment d'un outil d'interaction individuel, quelle que soit leur localisation dans l'espace physique. Cette configuration offre par exemple la capacité de disposer d'un affichage persistant du projet sur un écran de grande taille, offrant une représentation complète apte à recevoir des événements d'interaction provenant de tous les utilisateurs. La distribution des composantes logicielles permet de fournir ces fonctionnalités en discernant le code utile à l'interaction, du reste de l'application : le terminal mobile se voit donc assigné la gestion de l'outil, tandis que le poste doté d'un affichage public prend en charge la gestion de l'objet logiciel, à travers le stockage et l'exécution des composantes modèle et vue. Ce poste de travail peut d'ailleurs disposer de ses propres dispositifs d'interaction mono-utilisateur et la distribution logicielle permet alors à deux utilisateurs de collaborer sur la même représentation sans souffrir des limitations matérielles du poste public.
- **Centralisation des données.** Comme nous l'avons évoqué plus haut, la manipulation d'objets constitués d'une importante masse de données peut s'avérer trop lourde à manipuler pour les différents postes de travail. Cette contrainte peut par exemple intervenir dans le cas d'une collaboration autour de calculs complexes issus d'une simulation mécanique. La plateforme est donc capable d'accueillir dans son installation physique un poste de travail dédié, de type serveur de calcul. La configuration logicielle prenant place dans ce contexte reprend alors un concept similaire aux architectures centralisées, en affectant à la machine serveur la charge de stocker les données de la simulation, et de les fournir

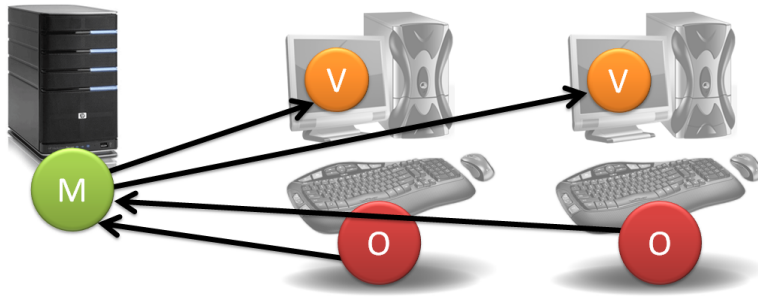


FIGURE 4.3 – Déploiement de l'application selon un modèle client-serveur

aux autres ordinateurs. Les postes de travail satellites sont alors dotés chacun d'une composante vue et les interactions issues des outils locaux sont envoyées au serveur (fig. 4.3).

- **Déplacement visuel de l'objet.** Nous avons montré dans la section 2.1.3 que les utilisateurs éprouvent des besoins de visualisation différents, variant notamment en fonction de la nature publique/privée de leur activité. Ainsi un utilisateur procédant à différents tests d'interaction sur l'objet éprouvera potentiellement le besoin de garder ces informations privées (en l'occurrence, sur un ordinateur portable) tant qu'il n'est pas parvenu à une proposition satisfaisante. Cependant, lorsque arrive le moment de présenter cette proposition à ses collaborateurs, il est nécessaire de diffuser la représentation sur un support plus grand, car plus confortable pour l'observation de plusieurs intervenants. En appliquant une distribution dynamique à la représentation visuelle de l'objet, cet utilisateur pourra travailler de façon privée sur l'objet, et lorsqu'il aura validé ses choix, envoyer virtuellement l'objet sur un écran de grande taille. En terme de distribution logicielle, cette action se traduira par le transfert de la composante vue de l'objet sur le second poste de travail (fig. 4.4).
- **vues simultanées semi-hétérogènes.** Notre modèle de conception, au même titre que le MVC classique, est capable de disposer simultanément de plusieurs composantes vue pour le même agent. L'algorithme utilisé par la vue pour procéder à la représentation visuelle des données du modèle peut s'avérer légèrement différent d'une vue à l'autre. Ainsi, lors de la réalisation d'un projet de conception d'un objet mécanique (un véhicule, par exemple), l'activité peut nécessiter l'affichage simultané du véhicule selon plusieurs modes d'affichage 3D. Ainsi, la plateforme peut proposer, pour la même composante modèle, une vue représentant un modèle 3D auquel sont appliqués des shaders complexes (couleur, spéculaire, réflexion), et une autre vue, plus technique, représentant la topologie du maillage 3D sous la forme d'un affichage de type "fil de fer" (fig. 4.5).
- **mise en commun de plusieurs objets sur le même espace visuel.** Lorsqu'une équipe pluridisciplinaire est impliquée dans l'activité de collaboration, chacun des participants est susceptible de n'avoir d'influence que sur l'un des objets constituant le projet. Si cet objet et les données qui le définissent disposent d'une grande indépendance de travail, il n'est pas nécessaire de tous les disposer dans le même environnement, mais au contraire de les distribuer entre les postes de travail de façon à les manipuler individuellement. Néanmoins, pour la cohésion du groupe et la gestion des conflits, il reste nécessaire d'avoir une vue d'en-

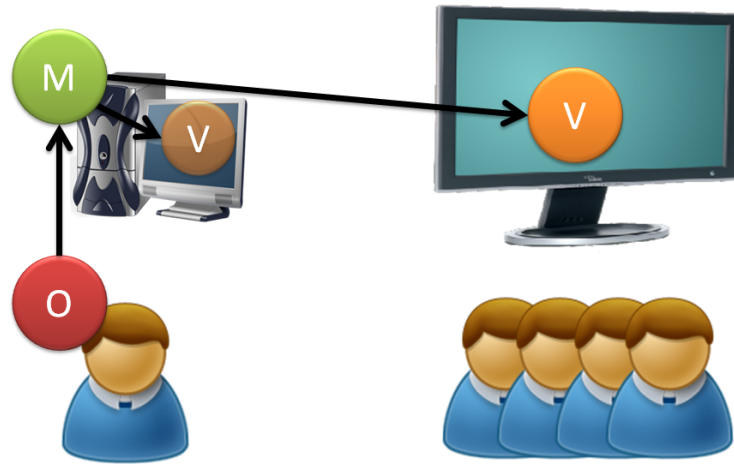


FIGURE 4.4 – Utilisation d'un écran tiers de grande taille pour la représentation visuelle. Lorsque la présentation est terminée, l'utilisateur isolé peut reprendre son travail individuel simplement en ramenant la composante Vue sur son poste de travail local

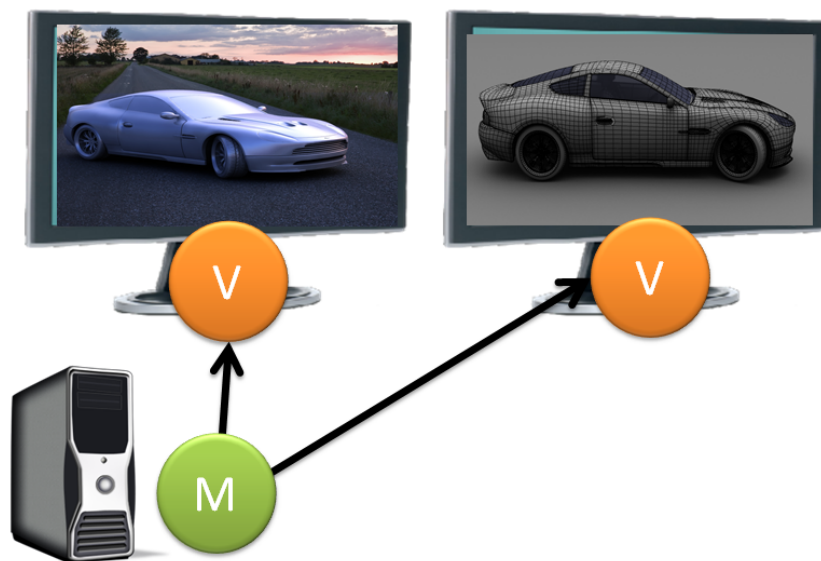


FIGURE 4.5 – Utilisation de deux composantes Vue concurrentes attachées au même modèle. Ces deux vues représentent simultanément le même objet selon des modes sensiblement différents

semble du projet, regroupant tous les objets sur un même affichage. Il est alors envisageable de proposer dans la war room un affichage persistant regroupant toutes les représentations simultanément. La distribution logicielle des agents peut alors consister en la duplication des composantes vue de chaque objet et le regroupement de tous ces duplicatas sur un même poste de travail, permettant une visualisation globale sur la surface d'affichage de ce poste (fig. 4.6). Cette approche pose néanmoins certains problèmes conceptuels. En effet, dans ce contexte, le poste de travail proposant l'affichage global ne permet pas de procéder

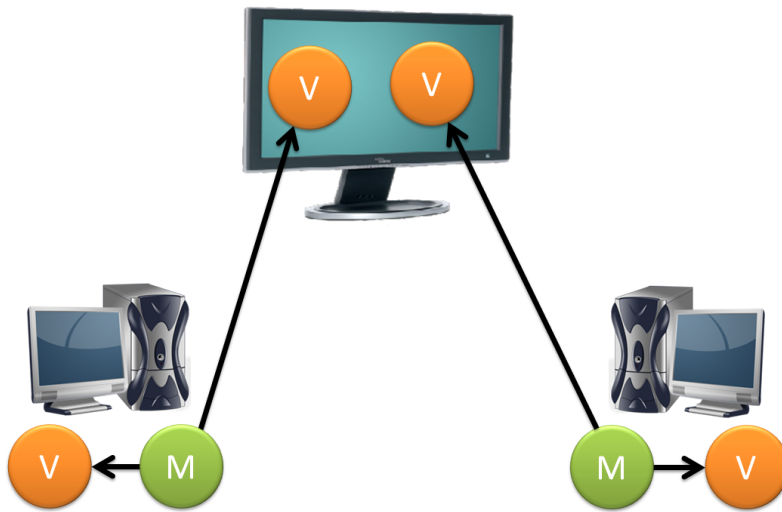


FIGURE 4.6 – Combinaison sur le même poste de deux vues issues de deux objets distincts conçus sur des postes de travail individuels. L'écran central peut ainsi être utilisé pour représenter l'assemblage de deux pièces mécaniques

à une activité d'assemblage des objets entre eux, puisque leur positionnement relatif les uns par rapport aux autres ne peut être stocké nulle part. Cette donnée ne peut légitimement pas être prise en charge individuellement par chaque composante modèle, puisque cette information de positionnement n'a aucune pertinence vis-à-vis des vues individuelles. La section 4.2 exposera une solution conceptuelle permettant entre autres de résoudre ce problème.

- **Intervention d'une application tierce.** Le support de l'intervention d'une application externe au sein de la war room peut également prendre part à la stratégie de distribution réseau de la plateforme. Ainsi, un utilisateur posté à une station de travail exécutant par exemple une application dédiée à la modélisation 3D peut présenter à ses collaborateurs le fruit de ses interactions directement dans le projet. L'utilisation d'un grand écran permet, là encore, de diffuser le modèle 3D de l'objet publiquement (fig. 4.7).

4.2 Support de canaux hétérogènes : la notion de modèle abstrait

La section 4.1 a montré l'intérêt d'utiliser une architecture de type MVO, et les avantages apportés par leur distribution réseau vis-à-vis de notre contexte d'utilisation. Cependant, la conceptualisation des objets sous forme d'agents MVO ne suffit pas à répondre complètement aux besoins relatifs à l'hétérogénéité de l'environnement virtuel qui doit prendre place dans la war room. Cette section a donc pour but de présenter un enrichissement de l'architecture existante pour lui permettre de prendre en charge un environnement non-uniforme tel que le notre.

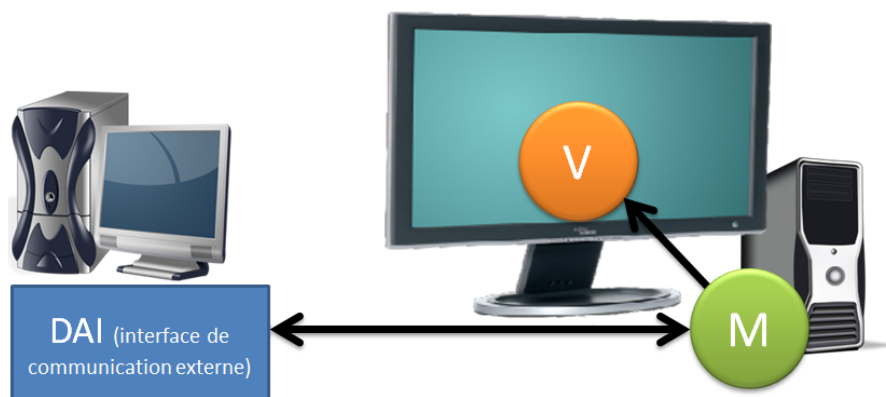


FIGURE 4.7 – Utilisation d’une application tierce et transmission des données produites au sein de la plateforme via la DAI

4.2.1 Le MVO : les limites du modèle

La principale fonctionnalité qui prend place dans notre war room réside dans sa capacité à présenter le projet sous différentes approches simultanées, que nous avons appelées les canaux (cf. section 2.3). Si l’on observe plus en détail les mécanismes contenus par les différentes composantes d’un agent MVO, nous pouvons constater que le lien qui est établi entre la définition fonctionnelle d’un objet (en l’occurrence le modèle) d’une part, et la ou les représentations visuelles qui en sont faites dans l’environnement (les vues) d’autre part, est un lien direct (fig. 4.8). Concrètement, cela signifie que les données prises en charge par le modèle sont directement interprétables par la vue pour procéder à une représentation perceptible par les utilisateurs. En effet, la vue n’est pas considérée dans le concept original comme une composante capable de stocker et de manipuler des données. Elle ne doit être constituée que d’algorithmes de rendu, quelle que soit leur nature. D’un point de vue conceptuel, nous pouvons en conclure que pour que le paradigme MVO soit correct, le modèle doit stocker un ensemble de données de nature géométrique. Une composante vue est donc en quelque sorte prévue pour être contrôlée par un jeu de données précis. Si deux vues distinctes ne reposent pas sur les mêmes paramètres, elles ne peuvent pas être naturellement rattachées à la même composante modèle.

Il n’est toutefois pas exclu que la composante Vue dispose de mécanismes de plus haut-niveau. Elle reste cependant très dépendante des données du modèle, puisque ce sont celles-ci qui vont "piloter" le comportement de la Vue. Les données en question sont donc dans tous les cas des données concrètes ayant un impact immédiat sur la représentation graphique. Si cette solution peut convenir à une application classique, elle n’est cependant pas compatible avec notre approche, qui vise à diversifier autant que possible les modes de représentation des objets virtuels. Selon ce modèle, l’hétérogénéité visuelle permise à travers les différentes vues s’avère donc restreinte.

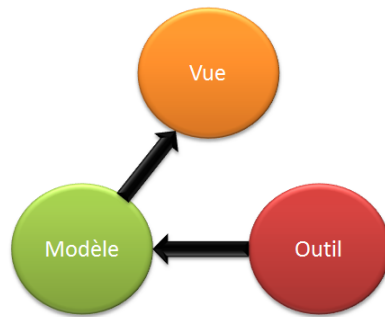


FIGURE 4.8 – Le modèle MVO de SPIN|3D.

Dans le contexte d'un environnement non-uniforme tel que notre plateforme, la notion de "données définissant un objet" revêt en fait une forme plus complexe. La définition d'un objet sous la forme d'un ensemble d'informations concrètes n'a de sens que dans le cadre d'un environnement dans lequel les représentations visuelles qui en sont faites sont homogènes, et peuvent ainsi constituer un référentiel sémantique pour tous les utilisateurs.

4.2.1.1 Données concrètes et données abstraites

Prenons l'exemple simple de l'observation d'une horloge par un groupe de personnes : si cet objet est systématiquement représenté sous la forme d'une montre analogique, le positionnement des aiguilles sur le cadran perçu visuellement par les utilisateurs n'est constitué que de valeurs représentant des angles de rotation. Cependant, les individus observant l'objet assimilent immédiatement cette information visuelle sous la forme d'une notion plus abstraite : l'heure. Les angles de rotation perdent alors une partie de leur aspect concret pour prendre dans l'esprit des participants une valeur sémantique plus profonde. La prise en charge de l'information "heure" par le logiciel s'avère par conséquent moins importante dans cet environnement uniforme, puisqu'elle est instantanément interprétée par les utilisateurs, qui peuvent alors communiquer entre eux naturellement sans perturber la compréhension du groupe. L'objet peut ainsi être défini dans l'application uniquement par des informations graphiques.

Cependant, dès lors qu'il est question d'observer et d'analyser un objet à travers une représentation différente, les données graphiques que nous venons d'évoquer perdent leur sens sémantique, puisqu'elles ne sont pas assimilables à un élément visuel perceptible par tous les participants. Elles ne sont alors plus significatives pour définir l'objet à une échelle globale. Dans le cas de l'horloge, si le groupe d'individus est placé dans un environnement impliquant la représentation de l'heure sous la forme à la fois d'une horloge analogique et d'une horloge à affichage numérique, la définition de l'heure sous la forme de valeurs concrètes nécessite deux jeux d'informations différents : les angles de rotation des aiguilles pour l'horloge analogique, et des valeurs booléennes pour l'horloge numérique ; ces valeurs booléennes correspondant aux interrupteurs contrôlant l'illumination de chaque zone des "afficheurs 7 segments". Dans ce cas de figure, les participants gardent la même faculté à communiquer autour de l'aspect sémantique de l'objet. Cependant, au sein du système

de visualisation, pour conserver la cohérence entre les deux représentations, les deux jeux d'informations graphiques doivent être liés à la même information abstraite "heure". Cette donnée n'est pas représentable directement, mais dans ce contexte, elle est indispensable au bon fonctionnement de l'application.

Cet exemple est symptomatique de la nécessité de manipuler la définition d'un objet selon plusieurs degrés d'abstraction. La notion de canal que nous proposons en section 2.3 repose sur cette constatation, qui nous pousse à considérer les objets sous leur forme globale, intelligible pour tous les canaux, c'est-à-dire sous la forme de données abstraites qui ne sont pas toujours immédiatement représentables. La composante modèle de notre architecture doit par conséquent prendre en charge des données abstraites. Il devient alors nécessaire de mettre en place les mécanismes permettant de manipuler les informations abstraites pour leur donner une dimension concrète indispensable à la constitution de représentations visuelles.

4.2.1.2 Cohabitation des données concrètes et abstraites

Prenons maintenant un deuxième exemple : nous avons évoqué plus haut la possibilité de représenter un modèle 3D selon différentes formes (avec shaders ou en mode wireframe). Cette solution reste valide dans la mesure où ces paramètres de représentation des surfaces 3D constituent des constantes qui font partie intégrante de l'algorithme de rendu exécuté par la vue. Dans ce contexte, les deux représentations reposent sur la même donnée concrète : la topologie 3D du modèle.

Cependant, la diversité des compétences qui peuvent prendre part à l'activité collaborative au sein de la war room nécessite l'utilisation de modes de représentations plus divers et ne reposant donc pas sur les mêmes données bas niveau : dans le cas de la représentation d'une voiture dans l'environnement, une représentation 2D sous forme de plans techniques de type blueprint ne peut s'appuyer sur un modèle fournissant uniquement des informations 3D. En effet, la conversion de la topologie 3D en une représentation 2D impliquerait la mise en place de mécanismes trop complexes pour être justifiés dans le code de la vue 2D.

La première solution avancée pour assurer le support de plusieurs représentations différentes pourrait consister en l'assignation au modèle de la totalité des données utilisables par les différentes vues. Cependant, cette approche aboutit à plusieurs contraintes :

- le modèle perd sa fonction de définition sémantique de l'objet et se voit attribué un rôle de serveur de données pour les multiples algorithmes exécutés par les vues. Notre approche vise au contraire à exploiter la composante modèle comme une description globale sous la forme d'une abstraction haut-niveau de l'objet.

- le maintien de la cohérence des données est complexe. En effet, dans notre exemple, si la représentation 2D de la voiture repose sur une forme définie par des courbes paramétrées, cela signifie que le modèle prendra en charge deux données : une liste de polygones pour la vue 3D et une liste de points de contrôles pour la vue 2D. Selon une telle configuration, toute modification de l'un de ces deux ensembles doit immédiatement être retranscrite sur son homologue. Ce choix aurait pour conséquence que le modèle devrait intégrer la totalité

des mécanismes de retranscription pour toutes les données qu'il prend en charge. Il serait d'ailleurs nécessaire de modifier le modèle pour chaque nouvelle vue qui lui serait attachée. Cette approche s'éloigne de la structuration précise du modèle MVC et de la spécialisation des composantes qu'il implique.

Cette solution ne s'avère par conséquent pas pleinement satisfaisante pour combler les besoins d'un environnement non-uniforme tel qu'il doit apparaître dans notre war room.

La seconde proposition envisageable pour prendre en charge l'hétérogénéité de notre plateforme serait alors d'enrichir la composante vue, de façon à la doter de la faculté de stocker les données visuelles qu'elle nécessite. Cependant, cette approche implique la nécessité d'intégrer à la fois des fonctions de stockage de champs, des algorithmes de conversion visant à retranscrire les données du modèle en données visuelles, ainsi que les algorithmes classiques de représentation de ces données visuelles. La vue deviendrait alors une composante hybride remplissant à la fois les rôles de modèle, de vue et de contrôleur. Cette solution souffre de ses limites en terme de factorisation du code. En effet, deux représentations visuelles proches reposant sur des données géométriques similaires reposeraient sur deux composantes vue distinctes, mais implémentant les mêmes mécanismes.

La solution que nous pouvons proposer consiste donc à scinder cette Vue enrichie sous la forme de deux sous-composantes. La première serait similaire à une Vue MVC classique et serait chargée uniquement de procéder à l'affichage des données. La seconde sous-composante serait quant-à elle similaire à un modèle dans la mesure où elle serait en charge des données concrètes. De cette manière, l'ajout dans l'application d'une nouvelle vue reposant sur les mêmes données ne nécessiterait d'implémenter que cette dernière.

Comme nous l'expliquerons plus loin dans ce chapitre (cf. 4.2.5 page 134), cette solution ne se limite pas à l'utilisation de deux niveaux d'abstraction. Dans le cadre d'une application proposant un ensemble de vues qui peuvent être regroupées en plusieurs catégories, il est envisageable de définir les objets sous la forme de trois (ou plus) niveaux d'abstraction. Par exemple, pour une application proposant deux canaux 3D distincts, il est possible de déterminer trois niveaux d'abstraction pour les données : un niveau abstrait global, un niveau d'abstraction intermédiaire constitué de données 3D générales et un niveau concret, qui lui, prendra en charge des données concrètes propres à chaque canal.

4.2.2 Les modèles existants

Comme nous l'avons souligné plus haut, les différents canaux impliqués dans l'application collaborative peuvent être étroitement liés. Les conséquences d'une interaction réalisée à travers un canal peuvent impacter les autres canaux de la war room. En effet, l'objet virtuel concerné, même s'il est perçu de façon différente entre les postes de travail, reste le même objet sémantique. S'il n'existe pas de lien concret entre les données manipulables et représentables dans chaque canal, la définition abstraite de l'objet d'interaction constitue le lien immuable permettant de maintenir une cohérence globale de l'objet malgré le caractère hétérogène des canaux à travers lesquels il est perçu. Pour fonctionner correctement, notre plateforme logicielle se doit par conséquent de rétablir ce lien abstrait et lui donner une certaine consistance dans l'implémentation des canaux.

Si l'on observe le modèle de conception PAC décrit dans l'état de l'art, nous pouvons en dégager sa capacité à manipuler un certain niveau d'abstraction dans les données. Il constitue par conséquent une forme de conceptualisation intéressante des objets interactifs. En effet, contrairement au MVC, la communication établie entre la définition fonctionnelle de l'objet (c'est-à-dire les données stockées par la composante "abstraction") et sa représentation visuelle (la composante "Présentation") passe par une composante "intelligente", le Contrôle. Ce Contrôle permet de retranscrire à la volée les informations de l'abstraction pour les rendre utilisables par les algorithmes de rendu exécutés par la présentation. Ainsi, pour reprendre l'exemple de l'horloge, la représentation d'une donnée abstraite "heure" est convertie par le contrôle en valeurs d'angles, qui une fois transmises à la présentation permet de visualiser cette heure sous une forme concrète, à savoir une horloge à aiguilles. Cette architecture permet par conséquent de décrire l'objet virtuel sous une forme abstraite, tout en bénéficiant de représentations visuelles concrètes. Ce modèle possède toutefois quelques inconvénients :

- les informations concrètes fournies par le contrôle à la présentation ne possèdent pas de consistance en terme de stockage de données. Dès lors, le contrôle se voit contraint d'exécuter ses algorithmes de conversion successivement pour toutes les présentations impliquées dans l'affichage de l'objet, même si certaines de ces présentations reposent sur des données concrètes similaires.
- il ne permet nativement de manipuler que deux niveaux d'abstraction : les données abstraites stockées par l'abstraction et les données concrètes générées par le contrôle. La diversité des représentations visuelles peut cependant nécessiter des niveaux plus nombreux, chaque degré d'abstraction pouvant définir un nouveau degré plus concret, utilisable directement par les présentations.

Ces deux contraintes peuvent être contournées par la mise en place d'une hiérarchie d'agents PAC, chacun étant en charge de la gestion d'un niveau d'abstraction (fig. 4.9). Les contrôles prennent alors en charge la conversion de ces différents niveaux, les différentes composantes abstraction stockant les données relatives à chaque niveau. Cependant, selon cette architecture, l'objet virtuel pourtant unique en terme de sémantique se voit constitué de plusieurs agents interactifs. L'arbre hiérarchique des agents PAC représentant l'environnement et ses différents objets peut par conséquent s'avérer plus difficile à analyser, puisqu'aucune distinction précise n'est faite entre les agents issus d'objets différents.

Un autre modèle, ou plus précisément un métamodèle, constitue une conceptualisation intéressante des objets d'un environnement virtuel. Le métamodèle de Dewan [Dew99] consiste justement en la segmentation de la totalité de l'application sous la forme de différents degrés d'abstraction. Ces degrés sont présents en un nombre indéfini, comme peuvent l'être les degrés d'abstraction nécessaires à la gestion des canaux de notre war room. Si la spécification de ce modèle n'apporte aucune information précise concernant le contenu de ces différents niveaux (c'est d'ailleurs ce qui le qualifie de métamodèle), il apporte une vision de la structuration logicielle proche de nos besoins. La critique que nous pouvons établir vis-à-vis de ce modèle réside dans sa gestion des échanges avec l'utilisateur. Selon la structuration de Dewan, la totalité des échanges entre l'application et l'utilisateur est réalisée au niveau de la couche la plus basse, c'est-à-dire au niveau le plus concret (qui constitue la gestion des périphériques d'interaction et de visualisation). Cependant, dans

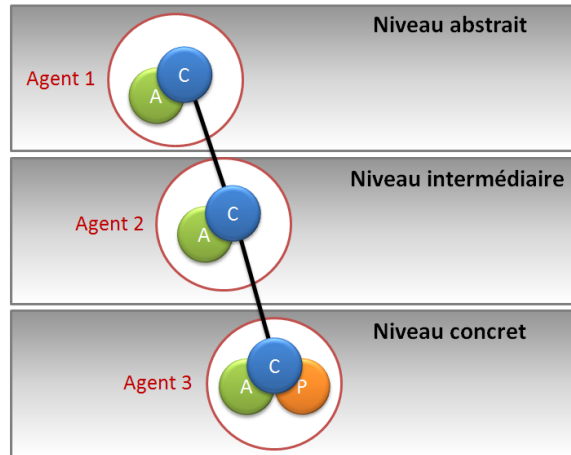


FIGURE 4.9 – Hiérarchie d'agents PAC pour gérer plusieurs degrés d'abstraction

la plateforme logicielle de notre war room, le concept de canaux n'est pas opposable aux interactions relatives à un degré abstrait. Chaque degré présent est en effet susceptible d'être exploité directement par une représentation visuelle. Dans ce contexte, si un objet implique la présence de canaux reposant sur des degrés d'abstraction différents, les échanges entre l'application et l'utilisateur peuvent intervenir sur toutes les couches conceptuelles de l'objet.

Nous proposons d'enrichir le modèle de conception MVO, de façon à le doter de différents degrés d'abstraction permettant ainsi d'intégrer les différents canaux nécessaires à l'activité collaborative. Pour ce faire, nous nous sommes appuyés sur les points forts du modèle PAC, en particulier sur sa composante contrôle et sur l'application du métamodèle de Dewan à notre architecture pour faciliter la gestion des données nécessaires à chaque canal.

4.2.3 M_NV-O : Adaptation de la structure MVO

L'architecture que nous proposons reprend ainsi l'idée de PAC, c'est-à-dire l'utilisation d'une composante dédiée aux échanges inter-composantes, dans l'objectif d'établir un lien consistant entre les données concrètes propres à chaque canal et les données abstraites constituant l'objet virtuel. Nous proposons d'adapter cette notion de composante contrôle en la dotant de possibilités de stockage analogues à la composante modèle. Pour cette raison, nous appelons cette nouvelle composante le "modèle concret". Notre modèle de conception, que nous appellerons M_NV-O repose ainsi sur la décomposition fonctionnelle de l'application sous la forme de quatre types de composantes (fig. 4.10) :

1. **l'outil**, constitué d'un automate d'opérateurs : cette entité logicielle est utile à la gestion de l'interaction. Elle peut être considérée comme externe au reste de l'architecture puisqu'elle est générique et permet d'accéder aux données de n'importe quel objet virtuel. Elle n'est donc pas à proprement parler un élément constitutif des agents.
2. **la vue**, chargée de l'affichage des objets virtuels à partir de leurs informations concrètes.

3. **le modèle abstrait** : cette composante remplit le même rôle que le modèle classique, mais il est utilisé de façon sensiblement différente. Dans notre cas, il constitue la base de données abstraites qui sert de patron persistant commun à l'ensemble des canaux d'un objet. Cela signifie qu'il est utilisé pour stocker des informations suffisantes pour définir un objet sous la forme de données qui ne sont pas nécessairement représentables graphiquement. Par conséquent, contrairement au modèle classique du paradigme MVC, le modèle abstrait n'est pas nécessairement lié à une composante vue de façon directe. Dans les cas d'utilisation impliquant plusieurs canaux (donc plusieurs degrés d'abstraction) le modèle abstrait n'aura aucun impact direct sur les représentations graphiques formulées par les vues. Il ne possèdera de lien qu'avec notre nouvelle composante, le modèle concret. Cependant, dans les cas d'utilisation simples ne nécessitant pas de distinction entre données abstraites et données concrètes, le modèle abstrait se comportera comme un modèle classique, c'est-à-dire qu'il sera relié de façon directe à une composante vue.

4. **le modèle concret** : cette nouvelle composante constitue une entité hybride, qui reprend à la fois les concepts du modèle et ceux de la composante " contrôle " de PAC. Il s'agit ici d'une composante " intelligente ", qui a connaissance de la définition concrète d'un objet par rapport à un canal donné, ainsi que des mécanismes liant son propre degré d'abstraction à celui du modèle abstrait. Ce modèle prend place au centre des communications, en s'interposant entre le modèle abstrait et la vue du canal concerné. La relation établie entre le modèle concret et le modèle abstrait est alors la même que celle qui existe entre un modèle et une vue dans le paradigme MVC : chaque modèle concret de l'architecture s'enregistre en tant qu'observateur auprès du modèle de niveau d'abstraction supérieur et se voit alors notifié de toute modification abstraite appliquée sur l'objet.

En terme de décomposition, ce modèle concret permet de factoriser les communications en fonction du degré d'abstraction. Il est important de souligner que les mécanismes d'adaptation entre données abstraites et données concrètes sont à double sens. Lors de l'action effectuée par un outil sur une donnée concrète, le modèle concret utilisera ces mécanismes pour retranscrire les modifications sur le modèle abstrait. A l'inverse, lorsque le modèle abstrait subit une modification de ses informations (à partir d'un outil ou d'un modèle concret) et qu'il en notifie ses modèles concrets, ces derniers exploiteront ces mêmes mécanismes pour en répercuter les modifications sur les données concrètes (fig. 4.10).

En regard du processus d'adaptation entre les données issues de deux niveaux d'abstraction, il est nécessaire de préciser que l'invocation de ces mécanismes n'est pas systématique. En effet, en dépit du besoin de maintenir une certaine cohérence entre les vues, il est parfaitement envisageable que les données concrètes soient exclusives à un canal. Certaines interactions sur des données graphiques n'ont pas d'influence sur la facette " production " de l'application. C'est notamment le cas de la réorganisation des éléments à l'écran par l'utilisateur, pour une lisibilité accrue. Dans ce cas précis, l'impact sur les données abstraites sera nul et le modèle concret n'opèrera aucune modification sur le modèle abstrait. A l'inverse, si un canal ne permet pas de représenter toutes les caractéristiques abstraites d'un objet virtuel, il est possible que certaines données prises en charge par le modèle abstrait n'aient pas d'influence directe sur le modèle concret. La modification de ces in-

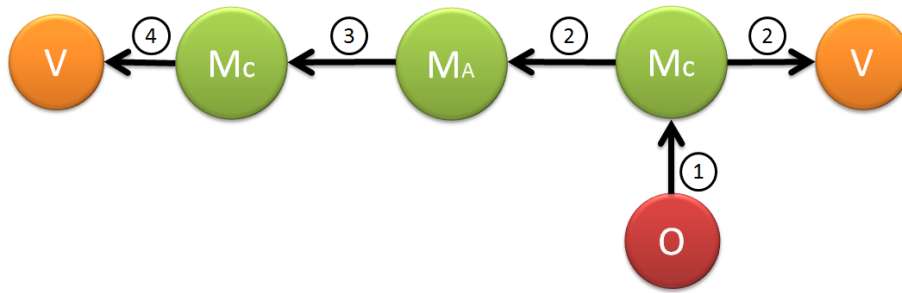


FIGURE 4.10 – Déroulement des échanges inter-composantes :

- 1) l'outil modifie les données visuelles du modèle concret
- 2) le modèle concret notifie sa vue des changements et retranscrit l'interaction sous la forme d'un évènement abstrait transmis au modèle abstrait
- 3) suite à la réception d'un évènement, le modèle abstrait modifie ses champs et en notifie les modèles concrets qui lui sont attachés
- 4) le second modèle concret ayant reçu notification du modèle abstrait retranscrit cette interaction sur ses propres données et notifie sa vue.

formations abstraites n'entraînera alors pas l'exécution d'un mécanisme de retranscription concrète.

Pour aller plus loin dans l'idée d'adaptation des évènements concrets en évènements abstraits, il faut préciser que le modèle concret peut inclure des mécanismes complexes, en s'appuyant sur plusieurs données concrètes à la fois. La composition des multiples informations graphiques peut aboutir à la génération d'un évènement abstrait particulier : pour reprendre l'exemple de l'horloge évoqué plus haut, la combinaison des angles des deux aiguilles correspond à la donnée abstraite heure. Une unique information abstraite peut ainsi voir sa valeur dépendante de plusieurs informations visuelles à la fois. La réciproque est bien sûr valide et la modification de la valeur abstraite se voit retranscrite sur plusieurs données concrètes.

Le choix d'avoir confié la prise en charge des données concrètes à une composante distincte plutôt qu'à la vue permet de tirer parti de la distribution réseau de l'architecture, de façon à factoriser les informations : lorsque la war room est configurée pour exécuter simultanément des vues issues du même canal, il n'est pas nécessaire de dupliquer ces données. Le modèle concret associé à ce canal est alors unique et fournit aux vues qui le concernent les données nécessaires aux représentations graphiques, indifféremment de leur répartition sur le réseau (fig. 4.11).

4.2.4 Transparence de l'adaptation et distribution distante

Nous avons indiqué que le modèle concret implémente à la fois un composant modèle standard (pour remplir les besoins de sa relation avec la vue) et un composant vue (pour prendre en charge la relation allant du modèle abstrait au modèle concret). Cette approche

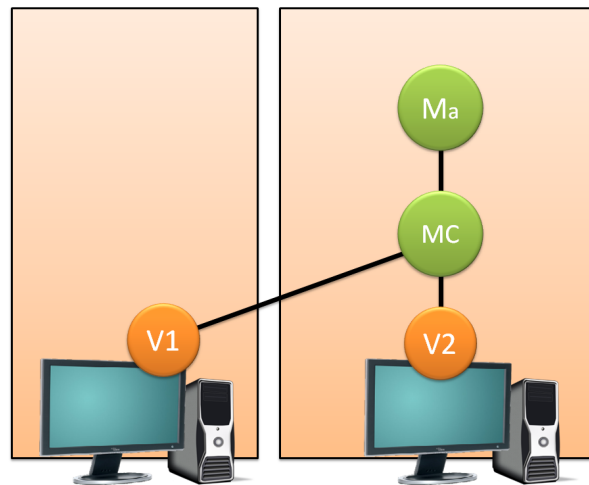


FIGURE 4.11 – Factorisation des données pour plusieurs vues appartenant au même canal, mais exécutées sur des postes différents.

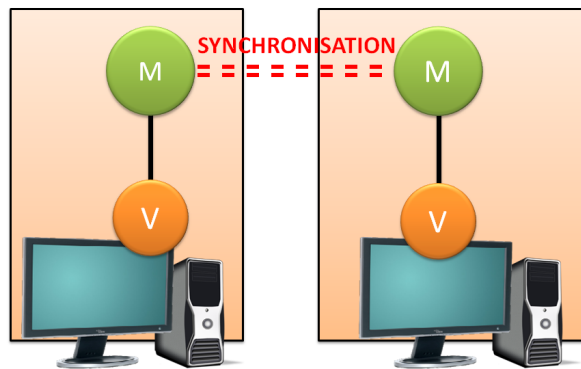


FIGURE 4.12 – Communication entre deux sites distants selon le modèle MVO classique : deux modèles similaires sont instanciés de part et d'autre.

permet une transparence totale vis-à-vis de l'architecture MVO existante, sans nécessiter de modification de l'API : l'architecture s'adapte donc sans modification des mécanismes de communication internes des agents MVO. La seule relation nouvelle nécessitant un travail d'implémentation réside dans la mise en place des mécanismes d'adaptation des données entre les différents degrés d'abstraction.

La transparence qui résulte de cette architecture permet de préserver également les systèmes de communication distante. En effet, le mode de communication distante repose sur la synchronisation de deux modèles. En l'occurrence, le modèle concret étant une forme dérivée du modèle abstrait (et donc du modèle classique), son insertion dans la relation distante ne nécessite pas l'implémentation de mécanismes particuliers. Selon l'architecture MVO originale, le déploiement des composantes entre les sites distant doit respecter une certaine symétrie. En effet, la synchronisation qui opère dans ce cas de figure a lieu au niveau du stockage des données. Les composantes modèle instanciées entre les deux sites doivent par conséquent être les mêmes (fig. 4.12).

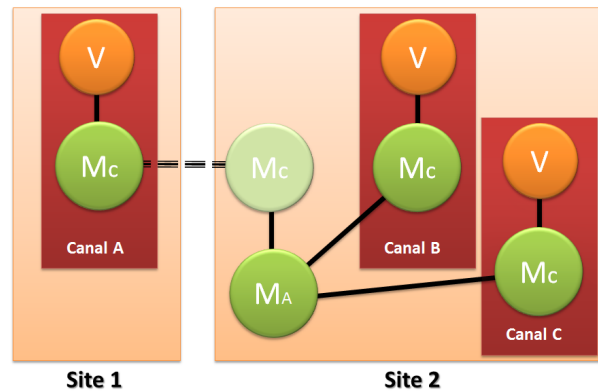


FIGURE 4.13 – Exemple de répartition asymétrique entre deux sites distants.

L'insertion des différents niveaux d'abstraction permet à l'utilisateur de la plateforme de s'affranchir de cette contrainte de symétrie. En effet, si un site distant nécessite la symétrie des données qu'il cherche à synchroniser sur le réseau, il n'est pas nécessaire que ce site ait conscience de tous les niveaux d'abstraction d'un objet (fig. 4.13). En terme de distribution des composantes, cela signifie que pour un objet donné, le site distant ne procédera à l'instanciation que de son modèle concret, tandis que le site local prendra en charge l'exécution du modèle abstrait.

4.2.5 Multiplicité des niveaux d'abstraction

Concernant la notion de niveaux d'abstraction d'un objet virtuel, nous pouvons dire que la couche correspondant au modèle abstrait, que nous appelons le plus haut niveau d'abstraction, constitue une notion qui n'est pas précisément définissable. En effet, il existe une multitude d'abstractions possibles pour définir un objet. En ce sens, notre proposition n'est pas nécessairement limitée aux seuls deux niveaux évoqués dans les exemples jusqu'à maintenant. Si nous considérons une donnée abstraite comme étant une donnée ayant un impact direct sur plusieurs approches (ie. canaux), il est envisageable qu'un cas d'utilisation exploite trois (ou plus) degrés d'abstraction. Chacun de ces degrés abstraits constitue alors un facteur commun entre plusieurs modèles concrets de plus bas niveau d'abstraction. Les composantes constituant un agent de notre plateforme peuvent donc être organisées de façon hiérarchique : entre une composante vue et le modèle abstrait de l'objet concerné, il peut ainsi exister une chaîne plus ou moins longue de modèles concrets.

La figure 4.14 présente l'exemple d'un objet complexe de ce type : l'agent est ici constitué de trois niveaux d'abstraction. Le modèle abstrait constitue le plus haut niveau possible, commun à toutes les vues. Dans cet exemple, quatre vues sont disponibles, caractérisées par leur appartenance à l'un des trois canaux déployés (en l'occurrence, les vues 1 et 2 sont similaires et reposent sur les mêmes données concrètes). Le canal 3 fonctionne sur le principe évoqué plus haut, dans la mesure où les données graphiques exploitées par la vue sont directement liées à celles du modèle abstrait. En revanche, les canaux 1 et 2 ne sont que partiellement indépendants entre eux. En effet, il existe un niveau d'abstraction permettant de factoriser certaines informations communes à ces deux canaux. Ce degré

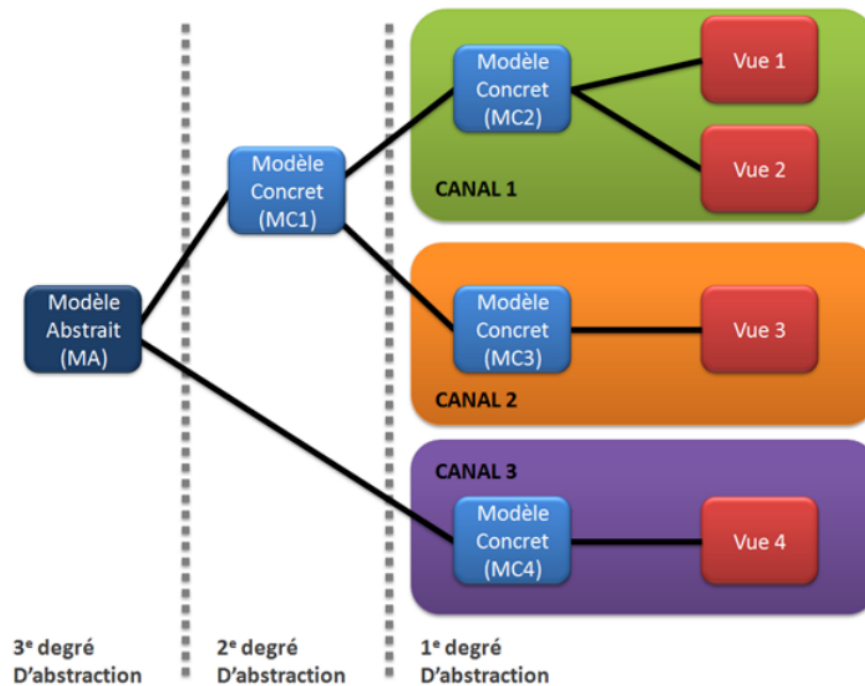


FIGURE 4.14 – Exemple de hiérarchie de plusieurs degrés d'abstraction. Le premier degré constitue le niveau "concret".

supplémentaire se reflète dans notre exemple, à travers le modèle concret 1 : celui-ci peut alors être considéré à la fois comme abstrait et concret. Il est en effet concret vis-à-vis du modèle le plus abstrait et abstrait vis-à-vis des Modèles concrets 2 et 3.

La conséquence de cette hiérarchisation est que si une donnée concrète bas niveau est modifiée par un outil sur une des deux premières vues, les conséquences de cette interaction sur le niveau d'abstraction supérieur (MC1) peuvent potentiellement avoir des répercussions sur le second canal et donc sur la vue 3. Dans cette situation, si ces modifications ne possèdent pas de répercussions sur le plus haut niveau d'abstraction (MA), cela signifie que la vue 4 n'est pas concernée par ces modifications. Elle ne sera par conséquent pas notifiée inutilement.

Bien sûr, le rôle du second niveau d'abstraction (en l'occurrence, le modèle concret 1) ne s'arrête pas nécessairement à une simple activité de lien entre ses modèles voisins. En effet, de part l'aspect générique de la composante modèle utilisée, il est envisageable d'intégrer à ce système une composante vue supplémentaire, capable de représenter l'objet virtuel directement à partir des données de ce modèle, sans nécessiter de composante intermédiaire. Cette situation n'est bien sûr possible que si les données prises en charge par ce modèle possède une dimension graphique, permettant leur représentation directe selon un canal particulier.

4.2.6 Insertion d'une application tierce dans le contexte multi niveaux d'abstraction

L'insertion d'une application tierce dans le contexte logiciel de la war room bénéficie également de la hiérarchisation sous forme de plusieurs degrés d'abstraction. En effet, une application externe prenant place dans l'activité collaborative de l'équipe est assimilable à un canal, à travers lequel les utilisateurs peuvent modifier l'objet avec des outils particuliers.

L'objet virtuel, une fois représenté dans cette application externe, sera défini au sein de l'application par des données qui sont propres au logiciel et aux fonctions d'interaction qu'il offre. Ces données sont ainsi peu susceptibles d'être les mêmes que celles employées pour définir l'objet dans la plateforme logicielle de la war room. En particulier dans le cadre d'une application propriétaire, il est difficile de connaître précisément la nature des informations manipulées par le logiciel. Ces informations nécessitent par conséquent d'être traitées par certains mécanismes visant à les retranscrire sous la forme de données valides pour l'encodage de l'objet choisi dans la war room.

S'il est possible d'implémenter la couche communicante de cette application externe de façon à intégrer directement les traitements nécessaires à la compatibilité avec la plateforme, il peut s'avérer plus cohérent de considérer l'intervention de cette application comme un canal supplémentaire à la war room. Ainsi, les mécanismes intervenant dans la transmission de l'interaction entre les données peuvent être implémentés de la même façon que pour les autres canaux, et les données résultantes peuvent être utilisées pour la représentation de canaux similaires. Par exemple, une application de modélisation 3D stockant les données géométriques de l'objet sous la forme d'un fichier propriétaire établira une communication avec un modèle concret possédant un champ dédié au stockage de la liste des polygones de l'objet. Ce champ sera utilisable dans la war room par une composante vue permettant une autre représentation 3D du même objet ou même par une autre application externe permettant de manipuler des données similaires (fig. 4.15).

4.2.7 Intégration dynamique d'un nouveau dispositif

Parmi les avantages de cette architecture vis-à-vis de l'activité collaborative, il est important de souligner la dynamique que peut offrir une telle configuration : les différents niveaux d'abstraction communiquent entre eux par une relation de type observateur/observable, qui consiste à notifier les composantes qui se sont enregistrées à chaque détection d'un événement significatif. Concrètement, cela signifie que les modèles, quel que soit leur niveau d'abstraction, ne sont dépendant du niveau supérieur que lorsqu'ils sont enregistrés auprès de lui. Dans ce contexte, cela signifie que les modèles possèdent une certaine autonomie, indépendamment du niveau d'abstraction qu'ils représentent. Dans le cadre de l'activité collaborative, ce constat implique une certaine dynamique dans la prise en compte des canaux.

Prenons un exemple selon lequel une équipe travaille dans la war room sur un projet impliquant plusieurs objets requérant des domaines d'expertise variés. Un utilisateur spécialisé, ayant travaillé sur l'un des objets de façon individuelle peut décider d'intégrer son ordinateur portable dans l'installation matérielle de la pièce, connecter son canal personnel au

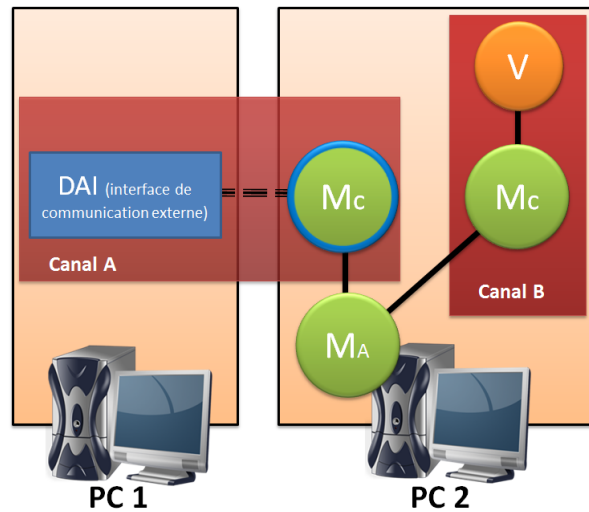


FIGURE 4.15 – Utilisation d'un modèle concret comme une interface communiquant avec la DAI. Le modèle concret (cercle de bleu sur le schéma) prend en charge les mécanismes de retranscription concret/abstrait, et est ainsi réutilisable par exemple en y ajoutant une composante vue.

reste de la plateforme et mettre à jour le projet automatiquement. En terme d'intégration des composantes, ce scénario implique de la part de cet utilisateur qu'il dispose sur son poste de travail d'un canal à part entière, c'est-à-dire d'une vue, et d'un ou plusieurs modèles concrets associés. Le comportement autonome de ce canal lui permet par conséquent de travailler " hors ligne ", c'est-à-dire sans être connecté au reste de la plateforme. De même, à l'instant où il décide d'intégrer son travail dans la war room, il suffit au modèle de son canal personnel de s'enregistrer auprès du modèle de niveau d'abstraction supérieure dans la war room, activant ainsi le maintien de la cohérence entre son travail individuel et les objets de la plateforme. Bien sûr, ce scénario implique également la nécessité pour les composantes locales de son ordinateur de connaître les mécanismes de retranscription nécessaire à la communication entre ces deux niveaux d'abstraction (fig. 4.16). Il faut cependant préciser que la synchronisation dynamique soulève la question d'assurer la mise en cohérence des données entre les deux sites distants au moment de la connexion.

Selon une approche similaire, il est possible d'intégrer à l'application reposant sur notre plateforme la capacité de gérer un mode de travail privé. Au cours de la collaboration, si certains utilisateurs décident d'interagir sur un objet de façon autonome à travers un canal, c'est-à-dire sans influencer le déroulement du reste de l'activité, il est éventuellement possible de " déconnecter " le modèle concret associé à ce canal du reste de l'application. Cette situation équivaut à désactiver les échanges entre le modèle concret et son modèle abstrait, sans pour autant perturber la mise à jour de la vue. Les utilisateurs auront ainsi la capacité de tester différentes manipulations sur l'objet sans que les représentations de cet objet sur les autres canaux ne soient influencées. Une fois parvenus à une proposition satisfaisante, ils pourront alors injecter leurs modifications dans la plateforme pour les soumettre à une validation collective par l'équipe. Cette reconnexion se reflète alors au niveau logiciel par le réenregistrement du canal auprès du modèle abstrait et l'envoi des données concrètes retranscrites pour rétablir la cohérence entre tous les canaux.

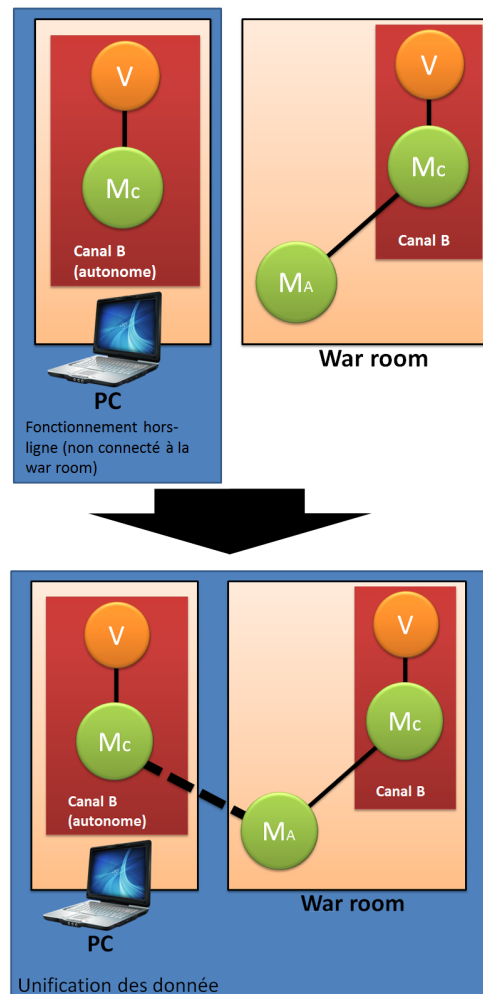


FIGURE 4.16 – Partie supérieure : poste de travail autonome fonctionnant en mode hors-ligne
 Partie inférieure : connexion du modèle concret du poste autonome au reste de la war room

Ce scénario n'est toutefois que théorique, puis qu'il s'avère problématique vis-à-vis de la reconnexion : comme pour l'exemple précédent, le rétablissement de la cohérence globale lorsque la liaison réseau est à nouveau activée, est sujet aux problèmes relatifs à la fusion de deux modèles de données dont les valeurs divergent.

4.2.8 Choix des données dans le cadre du multicanal

Comme l'illustre notre architecture logicielle, le choix des données définissant les objets du projet revêt une importance capitale pour la bonne marche de l'application. Cette tâche doit être réalisée avec précaution, puisque de nombreux facteurs influencent les décisions à prendre dans ce domaine.

Par exemple, le choix des données permettant la définition abstraite de l'objet constitue un point qui nécessite une attention particulière. En effet, selon notre approche, toute modification d'une information abstraite implique naturellement un impact sur les données

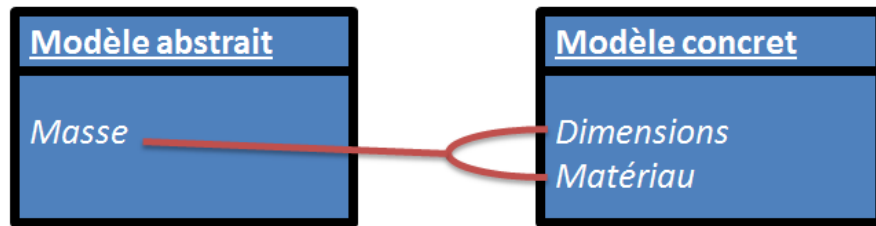


FIGURE 4.17 – Utilisation de deux données concrètes dans le même canal : le mécanisme de retranscription concret/abstrait peut prendre en compte les deux paramètres concrets.

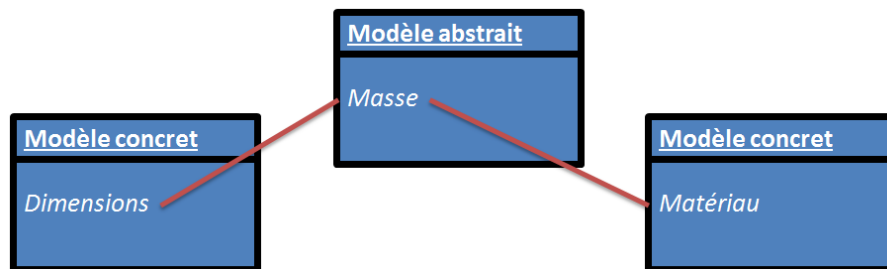


FIGURE 4.18 – Séparation des deux données concrètes en deux canaux distincts. En cas de modification du matériau, la masse de l'objet devrait être affectée, mais n'est pas calculable en raison de la méconnaissance du volume de l'objet. Ce cas de figure est invalide.

concrètes qui lui sont liées. Cependant, cette situation peut parfois s'avérer problématique.

Pour illustrer cette contrainte, prenons l'exemple simple d'un objet physique défini de façon abstraite par sa masse. Imaginons que cet objet soit intégré dans un environnement permettant d'interagir visuellement sur cette masse. Cette interaction peut par exemple avoir lieu à travers deux informations concrètes constituées respectivement :

- des dimensions de l'objet (son volume possède en effet un impact direct sur la masse)
- du matériau constituant l'objet (d'un matériau à l'autre, la masse volumique d'un objet est variable)

Si ces deux informations concrètes sont représentées à travers le même canal, aucun problème n'apparaîtra : les deux données concrètes seront stockées au sein du même modèle concret, et le mécanisme de retranscription pourra prendre en compte ces deux informations pour en traduire les conséquences sur la masse stockée par le modèle abstrait (fig. 4.17).

Imaginons au contraire que le concepteur de l'application désire introduire deux canaux distincts pour gérer ces deux données concrètes. La configuration instinctive qui serait proposée consisterait à définir l'application sous la forme de 3 composantes : le modèle abstrait prenant en charge la masse et deux modèles concrets, gérant respectivement les dimensions, et le matériau (fig. 4.18).

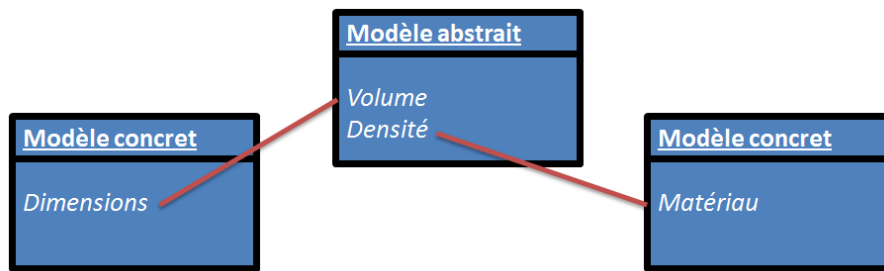


FIGURE 4.19 – Encodage de la masse sous la forme de deux informations distinctes : Volume et Densité.

Cette solution s'avère cependant invalide : si un utilisateur modifie le matériau de l'objet via le canal dédié, cette interaction devra avoir un impact sur la masse de l'objet. Or, le modèle abstrait recevant la notification de cette interaction n'aura pas connaissance des dimensions courantes de l'objet. Pour résoudre cette problématique, plusieurs solutions sont possibles :

- une première consisterait à supprimer l'information abstraite "Masse" pour la remplacer par deux autres données abstraites : le volume, et sa densité. Ainsi, l'information "Masse" existera toujours dans le modèle abstrait, mais sous la forme du couple "Volume" et "Densité" (fig. 4.19). Grâce à cette approche, si un utilisateur modifie le matériau de l'objet, l'impact au niveau de la définition abstraite aura lieu sur la propriété "Densité". Cela n'entraînera par conséquent pas de modification sur les dimensions de l'objet

- une seconde solution consiste à insérer un nouveau modèle concret servant d'intermédiaire entre les canaux et le modèle abstrait. Ce modèle intermédiaire est alors utilisé pour factoriser les deux informations concrètes au sein du même modèle, permettant ainsi une parfaite cohérence dans les données (fig. 4.20). Dans notre exemple, l'architecture se verrait constituée :

- d'un modèle abstrait, prenant en charge l'information "Masse".
- d'un modèle concret intermédiaire, fils direct du précédent, prenant en charge les deux informations concrètes "Matériau" et "Dimensions".
- des deux modèles concrets de plus bas niveau, gérant respectivement le matériau et les dimensions, et étant définis comme fils directs du modèle concret intermédiaire.

Grâce à cette solution, toute modification du matériau à travers le canal dédié entraînera une modification de l'information "Matériau" du modèle intermédiaire, et ce dernier pourra prendre en compte la valeur des dimensions pour retranscrire une modification cohérente de la donnée abstraite "Masse". Il est en outre intéressant de souligner que cette solution présente une répétition des mêmes données entre plusieurs composantes. En effet, cet exemple révèle que la hiérarchisation des modèles de notre architecture ne repose pas uniquement sur les niveaux d'abstractions nécessaires, mais peut également servir à structurer les informations concrètes entre elles pour assurer la conservation de la cohérence globale.

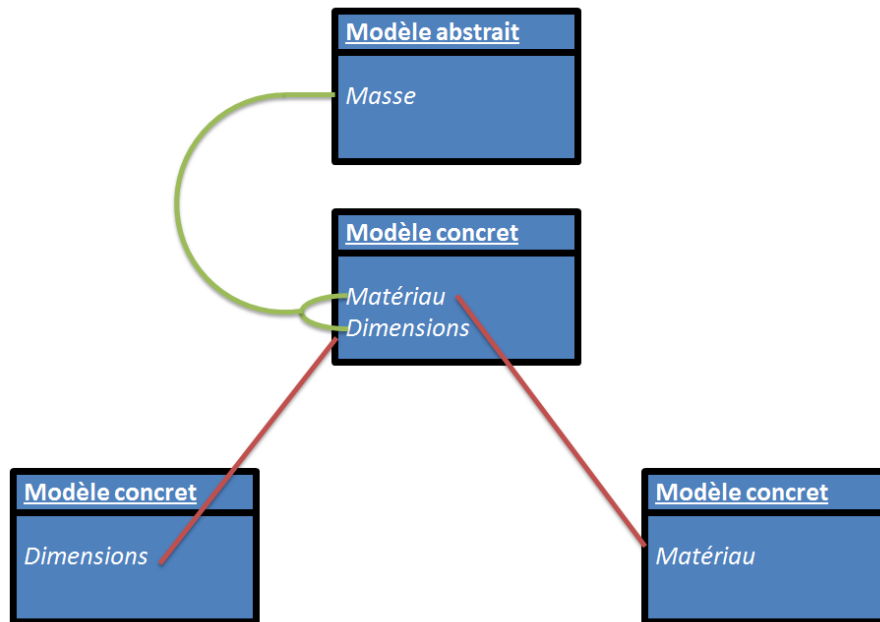


FIGURE 4.20 – Factorisation des données concrètes au sein d’un unique modèle concret. Cette solution permet de structurer correctement les mécanisme de retranscription concret/abstrait.

Dans le cadre d’une hiérarchisation complexe de plusieurs modèles concrets, le problème de définition des données s’avère donc plus délicat qu’une simple différenciation entre données visuelles et données non-visuelles. Il est important de définir précisément quel modèle prendra en charge quelle donnée, et dans quelle mesure cette donnée sera utilisable par les modèles de plus bas niveau d’abstraction. Cette contrainte est d’ailleurs caractérisée par le fait qu’il existe de multiples moyens possibles pour réaliser une même application. La structure interne choisie sera toutefois primordiale dans la capacité à faire évoluer l’application par l’ajout de nouveaux canaux. Il n’existe donc pas de solution universelle, puisque chaque application possède des besoins qui lui sont propres. Ces informations sont en effet dépendantes de plusieurs facteurs, tels que :

- le type de projet, c’est-à-dire le choix des objets qui prendront place dans l’environnement virtuel déployé.
- l’activité qui prend place autour de ce projet. S’agit-il d’une tâche d’assemblage ? De modélisation d’objets mécaniques ? D’agencement d’un plan ? D’analyse de données ? De planification d’un projet ? La nature de l’activité réalisée autour de ce projet possède un impact direct sur le panel d’interactions qui seront disponibles dans la war room. Ces interactions opérant sur les données de l’objet, il est important de définir un jeu de données pertinent.
- les canaux d’interaction qui prendront place dans la war room. Le choix de ces canaux dépend directement des domaines de compétences de l’équipe, mais aussi de l’installation matérielle disponible. La question à se poser, ici, est de savoir quelles seront les représentations visuelles nécessaires dans la war room, et sur quelles informations relatives à l’objet

elles reposent. Si ces canaux sont multiples, quels sont les interactions qu'ils proposent, et sur quel(s) objet(s) ? Quel impact ces interactions doivent-elles avoir sur les représentations des autres canaux ? Deux canaux possèdent-ils des caractéristiques communes, qui permettraient de factoriser certaines données sous la forme d'un modèle concret commun et de les structurer sous la forme d'un arbre hiérarchique ? L'un des canaux proposés constitue-t-il une forme dérivée d'un autre canal ? Dans quelle mesure les données associées à un canal impacteront la définition abstraite de l'objet ?

- le déroulement de l'activité dans la war room. Dans quelle mesure l'activité impliquera-t-elle de pouvoir déplacer visuellement les objets ? Cette activité implique-t-elle la nécessité de disposer d'affichages simultanés du même objet ? Sera-t-il nécessaire de permettre aux utilisateurs d'interagir à distance ? Comment les participants sont-ils susceptibles d'exploiter les dispositifs proposés ?

Cette liste non-exhaustive illustre bien qu'il est difficile de définir une stratégie universelle permettant de définir avec précision la nature des données à manipuler, ainsi que la place qu'elles doivent prendre dans les différents degrés d'abstraction. Le choix de cette stratégie dépend d'ailleurs de choix personnels propres au concepteur de l'application collaborative. En effet, pour un cas d'utilisation et une application donnée, il n'existe pas une unique stratégie de distribution des données. Chaque cas d'usage peut voir ses besoins comblés par des applications similaires en apparence, mais basées sur une structure interne différente.

Dès lors, la démarche qui vise à définir la structuration des données dans l'application repose sur l'utilisateur de la plateforme et sur la manière dont il conceptualise les canaux de son application. Par exemple, si ce concepteur décide que l'un des canaux qu'il veut mettre en place sera susceptible d'être enrichi d'un canal dérivé aux caractéristiques proches, il prendra soin de faire en sorte que ce canal d'origine repose sur un modèle concret suffisamment complet pour répondre aux besoins du nouveau canal. De même, pour rendre un poste de travail capable de travailler en mode "hors ligne", c'est-à-dire dégagé de toute communication avec les autres canaux, il est nécessaire de confier à ce poste de travail la charge d'un modèle concret autonome en terme de données pour répondre à tous les besoins du canal en question, même si cela engendre des doublons dans la hiérarchie de données. D'une manière générale, le contenu de chaque modèle d'un objet doit être pensé en fonction des canaux qui viendront s'y greffer.

Dans la pratique, la définition de notre architecture comme étant une superposition de différents degrés d'abstraction constitue un abus de langage. En effet, si cette structure logicielle permet effectivement de hiérarchiser les informations d'un objet en fonction de leur niveau d'abstraction, elle peut en outre permettre de structurer des informations concrètes entre elles. Il est en effet possible, comme l'a illustré notre exemple précédent, qu'un modèle stockant des informations purement visuelles soit le père d'un ou plusieurs autres modèles concrets reprenant des données similaires.

La conceptualisation de l'objet virtuel d'un point de vue sémantique constitue également un facteur pouvant influencer le choix des données. En effet, le concepteur d'une application utilisant notre plateforme peut aborder la définition de l'objet de deux façons :

- D'une part, il est possible de considérer que l'objet sémantique est défini uniquement à

travers les données prises en charge par le modèle abstrait. Cette approche nécessite par conséquent d'adopter un degré d'abstraction très élevé et surtout suffisamment complet pour pouvoir reconstruire toutes les représentations possibles de l'objet à partir de ces seules données abstraites. Dans ces conditions, l'intégralité des données graphiques utilisées par les vues hétérogènes de la war room peuvent être déduites à partir des informations du modèle abstrait. L'intérêt de cette stratégie réside dans la centralisation de la définition globale de l'objet en une seule et même composante logicielle. Il est ainsi envisageable d'exporter les informations du modèle abstrait directement dans le pipeline de production, grâce à des mécanismes capables de reconstruire l'intégralité des documents électroniques nécessaires à cette phase de production. De même, la centralisation de tous les concepts au niveau du modèle abstrait permet une totale liberté vis-à-vis de l'ajout de nouveaux canaux (quels que soient les modes de représentation utilisés par ces canaux, la généralité du modèle abstrait permettra toujours de reconstruire les données concrètes nécessaires). Cependant, cette stratégie s'avère souvent complexe, voire impossible à mettre en oeuvre car elle implique de définir des données abstraites qui ont une portée complètement générale. Or, la diversité des canaux implique souvent l'existence de données annexes, qui ne sont pas systématiquement liées au modèle abstrait. Dans ces conditions, il est considéré comme acceptable que le modèle abstrait prenne en charge également des données concrètes, à condition qu'elles fassent partie intégrante de sa définition sémantique.

- La seconde approche consiste à considérer l'objet comme étant défini par l'intégralité des données qui le constituent, quels que soient les modèles qui les prennent en charge. Cette approche définit alors l'objet sémantique à travers les données stockées par son modèle abstrait, mais également par des données annexes stockées plus bas dans la hiérarchie de modèles. Selon cette conception, l'application ne repose pas sur l'absolue nécessité de centraliser tous les paramètres distinctifs de l'objet au sein de la même composante "modèle abstrait". Les modèles concrets qui s'y greffent peuvent en effet avoir pour fonction d'enrichir les données fondamentales de l'objet. L'inconvénient de cette approche réside dans le fait que toute évolution de l'application via l'ajout de nouveaux canaux sera dépendante de la structuration déjà présente.

Pour illustrer plus clairement ces deux approches, reprenons l'exemple d'une application permettant de travailler sur la conception d'un objet physique, défini par son coût de fabrication. Le modèle abstrait de cet objet prendra donc en charge l'information "Coût". Imaginons maintenant que dans cette application, l'utilisateur décide d'intégrer un canal de modélisation 3D permettant d'éditer la forme de cet objet via la modification d'un mesh 3D. Il est en effet envisageable que la complexité de la géométrie de l'objet, aie un impact direct sur son coût.

Étant donné que la connaissance du coût de l'objet est insuffisante pour supporter l'affichage 3D de l'objet par une vue, ce nouveau canal entraînera donc naturellement la création d'un modèle concret dédié prenant en charge la définition du mesh 3D de l'objet (par exemple, une liste de polygones). Cette information "mesh 3D" est une information purement visuelle, mais peut également être considérée comme une des données déterminantes pour définir l'objet de façon abstraite. En effet, le coût de l'objet ne suffit pas à la reconstruction d'un modèle 3D de celui-ci. Le modèle concret contient par conséquent des informations qui n'apparaissent pas dans les degrés d'abstraction supérieurs.

La question à laquelle le concepteur doit alors répondre dans ce contexte est "la donnée représentant la forme 3D de l'objet doit-elle également apparaître dans le modèle abstrait ?"

- Si la définition abstraite de cet objet repose uniquement sur sa masse, le mesh 3D ne constitue qu'une donnée annexe au projet. La manipulation de ce mesh 3D n'est alors considérée que comme un moyen d'impacter le coût de l'objet. Dans ces conditions, il n'est pas nécessaire d'inclure la donnée "mesh 3D" au sein du modèle abstrait.

Cependant, cette solution comporte un risque : si le concepteur de l'application décide à posteriori d'ajouter un nouveau canal reposant sur ce mesh 3D, les composantes logicielles propres à ce canal seront obligatoirement placées en tant que filles du modèle concret.

- Si au contraire, la forme 3D de l'objet constitue un paramètre critique pour définir l'objet final, alors les données relatives au mesh 3D trouvent naturellement leur place dans le modèle abstrait. En dépit du caractère très visuel de ces données polygonales, elles sont en effet considérées comme appartenant à la définition abstraite de l'objet.

Comme nous pouvons le constater ici, la conceptualisation de l'objet virtuel est déterminante pour définir le placement des données entre les modèles.

MISE EN OEUVRE DE LA PLATEFORME EN SITUATION RÉELLE

Sommaire

5.1	Mise en oeuvre de la distribution des composantes logicielles .	146
5.1.1	Utilisation de la DAI de SPIN 3D	146
5.1.1.1	Communication inter-composantes : le système de proxy	147
5.1.2	Déploiement et configuration XML	157
5.2	Démonstrateur de distribution logicielle	158
5.2.1	Scénario et fonctionnalités	159
5.2.2	Implémentation : les composantes impliquées	164
5.2.3	Résultats	167
5.2.4	Perspectives	168
5.3	Mise en oeuvre de l'abstraction $M_N V-O$	169
5.4	Mise en application de la plateforme dans un projet d'urbanisme	173
5.4.1	L'urbanisme : contexte de notre étude	173
5.4.2	L'urbanisme : un domaine complexe, propice à l'utilisation de la war room	174
5.4.3	Observation d'un projet concret	176
5.4.3.1	Phase de production	176
5.4.3.2	La phase de maquettage informatique	179
5.4.4	L'installation matérielle mise en oeuvre	179
5.4.5	L'environnement virtuel partagé	180
5.4.5.1	Le canal 2D	181
5.4.5.2	Le canal 3D global	183
5.4.5.3	Le canal de modélisation 3D	186
5.4.6	Distribution de la plateforme	187
5.4.6.1	Caractéristiques générales	187
5.4.6.2	Le déploiement des composantes et la configuration XML	188
5.4.6.3	Les données et leur signification	192
5.4.7	Bilan	194

Ce dernier chapitre propose une mise en application des concepts que nous avons présentés jusqu'à maintenant. L'objectif des sections suivantes est de décrire la manière dont nous avons implémenté notre proposition au sein de la plateforme SPIN|3D. Cette implémentation a fait l'objet de deux prototypes visant à valider notre plateforme en la confrontant à deux scénarios d'utilisation distincts. La première partie mettra l'accent sur la distribution des composantes et son utilisation dans le cadre d'une application de conception automobile. Nous pourrons ensuite développer l'aspect "abstraction" de notre modèle à travers la présentation d'une application de conception urbaine.

5.1 Mise en oeuvre de la distribution des composantes logicielles

La première étape visant à illustrer la mise en situation de notre plateforme collaborative dans le cadre d'une réelle activité met l'accent sur la distribution logicielle. En effet, le premier prototype que nous présentons ici est constitué d'un environnement mixte 2D et 3D réparti sur plusieurs dispositifs, et à travers lequel les utilisateurs peuvent collaborer sur une tâche de manipulation d'objets virtuels (en l'occurrence, une voiture et ses différentes pièces).

Le concept illustré dans cet section repose essentiellement sur la constitution des différents agents logiciels MVO, leur distribution entre les différents dispositifs, et les mécanismes informatiques qui sont mis en oeuvre pour assurer la communication inter-composantes et donc la cohérence globale de l'environnement virtuel.

Comme nous l'avons souligné dans le chapitre précédent, une plateforme de collaboration distante telle que SPIN|3D fournit des fonctionnalités de base qui peuvent constituer un support de départ fiable pour la construction de notre plateforme d'interaction collaborative colocalisée. Notre méthode de travail consiste donc à implémenter les fonctionnalités manquantes aux plateforme d'interaction distante et à les greffer à une solution existante. Cette implémentation constituera donc une extension de la plateforme SPIN|3D écrite en C++ et utilisant CORBA pour les communications réseau. L'un des objectifs de cette approche est d'apporter des modifications minimales à SPIN|3D de façon à conserver dans la mesure du possible ses facultés de collaboration distante.

5.1.1 Utilisation de la DAI de SPIN|3D

La plateforme SPIN|3D est équipée d'un mécanisme de communication externe pour communiquer par le biais d'événements et d'informations diverses avec une application tierce. Cette interface de communication permettant également de communiquer via le réseau, elle peut par conséquent trouver naturellement sa place dans le cadre de la collaboration colocalisée multi-dispositifs. En effet, nous pouvons nous réapproprier ce mécanisme de communication de façon à intégrer à la war room des dispositifs sur lesquels s'exécutent des applications interactives très spécialisées, et qui peuvent être utilisées par les participants pour interagir sur le projet partagé dans l'environnement virtuel de la war room.

En terme d'implémentation, le système de communication entre la plateforme logicielle et une application externe est constitué d'un service réseau dédié appelé DAI (Distributed Authoring Interface) [PIC03]. Cette interface de communication reprend les concepts

de l'EAI (External Authoring Interface) apportés par la norme VRML97. Elle repose sur la norme CORBA qui offre en effet une infrastructure générique permettant aux applications de communiquer avec les composantes de notre plateforme de façon transparente. Les échanges entre les deux entités logicielles sont par conséquent formulés sous la forme d'appels de méthodes classiques. De plus, la norme CORBA nous offre également un support multi-plate-formes et multi-langages, pour le développement des applications externes communicantes. Le langage de description IDL (Interactive Data Language) permet au développeur de définir les différents services proposés, et d'en générer des souches utilisables dans divers langages de programmation.

Le référencement des objets (au sens « programmation » du terme) distribués sur le réseau est pris en charge automatiquement par l'interface CORBA. À la connexion, l'entité cliente établit ainsi une référence (IOR, selon la terminologie CORBA) vers l'objet partagé, et peut procéder à des invocations de méthodes sur cet objet de façon transparente, c'est à dire de la même manière que s'il s'agissait d'un objet instancié localement dans l'espace mémoire.

Au même titre que pour la communication entre la composante modèle et ses vues, une relation de type observateur/observable est mise en place. Ainsi, l'application externe est capable de placer des observateurs sur les champs du modèle, en vue d'être notifiée de toute modification de leur valeur. À la création de cette relation d'observation, l'application externe fournit à la plateforme une référence vers l'observateur qui se voit enregistré auprès de l'objet observé. À chaque modification d'un champ observé, la plateforme notifiera alors tous les observateurs enregistrés, les invitant ainsi à mettre à jour leurs données locales. La tâche de notification passe, elle aussi, par une invocation de méthode à distance sur l'observateur.

Dans le cadre de notre plateforme, l'utilisation de cette interface CORBA a lieu pour étendre le niveau de spécialisation des terminaux. En effet, il nous est possible d'intégrer à l'activité les environnements logiciels que les utilisateurs ont l'habitude d'utiliser lors de leur travail individuel. Par exemple, lorsqu'une équipe travaillant sur la conception d'une pièce mécanique comprend un technicien spécialisé dans la modélisation 3D, il devient possible grâce à la DAI de procurer à cet utilisateur un poste de travail exécutant l'application dont il est coutumier, et sur laquelle il est le plus productif. Cette particularité permet ainsi aux utilisateurs de s'affranchir de la nécessité d'apprendre à réaliser leurs tâches via une nouvelle application.

Pour bénéficier de cette fonctionnalité, il est bien sûr nécessaire d'ajouter à l'application tierce le module logiciel DAI qui lui permettra de se connecter au bus CORBA et d'échanger des données avec SPIN|3D. L'intégration de ce module peut se faire directement dans le code source de l'application mais il est également possible, dans le cas d'applications propriétaires, d'intégrer l'interface DAI dans un plugin pour cette application (sous réserve que cette fonctionnalité soit proposée).

5.1.1.1 Communication inter-composantes : le système de proxy

Dans un mode d'utilisation mono-poste, ou dans le cadre d'une collaboration distante impliquant une duplication et une synchronisation de toutes les composantes entre les postes

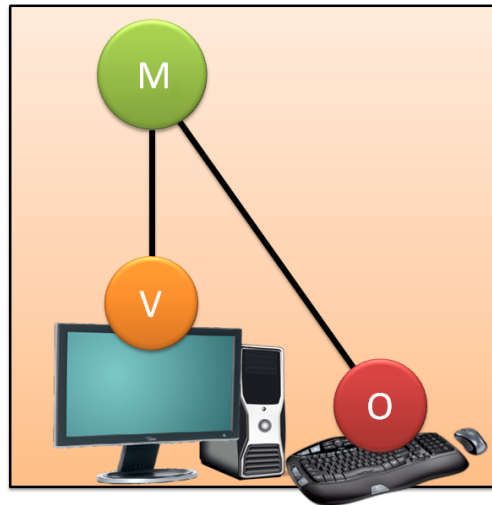


FIGURE 5.1 – Illustration des relations entre les composantes dans la plateforme SPIN|3D. Les liens entre ces composantes sont directs.

de travail, les échanges qui ont lieu entre les composantes modèle, vue et outil sont directs (fig. 5.1). En effet, dans ce type de situation, l'ensemble des composantes constituant un même agent interactif est localisée sur le même poste. La transmission d'événements entre ces composantes passe par conséquent par un simple mécanisme d'invocation de méthodes.

Dans le cadre de notre war room, les composantes sont au contraire distribuées entre les différents ordinateurs du réseau local. Il importe par conséquent de mettre en place un mécanisme de communication inter-composantes capable de transiter sur le réseau sans perturber le fonctionnement normal des échanges.

Pour conserver le caractère simple de ces liens directs et éviter de mettre en péril les facultés de collaboration distante de SPIN|3D, nous avons décidé de baser les mécanismes de communication inter-composantes sur l'utilisation d'un BUS CORBA. Ainsi, au même titre qu'une application externe connectée à la war room via l'utilisation de la DAI, les échanges qui ont lieu entre deux composantes situées sur des ordinateurs différents restent basés sur un simple mécanisme d'invocation de méthode. Cette invocation à distance intervient en remplacement des invocations locales.

La distribution d'un agent pouvant avoir lieu au niveau de n'importe quelles composantes, il est nécessaire, pour assurer une structuration simple de l'application, de conserver une certaine transparence dans la nature des échanges inter-composantes. En ce sens, le mécanisme de communication ne doit pas être une caractéristique notable de l'agent, mais au contraire s'intégrer à celui ci de la même manière que pour un objet purement local. Pour ce faire, nous nous sommes tournés vers la création d'un système de proxy. Ce procédé consiste alors en deux temps :

- Doter les composantes de base des processus de partage sur le bus CORBA, de façon à les rendre disponibles à distance pour toute invocation de méthode.

- Ajouter à l'implémentation de chaque type de composante l'intégration d'une nouvelle classe dérivée, qui constituera l'objet proxy dédié à la communication. Cette classe dérivée possède les mêmes caractéristique que la composante de base, à la différence que ses méthodes liées aux échanges inter-composantes sont implémentées de façon à transmettre directement les informations à leur homologue distant. Ces composantes proxy sont, dans notre application, qualifiée de "pseudo-composantes".

Pour clarifier la démarche d'implémentation que nous avons suivi, il importe de préciser certaines caractéristiques logicielles de SPIN|3D. Une composante MVO, quel que soit son type, implémente deux interfaces relatives à des fonctionnalités différentes :

- **une interface de service** qui définit les méthodes relatives à la mécanique interne de la composante. Ces fonctions, utiles au fonctionnement de la plateforme SPIN|3D permettent d'assurer les traitements logiques au sein de l'application. L'invocation de ces méthodes doit par conséquent rester locale, y compris dans le cadre d'une distribution logicielle telle qu'elle apparait dans la war room. Par exemple, lorsque la composante modèle d'un agent est déployée sur un autre ordinateur, les méthodes de l'interface de service de ce modèle doivent tout de même être appelées localement sur chacun des postes de travail.
- **une interface de dialogue** qui est dédiée au processus d'échange établi entre les composantes (c'est par exemple le cas de la notification de la composante vue par son modèle). Les méthodes définies dans l'interface de dialogue constituent les liens entre les différentes composantes d'un agent. C'est donc sur ces méthodes que doit être appliqué le système de communication réseau, de façon à assurer le bon fonctionnement de la distribution logicielle entre les postes de travail de la war room.

Dans la plateforme distribuée de la war room, le processus de dialogue qui est établi entre deux composantes peut être définir par deux cas de figure. Ce dialogue repose sur un lien direct lorsque les deux composantes sont situées sur le même poste de travail, et sur un lien réseau dans le cas de deux composantes distribuées sur des postes distincts. La solution consiste donc à définir chaque composante sous la forme d'une classe polymorphe qui sera soit purement locale, soit "proxy", c'est à dire n'exécutant en local que l'interface de service et qui redéfinit l'interface de dialogue de façon à ce qu'elle puisse passer par le réseau. La figure 5.2 illustre la nuance entre ces deux formes. Le poste 2 instancie un modèle local. C'est cette composante qui sera le modèle réel de l'objet virtuel. Le poste 1 doit communiquer avec cette composante par l'intermédiaire d'un modèle "proxy", qui procédera aux invocations de méthodes sur le modèle réel distant.

Dans la mesure où les communications réseau transitent sur le BUS CORBA, il est nécessaire de définir en langage IDL la composition des objets partagés et les méthodes qui pourront être invoquées à distance. La compilation IDL entraîne ainsi la génération de deux souches distinctes : le "Skeleton", qui correspond à l'objet partagé sur le bus, et le "Stub" qui fournit les ressources nécessaires à l'invocation distante des méthodes.

Dans notre cas de figure, une composante MVO locale doit implémenter l'interface "Skeleton" pour être accessible par les autres postes de travail, tandis que la composante proxy instanciée sur ces autres ordinateurs implémente l'interface "Stub".

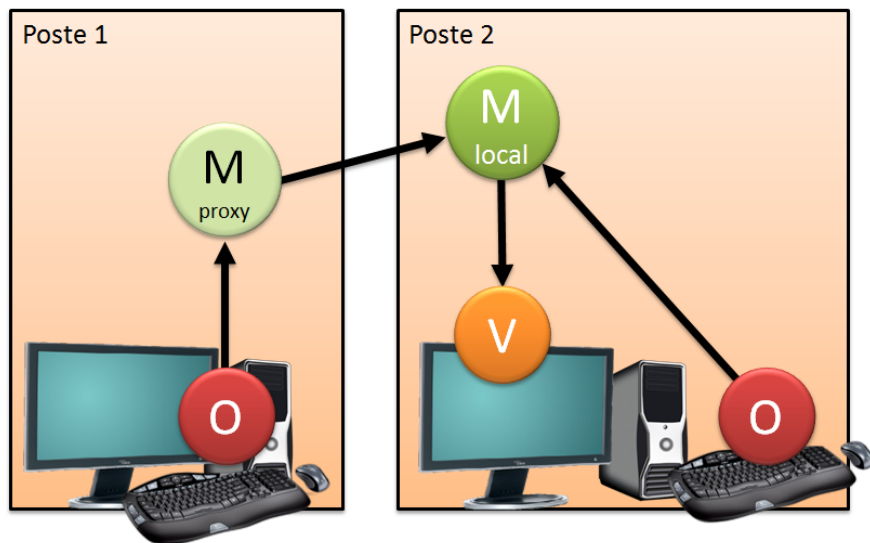


FIGURE 5.2 – Illustration des deux types de composants : à gauche, un modèle "proxy", et à droite un modèle "local".

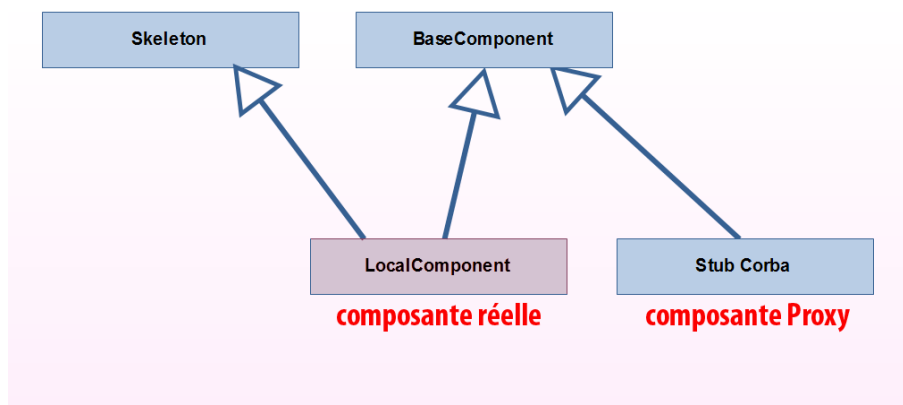


FIGURE 5.3 – Première proposition d'implémentation des composants polymorphes.

Une première stratégie d'implémentation est présentée sur le diagramme de la figure 5.3. Cette approche consiste à définir une classe de base commune aux deux types de composants et qui implémente les méthodes de service. De cette classe dérivent deux autres classes :

- une classe "locale", qui est équivalente à la classe de base à la différence que celle-ci implémente en sus des méthodes de dialogues pouvant être invoquées à distance. Pour activer cette fonctionnalité, la classe locale hérite donc du Skeleton généré par la compilation IDL. Cette classe locale est équivalente à la composante "classique" de SPIN|3D.
- une classe "proxy" qui elle, redéfinit les méthodes de dialogue de façon à les transmettre à la composante "réelle". Dans le cadre de CORBA, cette classe est similaire au Stub généré par la compilation IDL.

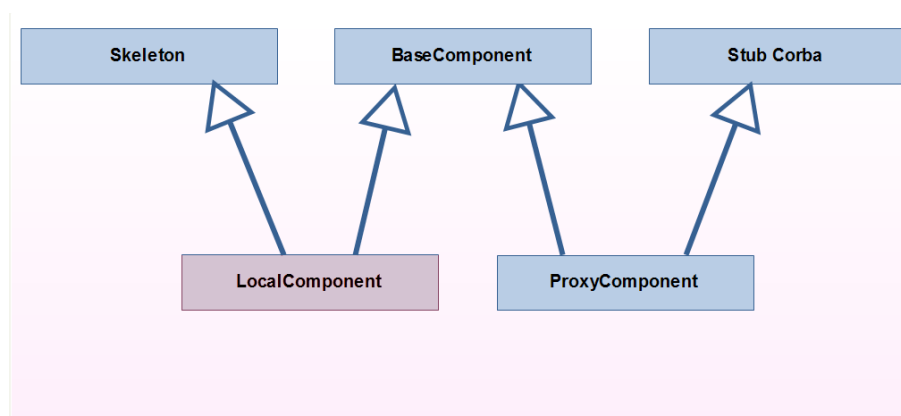


FIGURE 5.4 – Seconde proposition d’implémentation des composantes polymorphes.

Cette méthode est cependant confrontée à une contrainte technique de la plateforme : le compilateur IDL de SPIN|3D, qui repose sur la bibliothèque TAO, ne permet pas aux objets définis dans la classe IDL d’hériter d’une interface C++. Une réponse à cette problématique pourrait être de dégager le Stub de cette hiérarchie, et de proposer un héritage multiple du proxy similaire à celui qui s’applique à classe locale (fig. 5.4). Cette seconde méthode, bien que valide, implique cependant de définir une classe de base pour la composante, et donc de modifier légèrement la structure originale de SPIN|3D. Pour limiter cet impact, nous avons donc décidé de conserver la composante classique de SPIN|3D et de lui ajouter simplement une filiation avec le Skeleton de CORBA. La classe proxy est alors dérivée de cette classe locale, et redéfinit uniquement les méthodes de dialogue (fig. 5.5).

Il est important de souligner que la composante "Outil" du MVO n’est pas soumise à ces problématiques d’implémentation : l’Outil est en effet toujours exécuté localement sur un poste de travail, et ne nécessite donc pas d’implémenter les interfaces générées par CORBA. Un outil peut cependant disposer d’une représentation visuelle distante telle qu’un télépointeur, et agir sur une composante modèle distante par l’intermédiaire de son proxy.

Le choix de ce mécanisme basé sur l’utilisation de proxy nous permet notamment d’envisager l’implémentation d’un cas d’utilisation utilisant un transfert réel de données lors du déplacement d’une vue. En effet, la classe proxy héritant directement de la composante originale elle en possède toutes les caractéristiques vis-à-vis du reste de l’application. Il est par conséquent possible d’inclure à la plateforme un système permettant, lors d’une demande de déplacement de la part d’un utilisateur, de désactiver la redirection opérée par cette classe proxy de façon à la remplacer par les algorithmes d’affichage que doit exécuter la « vraie » vue. Ce système serait plus dynamique vis-à-vis de la distribution, mais n’apporterait cependant que peu d’avantages, si ce n’est la possibilité de déplacer le modèle dynamiquement entre les postes de travail.

Pour entrer plus en détail dans l’implémentation des mécanismes de communication par proxy, nous pouvons dire que les modifications apportées à la plateforme SPIN|3D reposent essentiellement sur l’insertion de nouvelles classes dérivées qui implémentent ces composantes proxy. Pour apporter aux classes la faculté de communiquer sur le bus CORBA,

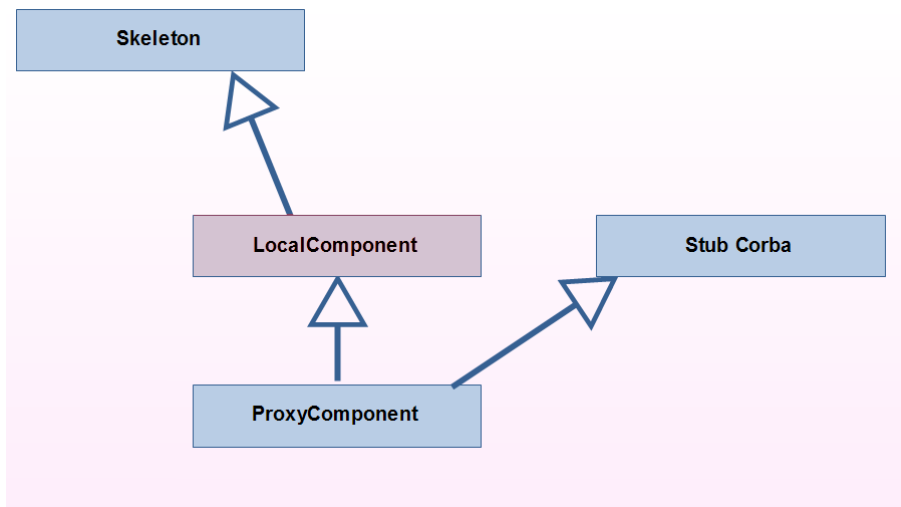


FIGURE 5.5 – Troisième proposition d'implémentation des composants polymorphes.

plusieurs interfaces ont d'abord été définies via le langage IDL. Sur le code IDL de la figure 5.6, nous pouvons distinguer trois interfaces distinctes.

- **WidgetView_corba.** Cette interface permet de définir les méthodes et fonctions de la composante vue qui seront accessibles sur le réseau. Il s'agit de `setDistantInteractionObject` (qui permet au système de définir sur quel objet l'outil va interagir en fonction de la position du pointeur souris) et de `remoteNotify`. Cette méthode, issue du comportement "observateur" du design pattern "observer/observable", est appelée par le modèle pour notifier la vue qu'une donnée de ce modèle a fait l'objet d'une modification. Nous pouvons d'ailleurs noter, dans ce fichier IDL, la définition d'une structure `EventFieldChanged_serialized`. Cette structure constitue une chaîne binaire encodant le contenu de l'événement généré par cette interaction. L'utilisation d'une telle chaîne binaire n'est pas toujours nécessaire lors de l'utilisation de CORBA. Il aurait en effet été possible de définir directement l'objet "Event" en IDL. Cette méthode implique cependant de définir également tous les types composés de l'objet, et ce, de façon récursive jusqu'à atteindre la définition des types de base (entier, flottant, chaîne de caractère). L'objet "Event" qui doit être transmis dans notre cas étant d'une importante complexité (puisque composé de plusieurs sous objets, eux-même à l'origine d'une composition), nous avons fait le choix de passer par une sérialisation pour le transfert de celui-ci.
- **WidgetModel_corba.** Cette interface définit les méthodes partagées de la composante "Modèle". Ces méthodes sont toutes les trois liées à l'interaction par l'utilisateur. Nous détaillerons leur comportement respectif dans l'analyse du diagramme de classe ci-dessous.
- **UserView_corba.** Cette interface est uniquement liée à l'affichage du pointeur de souris d'un utilisateur distant (ie. situé sur un autre poste de la war room).

La définition des interfaces dans le fichier IDL permet à CORBA de générer des classes virtuelles qui une fois étendues, fournissent tous les mécanismes nécessaires à la communi-


```

module MVC_Corba
{
    struct EventFieldChange_serialized
    {
        string efcFieldName;
        unsigned long dataType;
        sequence<octet> EventSerialized;
    };

    interface WidgetView_corba
    {
        boolean setDistantInteractionObject(in float x, in float y, in float z, in boolean propagate);
        oneway void remoteNotify(in EventFieldChange_serialized e);
    };

    interface WidgetModel_corba
    {
        oneway void remoteProcessEvent(in EventFieldChange_serialized e);
        oneway void remoteSetState(in char stateAndBool);
        oneway void remoteSensorProcess(in float rx, in float ry, in float rz, in boolean raw, in float tx, in float ty, in float tz);
    };

    interface UserView_corba
    {
        oneway void remoteNotify(in EventFieldChange_serialized e);
    };
};

```

FIGURE 5.6 – Définition en IDL des interfaces partagées sur le bus CORBA

cation réseau entre une classe cliente et une classe serveur.

La figure 5.7 illustre l'implémentation de la composante modèle d'un objet de l'environnement virtuel. Dans SPIN|3D, toute composante modèle est de type `WidgetModel`. Cette classe fournit tous les mécanismes de base nécessaires au fonctionnement de l'application dans un contexte mono-poste (ou purement distant, avec duplication des composantes sur chaque site). Le modèle d'un objet dans la war room hérite donc naturellement de cette classe : il s'agit de la classe `InteractiveObjectModel`. Le choix d'un héritage plutôt qu'une composition s'explique simplement par le fait que cette nouvelle classe doit se greffer le plus naturellement possible à la plateforme SPIN|3D originale, en impactant au minimum sa structure de base.

De part l'aspect distribué de notre plateforme, ce modèle doit également être accessible sur le réseau. La classe `InteractiveObjectModel` hérite donc également de la classe virtuelle générée par le compilateur IDL, en l'occurrence `WidgetModel_corba`. Cette classe fournit à la fois les souches serveur (skeleton) et client (stub), puisqu'en fonction du type de modèle (réel ou proxy) le comportement de l'objet sur le bus est différent.

La classe `InteractiveObjectModel` constitue donc le modèle réel de l'objet, c'est à dire celui qui, présent sous la forme d'une seule instance dans toute la war room, prendra en charge les données inhérentes à l'objet virtuel. Les méthodes dont le nom est préfixé de "remote", quant à elles, sont issues de la classe virtuelle générée par le compilateur IDL. Leur rôle est de réceptionner les messages reçus par les modèles "proxy", et de les injecter à distance dans les méthodes du modèle réel, comme si ces messages avaient été transmis directement (fig. 5.9).

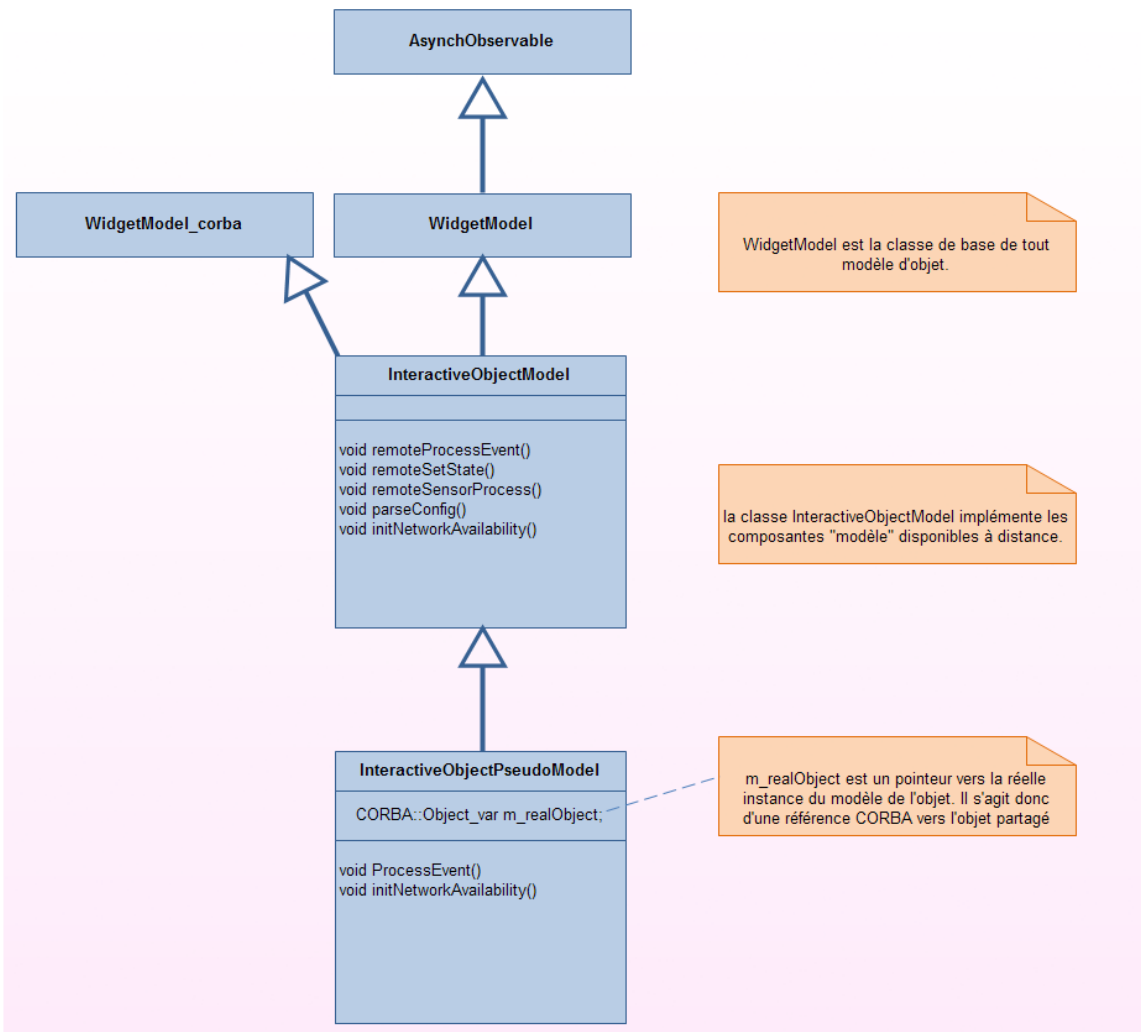


FIGURE 5.7 – Diagramme de classes présentant l'implémentation du système de proxy intégré à la composante modèle

La seconde classe importante qui apparaît sur le diagramme est la classe `InteractiveObjectPseudoModel` : celle-ci hérite directement de `InteractiveObjectModel`. Cette classe constitue la version "proxy" de la composante modèle. La méthode `ProcessEvent` est ici redéfinie : lorsqu'elle est appelée suite à une interaction sur l'objet virtuel, elle transmet cette interaction sous forme d'évènement au modèle réel, via l'appel à distance de la méthode `remoteProcessEvent`.

La figure 5.8 illustre quant à elle l'implémentation des composantes vue. La structure de cette implémentation est très similaire à celle qui a été mise en place pour les modèles (le fonctionnement de la distribution pour ces composantes est en effet le même). Nous pouvons toutefois souligner que la classe proxy de la vue redéfinit la méthode `display` de façon à la rendre inactive : lorsque la vue est instanciée sous cette forme, il ne s'agit en effet que d'une composante servant de relai aux échanges inter-composantes. Elle ne peut donc pas légitimement avoir un impact visuel sur le poste de travail sur lequel elle est instanciée.

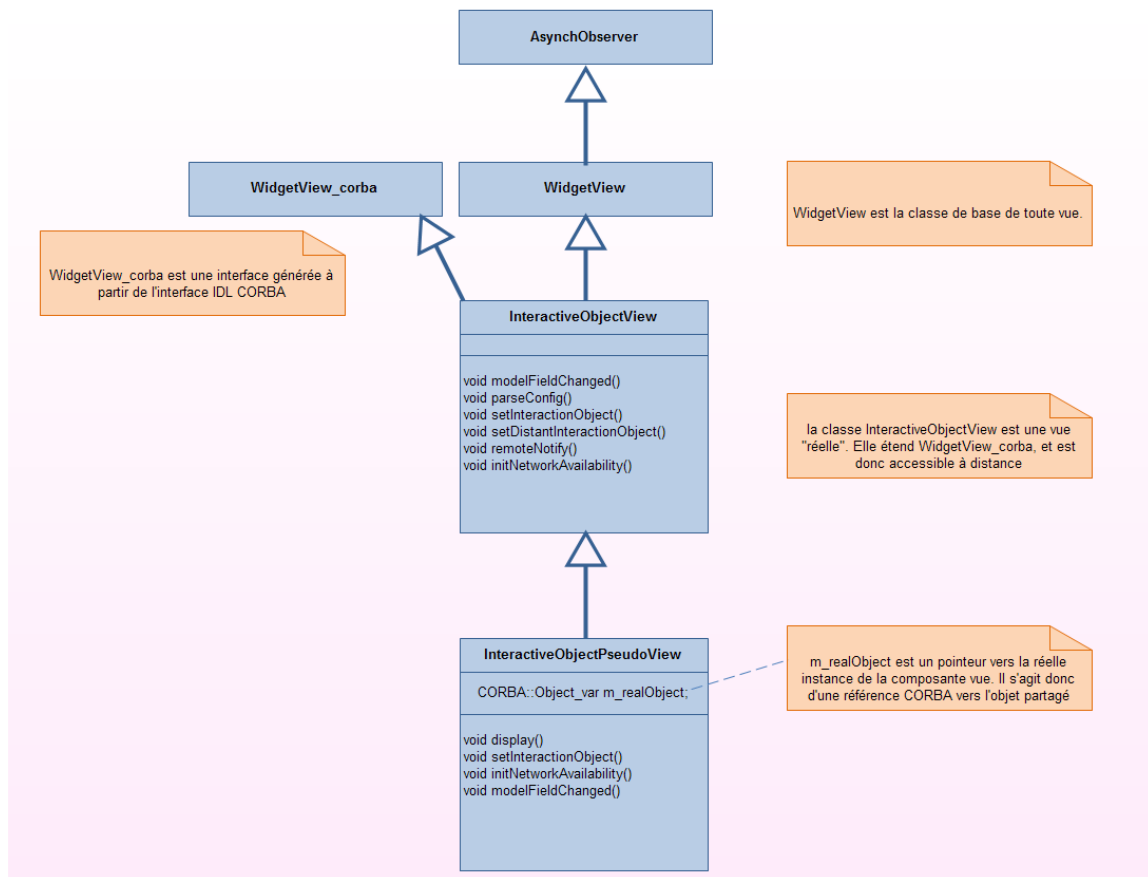


FIGURE 5.8 – Diagramme de classes présentant l'implémentation du système de proxy intégré à la composante vue

Pour illustrer le fonctionnement concret de cette structure logicielle, nous pouvons observer la figure 5.9. Cette séquence présente les échanges qui ont lieu entre les différentes classes lors d'une interaction dans notre démonstrateur : en l'occurrence, il s'agit de deux postes de travail, l'un prenant en charge le modèle de l'objet, tandis que le second prend en charge la composante vue. L'outil manipulé par l'utilisateur fournit une rotation qui doit être appliquée à l'objet présenté par la machine distante. Sur cette séquence, nous pouvons constater qu'en premier lieu le modèle est notifié des événements de rotation qui lui sont appliqués. Pour ce faire, le premier poste de travail va appeler la méthode `processEvent` sur son modèle local, en l'occurrence un proxy. L'évènement est alors sérialisé par le modèle proxy, puis transmis par le bus corba jusqu'au modèle réel grâce à la méthode `remoteProcessEvent`. Lors de la réception de l'évènement, le modèle réel reconstituera l'objet `Event`, et traduira son impact directement sur les données de l'objet (`processEvent`). Dès lors, la relation de type observateur/observable qui est établie entre le modèle et la vue va entraîner une notification générale de toutes les vues du modèle (`notify`). Cette action constituera alors une mise à jour de la vue locale (une vue proxy) par l'intermédiaire de `modelFieldChanged`. Ce dernier proxy répercutera alors la notification vers la vue réelle, via l'appel distant de la méthode `remoteNotify`.

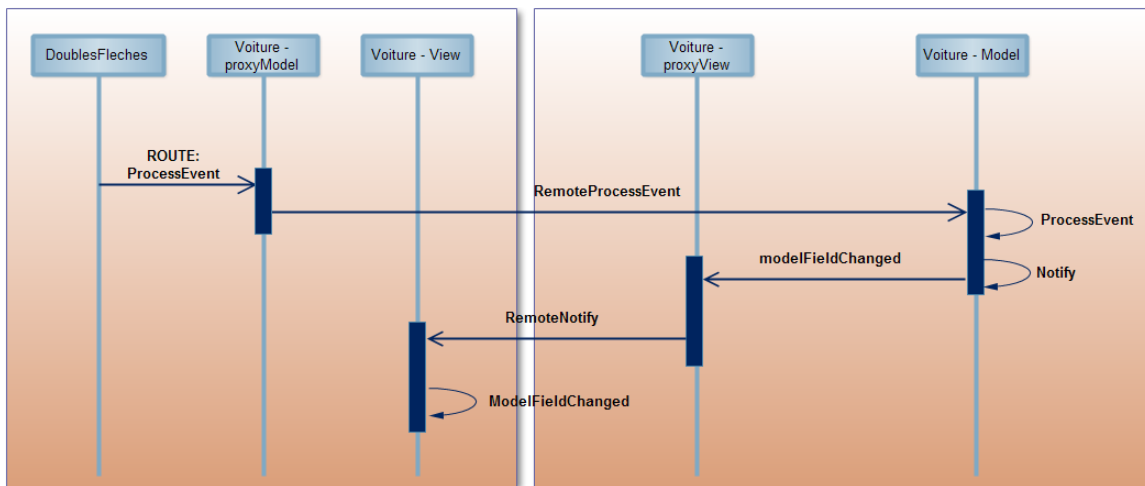


FIGURE 5.9 – Diagramme de sequence d’une interaction visant à opérer une rotation sur l’objet, par l’intermédiaire d’un item visuel

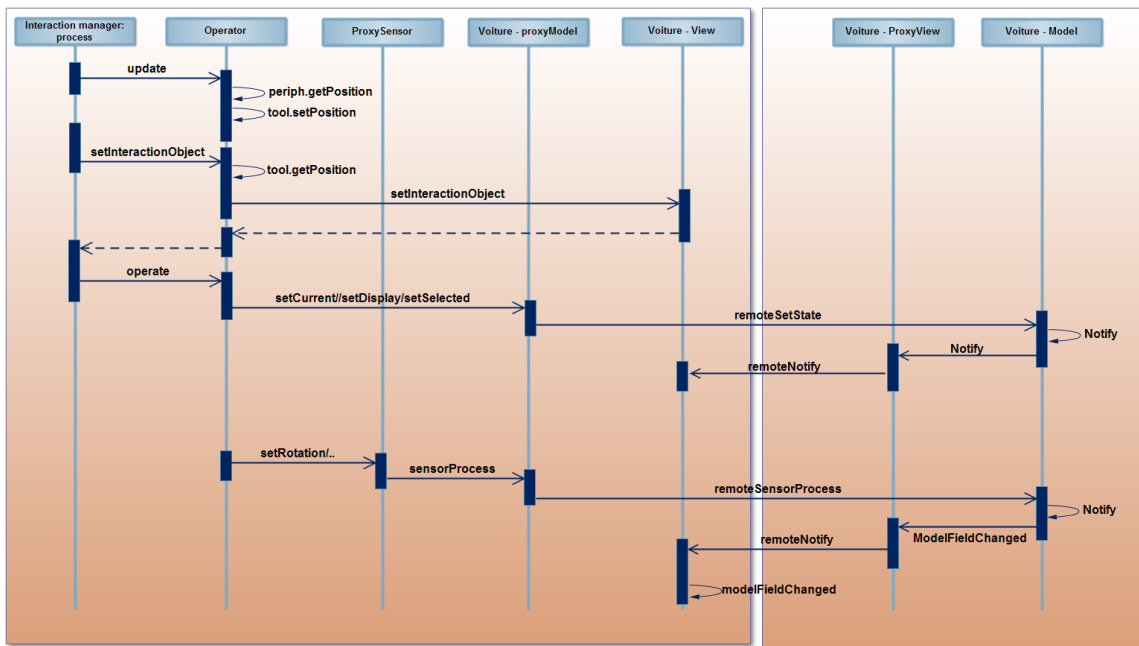


FIGURE 5.10 – Diagramme de sequence d’une interaction réalisée grâce à un Outil.

Pour prendre un second exemple, la figure 5.10 illustre deux étapes d’un second scenario d’interaction. Dans un premier temps l’outil dialogue avec le graphe de scène pour établir l’objet sur lequel porteront ces interactions, via `setInteractionObject` (dans le cadre d’une vue distante, cet appel serait bien sûr pris en charge par un proxy qui fera appel à `setDistantInteractionObject`). L’opérateur peut alors transmettre au modèle diverses informations relatives à son état (sélectionné, pointé, etc.). Dans un second temps, lorsque la phase de sélection est terminée, l’envoi d’un évènement de manipulation (en l’occurrence une rotation) est transmis au modèle local (le proxy). Ce proxy transmet alors l’évènement au modèle réel via `remoteSensorProcess`, puis la séquence de notification observateur/observable est réalisée de la même manière que pour la figure 5.9.

5.1.2 Déploiement et configuration XML

De part la nature multiple des cas d'utilisation potentiels de notre war room, et de la diversité qu'elle implique dans la détermination de ses fonctionnalités, notre plateforme doit être capable de s'adapter à de nombreux besoins. Pour fournir aux utilisateurs la capacité de construire le plus simplement possible les outils d'interaction, les modes de représentation, et la dynamique de collaboration qui doit prendre place dans cet espace, nous avons choisi de définir cette construction sous la forme de fichiers de configuration au format XML. Ce format d'encodage des informations permet ainsi au concepteur d'éditer les caractéristiques de son application sans nécessiter de compétences particulières en programmation. L'édition de ces fichiers se déroule en effet par simple utilisation d'un éditeur de texte standard.

La partie essentielle de la description d'une mise en application de notre plateforme réside dans la distribution des composantes entre les machines du réseau. En effet, c'est à travers cet aspect que l'utilisateur pourra déterminer la manière dont les objets seront accessibles sur chaque poste, tant en terme de visualisation qu'en terme d'interaction. La configuration XML de la plateforme est par conséquent constituée, entre autres, d'un fichier de déploiement réseau. A l'heure actuelle, la configuration est statique : notre étude porte en effet sur la distribution des composantes. Nous avons donc fait le choix le plus simple en terme d'identification des postes de travail (adresses IP plutôt que FQN). Un mécanisme plus dynamique d'identification des ordinateurs de la war room peut néanmoins être ajouté. Le fichier de déploiement XML permet de spécifier les caractéristiques suivantes (listing 1) :

- La liste des ordinateurs impliqués dans l'activité de collaboration de la war room. Cette liste définit chaque poste de travail à travers deux caractéristiques : son adresse IP, ainsi qu'un alias. L'adresse IP est présente de façon à permettre à l'interface de communication réseau de construire les références vers les composantes partagées sur le bus CORBA. En effet, pour localiser un objet distant et ainsi être en mesure d'invoquer des appels de méthode sur cet objet, l'interface réseau gérant la distribution des composantes construit ses références sous la forme d'adresses CORBALOC. Il s'agit d'une simple chaîne de caractère dont l'encodage permet de retrouver immédiatement un objet distant. Cette chaîne de caractère est présentée sous la forme "corbaloc :adresseIP :port/NomDeLObjetPartagé". Le second paramètre définissant un ordinateur de la war room, son alias, est une chaîne de caractère choisie arbitrairement, permettant de donner un nom à cet ordinateur. La plateforme interprète par conséquent directement chaque alias avec l'adresse IP qui lui est associée dans cette partie de la configuration. L'alias est ainsi utilisé plus loin dans le fichier de déploiement de l'application. Il est important de souligner que cette partie de la configuration est globale, puisqu'elle est commune à tous les ordinateurs installés dans la war room.
- L'alias du poste de travail courant. Ce paramètre est spécifique à chaque ordinateur. Il constitue une simplification de la syntaxe du fichier de configuration, permettant d'identifier la machine courante. Il aurait cependant été possible de déduire cette information directement à partir de son adresse IP.
- La liste des composantes de l'environnement logiciel. Cette dernière partie du fichier XML est identique pour tous les postes de travail. Elle constitue la définition des différents agents qui seront distribués dans l'environnement virtuel, ainsi que leur stratégie

de déploiement. Les noms de composantes employés sont ceux définis dans le fichier de configuration de l'application non fourni ici.

```
<site name="fixe" addr="134.206.10.59" />
<site name="portable" addr="134.206.10.51" />

<localhost name="fixe" />

<widget name="MeganeW">
  <model name="MeganeWModel" location="portable" />
  <view name="light" location="portable" />
  <view name="heavy" location="fixe" />
</widget>

<widget name="DistantUserWidget91">
  <model name="DistantUserModel" location="portable" />
  <view name="default" location="fixe" />
</widget>
```

Listing 1 – Exemple de configuration XML pour le déploiement des composantes

Nous pouvons constater, en observant le contenu de ce fichier de déploiement, qu'il ne contient pas d'information relative au système de proxy nécessaire à la communication inter-composantes. En effet, lors de la phase d'initialisation de l'application, le parsing du fichier XML permet à la plateforme de définir automatiquement la structuration des composantes et de leur proxy. Ce mécanisme de communication est par conséquent invisible pour l'utilisateur, qui ne doit alors se préoccuper que de la distribution effective des composantes MVO. Le second point observable sur cette configuration est qu'elle n'implique pas de la part de l'utilisateur de déclaration des éventuelles applications externes qui interviendront dans la war room. En effet, l'interface de communication qui prend place entre ces applications tierces et la plateforme logicielle n'est pas soumise aux contraintes de déploiement des composantes. La raison en est que l'application externe n'établit de communication qu'avec le modèle de l'objet virtuel. Dans ces conditions, il peut établir son lien avec n'importe lequel des modèles instanciés dans la pièce, puisque le mécanisme de proxy permettra de toutes façons de transférer automatiquement les requêtes au modèle réel.

5.2 Démonstrateur de distribution logicielle

Après avoir présenté la structuration logicielle et les mécanismes mis en oeuvre pour la distribution de l'environnement entre les différents dispositifs de la war room, cette section va maintenant décrire de façon plus concrète l'utilisation qui en est faite dans un premier démonstrateur, consistant en l'exploration des méthodes de travail industrielles relatives à la revue de projet automobile.

La réalisation de ce démonstrateur assez simple est essentiellement orientée vers l'aspect hétérogène des installations matérielles, ainsi que sur la dynamique de distribution de l'environnement virtuel entre ces postes de travail. Ce prototype ayant été implémenté antérieurement aux travaux relatifs à la gestion des canaux hétérogènes, il ne fait pas apparaître de hiérarchisation des données sous forme de plusieurs niveaux d'abstraction.

5.2.1 Scénario et fonctionnalités

L'installation matérielle

Dans son contexte d'utilisation multi-dispositifs, la war room est constituée de trois postes de travail différents (fig. 5.11) :

- **Un poste de visualisation et manipulation 3D** : Il s'agit ici d'un ordinateur dont l'assemblage matériel est en particulier dédié aux performances d'affichage 3D. Ce poste de travail est doté d'une carte graphique haute-performances, capable de calculer le rendu temps-réel d'objets complexes, constitués d'un important nombre de polygones. Pour accentuer le bénéfice de cette puissance en terme de calculs graphiques, le poste de travail est connecté à un dispositif d'affichage de grande taille. Cette station diffuse ainsi les images haute résolution à travers une installation constituée d'un écran de projection rétro-projeté. Pour bénéficier de capacités d'interaction les plus naturelles possibles, le poste de travail dispose d'un ensemble de périphériques de manipulation permettant aux utilisateurs de procéder à une interaction à deux mains libres, chaque main étant utilisée pour l'une des deux phases nécessaires à la manipulation des objets. La station dispose d'abord d'un système de tracking 3D de type ARTrack fournissant la position de la main de l'utilisateur dans l'espace. Cette modalité d'interaction s'avère très naturelle pour procéder à une tâche de pointage d'un objet de l'environnement virtuel. Le second périphérique d'interaction est constitué d'un système de manipulation absolue sous la forme d'un « stilet d'interaction », prototype développé par l'ESIEA dans le cadre du projet ANR Part@ge [DPC⁺09]. Cette centrale inertielle sans fil permet à l'utilisateur de manipuler en rotation l'objet virtuel en manipulant le périphérique réel, qui constitue alors un clone tangible de l'objet.
- **Un poste de travail mobile** : Ce dispositif est constitué d'un ordinateur portable (et donc, transportable dans la war room) aux performances standards. En terme d'interaction, cet ordinateur est doté du traditionnel couple clavier-souris.
- **Un poste de travail tactile** : Il s'agit d'un dispositif constitué d'une table tactile de type diamond touch. Cette table permet à plusieurs utilisateurs de travailler simultanément sur la même interface, à travers des modalités d'interaction tactile.

Le scénario d'utilisation

Le scénario d'utilisation illustré par cette installation vise à représenter le déploiement dynamique de l'activité dans la war room, et à mettre à l'épreuve le système de distribution des composantes de l'architecture sur laquelle repose notre support logiciel. La plateforme propose ainsi à une équipe d'utilisateurs de manipuler une voiture virtuelle et les principales pièces mécaniques la constituant, et de procéder à diverses interactions simples. Selon le scénario fictif imaginé, le déploiement de l'activité consiste en l'intégration dans la salle de l'ordinateur portable. Cet ordinateur, utilisé précédemment de façon individuelle pour concevoir la voiture, est ainsi connecté au reste des dispositifs via une connexion sans fil, et la plateforme permet alors de distribuer l'espace de travail de façon à faire bénéficier à l'équipe de la diversité matérielle mise en place dans la war room (fig. 5.11)

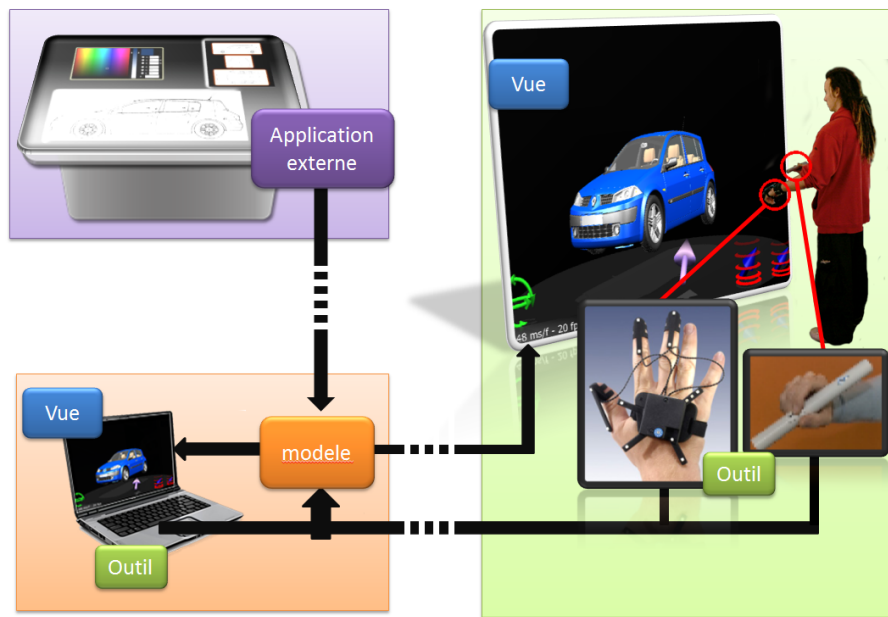


FIGURE 5.11 – Les trois postes de travail du démonstrateur de manipulation d’une maquette 3D.

L’environnement virtuel partagé

L’environnement virtuel partagé est constitué d’un seul objet : la voiture. Cette voiture est composée de plusieurs pièces majeures, c’est à dire les portières, le capot, les zones vitrées (pare-brise et fenêtres) ainsi que des différentes parties de la carrosserie. Chacune de ces parties peut être sélectionnée individuellement pour l’interaction. Cet environnement constitue une première approche du support de la pluridisciplinarité de l’équipe de collaborateurs. En effet, pour tirer parti des possibilités des différents postes de travail, tant en terme de visualisation qu’en terme d’interaction, deux types de représentations sont disponibles simultanément. Deux postes de travail présentent ainsi le projet sous la forme d’une interface de manipulation 3D (fig. 5.12), tandis que la table multitouch propose une interface 2D (fig. 5.13).

- **L’interface 3D** est constituée d’un modèle 3D de la voiture, permettant une visualisation globale de celle-ci (fig. 5.12). Ce mode de représentation ne constitue pas une approche très technique de la voiture, mais fournit un support plus adapté à l’observation et à la discussion autour des caractéristiques générales du projet. Cette interface fournit plusieurs possibilités en terme de manipulation. Il est notamment possible de tourner la voiture pour changer de point de vue visuel. De même, certaines des pièces constitutives de la voiture sont sélectionnables et manipulables, de manière à pouvoir appréhender leur fonctionnement en 3D. Il est ainsi possible d’ouvrir et de fermer les portières, et d’observer l’ergonomie de celles-ci ainsi que leur cohérence visuelle vis-à-vis du reste de la voiture.
- **L’interface 2D** quant à elle, propose une représentation plus symbolique de la voiture (fig. 5.13). Elle est constituée de plusieurs plans techniques, également appelés blueprints. Ces plans techniques permettent d’observer la voiture selon plusieurs points de vue re-

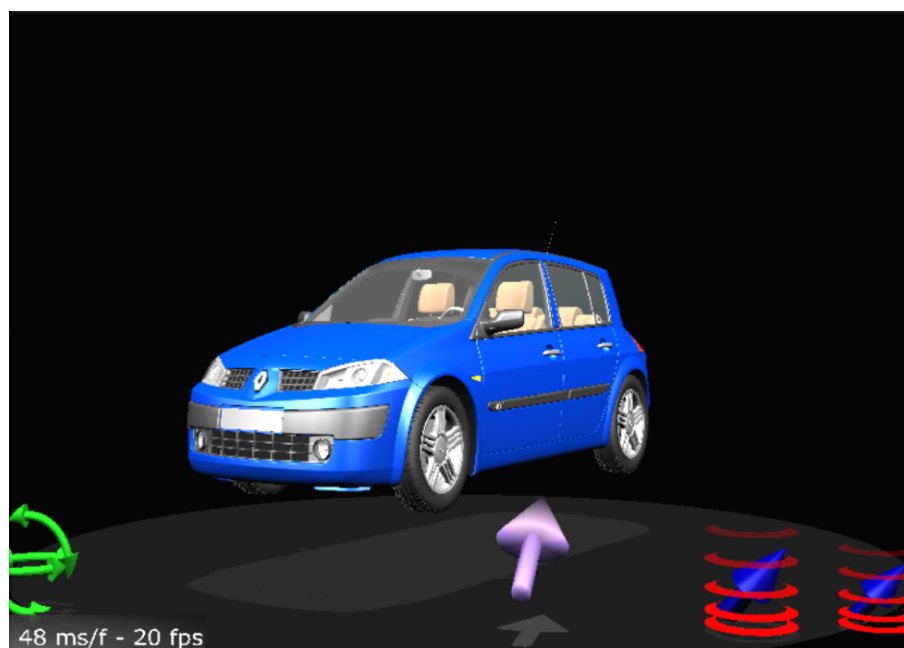


FIGURE 5.12 – L’interface 3D de la war room

présentant des vues de face, dos, profil, et dessus. Ces différents plans peuvent alors être organisés librement dans l’interface, par un simple glissement. Le choix de cette interface, outre le fait qu’elle apporte aux utilisateurs une vision différente du même projet, est conçue pour exploiter de façon plus naturelle le mode d’interaction d’une table tactile. Parmi les outils d’interaction mis à disposition sur cette interface 2D, les utilisateurs ont la possibilité de sélectionner les différentes sous parties de l’objet, et de modifier leur couleur grâce à une palette apparaissant à chaque sélection. L’interface tactile propose également aux utilisateurs la possibilité de manipuler les portières de la voiture. Pour ce faire, les participants sont invités à sélectionner les poignées de porte, occasionnant l’ouverture d’un item de manipulation dans l’interface. Cet item peut prendre la forme d’un slider traditionnel (permettant de choisir le degré d’ouverture de la portière), ou d’une checkbox. Si ce mode de manipulation n’est pas le plus naturel comparé à la manipulation 3D directe, il constitue cependant une forme d’adaptation de la modalité d’interaction inhérente à la table, de façon à la rendre compatible avec l’objet d’interaction.

La cohérence globale du projet est bien sûr maintenue entre les deux interfaces, et chaque modification (de manipulation ou de changement de couleur) est immédiatement répercutée sur tous les postes de travail de la pièce, indépendamment de l’interface à l’origine de cette modification.

Fonctionnalités relatives à la collaboration colocalisée

Au delà de la diversité des modalités d’interaction et des dispositifs matériels installés dans cette version de la war room, la plateforme logicielle présentée précédemment a été configurée pour répondre efficacement aux contraintes relatives à l’utilisation d’un système d’interaction collaborative en situation de coprésence. L’application logicielle résultant de cette configuration offre par conséquent à l’équipe de collaborateurs plusieurs fonctions

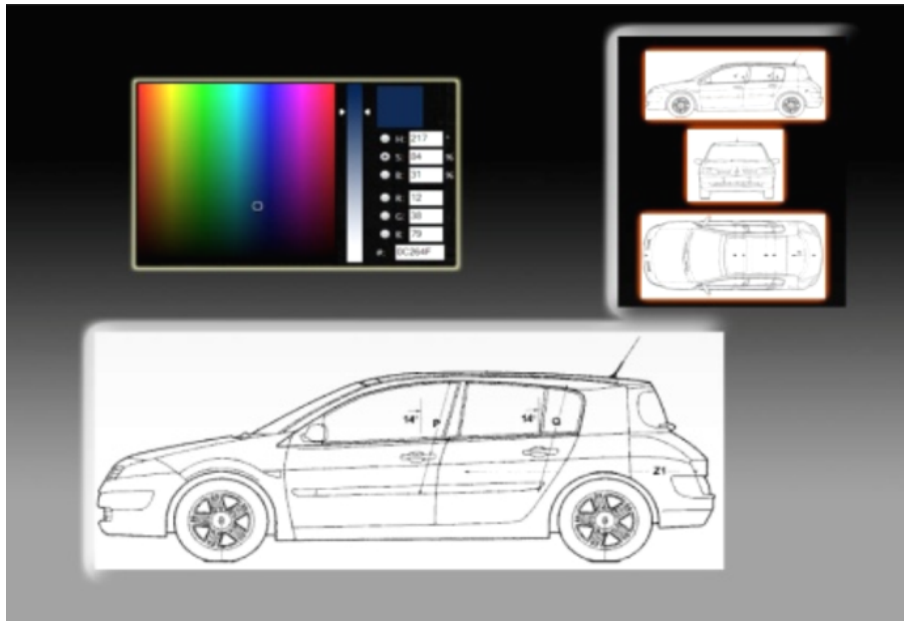


FIGURE 5.13 – L'interface 2D de la war room

permettant de répondre à leurs besoins dans un tel contexte :

- **distribution de l'activité : l'affichage dynamique de la maquette.**

Ce démonstrateur illustre notamment la réponse apportée vis-à-vis de la perception spatiale des utilisateurs. En effet, la représentation 3D de la voiture possède une consistance proche d'un document réel, dans la mesure où elle est unique, et mobile dans l'environnement :

Le modèle 3D n'est ainsi diffusé que sur un seul poste de travail à la fois, facilitant l'assimilation de l'objet par les participants. De plus, au même titre qu'un document physique (une feuille de notes sur papier par exemple), l'affichage du modèle 3D peut être déplacé d'un écran vers un autre en fonction des besoins des collaborateurs. Pour ce faire, un objet particulier est disposé au sein des interfaces 3D. Cette objet, lorsqu'il est sélectionné par l'utilisateur, entraîne la disparition du modèle 3D de l'affichage courant, et la réapparition de ce même modèle sur l'écran de l'autre poste de travail. Lorsque l'ordinateur portable est connecté à la plateforme, il est donc possible pour les participants de basculer l'affichage 3D de la voiture vers la station dotée d'un écran de grande taille. De même, pour procéder à une interaction plus individuelle, un utilisateur peut décider de rétablir l'affichage sur le portable, en utilisant le même artefact visuel(fig. 5.14)

- **distribution de l'activité : l'interaction dynamique**

Pour appuyer cette dynamique, l'application est également configurée pour prendre en charge l'interaction distante. En effet, les utilisateurs ont la capacité d'interagir directement sur le modèle 3D affiché sur l'écran géant en utilisant les périphériques d'interaction de l'ordinateur portable. Cette fonctionnalité vise entre autres à répondre à une contrainte liée aux dispositifs matériels. Comme nous l'avons précisé plus haut, la station de travail dédiée à l'affichage 3D sur grand écran est dotée de dispositifs d'interaction basés sur la détection d'un gant de données. En dépit de la qualité de la modalité



FIGURE 5.14 – Déplacement de l’affichage par clic sur un item visuel

d’interaction fournie, ce dispositif souffre d’un inconvénient majeur vis-à-vis de la collaboration : son usage est restreint à une utilisation individuelle, puisque le gant ne peut être porté que par un seul participant à la fois. Sans distribution de l’interaction, la plateforme confronterait les collaborateurs à l’impossibilité d’interagir simultanément sur la représentation 3D. Au contraire, la distribution de la plateforme mise en place ici permet à un second utilisateur d’utiliser le couple clavier-souris de l’ordinateur portable pour manipuler la voiture lorsqu’elle est affichée sur le grand écran.

La tâche de manipulation reposant sur une séquence d’actions impliquant une sélection de l’objet à manipuler, il est nécessaire pour l’utilisateur de pouvoir percevoir sur l’interface visuelle un artefact permettant d’avoir un retour visuel sur l’action de pointage. Ainsi, la plateforme logicielle inclus dans ses interfaces 3D un troisième objet, dont le fonctionnement est comparable à celui permettant de basculer l’affichage de la voiture. Lorsque l’utilisateur sélectionne cet objet à partir du PC portable, son pointeur de souris est immédiatement transféré vers l’autre écran 3D. Sur l’interface 3D visible sur le grand écran, il apparaît alors deux pointeurs, l’un étant directement contrôlé par le système de tracking local, tandis que l’autre est dirigé par la souris de l’ordinateur portable.

- **distribution de l’activité : adaptation au contexte matériel**

Pour aller plus loin dans l’adaptation au contexte matériel de la war room, la plateforme est également configurée pour prendre en compte les performances de chaque poste de travail. En effet, les deux ordinateurs utilisés pour procéder à la représentation 3D de la voiture possèdent des caractéristiques très différentes en terme de puissance de calcul graphique. Ainsi, tandis que la station dotée d’un grand écran est capable de gérer le tracé d’un modèle très haute définition de façon fluide, l’ordinateur portable ne permet

de prendre en charge qu'un affichage plus modeste en terme de qualité. Les composantes distribuées entre ces deux ordinateurs sont ainsi définies en conséquence. Lorsque l'affichage 3D de la voiture est confié à l'ordinateur portable, le modèle utilisé est composé de quelques milliers de polygones. Au contraire, lorsque cette tâche est confiée à la station dédiée à l'affichage haute qualité, le modèle 3D utilisé est une version subdivisée, comportant un beaucoup plus grand nombre de polygones, possédant des contours plus lisses, et des matériaux plus complexes (transparence des vitres par exemple).

5.2.2 Implémentation : les composantes impliquées

Composition de l'objet interactif

Pour permettre ces fonctionnalités supportant la dynamique de collaboration et l'hétérogénéité des spécifications matérielles, la construction logicielle de la plateforme a été configurée en conséquence. L'objet d'interaction est ici unique, puisqu'il est constitué uniquement de la voiture, les différentes pièces étant considérées comme des sous-parties du même objet. L'objet virtuel manipulé à travers l'application est ainsi représenté sous la forme d'un unique agent, lui même constitué de trois composantes :

- le modèle, prenant en charge tous les paramètres intrinsèques de l'objet, c'est à dire les coordonnées en position et en rotation de chaque pièce, ainsi que leur code couleur RGB.
- une première vue, permettant de représenter la voiture sous la forme d'un modèle 3D léger, grâce à un nombre de polygones plus faible.
- une seconde vue, fournissant une représentation similaire, mais cette fois à travers une topologie 3D plus dense, et des matériaux plus complexes.

Il est intéressant de souligner le choix du jeu de données pris en charge par le modèle. En effet, nous pouvons constater que la topologie 3D du modèle de la voiture n'est pas incluse dans ces données. Ce choix est justifié par plusieurs raisons. Premièrement, nous avons précisé plus haut que les deux postes de travail ne sont pas en mesure d'afficher des représentations aussi complexes l'une que l'autre. Par conséquent, le degré de complexité du modèle 3D affiché doit être propre à chaque poste de travail. Un agent étant ici constitué d'un unique modèle, qui est alors commun à toute la plateforme, il n'apparaît donc pas pertinent de stocker la topologie 3D sous la forme d'un champ de ce modèle (la décimation des polygones du modèle 3D pour adapter son affichage à la volée en fonction de l'ordinateur s'avèrerait dans notre contexte une solution trop lourde à prendre en charge pour la vue). La charge de la description de cette topologie à afficher est par conséquent intégrée directement dans les vues. Cette contrainte de modèle unique peut bien sûr être contournée par l'utilisation de canaux hétérogènes, entraînant l'existence de plusieurs modèles concrets pour définir l'objet. L'application exploiterait ainsi deux canaux 3D. Cependant, cette approche n'est pas indispensable dans le contexte présent, ce qui nous amène à la seconde raison : le scénario d'utilisation ne prévoit pas de fournir aux utilisateurs d'outils permettant de modifier la topologie des modèles 3D de la voiture. Ce paramètre ne nécessite par conséquent pas l'utilisation d'un champ de données sur lequel les composantes seraient en mesure d'interagir. Le modèle 3D utilisé pour la représentation est alors une constante

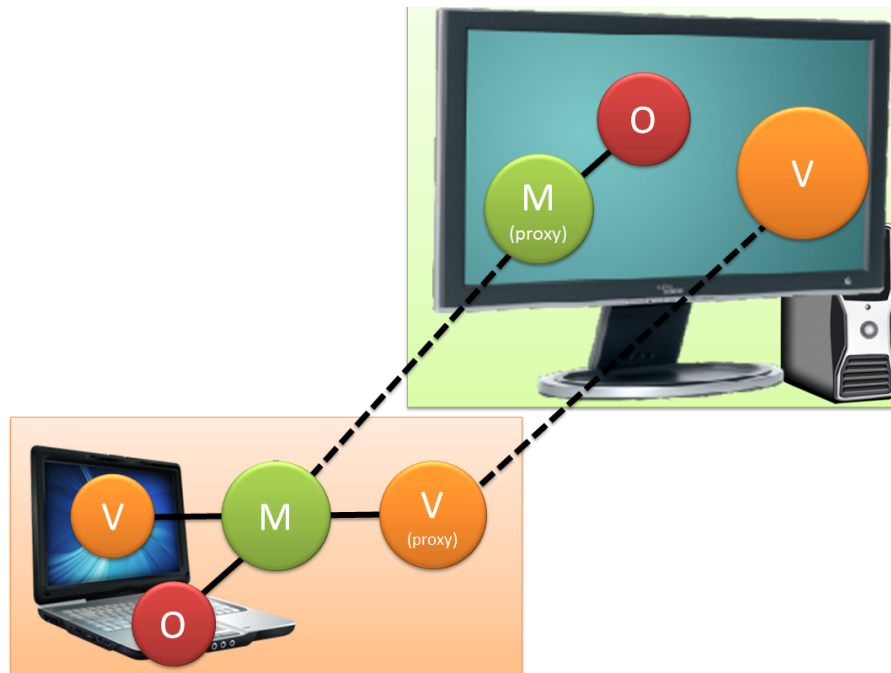


FIGURE 5.15 – distribution des composants dans le premier prototype

intégrée directement à la vue.

Ces composantes sont distribuées entre les stations de travail de la façon suivante (fig. 5.15) :

- Le modèle est pris en charge par l'ordinateur portable. En effet, d'après le scénario d'utilisation prévu, la voiture a potentiellement été sujet à un travail individuel avant d'être connectée au reste de la war room. Ce modèle sert alors de brique constitutive pour importer le projet au sein de l'environnement.
- La première vue, associée au modèle 3D le plus léger, est également assignée à l'ordinateur portable. Outre la capacité d'affichage sur l'un ou l'autre des postes 3D, cette configuration fournit également une certaine autonomie à l'ordinateur portable, qui coïncide avec le scénario défini. Ainsi, même lorsque cet ordinateur est déconnecté de la plateforme collaborative, il dispose des composants nécessaires à un travail individuel « hors ligne » : il dispose en effet du jeu de données définissant l'objet, ainsi que d'une vue apportant un support visuel pour l'activité.
- La seconde vue, plus complexe, est naturellement disposée sur le poste de travail le plus puissant.

En observant cette distribution, et la multiplicité des vues, nous pouvons constater que le transfert dynamique de la vue repose sur un mécanisme différent de celui qui apparaît visuellement aux utilisateurs. En effet, lorsque l'utilisateur « envoie » l'objet 3D vers un autre poste de travail, le fonctionnement interne de l'application réside dans la désactivation de la vue locale, et l'activation de la vue distante. La seule information transitant réellement sur le réseau lors de cette action est composée de l'évènement commandant l'ac-

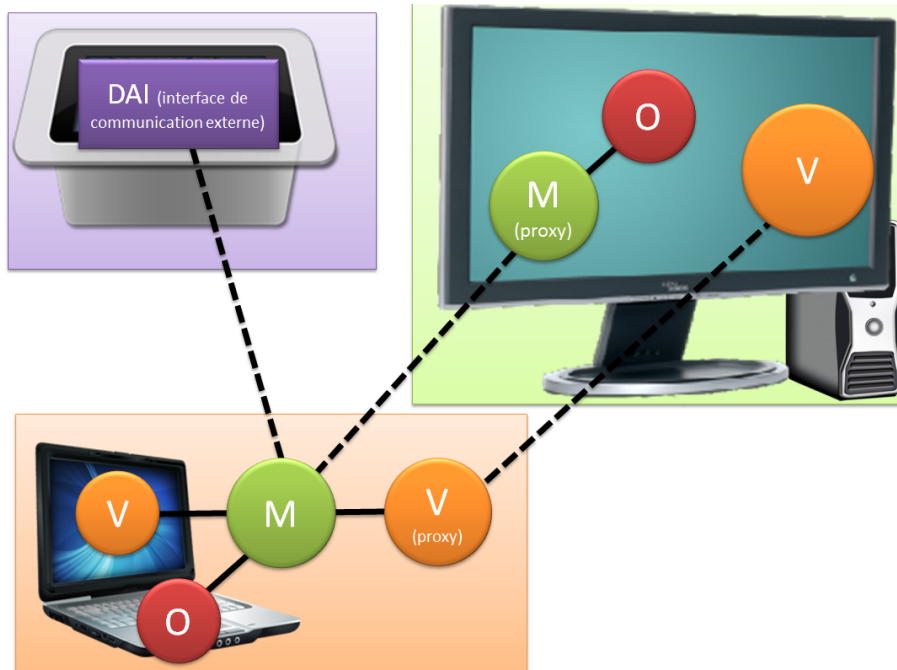


FIGURE 5.16 – distribution des composants dans le premier prototype

tivation de la vue distante. Ce mécanisme apporte aux utilisateurs la possibilité de changer facilement la stratégie d’affichage. En effet, par le simple changement d’une ligne du fichier de configuration, ils ont la faculté de passer d’une activité selon laquelle l’affichage 3D de la voiture est unique, à une situation proposant deux affichages simultanés sur deux postes de travail distincts, apportant une approche plus individuelle de l’interaction. La plateforme peut ainsi être configurée simplement pour autoriser l’activation simultanée des vues sur les deux ordinateurs 3D.

Gestion de l’interaction

Chaque poste de travail installé dans la war room comporte ses propres outils d’interaction. Ainsi, lorsque deux utilisateurs décident par exemple de travailler simultanément sur l’affichage 3D de grande taille, mais en conservant leurs périphériques d’interaction locaux, l’outil localisé sur l’ordinateur portable agit à distance sur la vue. L’intégralité du processus associé à la logique d’interaction de cet outil est par conséquent exécutée sur cet ordinateur portable. De même, le collaborateur utilisant la centrale inertielle et le gant de données verra ses mouvements d’interaction interprétés par son outil local. Ces deux outils appliqueront alors les potentielles modifications issues de ces interaction directement sur la composante modèle prise en charge par l’ordinateur portable.

L’interface 2D : un module complémentaire

Dans ce démonstrateur, la plateforme logicielle propose une interface d’interaction 2D. Cette interface est ici programmée sous la forme d’une application externe (fig. 5.16). En effet, l’application tactile exécutée sur la table interactive est un logiciel autonome pré-existant, auquel une couche de communication a été ajoutée pour permettre les échanges

avec le support logiciel déployé dans la war room.

En terme de distribution de l'activité, cette application externe est bien sûr prise en charge par la table tactile, et ses liens avec la plateforme logicielle sont établis directement avec le modèle de la voiture, stocké sur l'ordinateur portable. A chaque interaction réalisée à travers la table tactile (ouverture/fermeture d'une portière, changement de couleur d'une pièce), l'application convertit cette interaction sous forme d'évènement de modification de champ du modèle, qu'elle transmet alors au modèle de l'objet par la connexion réseau.

Au même titre que pour les interfaces 3D des deux autres postes de travail, la représentation de l'objet dans cet environnement est directement incluse dans l'application 2D. Cependant, la gestion de l'interaction est différente des interfaces 3D. En effet, contrairement aux postes 3D, qui proposent une manipulation directe des éléments de la voiture (les interactions de manipulations sont directement associées aux données du modèle), les tâches d'interactions relatives à l'ouverture et à la fermeture des portières n'est pas compatible avec le mode de représentation choisi dans cette interface 2D : une automobile représentée sous la forme de plans techniques n'offre pas de métaphore d'interaction permettant ce type de manipulation. L'application tactile passe par conséquent par l'utilisation de divers artefacts visuels (en l'occurrence, des sliders et checkboxes), qui reposent sur des données élémentaires, sans lien direct avec les données du modèle (un coefficient pour le slider, et un booléen pour la checkbox). La logique d'interaction implique par conséquent une retranscription de ces données élémentaires en interne à l'application tierce, de façon à les convertir en évènements liés directement au modèle.

D'une certaine manière, les informations concrètes liées aux artefacts d'interaction 2D de l'application tierce représentent des données qui pourrait prendre place dans le cadre de la mise en place du modèle concret d'un canal dédié. Cette solution offrirait une approche plus structurée de la diffusion de l'interaction, pouvant permettre notamment l'intégration d'un nouveau poste de travail disposant d'une nouvelle interface différente, mais reposant sur les mêmes données concrètes. Cependant, la réalisation de ce prototype ayant eu lieu antérieurement à l'implémentation du système de canaux hétérogènes de notre architecture, cette stratégie n'a pas pu être mise en application dans ce cas précis.

5.2.3 Résultats

Comme nous l'avons illustré dans les schémas de distribution de l'application, notre war room répartit chaque composante logicielle sur des postes de travail distincts. Par conséquent, le nombre d'évènements et de données qui transitent sur le bus CORBA s'avère important. Pour illustrer la densité des échanges intercomposantes, la figure 5.9 établit le diagramme de séquence répertoriant les appels de méthode qui ont lieu suite à la modification d'un des champs du modèle.

Le mode de manipulation utilisé sur le modèle 3D de la voiture constitue d'ailleurs une source de transmission d'évènements relativement lourde. En effet, dès lors que ce modèle est sélectionné sur le poste de travail équipé de périphériques de manipulation absolue, ces périphériques émettront à chaque pas de temps leur position et leur orientation. Etant donné l'importante précision de ces périphériques, chaque pas de temps se verra associé

à la génération d'un ou plusieurs évènements de manipulation de l'objet virtuel qui sera diffusé dans la structure de composantes réparties.

Les appels de méthode distants, même lorsqu'ils sont fréquents, ne remettent cependant pas en question la stabilité de l'ensemble de la plateforme, ni la fluidité de son exécution. En effet, lors des tests en conditions réelles, aucun ralentissement n'est à déplorer. Différents tests préliminaires ont d'ailleurs été réalisés, visant à mettre à l'épreuve le bus CORBA face à l'envoi massif et simultané de nombreux messages réseau de petite taille. Là encore, aucune perte ni ralentissement n'a été détecté.

Ce démonstrateur sur projet automobile a été présenté aux journées professionnelles de Laval Virtual 2008, nous permettant ainsi de recueillir les impressions et suggestions d'acteurs fortement impliqués dans l'utilisation industrielle de la réalité virtuelle. Les visiteurs ont exprimé un vif intérêt en regard de la faculté de collaborer simultanément sur des dispositifs différents. Outre la diversité des modalités d'interactions et l'hétérogénéité matérielle, la proposition visant à exposer à l'équipe d'utilisateurs plusieurs modes de visualisation simultanés a suscité un certain enthousiasme.

De nombreux professionnels du domaine industriel ont exprimé l'intérêt qu'ils trouveraient à disposer de ce type de plateforme pour l'insérer dans le contexte d'une activité de collaboration distante.

5.2.4 Perspectives

L'interface tactile 2D proposée dans cette mise en situation est assimilable à un canal singulier. En effet, elle propose d'interagir sur l'objet virtuel à travers un mode de visualisation et d'interaction particulier. De même, les deux représentations 3D de la voiture dans l'environnement peuvent être considérées comme appartenant à un unique canal 3D. Cette approche multi-canaux n'est cependant pas perceptible dans la structuration logicielle du démonstrateur tel qu'il est implémenté. L'intégration d'un nouveau terminal permettant d'aborder la voiture sous la forme d'une représentation 2D au sein de cette plateforme implique en effet d'implémenter ce terminal sous la forme d'une application autonome, c'est à dire reposant sur des données purement locales. Cette situation provoquerait alors la formation de doublons dans les données : les informations 2D déjà présentes au sein de l'application tactile devraient en effet être dupliquées au sein des deux applications 2D.

Pour répondre à cette contrainte, et ainsi proposer une hiérarchisation des données de façon à les rendre unique, nous pouvons proposer d'implémenter la même plateforme mais cette fois en prenant en compte la notion de canaux M_NV-O . Pour les retranscrire, nous pouvons considérer la voiture selon deux canaux : 2D, et 3D. Conformément à l'architecture que nous avons défini pour encoder ces canaux au niveau logiciel, l'objet virtuel de la voiture se verra alors constitué de deux modèles concrets, respectifs à ces deux canaux. Pour conserver la cohérence de l'ensemble, ces deux modèles concrets sont alors parentés à un modèle abstrait commun (fig. 5.17).

De cette manière, chaque définition concrète de la voiture est constituée de données directement interprétables dans le canal qui lui est associé. Par exemple, le positionnement de la

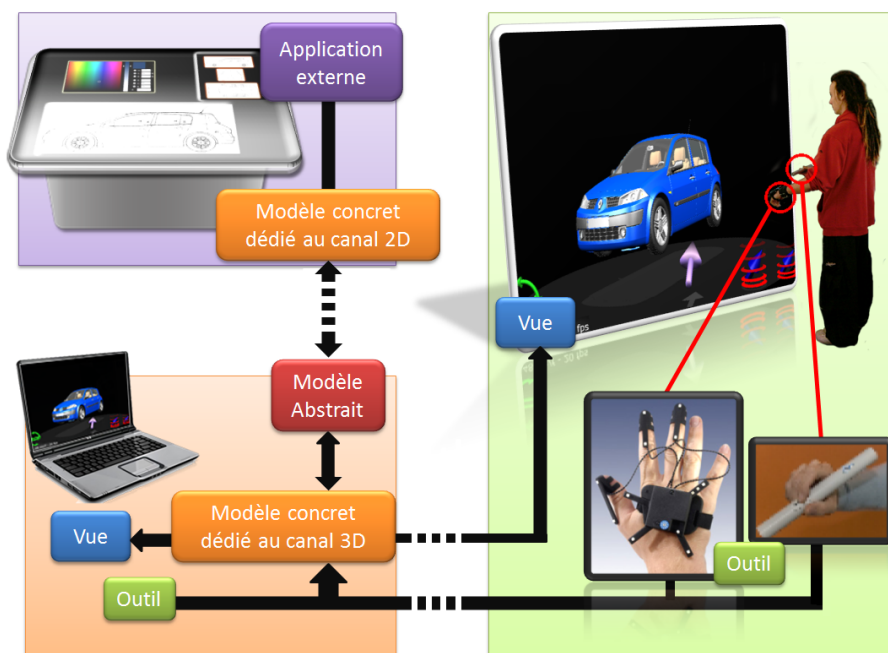


FIGURE 5.17 – Proposition de remaniement du prototype, exploitant différents degrés d'abstraction

portière (c'est à dire son degré d'ouverture) se verra encodé sous deux formes différentes : une matrice de transformation dans le cadre du canal 3D, tandis que le canal 2D verra ce positionnement représenté par une valeur numérique correspondant au positionnement du curseur de manipulation du slider. D'un point de vue abstrait, ce positionnement correspond à un angle de rotation, c'est par conséquent sous cette forme que l'information peut apparaître au sein du modèle abstrait.

5.3 Mise en oeuvre de l'abstraction $M_N V-O$

La seconde partie de ce chapitre est dédiée à la mise en oeuvre du système d'abstraction que nous avons proposé. Comme nous l'avons développé dans le deuxième chapitre de ce document, l'hétérogénéité de la plateforme d'interaction colocalisée repose sur une structuration des données particulière. Cette structuration est basée sur la superposition de plusieurs couches, chacune dédiée à un niveau d'abstraction différent. Les données intrinsèques d'un objet virtuel sont alors représentées sous différentes formes simultanément, dépendantes des canaux déployés.

Ce concept a fait l'objet d'une mise en application sous la forme d'un second prototype logiciel, permettant ainsi de mettre à l'épreuve la validité de cette structuration des données. Avant d'exposer concrètement le scénario et le déploiement des composantes mis en place dans ce prototype, nous allons procéder à une description détaillée de l'implémentation que nous avons réalisé sur notre architecture logicielle.

Dans notre implémentation précédente, la totalité des données relatives à un objet virtuel étaient prise en charge par une unique composante modèle. Pour intégrer la multiplicité des niveaux d'abstraction des données, notre proposition consiste à modifier le schéma modèle-vue pour lui donner la forme modèle abstrait-modèle concret-vue. Le lien entre le modèle

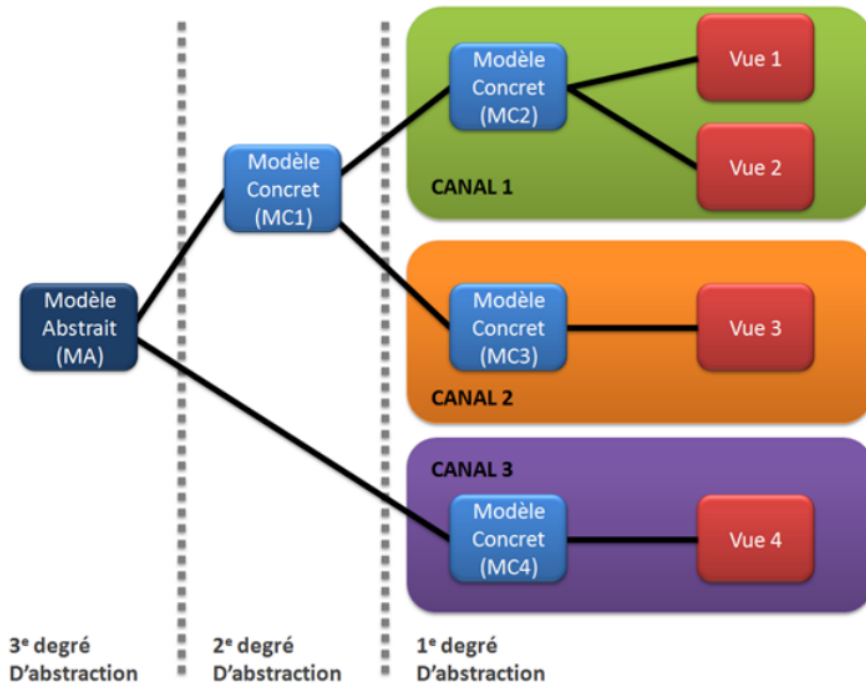


FIGURE 5.18 – Exemple de hiérarchie de plusieurs degrés d'abstraction. Le premier degré constitue le niveau "concret".

et la vue est alors présentée sous la forme d'une chaîne de modèles concrets. Le nombre de modèles concrets est variable, et l'encodage global d'un objet virtuel vis-à-vis des canaux d'interaction prend alors la forme d'un arbre, comme sur la figure 5.18.

En terme d'implémentation cela se traduit par l'insertion d'une nouvelle classe dans l'architecture. Cette classe, le "modèle concret" constitue une forme hybride entre la vue et le modèle. En effet, le modèle concret se comporte comme un modèle classique vis-à-vis de la vue, et comme une vue vis-à-vis du modèle plus abstrait. Une relation de type Observateur/Observable est donc mise en place entre ces composantes. Une relation supplémentaire, spécifique à la hiérarchie des modèles, est également à intégrer. Elle permet de transmettre les modifications du modèle concret vers le modèle abstrait. La figure 5.19 présente le diagramme de classes dans lequel s'insère le modèle concret.

Sur cette extension de l'architecture, le modèle "classique" InteractiveObjectModel devient ce que nous appelons le modèle abstrait, qui constitue la racine de l'arbre. Cette classe présente en effet déjà toutes les caractéristiques nécessaires à la notion de modèle abstrait, puisque ce dernier ne nécessite aucun besoin particulier en terme d'implémentation.

Le modèle concret, présenté quant-à-lui sous la forme de la classe InteractiveObjectConcreteModel, est ici une forme dérivée du modèle abstrait. De part cette relation, le modèle concret hérite ainsi naturellement de toutes les fonctionnalités d'un modèle classique, ainsi que de sa faculté à se connecter au bus CORBA pour être accessible à distance. Les méthodes permettant de réceptionner les événements d'interaction sont alors surchargées de façon à ajouter une notification du modèle abstrait correspondant.

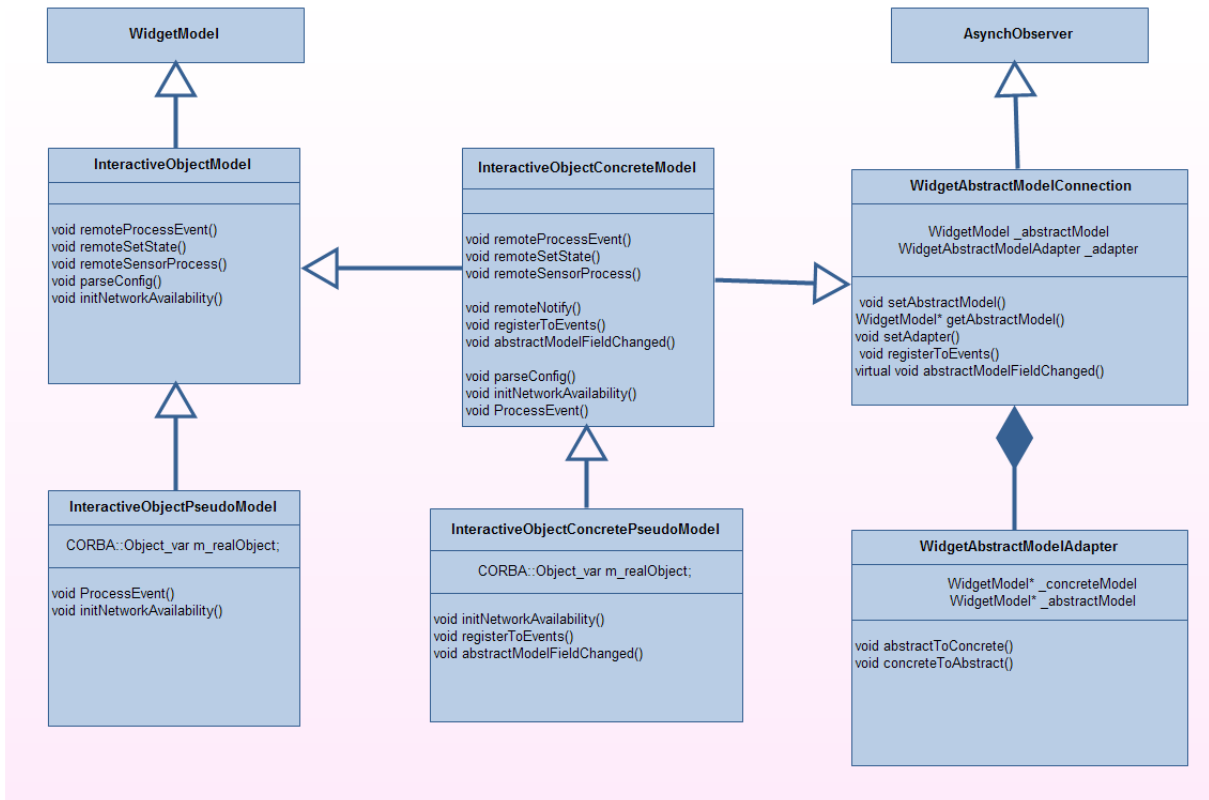


FIGURE 5.19 – Diagramme de classe illustrant l'implémentation du modèle concret dans l'architecture logicielle.

Nous pouvons également constater que la classe `InteractiveObjectConcreteModel` hérite également d'une autre classe `WidgetAbstractModelConnection`, elle-même dérivée de `AsyncObserver`. Cette branche du diagramme illustre la manière dont le modèle concret bénéficie, au même titre que la vue, du caractère "observateur" du patron de conception Observateur/Observable.

Pour tirer parti de son caractère "observateur", la classe `InteractiveObjectConcreteModel` implémente des méthodes analogues à celles de la composante vue à travers les méthodes `remoteNotify` ou `abstractModelFieldChanged`. Ceci permet au modèle concret de s'enregistrer auprès d'un modèle abstrait pour en suivre les changements de données.

Le caractère "observable", nécessaire pour la relation avec la vue, est obtenu naturellement par héritage puisque le modèle abstrait `InteractiveObjectModel` implémente déjà cette fonctionnalité. Ce caractère "observable" permet donc à une composante Vue de s'enregistrer auprès du modèle concret, et d'être notifiée des modifications des données concrètes.

De même que pour l'implémentation du précédent démonstrateur, le modèle concret dispose également d'une forme dérivée appelée `InteractiveObjectConcretePseudoModel`. Cette classe, comme son nom l'indique, constitue la forme "proxy" du modèle concret, et prend en charge l'invocation des méthodes distantes à travers le réseau.

Pour entrer plus en détail sur la classe `WidgetAbstractModelConnection`, nous pouvons dire qu'elle remplit deux rôles. D'abord, comme nous l'avons précisé, elle fournit au mo-

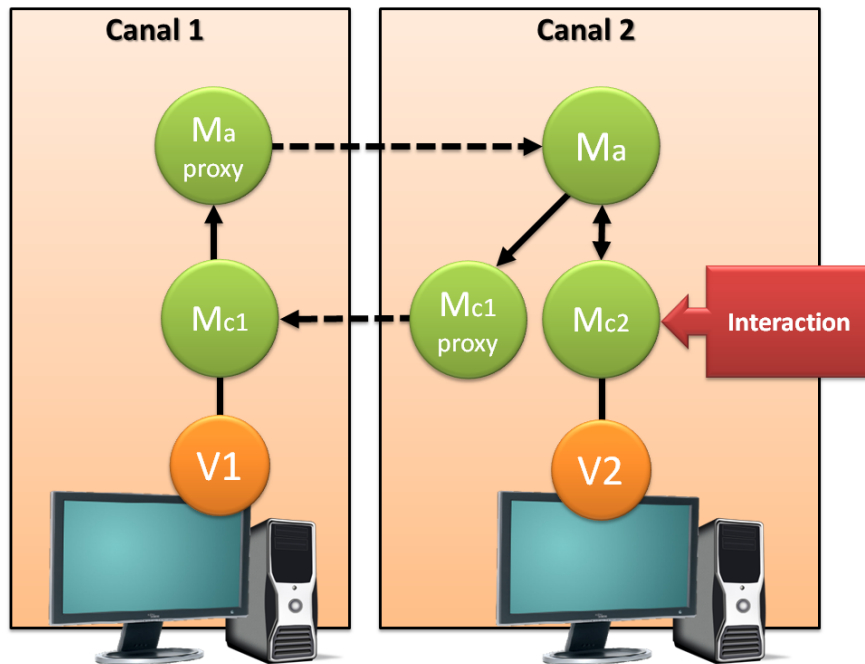


FIGURE 5.20 – Composantes déployées dans un contexte multicanal

dèle le caractère "Observateur", ou plus généralement des fonctionnalités similaires à la composante vue de SPIN|3D : elle inclut ainsi dans ses attributs une référence vers le modèle de plus haut niveau d'abstraction, permettant ainsi de transmettre les interactions aux niveaux supérieurs. Le second rôle joué par cette classe consiste en la mise en place d'une relation de composition : cette composition porte sur l'inclusion parmi ses attributs d'un objet de type `WidgetAbstractModelAdapter`. Cet objet prend en charge les algorithmes de retranscription permettant de convertir une information abstraite en donnée concrète, et vice-versa : c'est le rôle des méthodes `concreteToAbstract` et `abstractToConcrete`.

Pour illustrer avec précision la nature des échanges entre les différents degrés d'abstraction, considérons le déploiement de composantes tel qu'il apparaît sur la figure 5.20 : ici, deux canaux d'interaction/visualisation sont disponibles, chacun déployé sur un poste de travail. Nous avons ici fait le choix de constituer l'objet de deux définitions concrètes, propres à chaque canal (ces définitions concrètes représentent alors des données visuelles interprétables par leurs vues respectives). Ces deux définitions concrètes, visibles sous la forme de deux modèles concrets (`Mc1` et `Mc2`), maintiennent leur cohérence grâce à un modèle abstrait commun (`Ma`). De plus, la communication réseau entre les composantes est assurée par des composantes proxy situées sur chaque poste.

Imaginons maintenant qu'un utilisateur procède à une interaction sur l'objet à travers le second canal. Outre des modifications immédiates sur le modèle concret concerné (`Mc2`), cette interaction sera à l'origine de toute une séquence d'appels de méthodes, visible sur la figure 5.21 :

Dans un premier temps, le modèle concret `MC2`, dès la réception de l'évènement d'interaction, procédera à une mise à jour de sa/ses vues via la relation Observateur/Observable (appel à `modelFieldChanged`). Ce même modèle concret invoquera ensuite l'algorithme de

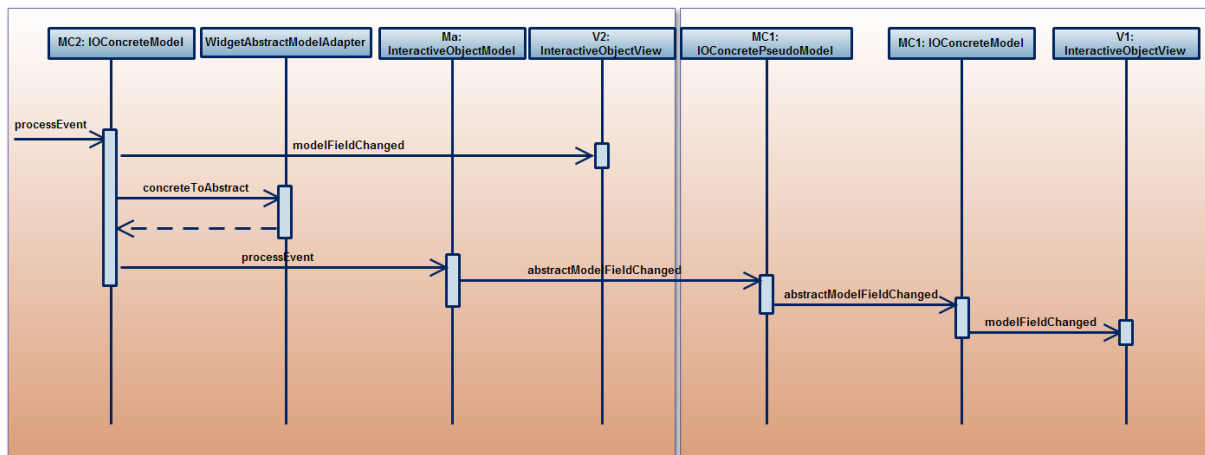


FIGURE 5.21 – Diagramme de séquence dans le cadre d'une utilisation multicanal

retranscription abstraite de son `WidgetAbstractModelAdapter`. Il pourra alors transmettre un nouvel évènement, fruit d'une conversion concret/abstrait, vers le modèle abstrait `Ma`. Dès la réception de cet évènement, le modèle abstrait notifiera tous les modèles concrets qui lui sont attachés (à l'exception de `MC2`, qui est la source de l'évènement) d'un changement dans la définition abstraite de l'objet virtuel. Le modèle concret concerné (`MC1`) sera alors notifié à distance de ces modifications via sa composante proxy, et pourra à son tour invoquer l'algorithme de conversion abstrait/concret correspondant, modifier ses propres données concrètes, et enfin mettre à jour sa/ses vues en conséquence.

5.4 Mise en application de la plateforme dans un projet d'urbanisme

Après implémentation de l'architecture M_NV-O prenant en charge de multiples degrés d'abstraction, nous avons orienté nos travaux vers l'utilisation de notre plateforme logicielle en situation réelle. L'objectif est ici de mettre à l'épreuve la structure logicielle globale de la war room, tout en mettant l'accent vers l'idée de répondre efficacement aux besoins d'une activité collaborative en situation réelle. Pour illustrer cette proposition, notre choix s'est tourné vers le domaine de l'architecture, ou plus précisément vers l'urbanisme et le ré-aménagement de territoire. Nous allons dans un premier temps développer le travail d'analyse que nous avons réalisé vis-à-vis de ce domaine, puis nous illustrerons la solution logicielle et matérielle que nous avons bâtie pour assister les professionnels urbanistes dans la conception d'un nouvel aménagement urbain.

5.4.1 L'urbanisme : contexte de notre étude

Le travail qui est effectué autour d'un projet de ré-aménagement urbain constitue un enchevêtrement de tâches complexes, qui nécessite au préalable une étude analytique des besoins associés à ce type d'activité. Pour la réalisation de notre prototype, notre travail d'analyse du problème s'est déroulé en deux étapes. Dans un premier temps, nous avons discuté avec la communauté urbaine de la métropole Lilloise (<http://www.lillemetropole.fr>), qui nous a permis d'assister aux réunions préliminaires dans lesquelles les élus locaux et les professionnels définissent les projets à réaliser, leurs enjeux majeurs, ainsi que la

démarche nécessaire à leurs réalisation. Cette étape nous a permis de mettre en avant la diversité des domaines impliqués simultanément dans ce type d'activité, ainsi que les différents documents servant de ressource à la collaboration. La seconde étape d'analyse, plus concrète, s'est déroulée auprès de l'UFR de géographie de l'université de Lille 1. Cette observation a pris la forme d'une collaboration avec les enseignants responsables de l'UFR, nous permettant de procéder au suivi d'un ensemble d'étudiants de master 2 issus de diverses spécialités de la mention AUDT (Aménagements, Urbanisme et Développement des Territoires) travaillant en équipe sur leur projet de fin d'études.

5.4.2 L'urbanisme : un domaine complexe, propice à l'utilisation de la war room

L'une des caractéristiques majeures d'un projet d'urbanisme réside dans la diversité des domaines de compétences qui sont mis en oeuvre. En effet, chaque projet nécessite l'intervention de diverses personnes possédant des spécialités très différentes. De plus, l'interconnexion qui apparaît entre les travaux réalisés par ces différents spécialistes est élevée, et chaque décision prise par l'un des participants peut avoir un impact fort sur le travail des autres collaborateurs. Chaque projet doit par conséquent être abordé de façon à prendre en compte une multitude de facteurs, tels que :

- Le coût de réalisation
- L'aspect esthétique du résultat final (mobilier urbain, végétation, etc.)
- Le respect des arrêtés en vigueur définissant les règles d'harmonisation des habitations d'un quartier, les contraintes de respect du COS (coefficient d'occupation du sol), etc.
- Le type de bâtiments constituant la zone (commerces, habitations collectives et individuelles, etc.)
- L'accessibilité des usagers (piétons ou motorisés)
- Le contexte économique et social de la zone (par exemple, à travers la mise en place d'un réseau de transports en commun suffisamment dense si le projet est relatif à une zone d'emploi)
- Les objectifs généraux du projet, décidés en amont (réduire l'intrusion des véhicules personnels dans le secteur, encourager l'utilisation des transports en commun, relancer l'activité économique, etc.)

Mais un projet de ré-aménagement urbain doit également tenir compte de contraintes plus concrètes, telles que la gestion des réseaux souterrains, la fluidité des infrastructures routières, l'obstruction de l'ensoleillement par les bâtiments, ou la stratégie relative à la disposition des éclairages artificiels.

Pour prendre un exemple concret, la réalisation d'un simple carrefour est une activité qui repose sur de nombreuses problématiques. En effet, ce projet implique de définir une structure fiable en terme de circulation routière. Cela nécessite de définir les règles de

circulation les plus pertinentes possible. Ces règles définissent entre autres la disposition de panneaux de signalisation ou de feux tricolores. L'alimentation électrique de ces feux tricolores constitue à son tour une problématique relative à la gestion du tracé des réseaux câblés. La gestion des eaux grises (écoulement d'eau sur les chaussées) possède également un impact sur la conception de la route, et sa cohérence vis-à-vis des réseaux souterrains d'évacuation. De même, le tracé des réseaux souterrains doit conserver une parfaite cohérence vis-à-vis de l'aménagement routier, et rester disponible pour toute intervention. Les choix relatifs au mobilier urbain et à la végétation sont également étroitement liés à tous ces domaines : il est nécessaire de s'assurer que le type et la disposition de ces mobiliers n'entrave par la circulation ni la bonne visibilité des panneaux de signalisation. Chacune de ces problématiques est prise en charge individuellement par un ensemble de spécialistes, plus à même de prendre des décisions pertinentes. Cependant, il est difficile d'assurer la coordination de tous ces spécialistes et d'éviter les conflits dans les propositions.

Étant donnée la grande complexité des facteurs potentiels à prendre en compte pour la réalisation d'un projet d'urbanisme, et étant donné le caractère unique de ces contraintes relatives à chaque projet, il ne peut exister de solution universelle permettant de structurer les tâches de chacun de façon à assurer la cohérence de toutes les décisions entre elles. Pour cette raison, le contrôle de cette cohérence est confié à une infrastructure appelée BET (Bureau d'Etude Technique). Ce bureau d'étude procède à l'analyse détaillée des solutions proposées dans les différents domaines d'expertises, et détecte les conflits entre ces solutions. Le projet est alors remis dans les mains des experts, qui doivent trouver un compromis acceptable pour chaque conflit survenant au cours de la réalisation du projet. Le projet est par conséquent soumis à une évolution incrémentale dont chaque passe vise à corriger les incohérences détectées suite à la passe précédente. Notre intervention dans ce contexte vise à fournir à l'équipe d'experts un support de collaboration utilisant la réalité virtuelle, leur permettant de proposer des solutions et de contrôler en temps réel leur impact sur le projet global.

Le second point qui attire notre attention dans la réalisation d'un projet d'urbanisme réside dans la diversité des approches de chaque spécialiste. En effet, il n'existe pas de représentation universelle permettant de répondre aux besoins de chaque domaine d'expertise. Un participant travaillant sur l'aspect géographique de la zone sera plus à même d'interagir sur un document technique de type SIG (Système d'Information Géographique), tandis qu'une autre tâche plus orientée sur la volumétrie de la zone à ré-aménager (un paysagiste, par exemple) sera plus susceptible d'observer l'environnement sous la forme d'une représentation 3D.

De même, la discussion avec les élus locaux qui valident la proposition nécessite une représentation plus pédagogique du projet. En effet, la visualisation de la proposition sous la forme de plans techniques ne s'avère pas assez intuitive pour être facilement perceptible par tous les intervenants. Une représentation 3D réalisée à posteriori ne possède pas, quant-à-elles, les facultés d'interactivité suffisante pour une réelle collaboration autour du projet. La diversité des approches qu'apporte notre plateforme à travers le système de canaux d'interaction s'avère par conséquent particulièrement adaptée à cette diversité des domaines d'expertise, tout en offrant une interactivité capable d'optimiser la collaboration.



FIGURE 5.22 – Exemple de maquette réalisée dans le cadre d’un projet de réaménagement urbain

5.4.3 Observation d’un projet concret

La phase d’observation du groupe d’étudiants nous a permis d’extraire un modèle de workflow significatif pour la réalisation d’un projet de ré-aménagement. Cette équipe, massivement pluridisciplinaire, avait pour objectif de prendre en charge la totalité d’un projet d’urbanisme, à la manière d’un bureau d’étude.

Dans un premier temps, les urbanistes procèdent à une analyse de l’existant. Ils répertorient toutes les caractéristiques du projet, à travers les données statistiques significatives (COS, types de surfaces et proportionnalités de celles-ci sur la zone, etc.) et les photographies prises directement sur le site. L’équipe analyse alors ces données, de façon à extraire oralement leurs objectifs vis-à-vis du projet.

Cette étape du projet est essentiellement constituée d’activités de fouille de données et de communication interpersonnelle. Elle ne s’avère par conséquent que peu dépendante d’activités d’interaction.

5.4.3.1 Phase de production

Vient alors une phase de travail de conception qui s’avère plus orientée sur la production. Il est ici question de définir la manière dont les objectifs s’inséreront dans le contexte de la zone géographique existante.

Pour ce faire, les collaborateurs vont procéder à un maquettage de leur proposition (fig. 5.22). Cette maquette physique constitue alors l’objet central de leur collaboration. Elle



FIGURE 5.23 – Photographie aérienne servant de support à la réalisation de la maquette

aura pour rôle de servir de support visuel permettant à chaque spécialiste de présenter ses propositions aux autres collaborateurs, et de discuter de la pertinence de chacune de ces propositions. La construction concrète du projet prend place dans cette phase du travail, et est sujette à d'importantes activités de collaboration, tant en terme de communication interpersonnelle qu'en terme d'interaction sur la maquette.

La réalisation d'une telle maquette s'avère cependant à l'origine de tâches de création fastidieuses qui constituent un important frein à la fluidité de la collaboration. Notre plateforme d'interaction prend par conséquent place dans ce contexte, fournissant un outil de réalité virtuelle permettant à l'équipe de se focaliser uniquement sur la conception du projet.

La réalisation concrète d'une maquette d'urbanisme a lieu à partir d'un document de référence : une photographie aérienne de la zone. Cette photographie sert de support à la maquette, et fournit une représentation persistante de l'état actuel de la zone géographique (fig. 5.23).

A partir de ce repère, imprimé à l'échelle de la maquette, les urbanistes réalisent sur calque le tracé des plans techniques de type SIG présentant les surfaces qui prennent place dans la proposition (surfaces imperméables correspondant aux zones goudronnées, surfaces correspondant aux espaces végétalisés, ou zones d'habitation) (fig. 5.24). Ces différentes surfaces sont utilisées comme repères pour construire la volumétrie du projet.

l'étape suivante consiste en la réalisation des éléments tels que les bâtiments ou les arbres (fig. 5.25). En fonction des besoins et de l'échelle de la zone à réaménager, le niveau de détail de ces miniatures peut être différent. En effet, la majorité des bâtiments sont représentés par de simples cubes. Cependant, pour la prise en compte des bâtiments particuliers

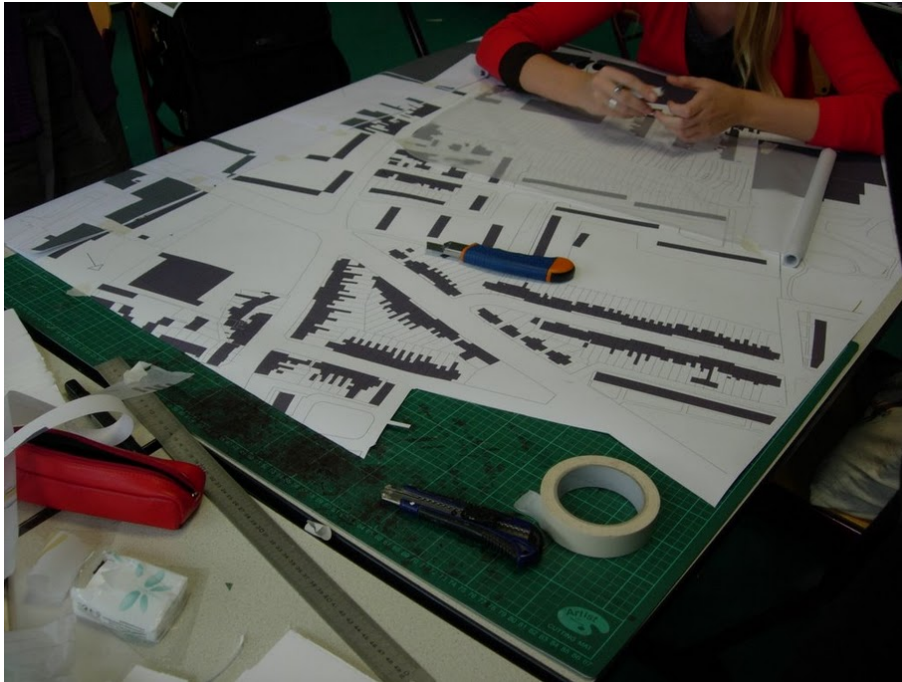


FIGURE 5.24 – Tracé des surfaces significatives de la maquette



FIGURE 5.25 – Construction de l'aspect volumétrique de la maquette

(bâtiment historique, architecture distinctive, statue, etc.), il est parfois nécessaire de représenter ces bâtiments de façon plus significative et plus détaillée.

Une fois la maquette réalisée, une légende peut être définie pour caractériser les types de

bâtiments ou les types de surfaces.

5.4.3.2 La phase de maquettage informatique

Une fois validée par l'ensemble des collaborateurs, cette maquette peut céder la place à une représentation 3D, permettant ainsi de simuler l'éclairage de la zone, ou de réaliser une vidéo de type "visite virtuelle", grâce à laquelle on peut observer le site selon une vue subjective proche de celle d'un usager marchant dans la zone réaménagée.

Cependant, les outils de représentation 3D de type CAD (Computer Aided Design) disponibles à l'heure actuelle révèlent un degré de complexité trop élevé pour servir de support à la discussion, et être soumis aux interactions simultanées de plusieurs utilisateurs. Notre plateforme peut ainsi être utilisée pour réaliser un prototypage rapide du projet d'urbanisme, et offrir aux utilisateurs de modifier à tout moment l'un des choix réalisés au cours de l'activité de conception.

5.4.4 L'installation matérielle mise en oeuvre

La collaboration autour de la maquette, constitue un candidat intéressant pour mettre à l'épreuve la diversité des canaux d'interaction/visualisation qui peut prendre place dans notre solution logicielle. En effet, si le travail de ré-aménagement d'une zone géographique repose en partie sur la conception de plans techniques en deux dimensions, la prise en charge de la volumétrie de la maquette s'avère indispensable, dans la mesure où elle offre un support de discussion plus concret, dans lequel les participants peuvent se projeter virtuellement. Ce contexte d'utilisation illustre donc de façon précise l'hétérogénéité des approches nécessaires sur le même projet.

Pour supporter au mieux la mixité des représentations, nous avons donc procédé à une installation matérielle comprenant quatre types de dispositifs (fig. 5.26) :

- **Un poste de travail 2D multi-utilisateurs.** Cette station de travail est constituée d'un ordinateur fixe sur lequel est connecté un écran tactile. Cet écran (de type 3M 22") est doté de fonctionnalités multitouch permettant la détection de 20 points de contact. Il possède en outre une taille suffisamment confortable pour permettre l'interaction simultanée de plusieurs utilisateurs. Dans notre war room, ce poste de travail est par conséquent dédié à une activité multi-utilisateurs comme peut l'être une table tactile.
- **Un dispositif tactile mobile.** La war room est également équipée d'un autre type de poste dédié à l'interaction bidimensionnelle, à travers l'intégration de dispositifs mobiles multitouch. Il s'agit ici d'ordinateurs portables de type tablet-PC (Dell XT2), dont l'écran dispose d'une détection des interactions multitouch. Ce dispositif n'a pour but que de fournir un mode d'interaction tactile individuel, pouvant éventuellement offrir plus de confort et de mobilité aux utilisateurs dans la pièce.
- **Un dispositif de navigation 3D.** Cet ordinateur possède des caractéristiques similaires au poste de travail 3D utilisé dans le précédent prototype. En effet, il possède, outre sa puissance de calcul graphique 3D, un écran de grande taille.

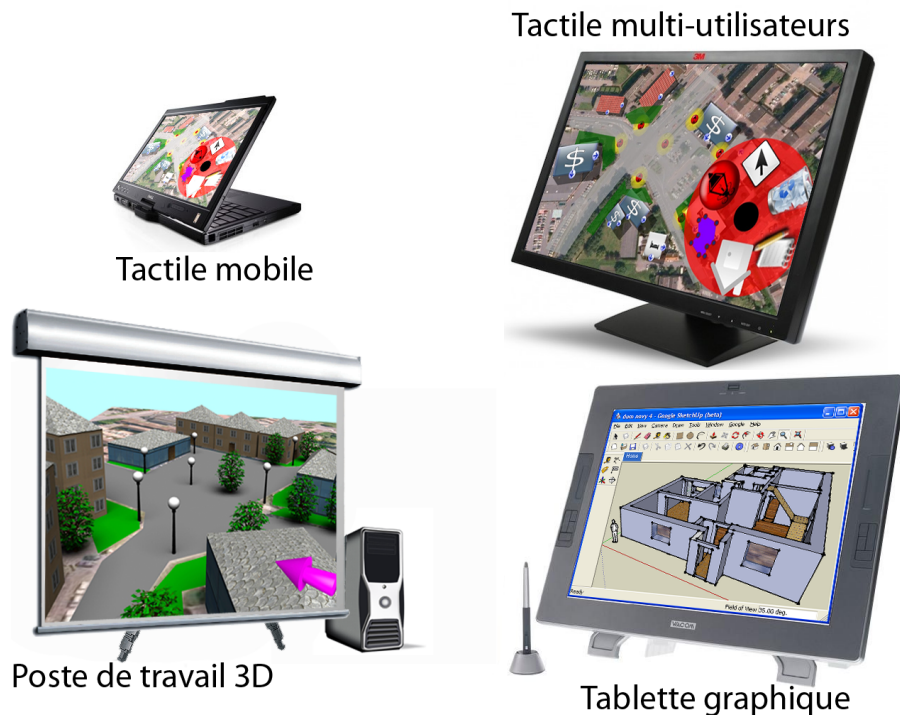


FIGURE 5.26 – Le matériel utilisé pour le démonstrateur d'urbanisme.

- **Un dispositif individuel de dessin.** Les postes de travail tactiles présents dans la war room offrent une interaction naturelle directement avec les doigts, qui permet aux utilisateurs de travailler rapidement et avec leurs deux mains. Cependant, cette modalité d'interaction reste relativement approximative, et peut s'avérer insuffisante lorsqu'il est question de réaliser un tracé précis. Pour combler ce besoin, nous avons doté notre war room d'un poste de travail individuel dédié à ce type de tâches. Pour ce faire, l'ordinateur en question est connecté à une tablette graphique (Wacom Cintiq 21U) permettant à l'utilisateur d'interagir à l'aide d'un stylet. Pour permettre un retour visuel direct, cette tablette graphique intègre une dalle LCD, fournissant ainsi à l'utilisateur une modalité d'interaction similaire au travail sur papier.

5.4.5 L'environnement virtuel partagé

L'environnement virtuel distribué entre les différents postes de travail revêt une forme plus complexe que celui du précédent démonstrateur. En effet, contrairement à l'exemple précédent, dans lequel l'environnement ne comportait qu'un seul objet (la voiture), il est ici question de fournir aux utilisateurs une maquette interactive d'une zone géographique. Par conséquent, l'environnement virtuel est ici constitué d'un ensemble d'objets distincts assemblés pour former la maquette. La singularité de ce contexte est basé sur le caractère non-prédéfini de la composition de cette maquette.

En effet, contrairement au précédent prototype, dans lequel la voiture était constituée d'un assemblage de sous-parties (entre autres, les portières) dont le nombre et le type étaient prédéfinis dans la configuration, la maquette virtuelle que nous proposons ici aux

urbanistes ne comporte aucune caractéristique pré-établie. Ainsi, le nombre, le type, le positionnement ou les paramètres intrinsèques des objets inclus dans cette maquette sont des facteurs libres qui ne dépendent que des choix de conception des collaborateurs. Au démarrage de l'application, l'environnement n'est donc constitué que d'un plateau vierge sur lequel l'équipe pourra ensuite agencer les objets qu'elle désire. Pour s'approcher au plus près des méthodes de travail habituelles des urbanistes, ce plateau est représenté sous la forme d'une photo aérienne du site à ré-aménager. Il constitue ainsi un repère visuel pour le placement des objets.

Les objets que les utilisateurs peuvent placer sur cette maquette sont multiples, et peuvent être associés à trois catégories :

-Les surfaces : Ce type d'objet a pour objectif de fournir aux collaborateurs la capacité de définir des zones sur la carte. Ces zones leur seront utiles pour définir les principaux types d'espaces tels que les espaces imperméables, les espaces végétalisés, ou les surfaces linéaires telles que les routes ou les pistes cyclables. Comme pour le travail sur SIG, ces surfaces sont constituées d'un ensemble de polygones, qui constituent le schéma général du projet. Elles sont par conséquent utiles à la définition du plan technique, mais servent également de repère visuel pour le placement des éléments en volume de la maquette.

-Les objets simples : Les objets dits « simples » sont des éléments dont les caractéristiques globales sont fixes. Les seuls paramètres interactifs relatifs à ces objets sont leurs coordonnées de positionnement et d'orientation dans l'espace. Dans le cadre de notre application, ces objets sont de diverse nature, en l'occurrence : station Velib', arbres, bancs publics, place de parking standard ou réservée aux personnes à mobilité réduite.

-Les objets complexes : Les objets complexes possèdent, contrairement aux objets simples, des caractéristiques qui vont au delà du simple positionnement dans l'espace. En effet, ces objets possèdent des paramètres supplémentaires dynamiquement modifiables que les utilisateurs pourront changer en fonction de leurs choix de conception. Dans notre application, cette famille d'objets est constituée des bâtiments et des éclairages publics.

Pour répondre à la nécessité de diversité dans les tâches d'interaction des urbanistes, notre plateforme est déployée dans la war room de façon à proposer trois canaux différents : un premier, exclusivement en 2 dimensions, un deuxième en 3D, et un troisième, 3D également, mais doté d'outils d'interaction spécifiques.

5.4.5.1 Le canal 2D

Pour respecter les méthodes de travail classiques des urbanistes, l'interface de conception principale est représentée à travers un canal 2D (fig. 5.27). Cette interface tend à se rapprocher d'une représentation technique de type SIG. Elle propose ainsi la carte de la zone sous la forme d'une vue aérienne. Les utilisateurs sont alors amenés à disposer les éléments directement sur cette carte, sous la forme d'icônes symboliques.

Ce canal d'interaction étant dédié à être exécuté sur les postes de travail multitouch, nous avons fait le choix de développer l'interface utilisateur sous la forme d'une application distincte. Nous nous sommes pour cela appuyés sur la technologie Microsoft .NET 4, qui



FIGURE 5.27 – Interface associée au canal 2D. (notez la présence du menu circulaire sur la droite)

offre un support natif des dispositifs d'interaction multitouch. Cette interface peut ainsi être exécutée sur n'importe quel ordinateur équipé du système d'exploitation Windows 7 et d'un dispositif matériel adapté. Cette technologie propose également une compatibilité avec les tables interactives de type Surface. L'interface du canal 2D est par conséquent développée dans le langage C#, et s'appuie sur l'API WPF (Windows Presentation Foundation), qui offre les fonctions de bases permettant la manipulation d'éléments graphiques.

Les modalités d'interactions multitouch sont ici conçues de manière à respecter une gestuelle tactile la plus naturelle possible. Ainsi, l'interface est dénuée de boutons tels qu'on peut les voir dans les interfaces WIMP (Window, Icon, Menu, Pointing device) classiques, et repose avant tout sur les mouvements opérés par les points de contact de la main sur l'écran. De cette manière, la totalité du champ visuel est disponible pour l'affichage de la maquette 2D. Étant donnée la diversité des dispositifs tactiles en terme de taille d'écran, l'affichage complet de la carte sur l'interface peut rendre l'interaction laborieuse lorsque les utilisateurs utilisent un écran de petite taille comme en sont équipés les tablet PC. Cependant, l'affichage de la carte dans sa totalité constitue une nécessité pour les utilisateurs, dans la mesure où il permet d'avoir une vision plus globale du SIG. Dans cette optique, l'interface 2D de la war room est dotée d'un système de navigation permettant de se déplacer sur la carte, mais aussi de procéder à des zooms. Cette tâche de navigation repose alors sur une modalité multitouch simple. Le glissement d'un doigt sur la surface de l'écran déplace le plateau de la maquette. De même, l'opération d'un zoom consiste à placer deux points de contact sur l'écran, et à définir le niveau de grossissement en modifiant l'écartement de ces points de contact.

Pour procéder au placement des différents éléments sur la maquette, nous avons pris le parti de proposer à l'utilisateur l'affichage d'une palette qui définit le mode de création. S'il reste envisageable d'instaurer une modalité impliquant le tracé d'un motif arbitraire sur la surface pour invoquer la création de l'élément associé, cette stratégie souffre de l'important nombre de types d'objets disponibles. Il serait en effet trop complexe pour l'utilisateur de

mémoriser le mouvement associé à chaque type d'objet. Pour cette raison, le processus de création se base sur l'utilisation d'un menu circulaire que l'utilisateur peut invoquer à tout moment : lorsque celui-ci appuie un doigt immobile sur l'écran, le menu circulaire apparaît sous le doigt en question (visible sur la figure 5.27). Ce menu est virtuellement attaché au doigt, et suit ce dernier dans tous ses mouvements, à la manière d'une palette de peinture. Il disparaît lorsque l'utilisateur relève le doigt concerné. Le contenu de ce menu circulaire propose un choix de modes d'interaction constitué de :

-un mode de sélection/manipulation. Ce mode est le mode par défaut de l'application. Il permet de naviguer sur la carte, et de manipuler les différents objets qui y sont présents. Il permet en outre d'éditer les surfaces via leurs points de contrôle.

-un mode de suppression. Ce mode permet de supprimer des objets de la carte, ou d'enlever des points de contrôles sur une surface.

-un mode de création de surfaces. Ce mode permet de définir le tracé d'une surface. L'interaction se déroule ici en plusieurs temps. Dans ce mode, chaque contact entre la main et l'écran entraîne la création d'un nouveau point de contrôle. Le polygone créé entre ces points constitue alors la surface. Un toucher sur le premier point de contrôle d'une surface valide la création de celle-ci. Cependant, le polygone reste éditable à tout moment, via la manipulation de ses sommets.

-un mode de création d'objets. Ce dernier mode permet de placer des objets simples ou complexes directement en touchant la carte.

La modification des propriétés d'un objet passe également par des interactions multitouch. Outre les informations de position, d'orientation et de dimension, qui peuvent être modifiées directement par manipulation directe de l'objet, la modification de données possédant un impact visuel moins direct sur l'interface 2D est possible par l'intermédiaire d'un objet graphique dédié. En effet, l'appui prolongé sur un objet complexe ou une surface entraîne l'ouverture d'un panneau de propriétés. Ce panneau présente alors un ensemble d'éléments graphiques interactifs permettant de manipuler les données associées. Par exemple, le nombre d'étages d'un bâtiment est symbolisé par une barre verticale graduée sur laquelle l'utilisateur peut glisser son doigt (fig. 5.28).

Cette approche purement 2D sous forme de SIG permet de manipuler la majorité des éléments constitutifs du projet de ré-aménagement. Cependant, elle ne suffit à pas à répondre à toutes les problématiques de collaboration pluridisciplinaire. Notre plateforme dispose par conséquent de deux autres canaux disponibles sur les autres postes.

5.4.5.2 Le canal 3D global

Le second canal qui prend place dans notre war room repose sur une approche volumétrique du projet d'urbanisme. Cette représentation vise à fournir aux participants une approche de leur travail visuellement plus proche de ce à quoi ressemblera la zone géographique après le travail de ré-aménagement. Sans pour autant chercher une représentation photo-réaliste de l'environnement, elle offre un point de vue plus accessible sur le projet (fig. 5.29). Ainsi, cette interface 3D est développée selon trois objectifs :



FIGURE 5.28 – Surligné de vert : le contrôle du nombre d’étages d’un bâtiment sur l’interface 2D.

-D’abord, elle permet d’intégrer à l’activité collaborative l’intervention de participants dont le domaine de compétence n’est pas associé à une utilisation courante de plans techniques. C’est notamment le cas des élus locaux qui sont amenés à valider le choix du projet, et éventuellement à apporter leurs suggestions et requêtes. Cette interface 3D, plus intuitive, offre ainsi au commanditaire du projet la capacité de formuler ses idées et suggestions en interagissant directement sur la maquette virtuelle.

-La représentation 3D de la maquette offre à l’équipe de collaborateurs la capacité de détecter plus facilement les éventuelles incompatibilités dans leurs choix respectifs. Par exemple, si certains utilisateurs décident de placer un bâtiment sur la carte, une approche exclusivement basée sur un plan 2D ne permettra pas de distinguer des incohérences relatives à la collision de ce bâtiment avec un autre objet de la maquette. Cependant, si la placement de ce bâtiment est validé sur la vue 2D, il peut potentiellement être source de complications sur la proposition finale. En effet, si le bâtiment en question obstrue la visibilité de la route, cette contrainte ne sera pas directement visible. Avec une représentation 3D simultanée, les utilisateurs disposent de la capacité de se placer selon une vue subjective et d’identifier plus facilement cette contrainte de visibilité. De même, le nombre de logements défini pour un bâtiment représentant une habitation collective impacte de façon directe le nombre d’étages de celui-ci, et donc sa hauteur. Ce paramètre est susceptible de posséder un impact fort sur le reste du projet, mais n’est perceptible visuellement qu’à travers une représentation 3D de la zone. En effet, même si les experts ont la capacité d’évaluer cet impact directement à travers une représentation technique 2D, cette évaluation mentale entraîne une charge cognitive qui peut être évitée grâce à l’existence d’un retour visuel directement en 3D.

D’une manière générale, l’analyse d’une zone urbaine selon une vue subjective apporte plus rapidement les informations relatives à l’harmonisation du projet. Cette représentation subjective n’est cependant pas suffisante pour développer le projet, puisqu’elle propose une vision trop restreinte de la zone à ré-aménager. Dans le contexte présent, les approches 2D et 3D sont par conséquent complémentaires.

L’existence du canal 3D peut permettre en outre de procéder à des simulations de contrôle



FIGURE 5.29 – Interface 3D du prototype d'urbanisme.

qui ne sont possibles qu'à travers une reconstruction volumique de la zone. Par exemple, il est envisageable d'exploiter la représentation 3D pour analyser l'impact de l'agencement des bâtiments sur l'ensoleillement du secteur, en fonction de l'heure de la journée. De même, il est possible de mettre en place des simulation physiques visant à analyser les mouvements d'air et de s'assurer que la zone n'est pas soumise à des vents violents.

-La troisième raison justifiant l'intégration d'un canal 3D dans la war room est liée aux contraintes d'utilisation du canal 2D. La représentation 2D disponible selon le canal d'interaction tactile souffre de contraintes liées à son support matériel. En effet, les dispositifs d'interaction multitouch permettent une interaction naturelle et multi-utilisateurs, mais ne bénéficient malheureusement que d'une précision très approximative. Une modalité d'interaction utilisant les doigts ne permet pas de profiter d'une précision capable d'égaliser celle disponible via l'utilisation du traditionnel couple clavier-souris, ni même celle d'une tablette graphique. Si le système de zoom disponible dans le canal 2D permet d'améliorer la précision des interactions de l'utilisateur, il ne suffit pas à l'ajustement des éléments entre eux. L'intégration du canal d'interaction 3D permet de répondre à cette contrainte. En effet, le bénéfice apporté par la perception tridimensionnelle de la scène permet d'ajuster le positionnement des éléments de façon plus précise, via des périphériques de manipulation 3D.

La war room configurée pour les projets d'urbanismes dispose par conséquent d'un canal dédié à la représentation de la maquette sous la forme d'une vue 3D. Cette maquette virtuelle est également constituée d'un plateau similaire à celui du canal 2D, c'est à dire présentant une photographie aérienne de la zone. Les surfaces sont alors représentées par des ensembles de polygones disposés sur le plateau. Les objets, quant-à eux, sont affichés sous la forme de différents modèles 3D (fig. 5.29).

5.4.5.3 Le canal de modélisation 3D

Comme nous l'avons précisé plus haut, la plupart des bâtiments représentés lors des conditions réelles de constitution d'une maquette sont représentés de façon symbolique par de simples cubes. Cette méthodologie convient à la plupart des bâtiments à intégrer dans le projet. Cependant, lors d'un projet de ré-aménagement, le secteur existant est susceptible de comprendre un ou plusieurs bâtiments significatifs, dont la forme ou la structure nécessitent d'être respectées au mieux pour la lisibilité de la maquette (c'est par exemple le cas pour un bâtiment historique).

De plus, l'équipe d'urbanistes impliquée dans la réalisation du projet peut souvent être constituée de spécialistes en architecture immobilière. En effet, la proposition qui est soumise à l'issue de la collaboration peut s'articuler autour d'un bâtiment distinctif, dont la structure géométrique possède un impact fort sur les choix qui seront fait lors de la réalisation du projet. Pour harmoniser l'apparence esthétique de l'ensemble des propositions, il est par conséquent nécessaire de développer conjointement l'architecture de ce bâtiment et son environnement urbain. De même, l'intégration dans la maquette d'une structure de grande taille constitue un paramètre majeur dans le déroulement du projet. Par exemple, un aménagement comprenant l'intégration d'un pont nécessite de concevoir la structure de ce pont en fonction de l'environnement. L'existence du retour visuel fourni par le canal 3D présenté précédemment permet également à l'équipe de collaborateurs de prendre en compte la forme de ce pont afin d'éviter tout conflit (par exemple, une collision entre un bâtiment et le pont).

La plateforme telle que nous l'avons développée dans ce contexte propose par conséquent un canal 3D supplémentaire dédié aux architectes. Ce canal, essentiellement mono-utilisateur, est intégré à un poste de travail disposant d'une tablette graphique. Il est question ici de proposer des outils d'interaction qui n'impacte que partiellement la maquette virtuelle. En effet, ce canal de modélisation permet à un utilisateur de procéder à la modélisation complète d'un objet complexe. Au fur et à mesure de la modélisation, l'objet concerné est mis à jour dans les autres canaux, permettant ainsi d'intégrer le bâtiment directement dans la maquette.

Pour répondre aux besoins complexes liés à la phase de création d'un modèle 3D, nous avons fait le choix d'intégrer ce canal sous la forme d'une application externe existante : l'application de modélisation Google SketchUp. Ce logiciel offre en effet la possibilité de procéder au prototypage rapide d'un modèle 3D via l'utilisation d'outils d'interaction élémentaires tels que l'ajout de primitives ou l'extrusion de surfaces. L'approche proposée par SketchUp permet de travailler rapidement sans encombrer l'utilisateur des contraintes de topologies inhérentes à la modélisation 3D. L'utilisation de cette application en particulier plutôt qu'un autre logiciel de CAO comme Autodesk Autocad est justifiée par sa simplicité d'utilisation. En effet, l'objectif de ce canal n'est pas de fournir un outil complet permettant de réaliser la conception dans les moindres détails, mais au contraire de disposer d'une application de création rapide d'utilisation. Cette rapidité et cette facilité d'utilisation s'inscrit dans la dynamique de collaboration qui prend place dans une war room, et qui implique la nécessité d'avoir des échanges rapides entre les utilisateurs.

Pour intégrer SketchUp à notre plateforme, nous avons donc greffé à cette application un plugin développé grâce au langage Ruby. Ce plugin, lorsqu'il est exécuté par l'utilisateur,

invoque l'interface de communication CORBA de la DAI (gérée sous la forme d'une librairie dynamique C++) pour transmettre les données géométriques à la plateforme de la war room.

Il faut souligner que ce plugin n'est pas exécuté en tâche de fond de l'application, comme c'est le cas pour les autres canaux. En effet, la tâche de modélisation d'un objet 3D est une activité nécessitant de nombreuses actions intermédiaires, telles que l'ajout de primitives dédiées à l'application d'une opération CSG (Constructive Solid Geometry). Pour éviter de perturber l'environnement visuel du canal 3D avec ces primitives intermédiaires, nous avons par conséquent pris le parti de laisser le déclenchement de la mise à jour des données à la charge de l'utilisateur du canal de modélisation.

5.4.6 Distribution de la plateforme

5.4.6.1 Caractéristiques générales

Comme nous l'avons vu dans la section précédente, notre plateforme est, dans ce contexte d'utilisation, configurée pour proposer trois canaux : le canal 2D, le canal 3D, et le canal de modélisation 3D. Pour rendre les mécanismes de préservation de la cohérence des données les plus génériques possibles, nous avons tiré parti de l'architecture M_NV-O de notre plateforme, qui permet au logiciel de manipuler un objet à travers plusieurs niveaux d'abstraction. La structure logicielle peut être ici caractérisée par plusieurs points principaux :

- En terme de constitution des agents selon leurs composantes MVO, l'hétérogénéité des données qui entrent en jeu entraîne l'existence de deux niveaux d'abstraction pour chaque canal. Cela signifie que chaque objet perçu à travers un canal quel qu'il soit s'appuie ici sur deux composantes de prise en charge des données : le modèle concret d'une part, et le modèle abstrait d'autre part. Bien sûr, chaque objet perceptible à travers l'un des canaux d'interaction dispose d'une composante vue associée à ce canal.
- A l'inverse du démonstrateur de conception automobile, l'environnement virtuel est ici constitué d'une multitude d'objets. Par conséquent, l'architecture logicielle déployée dans la plateforme est ici constituée de plusieurs agents M_NV-O . Pour satisfaire les besoins nécessaires à l'existence des trois canaux présentés plus haut, chacun de ces objets virtuels est doté de différentes facettes, chacune de ces facettes étant associée à la représentation de cet objet dans un canal.
- De part la nature multi-dispositifs de la war room mise en place, les agents interactifs de l'application voient leurs différentes composantes distribuées entre les postes de travail disposés dans la pièce. Cette distribution fonctionne sur un principe analogue à la distribution détaillée dans la section traitant de l'application de conception automobile.
- Le canal de modélisation 3D, proposant d'interagir sur la maquette via l'application SketchUp, possède une certaine singularité vis-à-vis des autres terminaux. En effet, l'utilisation de ce canal n'est pas dédiée à la maquette virtuelle dans son ensemble, mais au contraire à un type d'objet particulier. Cela signifie qu'en terme de distribution des composantes logicielles, le poste de travail dédié à la tâche de modélisation 3D ne s'appuie que sur des composantes issues de ces objets, c'est à dire leur vue et leur(s) modèles. Les autres objets de l'environnement sont ainsi indépendants de ce canal. Par exemple,

un objet de type « banc public » disposé sur la maquette n'est pas éditable en terme de géométrie 3D. Il n'apparaît donc pas sur le canal de modélisation. Par conséquent, ses composantes logicielles ne possèdent aucune facette permettant sa représentation selon ce canal particulier. Le canal de modélisation est donc un canal autonome vis-à-vis de la plupart des objets de la maquette. Cette situation est caractéristique de la liberté de distribution des objets dans notre plateforme : la co-existence de plusieurs canaux dans la war room n'implique pas nécessairement la présence de tous les agents interactifs de l'environnement dans tous les canaux, que ce soit en terme de représentation visuelle ou en terme de stockage de données.

- En dehors du canal 3D standard, les interfaces utilisateurs des deux autres canaux sont ici représentées à travers deux applications externes à la plateforme. En effet, ce choix nous permet de définir pour chaque canal des outils d'interaction très spécifiques à la représentation visuelle. Le rôle de la plateforme logicielle de la war room est, dans ce contexte, de fournir un environnement interactif 3D pour le canal 3D, et surtout d'assurer un lien cohérent et persistant entre les interfaces de tous les terminaux disposés dans la pièce. Il serait cependant envisageable de développer le canal 2D directement dans la plateforme SPIN|3D. Dans le cas de ce démonstrateur, nous nous sommes tournés vers le choix d'une application externe pour le canal 2D dans le but de tirer parti des fonctionnalités multitouch natives de WPF4.
- Contrairement à l'application de conception automobile, la configuration de la plateforme telle qu'elle est formulée ici ne propose pas de déplacement dynamique des représentations (comme ce pouvait être le cas pour le modèle 3D de la voiture, qui pouvait être basculé entre l'ordinateur portable ou la station 3D). En effet, la représentation de l'environnement virtuel repose ici sur un affichage persistant de la maquette. Les objets ne peuvent ainsi pas être déplacés d'un ordinateur vers l'autre, dans la mesure où ils sont simultanément affichés à travers tous les canaux (en dehors du terminal correspondant au canal de modélisation, pour les raisons évoquées plus haut). Il n'est par conséquent pas nécessaire de basculer l'affichage d'un objet d'un ordinateur vers un autre.

5.4.6.2 Le déploiement des composantes et la configuration XML

L'environnement virtuel que nous proposons dans ce démonstrateur est distribué entre les différents postes de travail disposés dans la war room. Dans la mesure où l'environnement de travail est constitué de trois canaux, nous avons naturellement structuré les données de chaque objet en fonction de ces canaux. La distribution de ces composantes est réalisée selon le schéma illustré dans la figure 5.30. Ce schéma illustre la répartition des deux principaux types d'objets :

- Les objets standards, regroupant les objets manipulables classiques (bancs publics, arbres, ect..) ainsi que les surfaces.
- Les objets éditables, c'est à dire ceux dont la topologie du modèle 3D peut être modifiée par le biais du canal de modélisation.

Ces deux types d'objets, en terme de composantes M_NV-O , sont constitués légèrement différemment. En effet, les objets standards, dont la topologie n'est pas éditable via le canal

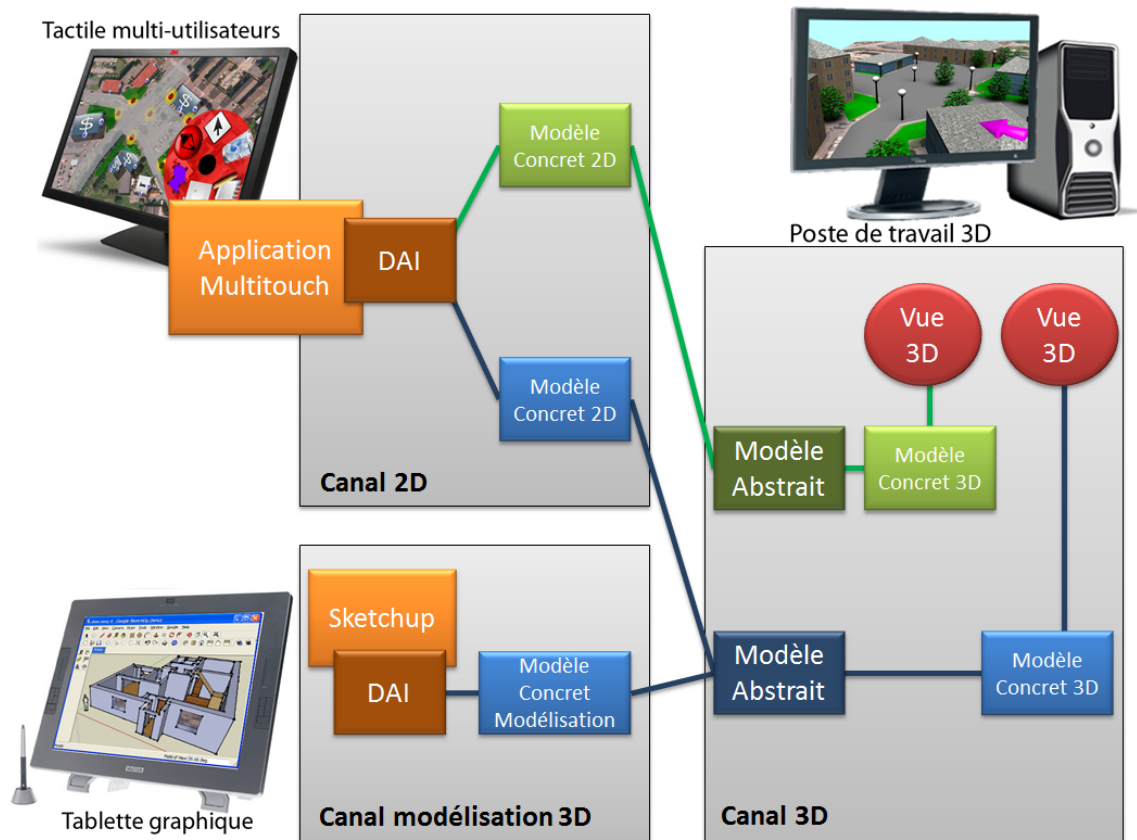


FIGURE 5.30 – Déploiement des composants pour le démonstrateur d'urbanisme. En bleu les objets éditables, en vert les objets standards.

de modélisation 3D, ne disposent pas d'une branche de données concrètes associée à ce canal. Ils possèdent en revanche toutes les composantes nécessaires à leur diffusion sur le canal de navigation 3D ainsi que sur le canal 2D. Les objets standards sont ainsi constitués de quatre composantes :

- Le modèle abstrait, définissant les informations génériques relatives à l'objet est pris en charge par le poste de travail 3D. Ce choix est purement arbitraire, puisque le modèle abstrait peut être déployé indifféremment sur n'importe quel ordinateur de la war room.
- Le modèle concret 3D, correspondant aux données visuelles de l'objet vis-à-vis d'une représentation tridimensionnelle. Dans notre exemple, ce modèle est également assigné au poste de travail 3D.
- Le modèle concret 2D, prenant en charge les informations visuelles de l'objet vis-à-vis de sa représentation dans l'environnement de type SIG. Cette composante est ici déployée sur le principal poste de travail multitouch.
- La vue 3D, permettant une représentation visuelle des informations prises en charge par le modèle concret 3D. Cette composante étant dédiée à l'affichage de l'objet sur le canal 3D, elle est assignée à la station de travail associée.

Il est à souligner qu'en dépit de l'existence de deux canaux distincts pour ces objets standards, seul l'un de ces canaux est attaché à une composante "vue". Cette absence est expliquée par le fait que le canal d'interaction 2D communique avec l'utilisateur par l'intermédiaire d'une application externe, rendant inutile l'exécution d'une telle composante sur les terminaux tactiles de la war room.

Les objets éditables, quant-à eux, possèdent une structure similaire, à la différence qu'ils sont constitué d'un modèle concret supplémentaire. Ce modèle concret est associé au canal de modélisation 3D, c'est-à-dire qu'il fournit à l'application sketchup des données géométriques se rapportant exclusivement à la topologie 3D de l'objet. Là encore, le placement de cette composante est libre, mais nous avons décidé de faire prendre en charge cette composante modèle au poste de travail dédié à ce canal de modélisation.

Pour mettre en place cette hiérarchie de composantes abstraites et concrètes au sein de notre plateforme, la constitution des fichiers de configuration XML a du être étoffée de plusieurs éléments. Pour une structure telle que celle de ce démonstrateur, plusieurs étapes sont nécessaires à la configuration de l'ensemble. Dans un premier temps, chaque objet de l'environnement virtuel doit être déclaré individuellement pour chaque canal. Par exemple, pour un bâtiment exploitant les trois canaux disponibles, l'objet doit être déclaré trois fois, comme s'il s'agissait de trois objets distincts. Chaque déclaration d'un objet peut donc être assimilée à la déclaration d'une couche concrète constituée d'un unique modèle et d'une vue.

L'exemple disponible sur le listing 2 illustre la manière dont est déclaré un objet 'lampadaire' exploitant deux canaux de l'environnement. Nous pouvons voir qu'un objet (widget) est défini par ses champs (ie. les données qui seront prises en charge par son modèle concret), par l'identifiant de son modèle, ainsi que par sa ou ses vues.

```

<widget name="UrbaLightW0">
  <fields>
    <field name="translation" type="SFVec3f" value="10.0_0.0_0.0" />
    <field name="rotation" type="SFRotation" value="0.0_1.0_0.0_0.0" />
    <field name="streetLight_scale" type="SFVec3f" value="1.0_1.0_1.0" />
    <field name="type" type="SFInt" value="0" />
  </fields>
  <model name="UrbaLightWModel0" instance="InteractiveObjectConcreteModel_id"
    proxyinstance="Proxy_ConcreteModel_id" />
  <views active="local">
    <view name="local" instance="InteractiveObjectView_id">
      <params>
        <param name="obj" value="wrl::urbaLampadaire_1" />
        <param name="links" value="['streetLight_Scale']*
          Lampadaire:streetLight_Scale" />
        <param name="localUserColor" value="0.0_0.0_0.9" />
        <param name="remoteUserColor" value="0.65_0.5_0.9" />
      </params>
    </view>
  </views>
</widget>

<widget name="2DUrbaLightW0">
  <fields>
    <field name="translation" type="SFVec2f" value="0.0_0.0" />
    <field name="rotation" type="SFFloat" value="0.0" />
    <field name="radius" type="SFFloat" value="1.0" />
    <field name="type" type="SFInt" value="0" />
  </fields>
  <model name="2DUrbaLightWModel0" instance="
    InteractiveObjectConcreteModel_id" proxyinstance="
    Proxy_ConcreteModel_id" />
  <views active="inactive">
    <view name="inactive" instance="InteractiveObjectView_id"/>
  </views>
</widget>

```

Listing 2 – Exemple de déclaration XML d'un lampadaire exploitant le canal 3D et le canal 2D

Après avoir défini les différentes facettes concrètes de chaque objet, le concepteur est amené à configurer les différentes couches d'abstraction qui sont nécessaires au bon fonctionnement de la plateforme. Pour ce faire, un élément 'superwidgets' est intégré au fichier de configuration (listing 3). Déclarer un élément "superwidget" équivaut à déclarer un modèle abstrait, qui se positionne alors comme père des objets concrets qui viennent d'être définis. Cette étape permet de définir les données abstraites à prendre en charge par le modèle, ainsi que les modèles concrets qui lui sont attachés (via l'élément 'concreteWidgets').

Dans ce fichier de configuration, nous pouvons d'ailleurs remarquer la présence d'un attribut "adapter" pour chaque modèle concret. Ces adaptateurs sont des classes dans lesquelles nous avons externalisé les traitements liés à la conversion des données. Ainsi, à chaque modèle concret est associé un adaptateur qui exécute les algorithmes de retranscription d'une donnée concrète en donnée abstraite, et vice-versa (fig. 5.31). Lors de la modification d'une des données d'un modèle concret, l'adaptateur qui lui est associé permettra la conversion de cette interaction sous la forme d'un événement abstrait que le modèle concret pourra transmettre à son modèle abstrait. De même, lorsque le modèle concret reçoit la notification d'un événement issu du modèle abstrait, cet événement est transmis à l'adaptateur pour obtenir en retour un événement applicable aux données concrètes.

Nous avons fait le choix d'implémenter ces algorithmes dans des classes externes de façon à pouvoir les réutiliser plus facilement d'une application à l'autre, et surtout pour apporter le moins de modifications possible à la classe "modèle" originale de la plateforme SPIN|3D.

```

<superwidgets>
  <superwidget name="UrbaLightSW0" instance="AbstractModel_id" proxyinstance=
    "proxy_AbstractModel_id">
    <fields>
      <field name="gps" type="SFVec2f" value="0.0_0.0" />
      <field name="range" type="SFFloat" value="1.0" />
      <field name="orientation" type="SFFloat" value="0.0" />
      <field name="type" type="CHAR" value="default" />
    </fields>
    <concretewidgets>
      <widget name="2DUrbaLightW0" adapter="Urba_Standard2DAdapter_id" />
      <widget name="3DUrbaLightW0" adapter="Urba_Standard3DAdapter_id" />
    </concretewidgets>
  </superwidget>
</superwidgets>

```

Listing 3 – Declaration d'un modèle abstrait commun à deux modèles concrets préalablement définis.

La dernière étape de la configuration consiste alors à définir le déploiement des composantes, comme c'était le cas pour le démonstrateur précédent. L'unique différence réside dans le fait qu'il est également nécessaire de définir le déploiement des 'superwidgets' définis ci-dessus, de façon à informer l'application du poste sur lequel doit être instancié le modèle abstrait, et des postes sur lesquels doivent être instanciés des modèles proxy.

5.4.6.3 Les données et leur signification

Les objets étant manipulés et affichés à travers des canaux différents, nous avons déterminé une définition abstraite de ces objets que nous avons placé à la charge du modèle abstrait. Ces informations abstraites ont ensuite été reformulées pour être interprétables graphiquement dans les différents canaux. Ce processus a donc donné naissance aux différents jeux de données qui sont intégrés dans chaque composante modèle. La liste ci-dessous établit un récapitulatif des champs et leur positionnement dans l'arbre des niveaux d'abstraction :

- **La position dans l'espace :** Chaque élément placé sur la maquette est défini notamment par sa position. Bien qu'il s'agisse d'une information concrète, il est nécessaire dans le modèle abstrait de définir cette position par rapport à un référentiel spatial précis, qui pourra être interprété efficacement par les différents modèles concrets. Le cadre de notre étude (la conception d'une maquette d'urbanisme) se rapportant à un espace du monde réel, l'encodage de la position des objets se prête facilement à l'utilisation de coordonnées GPS. Le modèle abstrait d'un objet possède donc une donnée "GPS", représentant la position générique de l'objet à travers une latitude et une longitude. Pour le modèle concret relatif au canal 2D, ces coordonnées GPS sont retranscrites en coordonnées dans l'espace 2D de l'application tactile, c'est à dire selon une unité comptée en pixels. Dans le même temps, les coordonnées GPS sont liées à un autre champ concret, relatif cette fois au canal de navigation 3D. La latitude et la longitude sont ici retranscrites sous la forme de coordonnées opengl en trois dimensions (fig. 5.31).

Il est à noter que dans le cas d'un objet complexe, c'est à dire tirant parti du canal

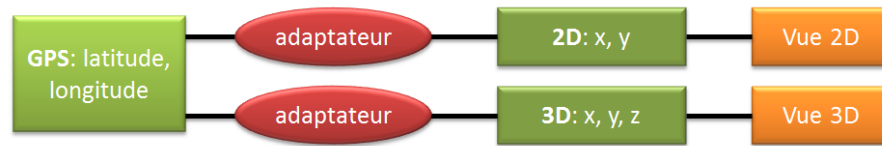


FIGURE 5.31 – Adaptation des données en fonction du canal : à gauche, la donnée abstraite.

de modélisation 3D, la position n'apparaît pas dans les données concrètes de ce canal. En effet, lors de l'édition de la topologie d'un objet, il n'est pas nécessaire de prendre en compte l'information de position. L'objet est donc naturellement placé au centre du repère.

- **L'orientation** : Il s'agit, au même titre que la position, d'une donnée élémentaire dans le cadre du maquettage. Ici encore, l'orientation est donnée par le modèle abstrait par rapport à un repère terrestre, et est convertie en angle selon un nouveau repère en fonction des modèles concrets qui lui sont liés. De même, pour le canal de modélisation 3D, l'orientation n'est pas à prendre en compte, et n'apparaît donc pas dans le modèle concret associé.
- **Le type** : Cette caractéristique peut être définie pour tous types d'objets. Elle constitue le type de l'objet, au sens large. Par exemple, pour un objet représentant une place de parking, le type permet de définir s'il s'agit d'une place classique ou dédiée aux personnes à mobilité réduite. De même, la donnée "type" peut définir pour un bâtiment s'il s'agit d'un logement collectif, individuel, ou d'une construction commerciale. Pour une surface, ce paramètre définit s'il s'agit d'un espace végétalisé, d'une surface imperméable, ou d'un point d'eau. Au niveau abstrait, cette information est codée sous la forme d'une chaîne de caractères.
 Au niveau des modèles concrets, le type est symbolisé par un nombre entier pour le canal 2D, tandis que pour le canal 3D, sa représentation est dépendante de l'objet virtuel. En effet, dans un environnement 3D, le type de l'objet peut se traduire par une topologie 3D, par une texture, ou par ses dimensions. Changer le type d'un bâtiment se traduit donc en 3D par la modification de sa texture, tandis que le type d'une surface est représenté dans le même canal par un code couleur.
- **Le nombre d'étages** : Cette information est propre aux bâtiments. Dans le cadre du canal 2D, la définition du nombre d'étages d'un bâtiment ne correspond qu'à la définition d'un nombre entier. Cependant, dans le canal 3D, cette information est directement liée à la hauteur du bâtiment en question. Ainsi, le nombre d'étages présenté sous la forme d'un entier dans le modèle abstrait est associé, dans le canal 3D, à la coordonnée 'Y' des dimensions du bâtiment 3D.
- **La portée** : Cette donnée est caractéristique d'un objet en particulier, le lampadaire. En effet, les utilisateurs sont amenés à définir les éclairages publics mis en oeuvre dans leur maquette. Dans le canal 2D, nous avons choisi d'offrir la possibilité de définir la

portée lumineuse d'un lampadaire à travers la manipulation d'un halo circulaire autour de l'objet. D'un point de vue visuel, la donnée que l'utilisateur manipule alors n'est qu'un nombre réel correspondant au rayon de ce halo lumineux. Cette valeur est propre à ce canal, et est bien sûr inutilisable directement par le canal 3D. Nous avons donc retranscrit cette portée dans le modèle abstrait sous la forme d'une valeur entière présentée par une unité arbitraire. Ainsi, le canal 3D peut s'appuyer sur cette information pour reconstruire une valeur de portée significative pour ce mode de visualisation. Dans notre application, nous avons pris le parti de représenter cette portée en 3D sous la forme d'une valeur représentant la hauteur du lampadaire. Dans le canal 3D, La portée influence donc, au même titre que le nombre d'étages pour les bâtiments, la coordonnées 'Y' des dimensions pour le lampadaire. Bien sûr, d'autres choix auraient pu être fait, notamment en appliquant à la maquette 3D une véritable simulation d'éclairage prenant en compte la portée des lampadaires.

5.4.7 Bilan

Ce second démonstrateur a fait l'objet d'une présentation au GestureCamp de Lille en 2010. La démonstration a permis d'illustrer un moyen de mettre en application des dispositifs aussi divers que les surfaces multitouch dans un contexte d'utilisation réel dédié à une activité collaborative. Les utilisateurs ont exprimé un vif intérêt pour la diversité matérielle de l'installation, et pour le système de canaux permettant d'élargir les possibilités en terme d'interaction et de visualisation. Plusieurs acteurs du domaine industriel ont également exprimé leur intérêt concernant la généricité de l'architecture logicielle mise en oeuvre, et la simplicité de sa configuration.

Plusieurs représentants de la communauté urbaine de Villeneuve d'Ascq ont également assisté à une démonstration de ce prototype. Le système d'adaptation des données a tout particulièrement retenu leur attention : l'utilisation d'une plateforme collaborative capable de manipuler et d'assurer la cohérence entre des données hétérogènes portant sur un même projet constitue selon eux un avantage significatif, en particulier pour permettre une meilleure collaboration entre des spécialistes ayant chacun leurs propres outils de travail.

Lors de cette mise en situation, nous avons pu illustrer la facilité de mise en place du déploiement. La conception de la distribution des différentes composantes via la configuration XML s'est avérée rapide et efficace pour répondre au cas d'utilisation réel.

Lors des tests réalisés en direct lors du salon, la plateforme s'est adaptée naturellement au contexte matériel, et s'est avérée fluide et robuste malgré l'utilisation d'une connexion réseau massivement sollicitée.

Concernant l'architecture logicielle, celle-ci a répondu correctement à nos attentes, et n'a pas nécessité la mise en place d'une hiérarchie complexe de niveaux d'abstraction puisque seuls deux niveaux (un abstrait et un concret) se sont avérés nécessaires. La tâche de conception relative au choix des données à confier à chaque modèle peut cependant s'avérer complexe pour une personne inexpérimentée. De même, l'intégration de l'interface de communication DAI au sein d'une application tierce sous la forme d'un plugin peut s'avérer complexe en terme d'implémentation, et peut éventuellement être impossible dans le cas d'applications propriétaires trop "fermées".

Le bilan de ce démonstrateur est positif puisqu'il a mis en évidence la validité de notre

proposition face à un scénario d'utilisation réel, et a été accueilli avec enthousiasme par plusieurs utilisateurs potentiels.

CONCLUSION

Sommaire

6.1	Résumé de la contribution	198
6.2	Questionnements et perspectives	200

6.1 Résumé de la contribution

Dans cette thèse, nous nous sommes intéressés aux particularités des situations de travail collaboratif colocalisé. Dans la première partie de ce travail réalisé en deux étapes, nous avons fait le choix d'aborder cette question en focalisant notre attention sur l'observation du comportement des utilisateurs confrontés à ce type de contexte.

L'analyse du rapport qu'entretiennent les collaborateurs avec la notion d'espace, que ce soit à travers leur communication interpersonnelle ou leur utilisation des supports de travail, nous a permis de mettre en évidence le caractère singulier de la situation de colocalisation. Cette singularité réside notamment dans le caractère "mixte" de l'espace. Contrairement aux plateformes de collaboration distante dans lesquelles l'espace de communication et d'interaction est exclusivement virtuel, l'utilisation d'une plateforme logicielle dans un contexte colocalisé scinde l'espace en deux notions concurrentes : l'espace physique d'une part, et l'espace virtuel d'autre part.

La coexistence de ces deux espaces au sein du même environnement ouvre des perspectives nouvelles quant-au contenu de l'espace virtuel, qui n'est plus soumis à l'uniformité stricte des applications de réalité virtuelle. De même, l'espace physique et tout particulièrement les dispositifs matériels ne sont plus contraints d'être similaires. Nous avons ainsi introduit l'idée de développer un environnement physique de type "war room" disposant d'un support informatisé.

Cette thèse met donc en avant une nouvelle approche de l'environnement de collaboration, qui met l'accent sur l'hétérogénéité. Cette hétérogénéité intervient au niveau :

- **matériel** : les dispositifs matériels disposés dans la war room peuvent être de types différents, et proposer ainsi des modes d'interaction et de visualisation hétérogènes.
- **humain** : les équipes qui sont amenées à collaborer dans la war room sont toutes différentes, de part leur nombre, leur organisation, et surtout la diversité de leurs domaines de compétences.
- **logiciel** : l'environnement virtuel proposé par la plateforme s'adapte à la diversité matérielle et humaine, et propose des modes de visualisation et d'interaction multiples couvrant les besoins individuels des collaborateurs.

Pour couvrir la diversité des besoins qui interviennent d'une session à l'autre, nous avons proposé le concept d'une plateforme logicielle généraliste pouvant être adaptée facilement à chaque situation. Cette plateforme permet une répartition dynamique des utilisateurs, la dissociation des dispositifs (ie. les périphériques d'interaction ne sont pas nécessairement liés à leur surface d'affichage), la distribution dynamique des représentation visuelles des objets, et la diversité des approches.

Pour conceptualiser cette diversité, nous avons introduit la notion de "canaux d'interaction et de visualisation", qui constituent les différents modes de représentation d'un objet virtuel au sein de la war room. Cette notion constitue la base de l'hétérogénéité logicielle de la plateforme, et permet aux collaborateurs issus de domaines de compétences différents de travailler simultanément sur le même objet virtuel à travers un environnement logiciel

plus proche de leurs besoins et habitudes.

La seconde partie de ce travail s'appuie sur les concepts et les contraintes extraits de la première partie, et consiste en une approche logicielle des problématiques liées à la collaboration colocalisée. Pour tirer parti des possibilités offertes par les solutions de collaboration distantes actuelles, nous proposons une méthode d'implémentation permettant d'adapter l'architecture multi-agents d'une plateforme collaborative existante : SPIN|3D, qui est basée sur une architecture MVO (modèle, vue, outil).

Notre proposition consiste alors à intégrer une interface de communication entre les différentes composantes des agents, permettant ainsi de dissocier les aspects "interaction", "visualisation" et "prise en charge des données" et de les distribuer entre les différents postes de travail de la war room. Pour répondre à la diversité des cas d'utilisation potentiels de notre plateforme et proposer une généricité optimale, cette interface de communication est "transparente" pour l'application et la distribution peut être personnalisée facilement par le concepteur.

Pour assurer le support des canaux d'interaction et de visualisation au sein de la plateforme, nous avons enrichi la structure MVO originale de SPIN|3D. La difficulté posée par l'aspect hétérogène des canaux reposant essentiellement sur la diversité des données et le maintien de la cohérence entre elles, nous avons introduit un système de gestion de plusieurs niveaux d'abstraction dans les données, à travers une structure que nous qualifions de M_NV-O (modèle abstrait, modèle(s) concret(s), vue, outil).

La nouvelle architecture que nous proposons pour SPIN|3D est donc constituée de plusieurs composantes "modèle" pour chaque agent logiciel (c'est à dire chaque objet virtuel) de l'application, organisés sous la forme d'un arbre dont la racine constitue la définition abstraite de cet agent. Chaque noeud fils du modèle abstrait représente donc une version plus concrète du modèle de l'objet. Chaque feuille de cet arbre est alors constituée d'un modèle concret dont les données sont directement représentables par une composante vue. Dans cette structure, chaque branche de l'arbre représente un canal à travers lequel les utilisateurs peuvent visualiser l'objet virtuel selon un mode particulier. La centralisation des canaux autour d'une même base abstraite permet de maintenir une cohérence globale entre ces canaux, et permet également d'intégrer dans l'environnement logiciel diverses applications tierces.

De plus, la nouvelle structure logicielle appliquée à SPIN|3D respecte la forme originale des agents, et permet de conserver les mécanismes de collaboration distante propres à la plateforme. La réutilisabilité du code est également maximisée par la structuration sous forme de composantes spécialisées pouvant être reprises directement dans d'autres cas d'utilisation.

Pour fournir aux utilisateurs de cette plateforme un environnement logiciel plus proche de leurs habitudes de travail, nous permettons également l'utilisation des multiples modèles concrets pour intégrer une application tierce à la war room. Une application quelconque peut donc être liée aux données d'un modèle concret dédié, permettant ainsi de maintenir une cohérence constante entre les données de la war room et celles de l'application tierce.

Nous avons poursuivi ce travail par une validation de la proposition consistant en la réalisation de deux prototypes logiciels visant à placer la plateforme développée dans des conditions d'utilisation réelle. Ces deux prototypes, respectivement dédiés à la conception automobile et à la collaboration autour de maquettes d'urbanisme, ont permis de mettre à l'épreuve l'interface de communication inter-composantes dédiée à la distribution, ainsi que la hiérarchisation des degrés d'abstraction relative à la prise en charge des canaux.

6.2 Questionnements et perspectives

La contribution que nous apportons dans cette thèse propose un environnement de collaboration nouveau de par sa nature hybride. Le fait de permettre la coexistence d'un environnement physique réel et d'un environnement virtuel, de même que la dynamique de distribution des informations qui est permise, peuvent donc soulever de nouvelles problématiques. De plus, même si la compréhension, la communication interpersonnelle ou le comportement des collaborateurs sont des questions déjà largement explorées dans l'étude des solutions de collaboration distante, la multiplicité des utilisateurs au sein de cet espace mixte peut faire réapparaître certains questionnements.

Etant donné que notre implémentation de la war room est non-destructive vis-à-vis de la communication distante disponible dans SPIN|3D, nous avons évoqué la possibilité de mettre en place une collaboration distante asymétrique. En terme de maintien de la "conscience sociale" entre les collaborateurs, la représentation des utilisateurs à distance est une problématique importante. Des solutions existent, consistant -entre autres- en l'affichage d'un avatar 3D (qu'il soit symbolique ou anthropomorphique) ou d'une fenêtre de visioconférence au sein de l'environnement virtuel. Cependant, ces solutions ne sont efficaces que lorsque l'on se trouve dans le cadre d'une collaboration symétrique dans laquelle chaque utilisateur est posté sur son propre terminal, et donc où chacun est représenté par son avatar personnel.

Dans une configuration proposant la connexion distante entre deux war rooms différentes peuplées par des équipes d'ampleurs différentes, le système de représentation distante peut avoir à représenter un utilisateur isolé ou une équipe de plusieurs personnes. La question soulevée par ce constat est "quel système de représentation proposer pour représenter un groupe?". Il est par exemple envisageable de ne proposer qu'un avatar par site, qui représenterait donc indifféremment un groupe ou un utilisateur isolé. Cependant, une personne confrontée à un unique avatar pourrait perdre la notion de collaborer avec un groupe, engendrant ainsi des incompréhensions. L'utilisation d'un système de visioconférence pourrait donc répondre à ce besoin, mais cette solution risque d'amplifier le phénomène d'isolement qui a lieu dans les situations de présence mixte. Une évaluation permettrait d'identifier le meilleur compromis.

Le fait de soumettre deux sites distants une situation asymétrique, à la fois au niveau du nombre de collaborateurs et au niveau des canaux de visualisation et d'interaction, soulève donc à nouveau les questionnements qui ont été levés pour les EVC classiques. La plateforme logicielle mettant en situation cette asymétrie étant désormais fonctionnelle, il serait intéressant de réaliser une série d'évaluations pour étudier l'impact d'un tel contexte

sur la communication interpersonnelle et la compréhension de l'activité. Ces évaluations pourraient mettre à jour de nouveaux comportements, et la plateforme de la war room pourrait constituer un support d'implémentation pour développer de nouvelles solutions permettant de minimiser l'impact de l'asymétrie sur la collaboration.

Lorsque l'activité est exclusivement colocalisée, la proximité physique des collaborateurs simplifie la perception de l'activité de chacun : Chaque poste de travail étant relativement spécialisé, il est possible à tout moment d'observer le positionnement des utilisateurs pour comprendre la nature de leurs interactions. Cependant, si la plateforme de la war room est configurée pour permettre l'utilisation d'un périphérique en tant que télécommande (ie. utiliser le périphérique d'un poste de travail pour interagir sur un objet affiché sur un autre), les informations récoltées par l'observation du groupe peuvent être faussées. Une solution pourrait être d'afficher sur les dispositifs un artefact visuel représentant un lien entre le périphérique et l'écran, à la manière de la plateforme table-centric de Wigdor et al. [WSFB06].

Les cas d'usage visés par le concept du travail en war room portent sur des travaux à réaliser sur une durée étendue. La tâche à accomplir par l'équipe de collaborateurs peut par conséquent être organisée sous la forme de plusieurs sessions de travail. De par la nature pluridisciplinaire des activités réalisables avec notre plateforme, une dynamique importante peut apparaître au niveau de la constitution de l'équipe. Ainsi, il est possible qu'un ou plusieurs participants se joignent à l'activité en cours de route.

Le niveau de spécialisation de ces participants peut alors engendrer la nécessité d'intégrer un nouveau canal -propre à ces utilisateurs- au sein de la plateforme logicielle de la war room. Ce cas de figure requiert une certaine dynamique dans l'enregistrement des canaux de visualisation et d'interaction.

Cette fonctionnalité est facilement implémentable dans notre plateforme en permettant aux composantes "vue" et "modèle-concret" de s'enregistrer auprès des modèles déjà existant de façon à pouvoir envoyer et recevoir les notifications des interactions. Le nouveau canal pourra donc être fonctionnel dès la connexion. Cependant, certaines situations liées à l'entrée d'un nouveau canal dans la war room peuvent s'avérer plus problématiques.

Par exemple, il est envisageable qu'un utilisateur se joignant à une session de collaboration en cours ait déjà effectué un travail individuel en amont, sur son poste de travail personnel. Il peut donc vouloir importer dans la war room les données issues de ce travail. La problématique majeure d'une telle situation réside dans le maintien de la cohérence globale des données de la plateforme : les données provenant du poste individuel doivent être fusionnées avec celles de la war room, sans remettre en cause la cohérence des canaux. Il est cependant impossible pour la plateforme de déterminer avec certitude la manière dont devront être fusionnés les deux jeux de données. La solution la plus naturelle serait donc de laisser le choix à l'utilisateur de décider s'il désire perdre ses modifications et de reprendre les données telles qu'elles sont dans la war room, ou s'il décide d'imposer ses modifications en "écrasant" les informations prises en charge par les autres postes de travail de la pièce.

D'une manière générale, la prise en charge du travail individuel est une fonctionnalité qui pourrait apporter des améliorations à l'activité dans la war room. Ainsi, de la même manière que pour un travail sur papier, un utilisateur pourrait décider de s'isoler provisoirement du reste du groupe pour travailler seul, et mettre au point une solution avant de la

soumettre au reste de son équipe.

Au niveau logiciel, cela pourrait se traduire par la "déconnexion" d'un poste de travail, c'est à dire la rupture provisoire des liens qui lie son canal au reste de l'application distribuée. Si cette fonctionnalité est possible dans certains cas (par exemple en coupant les liens établis entre le modèle concret du canal et son modèle abstrait), elle pose également le problème de la cohérence des données lors de la reconnexion : Les modifications apportées aux données durant la période de travail individuel auront rompu la cohérence entre les données du poste de travail et celles du reste de la war room.

Cette idée de rendre possible le travail individuel est cependant soumise à d'autres contraintes logicielles : pour qu'un poste de travail puisse travailler "en privé", c'est à dire de façon autonome sans connexion avec le reste de la war room, il est indispensable que ce poste dispose de toutes les composantes nécessaires au fonctionnement du canal, en l'occurrence une composante vue, et le modèle concret qui lui est associé. Si le poste de travail ne contient pas les composantes logicielles suffisantes pour une utilisation "hors ligne", il deviendrait nécessaire d'ajouter dans la plateforme la possibilité de réaliser une copie temporaire des composantes requises. Dans la mesure où les communications inter-composantes passent par un système de proxy et que ces proxys sont des classes dérivées des composantes classiques, il est envisageable qu'un modèle proxy importe les données du modèle réel pour y gagner un comportement autonome. Cette fonctionnalité pourrait donc être ajoutée à la war room sans remettre en question la structure logicielle de la plateforme.

Bibliographie

- [AC02] D. Armstrong and P. Cole. Managing distances and differences in geographically distributed work groups. In *Distributed work : New ways of working across distance using technology* (P. Hinds and S. Kiesler, editors, *MIT Press*, 2002).
- [All84] Thomas J. Allen. *Managing the Flow of Technology : Technology Transfer and the Dissemination of Technological Information Within the R&D Organization*. The MIT Press, January 1984.
- [ARH⁺05] J. Adams, B. Rogers, S. Hayne, G. Mark, J. Nash, and L. Leifer. The effect of a telepointer on student performance and preference. *Comput. Educ.*, 44 :35–51, January 2005.
- [Bar87] D. Barstow. Artificial intelligence and software engineering. In *Proceedings of the 9th international conference on Software Engineering, ICSE '87*, pages 200–211, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press.
- [BB04] Jacob T. Biehl and Brian P. Bailey. Aris : an interface for application relocation in an interactive space. In *Proceedings of Graphics Interface 2004, GI '04*, pages 107–116, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.
- [BDS⁺02] Boris B. Baltes, Marcus W. Dickson, Michael P. Sherman, Cara C. Bauer, and Jacqueline S. LaGanke. Computer-mediated communication and group decision making : A meta-analysis. *Organizational Behavior and Human Decision Processes*, 87(1) :156 – 179, 2002.
- [BF91] Eric A. Bier and Steven Freeman. Mmm : a user interface architecture for shared editors on a single screen. In *Proceedings of the 4th annual ACM symposium on User interface software and technology, UIST '91*, pages 79–86, New York, NY, USA, 1991. ACM.
- [BGBG95] Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, and Saul Greenberg, editors. *Readings in Human-Computer Interaction : Toward the Year 2000*. Morgan Kaufmann, San Francisco, CA, 2. edition, 1995.
- [BGG01] Kevin Baker, Saul Greenberg, and Carl Gutwin. Heuristic evaluation of groupware based on the mechanics of collaboration. In *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction, EHCI '01*, pages 123–140, London, UK, 2001. Springer-Verlag.

- [BHI93] Sara A. Bly, Steve R. Harrison, and Susan Irwin. Media spaces : bringing people together in a video, audio, and computing environment. *Commun. ACM*, 36 :28–46, January 1993.
- [BL97] M. Beaudouin-Lafon. *Interaction Instrumentale : de la manipulation directe à la réalité augmentée*. Actes des Neuvièmes Journées sur l'Interaction Homme Machine, IHM'97, Poitiers, Septembre 1997, Cépaduès-Éditions, 1997.
- [BL00] Hilary Bradbury and Benyamin M. Bergmann Lichtenstein. Relationality in organizational research : Exploring the space between. *Organization Science*, 11 :551–564, September 2000.
- [BL04] Michel Beaudouin-Lafon. Designing interaction, not interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 15–22, New York, NY, USA, 2004. ACM.
- [Bly88] Sara A. Bly. A use of drawing surfaces in different collaborative settings. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, CSCW '88, pages 250–256, New York, NY, USA, 1988. ACM.
- [BM08] Anastasia Bezerianos and Gregor McEwan. Presence disparity in mixed presence collaboration. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI '08, pages 3285–3290, New York, NY, USA, 2008. ACM.
- [BON⁺06] Nathan Bos, Judith Olson, Ning Nan, N Sadat Shami, Susannah Hoch, and Erik Johnston. Collocation bindness in partially distributed groups : is there a downside to being collocated ? In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 1313–1321, New York, NY, USA, 2006. ACM.
- [BOO95] Mathilde M. Bekker, Judith S. Olson, and Gary M. Olson. Analysis of gestures in face-to-face design teams provides guidance for how to use groupware in design. In *Proceedings of the 1st conference on Designing interactive systems : processes, practices, methods, & techniques*, DIS '95, pages 157–166, New York, NY, USA, 1995. ACM.
- [Bro00] Rupert Brown. Social identity theory : past achievements, current problems and future challenges. *European Journal of Social Psychology*, 30(6) :745–778, 2000.
- [BSO⁺04] Nathan Bos, N. Sadat Shami, Judith S. Olson, Arik Cheshin, and Ning Nan. In-group/out-group effects in distributed teams : an experimental simulation. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, CSCW '04, pages 429–436, New York, NY, USA, 2004. ACM.
- [BT91] Alan Borning and Michael Travers. Two approaches to casual interaction over computer and video networks. In *Proceedings of the SIGCHI conference on Human factors in computing systems : Reaching through technology*, CHI '91, pages 13–19, New York, NY, USA, 1991. ACM.
- [Bux92] William A. S. Buxton. Telepresence : integrating shared task and person spaces. In *Proceedings of the conference on Graphics interface '92*, pages 123–129, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
- [Car00] John M. Carroll. Making use : scenarios and scenario-based design. In *Proceedings of the 3rd conference on Designing interactive systems : processes,*

- practices, methods, and techniques*, DIS '00, pages 4–, New York, NY, USA, 2000. ACM.
- [CEG⁺87] Peter Cook, Clarence Ellis, Mike Graf, Gail Rein, and Tom Smith. Project nick : meetings augmentation and analysis. *ACM Trans. Inf. Syst.*, 5 :132–146, April 1987.
- [CFKL92] C. Cool, R. S. Fish, R. E. Kraut, and C. M. Lowery. Iterative design of video communication systems. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work, CSCW '92*, pages 25–32, New York, NY, USA, 1992. ACM.
- [CH84] Joëlle Coutaz and Mark Herrmann. *Adèle et le Médiateur-Compositeur ou Comment rendre une Application Interactive indépendante de l'Interface Usager*. Actes du deuxième colloque de Génie Logiciel (AFCET84), 1984.
- [Che53] E. Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the Acoustical Society of America*, 25(5) :975–979, 1953.
- [CHSH00] Trena C. Lilly Charles H. Sinex and Margaret A. Harlow. Using the war room process to explore network-centric warfare. *JOHNS HOPKINS APL TECHNICAL DIGEST*, 21(3) :368 – 377, 2000.
- [CNSD⁺92a] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, and John C. Hart. The cave : audio visual experience automatic virtual environment. *Commun. ACM*, 35 :64–72, June 1992.
- [CNSD⁺92b] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, and John C. Hart. The cave : audio visual experience automatic virtual environment. *Commun. ACM*, 35 :64–72, June 1992.
- [Cra02] C. Cramton. Attribution in distributed work groups. In *Distributed work : New ways of working across distance using technology* (P. Hinds and S. Kiesler, editors, *MIT Press*. 2002.
- [CSGCP99] M.-L. Viaud C.Dumas S.Degrande G.Saugis C.Chailou and P.Plenacoste. Spin :a 3d interface for cooperative work. *Virtual Reality Society Journal*, 1999.
- [DB96] Vial C. David B.T., Tarpin-Bernard F. Ergonomie du travail coopératif en conception. In *Actes de la conférence ERGO'IA 1996*, 1996.
- [DC01] Michael Dobbs and William D. Crano. Outgroup Accountability in the Minimal Group Paradigm : Implications for Aversive Discrimination and Social Identity Theory. *Personality and Social Psychology Bulletin*, 27(3) :355–364, 2001.
- [Dew99] P. Dewan. Architecture for collaborative applications. Computer Supported Cooperative Work. John Wiley Sons Ltd, 1999.
- [DF07] Mohamed Daassi and Marc Favier. Developing a measure of collective awareness in virtual teams. *Int. J. Bus. Inf. Syst.*, 2 :413–425, February 2007.
- [DFNA08] Thierry Duval, Cédric Fleury, Bernard Nouailhas, and Laurent Aguerreche. Collaborative exploration of 3d scientific data. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology, VRST '08*, pages 303–304, New York, NY, USA, 2008. ACM.

- [DL92] Paul DiZio and James R. Lackner. Spatial orientation, adaptation, and motion sickness in real and virtual environments. *Presence : Teleoper. Virtual Environ.*, 1 :319–328, July 1992.
- [DL01] Paul Dietz and Darren Leigh. Diamondtouch : a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 219–226, New York, NY, USA, 2001. ACM.
- [DPC⁺09] Nguyen-Thong Dang, Jean-Marie Pergandi, Franck Crison, Jerome Ardouin, and Daniel Mestre. Influence of orientation offset between control and display space on user performance during the rotation of 3d objects. In Michitaka Hirose, Dieter Schmalstieg, Chadwick A. Wingrave, and Kunihiro Nishimura, editors, *EGVE/ICAT/EuroVR*, pages 129–136. Eurographics Association, 2009.
- [DWM06] Declan Delaney, Tomás Ward, and Seamus McLoone. On consistency and network latency in distributed interactive applications : a survey—part i. *Presence : Teleoper. Virtual Environ.*, 15 :218–234, April 2006.
- [EE68] D. Engelbart and W. English. *A research center for augmenting human intellect*. AFIPS Fall Joint Computer Conference, 1968.
- [EGR91] Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. Groupware : some issues and experiences. *Commun. ACM*, 34 :39–58, January 1991.
- [ELW98] Robert Eckstein, Marc Loy, and Dave Wood. *Java Swing*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1998.
- [eOH93] C. Carlsson et O. Hagsand. Dive a platform for distributed multi-user virtual environments. In *Computer Graphics 17*, pages 663–669, 1993.
- [ESRF06] Katherine Everitt, Chia Shen, Kathy Ryall, and Clifton Forlines. Multispace : Enabling electronic document micro-mobility in table-centric, multi-device environments, 2006.
- [eTD02] D. Margery B. Arnaldi A. Chauffaut S. Donikian et T. Duval. Openmask : Multi-threaded or modular animation and simulation kernel or kit : a general introduction. In *Actes de VRIC02*, pages 101–110, 2002.
- [EW94] Clarence Ellis and Jacques Wainer. A conceptual model of groupware. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, pages 79–88, New York, NY, USA, 1994. ACM.
- [F97] Tarpin-Bernard F. *Travail coopératif synchrone assisté par ordinateur : Approche AMF-C*. Thèse de doctorat, Ecole Centrale de Lyon, France, Juillet 1997. 158 pages.
- [FKC90] Robert S. Fish, Robert E. Kraut, and Barbara L. Chalfonte. The videowindow system in informal communication. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, CSCW '90, pages 1–11, New York, NY, USA, 1990. ACM.
- [Fuc96] Philippe Fuchs. *Les interfaces de la réalité virtuelle*. Presses de l'école des mines de Paris, 1996.
- [Fuc06] Philippe Fuchs. *Le Traité de la Réalité Virtuelle*. March 2006.
- [GG98] Carl Gutwin and Saul Greenberg. Design for individuals, design for groups : tradeoffs between power and workspace awareness. In *Proceedings of the 1998*

- ACM conference on Computer supported cooperative work, CSCW '98*, pages 207–216, New York, NY, USA, 1998. ACM.
- [GG02] Carl Gutwin and Saul Greenberg. A descriptive framework of workspace awareness for real-time groupware. *Comput. Supported Coop. Work*, 11 :411–446, November 2002.
- [GKE⁺11] Tony Gjerlufsen, Clemens Nylandsted Klokmose, James Eagan, Clément Pillias, and Michel Beaudouin-Lafon. Shared substance : developing flexible multi-surface applications. In *Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11*, pages 3383–3392, New York, NY, USA, 2011. ACM.
- [GKEK04] Ana Cristina Bicharra Garcia, John Kunz, Martin Ekstrom, and Arto Kiviniemi. Building a project ontology with extreme collaboration and virtual design and construction. *Adv. Eng. Inform.*, 18 :71–83, April 2004.
- [GMA07] Stéphanie Gerbaud, Nicolas Mollet, and Bruno Arnaldi. Virtual environments for training : from individual learning to collaboration with humans. In *Proceedings of the 2nd international conference on Technologies for e-learning and digital entertainment, Edutainment'07*, pages 116–127, Berlin, Heidelberg, 2007. Springer-Verlag.
- [GNU97] T.C. Nicholas Graham, T. C. Nicholas, and Tore Urnes. Integrating support for temporal media into an architecture for graphical user interfaces, 1997.
- [GP02] Carl Gutwin and Reagan Penner. Improving interpretation of remote gestures with telepointer traces. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work, CSCW '02*, pages 49–57, New York, NY, USA, 2002. ACM.
- [Gru88] Jonathan Grudin. Why csw applications fail : problems in the design and evaluation of organizational interfaces. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work, CSCW '88*, pages 85–93, New York, NY, USA, 1988. ACM.
- [Gru94] Jonathan Grudin. Computer-supported cooperative work : History and focus. *IEEE Computer*, 27(5) :19–26, 1994.
- [Gru01] Jonathan Grudin. Partitioning digital worlds : focal and peripheral awareness in multiple monitor use. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '01*, pages 458–465, New York, NY, USA, 2001. ACM.
- [Gut98] Carl Andrew Gutwin. *Workspace awareness in real-time distributed groupware*. PhD thesis, Calgary, Alta., Canada, Canada, 1998. AAINQ31029.
- [Hal66] E.T. Hall. *Distances in Man : The Hidden Dimension*. Double Day, Garden City, NY, 1966.
- [HB99] Juan Pablo Hourcade and Benjamin B. Bederson. Architecture and implementation of a java package for multiple input devices (mid). Technical report, 1999.
- [HBR⁺94] Ralph D. Hill, Tom Brinck, Steven L. Rohall, John F. Patterson, and Wayne Wilner. The rendezvous architecture and language for constructing multiuser applications. *ACM Trans. Comput.-Hum. Interact.*, 1 :81–125, June 1994.

- [HCS06] Uta Hinrichs, Sheelagh Carpendale, and Stacey D. Scott. Evaluating the effects of fluid interface components on tabletop collaboration. In *Proceedings of the working conference on Advanced visual interfaces, AVI '06*, pages 27–34, New York, NY, USA, 2006. ACM.
- [HD96] Steve Harrison and Paul Dourish. Re-place-ing space : the roles of place and space in collaborative systems. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work, CSCW '96*, pages 67–76, New York, NY, USA, 1996. ACM.
- [HKR⁺05] Kirstie Hawkey, Melanie Kellar, Derek Reilly, Tara Whalen, and Kori M. Inkpen. The proximity factor : impact of distance on co-located collaboration. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work, GROUP '05*, pages 31–40, New York, NY, USA, 2005. ACM.
- [HM03] Elaine M. Huang and Elizabeth D. Mynatt. Semi-public displays for small, co-located groups. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '03*, pages 49–56, New York, NY, USA, 2003. ACM.
- [HMFG00] James D. Herbsleb, Audris Mockus, Thomas A. Finholt, and Rebecca E. Grinter. Distance, dependencies, and delay in a global collaboration. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work, CSCW '00*, pages 319–328, New York, NY, USA, 2000. ACM.
- [HP97] Edwin Hutchins and Leysia Palen. Constructing meaning constructing meaning from space, gesture, and speech, 1997.
- [IAHO10a] Zainura Idrus, Siti Z. Z. Abidin, R. Hashim, and N. Omar. Awareness in networked collaborative environment : a comparative study on the usage of digital elements. In *Proceedings of the 9th WSEAS international conference on Applications of computer engineering, ACE'10*, pages 236–241, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS).
- [IAHO10b] Zainura Idrus, Siti Z. Z. Abidin, R. Hashim, and N. Omar. Social awareness : the power of digital elements in collaborative environment. *W. Trans. on Comp.*, 9 :644–653, June 2010.
- [IBR⁺03] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, and Mia Underwood. Dynamo : A public interactive surface supporting the cooperative sharing and exchange of media. pages 159–168. ACM Press, 2003.
- [IKG93] Hiroshi Ishii, Minoru Kobayashi, and Jonathan Grudin. Integration of interpersonal space and shared workspace : Clearboard design and experiments. *ACM Trans. Inf. Syst.*, 11 :349–375, October 1993.
- [IM91] Hiroshi Ishii and Naomi Miyake. Toward an open shared workspace : computer and video fusion approach of teamworkstation. *Commun. ACM*, 34 :37–50, December 1991.
- [ITC08] Petra Isenberg, Anthony Tang, and Sheelagh Carpendale. An exploratory study of visual information analysis. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, CHI '08*, pages 1217–1226, New York, NY, USA, 2008. ACM.

- [JFW02] Brad Johanson, Armando Fox, and Terry Winograd. The interactive workspaces project : Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1 :67–74, April 2002.
- [JFW10] Brad Johanson, Armando Fox, and Terry Winograd. The stanford interactive workspaces project. In Saadi Lahlou, editor, *Designing User Friendly Augmented Work Environments*, Computer Supported Cooperative Work, pages 31–61. Springer London, 2010.
- [Joh88] Robert. Johansen. *Groupware : computer support for business teams / Robert Johansen; with contributions by Jeff Charles, Robert Mittman, Paul Saffo*. Free Press ; Collier Macmillan, New York : London :, 1988.
- [JOR⁺98] Lisa Covi Judith, Judith S. Olson, Elena Rocco, William J. Miller, and Paul Allie. A room of your own : What do we learn about support of teamwork from assessing teams in dedicated project rooms? In *in Dedicated Project Rooms? In : Proceedings of First International Workshop on Cooperative Buildings (CoBuild'98)*, pages 53–65. Springer-Verlag, 1998.
- [Kar94] A. Karsenty. Le collectifiel : de l'interaction homme-machine à la communication homme-machine-homme. In *Technique et Science Informatiques*, pages 105–127, 1994.
- [KBH⁺02] K. Kiyokawa, M. Billinghamurst, S. E. Hayes, A. Gupta, Y. Sannohe, and H. Kato. Communication behaviors of co-located users in collaborative ar interfaces. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, pages 139–, Washington, DC, USA, 2002. IEEE Computer Society.
- [KCSG03] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Saul Greenberg. How people use orientation on tables : comprehension, coordination and communication. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, GROUP '03, pages 369–378, New York, NY, USA, 2003. ACM.
- [Ken90] Adam Kendon. *Conducting Interaction : Patterns of Behavior in Focused Encounters (Studies in Interactional Sociolinguistics)*. Cambridge University Press, November 1990.
- [KFA⁺04] Azam Khan, George Fitzmaurice, Don Almeida, Nicolas Burtnyk, and Gordon Kurtenbach. A remote control interface for large displays. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 127–136, New York, NY, USA, 2004. ACM.
- [KP88] Glenn E. Krasner and Stephen T. Pope. A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *J. Object Oriented Program.*, 1 :26–49, August 1988.
- [LGS97] Alison Lee, Andreas Girgensohn, and Kevin Schlueter. Nynex portholes : initial user reactions and redesign implications. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work : the integration challenge*, GROUP '97, pages 385–394, New York, NY, USA, 1997. ACM.
- [LLO08] Chulhan Lee, Hohyun Lee, and Kyoungsu Oh. Real-time image-based 3d avatar for immersive game. In *Proceedings of The 7th ACM SIGGRAPH*

- International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '08, pages 48 :1–48 :2, New York, NY, USA, 2008. ACM.
- [LS92] M. Lea and R. Spears. Paralanguage and social perception in computer-mediated communication. *Journal of Organizational Computing*, 2 :321–341, 1992.
- [Lév90] P. Lévy. *Les technologies de l'intelligence*, Éditions La découverte. 1990.
- [M.90] Olson G. M. Collaborative Work as Distributed Cognition. *Unpublished Manuscript*, 1990.
- [MA04] G. Mark and S. Abrams. Sensemaking and design practices in large-scale group-to-group distance collaboration. In *ACM CHI 2004 Workshop on Designing for Reflective Practitioners*, 2004.
- [MAN03] Gloria Mark, Steve Abrams, and Nayla Nassif. Group-to-group distance collaboration : examining the "space between". In *Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 99–118, Norwell, MA, USA, 2003. Kluwer Academic Publishers.
- [Mar01] D. Margery. *Environnement logiciel temps-réel distribué pour la simulation sur réseau de PC*. PhD thesis, Rennes, France, 2001.
- [Mar02] Gloria Mark. Extreme collaboration. *Commun. ACM*, 45 :89–93, June 2002.
- [McA95] Daniel J. McAllister. Affect- and Cognition-Based Trust as Foundations for Interpersonal Cooperation in Organizations. *The Academy of Management Journal*, 38(1) :24–59, 1995.
- [McD95] D. M. McDonald. *Fixing blame in n-person attributions : a social identity model for attributional processes in newly-formed cross-functional groups*. St Lucie Press, Delray Beach, Florida, 1995.
- [MF98] Blair Macintyre and Steven Feiner. A distributed 3d graphics library. In *In Computer Graphics (Proc. ACM SIGGRAPH 98), Annual Conference Series*, pages 361–370, 1998.
- [MIEL99] Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Anthony La-Marca. Flatland : new dimensions in office whiteboards. In *CHI '99 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 346–353, New York, NY, USA, 1999. ACM.
- [mra92] A metamodel for the runtime architecture of an interactive system : the uims tool developers workshop. *SIGCHI Bull.*, 24 :32–37, January 1992.
- [MRM07] Gregor Mcewan, Markus Rittenbruch, and Tim Mansfield. Understanding awareness in mixed presence collaboration. In *Proceedings of the 19th Australasian conference on Computer-Human Interaction : Entertaining User Interfaces*, OZCHI '07, pages 171–174, New York, NY, USA, 2007. ACM.
- [Myn99] E.D. Mynatt. The writing on the wall. In *Proceedings of the ACM 2009 international conference on Supporting group work*, Proceedings of INTER-ACT'99 : Seventh IFIP Conference on Human-Computer Interaction, Laxenburg, Austria, 1999. International Federation for Information Processing.
- [ND86] Donald A. Norman and Stephen W. Draper. *User Centered System Design ; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986.

- [Nig94] Laurence Nigay. *Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales*. PhD thesis, Grenoble, France, 1994.
- [nyp] Nypd real-time crime center. <http://www.imtech.com/pdfs/imtechsnypd.pdf>.
- [OCR⁺98] Judith S. Olson, Lisa Covi, Elena Rocco, William J. Miller, and Paul Allie. A room of your own : what would it take to help remote groups work as well as collocated groups? In *CHI 98 conference summary on Human factors in computing systems*, CHI '98, pages 279–280, New York, NY, USA, 1998. ACM.
- [OW90] Olson G. M. Mack L. A. Olson, J. S. and P Wellner. User-Centered Design of Collaboration Technology. *Proceedings of Interact 90 - Third IFIP Conference on Human-Computer Interaction*, pages 835–840, 1990.
- [Pan64] R.R. Panko. Office work. office : technology and people, 1964.
- [Par03] Kyoung Shin Park. *Enhancing cooperative work in amplified collaboration environments*. PhD thesis, Chicago, IL, USA, 2003. AAI3111248.
- [Pat95] John F. Patterson. A taxonomy of architectures for synchronous groupware applications. *SIGOIS Bull.*, 15 :27–29, April 1995.
- [Paw06] *Multiple Mice for Computers in Education in Developing Countries*, 2006.
- [PF04] Ekaterina Prasolova-Forland. Supporting social awareness among university students in 3d cves : Benefits and limitations. In *Proceedings of the 2004 International Conference on Cyberworlds*, CW '04, pages 127–134, Washington, DC, USA, 2004. IEEE Computer Society.
- [PFD03] Ekaterina Prasolova-Førland and Monica Divitini. Collaborative virtual environments for supporting learning communities : an experience of use. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, GROUP '03, pages 58–67, New York, NY, USA, 2003. ACM.
- [PIC03] Stéphane LOUIS DIT PICARD. *Plate-forme de communication distribuée pour les Environnements Virtuels Collaboratifs 3D à fort couplage d'activité synchrone*. PhD thesis, Lille, France, 2003.
- [PMMH93] Elin Rønby Pedersen, Kim McCall, Thomas P. Moran, and Frank G. Halasz. Tivoli : an electronic whiteboard for informal workgroup meetings. In *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, CHI '93, pages 391–398, New York, NY, USA, 1993. ACM.
- [PST04] Thorsten Prante, Norbert Streitz, and Peter Tandler. Roomware : Computers disappear and interaction evolves. *Computer*, 37 :47–54, December 2004.
- [RE01] Hofer E.C. Herbsleb J.D. Rocco E., Finholt T.A. Out of sight, short of trust. In *Presentation at the Founding Conference of the European Academy of Management*, 2001.
- [RL04] Yvonne Rogers and Siân Lindley. Collaborating around vertical and horizontal large displays : Which way is best ? interacting with. *Computer*, 16 :2004, 2004.

- [Rob97] Toni Robertson. Cooperative work and lived cognition : a taxonomy of embodied actions. In *Proceedings of the fifth conference on European Conference on Computer-Supported Cooperative Work*, pages 205–220, Norwell, MA, USA, 1997. Kluwer Academic Publishers.
- [Rog97] Yvonne Rogers. Cognition and communication at work, engeström, y. and middleton, d. (eds). *Comput. Supported Coop. Work*, 6 :400–402, December 1997.
- [Rou00] Nicolas Roussel. *Support informatique à une communication médiatisée*. Thèse de doctorat, Université Paris-Sud, Orsay, France, Juillet 2000. 190 pages.
- [rrrb90] r, r, r, and b. Informal communication in organizations : Form, function, and technology. *Peoples reactions to technology in factories, offices and aerospace*, pages 145–199, 1990.
- [RV02] Leigh J Johnson A Park K Nayak A Singh R and Chowdhry V. Amplified collaboration environments. In *VizGrid Symposium*, Tokyo, 2002.
- [S.96] S. Bangay - *Modelling Parallel and Distributed Virtual Reality Systems for Performance Analysis and Comparison*, Thèse, Department of Computer Science, Rhodes University. 1996.
- [Sal95] Daniel Salber. *De l'interaction individuelle aux systèmes multiutilisateurs. L'exemple de la Communication Homme-Homme-Médiatisée*. PhD thesis, Grenoble, France, 1995.
- [SB92] Kjeld Schmidt and Liam Bannon. Taking cscw seriously : Supporting articulation work. *Computer Supported Cooperative Work*, 1 :7–40, 1992.
- [SB05] Wendy A. Schafer and Doug A. Bowman. Integrating 2d and 3d views for spatial collaboration. In *Proceedings of the 2005 international ACM SIG-GROUP conference on Supporting group work*, GROUP '05, pages 41–50, New York, NY, USA, 2005. ACM.
- [SBD99] Jason Stewart, Benjamin B. Bederson, and Allison Druin. Single display groupware : A model for co-present collaboration. pages 286–293. ACM Press, 1999.
- [SBF⁺87] M. Stefik, D. G. Bobrow, G. Foster, S. Lanning, and D. Tatar. Wysiwiw revised : early experiences with multiuser interfaces. *ACM Trans. Inf. Syst.*, 5 :147–167, April 1987.
- [SC95] Daniel Salber and Joëlle Coutaz. De l'observabilité et de l'honnêteté : le cas du contrôle d'accès dans la communication homme-homme médiatisée, 1995.
- [Sch70] Alfred Schutz. *On phenomenology and social relations; selected writings*. Edited and with an introd. by Helmut R. Wagner. University of Chicago Press Chicago,, 1970.
- [Sco05] Stacey D. Scott. *Territoriality in collaborative tabletop workspaces*. PhD thesis, Calgary, Alta., Canada, Canada, 2005. AAINR04617.
- [SD02] Edmondson A. Sole D. Situated knowledge and learning in dispersed teams. *British Journal of Management*, 13 :S17–S34(18), 1 September 2002.

- [SFB⁺87] Mark Stefik, Gregg Foster, Daniel G. Bobrow, Kenneth Kahn, Stan Lanning, and Lucy Suchman. Beyond the chalkboard : computer support for collaboration and problem solving in meetings. *Commun. ACM*, 30 :32–47, January 1987.
- [SGH⁺99] Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. i-land : An interactive landscape for creativity and innovation. In *CHI'99*, pages 120–127. ACM Press, 1999.
- [SGM03] Stacey D. Scott, Karen D. Grant, and Regan L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 159–178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.
- [Shn83] B. Shneiderman. Direct manipulation : A step beyond programming languages. *Computer*, 16 :57–69, August 1983.
- [SI03] S. D. Scott and K. M. Inkpen. Understanding children's collaborative interactions in shared environments. *Journal of Computer Assisted Learning*, 19 :220–228, 2003.
- [SK95] Hideyuki Suzuki and Hiroshi Kato. Interaction-level support for collaborative learning : Algoblockan open programming language. In *The first international conference on Computer support for collaborative learning*, CSCW '95, pages 349–355, Hillsdale, NJ, USA, 1995. L. Erlbaum Associates Inc.
- [SLV⁺02] Chia Shen, Neal B. Lesh, Frederic Vernier, Clifton Forlines, and Jeana Frost. Sharing and building digital group histories. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, CSCW '02, pages 324–333, New York, NY, USA, 2002. ACM.
- [Som69] R. Sommer. *Personal space : The behaviour basis of design*. Prentice-Hall, Englewood Cliffs, N.J. 1969.
- [Stu86] R. Stults. Media space, technical report, xerox parc,, 1986.
- [SUM⁺98] Greenberg S, M. Using, Room Metaphor, Ease Transitions, Saul Greenberg, Saul Greenberg, Mark Roseman, and Mark Roseman. Using a room metaphor to ease transitions in groupware. Technical report, In, 1998.
- [Taj78] Henri. Tajfel. *Differentiation between social groups : studies in the social psychology of intergroup relations / [edited by] Henri Tajfel*. Published in cooperation with European Association of Experimental Social Psychology by Academic Press, London ; New York :, 1978.
- [Tan91] John C. Tang. Findings from observational studies of collaborative work. *Int. J. Man-Mach. Stud.*, 34 :143–160, February 1991.
- [Tan04] Peter Tandler. The beach application model and software framework for synchronous collaboration in ubiquitous computing environments. *J. Syst. Softw.*, 69 :267–296, January 2004.
- [TAW90] H. Thimbleby, S. Anderson, and Ian H. Witten. Reflexive cscw : supporting long-term personal work. *Interact. Comput.*, 2 :330–336, November 1990.
- [TB94] T. Trowt-Bayard. *Videoconferencing : The Whole Picture*. Flatiron Publishing, 1994.

- [TBG04] Anthony Tang, Michael Boyle, and Saul Greenberg. Display and presence disparity in mixed presence groupware. In *Proceedings of the fifth conference on Australasian user interface - Volume 28*, AUIC '04, pages 73–82, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
- [TCKO00] Stephanie Teasley, Lisa Covi, M. S. Krishnan, and Judith S. Olson. How does radical collocation help a team succeed? In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, CSCW '00, pages 339–346, New York, NY, USA, 2000. ACM.
- [TG04] Edward Tse and Saul Greenberg. Rapidly prototyping single display groupware through the sdgtoolkit. In *Proceedings of the fifth conference on Australasian user interface - Volume 28*, AUIC '04, pages 101–110, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
- [TLGF09] Anthony Tang, Joel Lanir, Saul Greenberg, and Sidney Fels. Supporting transitions in work : informing large display application design by understanding whiteboard use. In *Proceedings of the ACM 2009 international conference on Supporting group work*, GROUP '09, pages 149–158, New York, NY, USA, 2009. ACM.
- [TM90] John C. Tang and Scott L. Minneman. Videodraw : a video interface for collaborative drawing. In *Proceedings of the SIGCHI conference on Human factors in computing systems : Empowering people*, CHI '90, pages 313–320, New York, NY, USA, 1990. ACM.
- [TM06] Goldie B. Terrell and D. Scott McCrickard. Enlightening a co-located community with a semi-public notification system. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, CSCW '06, pages 21–24, New York, NY, USA, 2006. ACM.
- [Weg97] Peter Wegner. Why interaction is more powerful than algorithms. *Commun. ACM*, 40 :80–91, May 1997.
- [Wel93] Pierre Wellner. Interacting with paper on the digitaldesk. *Commun. ACM*, 36 :87–96, July 1993.
- [WFDJ94] Steve Whittaker, David Frohlich, and Owen Daly-Jones. Informal workplace communication : what is it like and how might we support it? In *Proceedings of the SIGCHI conference on Human factors in computing systems : celebrating interdependence*, CHI '94, pages 131–137, New York, NY, USA, 1994. ACM.
- [WK00] Laurie A. Williams and Robert R. Kessler. All i really need to know about pair programming i learned in kindergarten. *Commun. ACM*, 43 :108–114, May 2000.
- [WSFB06] Daniel Wigdor, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. Table-centric interactive spaces for real-time collaboration. In *ACM AVI*, pages 103–107, 2006.
- [WSM⁺90] Kazuo Watabe, Shiro Sakata, Kazutoshi Maeno, Hideyuki Fukuoka, and Toyoko Ohmori. Distributed multiparty desktop conferencing system : Mermaid. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, CSCW '90, pages 27–38, New York, NY, USA, 1990. ACM.

- [YJsZ09] Teng Yingyan, Zheng Jun-sheng, and Gao Zhijun. Design and implementation of interactive 3d scenes based on virtools. In *Proceedings of the 2009 International Forum on Computer Science-Technology and Applications - Volume 02*, IFCSTA '09, pages 87–89, Washington, DC, USA, 2009. IEEE Computer Society.
- [ZG00] Ana Zanella and Saul Greenberg. *A Single Display Groupware Widget Set*. Number Section 2. 2000.