

UNIVERSITE JOSEPH FOURIER – GRENOBLE 1
U.F.R. EN INFORMATIQUE ET MATHEMATIQUES APPLIQUEES

THESE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER - Grenoble I

Discipline : Informatique

Présentée et soutenue publiquement

par

Frédéric Vernier

Le 21/02/2001

Titre :

**La multimodalité en sortie
et son application à la visualisation
de grandes quantités d'information**

Directeur de thèse : Laurence Nigay

JURY

Mr Michel Adiba,	Président
Mme Jocelyne Nanard,	Rapporteur
Mr Guy Melançon,	Rapporteur
Mme Joëlle Coutaz,	Examineur
Mr Giorgio Faconti,	Examineur
Mr Eric Lecolinet,	Examineur

Remerciements,

A celles et ceux qui ont contribué à l'aboutissement de ce document, à ma famille, à mes collègues et à tous mes amis. Tous les citer serait trop long mais comment pourrais-je ne pas remercier ... en tout premier Laurence qui m'a encadré pendant de nombreuses années, du magistère à la Thèse, toute l'équipe IIHM, Joëlle chef de l'équipe, Dan, Pépé, Eric, Dave, Manu, Gaëlle, Chaouki, Yann, Philou, Kris, Léon, Nick, Lisette, Julie et tous les autres ... tiens d'ailleurs j'en aurais pas oublié un ? ... Mais non Fanf, je ne vais pas oublier de te remercier de ta présence, au lab, rue Ampère, au bagatel, au transbo et à l'entrepôt mais aussi merci pour ton absence qui m'a permis de prendre le temps de rédiger ;-). Remercier les amis qui ne sont pas des collègues de travail serait encore plus long ... mais dans un mémoire de Thèse, il y a de la place. Merci donc à Delphine, Christine, Ninie (relectrice), Galou, Jean-Luc, Stella, Eric, Marie-Claude, Arnaud, Franck, Françoise, Jean-Philippe, Christèle, Mathias, Magalie, Olivier, Anne-Lyse, Nils, Alex, Doms, Laure, Nicole, Anne l'italienne, Anne la tahitienne, Céline l'américaine, Céline l'anglaise et Céline l'africaine (ma relectrice la plus acharnée) et puis tous les autres. Je voudrais enfin remercier ma famille qui m'a permis, que dis-je, poussé à faire des études et particulièrement à Papi Joe qui verra d'en haut la Thèse de son pillou. Merci aussi à mes parents, Hélène (ma plus fidèle supportrice) et André (mon plus fidèle relecteur), à ma sœur Isabelle, mon frère Ezana et à ma cousine Lili qui étudiait à Grenoble pendant ma Thèse et qui a participé à la relecture du manuscrit. A mes grands parents Mamie Paulo, Mami Simone et Papi Charlie. Aux cousins suisses, Marc, Barbara, Annick et Adrien. Aux cousins cannois, Annette, Olivier, Nino, Sydney, April, Nico et Françoise. Aux cousins montpelliérains, Flo, Thierry, Sophie, Violaine, Gabi, Pacôme et Valdo. Aux valentinois Joe et Françoise, ma marraine et enfin aux cousins bordelais.

Cette Thèse est un peu la vôtre.



Cette image a une longue histoire, je l'ai produite en mettant au point un de mes programmes. La plupart des personnes que je viens de remercier (et d'autres encore) ont joué le jeu et ont produit de telles grimaces (certaines sont de vraies œuvres d'art). Parce que ma Thèse fut aussi un plaisir et un amusement, voici l'adresse : <http://iihm.imag.fr/vernier/grimaces>. [;---]

La multimodalité en sortie et son application à la visualisation de grandes quantités d'information

CHAPITRE I INTRODUCTION	7
1 SUJET	9
2 OBJECTIFS	10
3 STRUCTURE DU MÉMOIRE	11
CHAPITRE II PATRONS DE CONCEPTION	15
4 HISTORIQUE	17
5 TERMINOLOGIE	20
5.1 Auteur et concepteur	20
5.2 Patron et langage de patrons	20
6 FORMAT DE PATRONS	22
6.1 Exemples illustratifs	25
6.2 Description du problème et du contexte	26
6.3 Motivations	26
6.4 Contre-exemple	26
6.5 Solution proposée	27
7 PRINCIPE D'ORGANISATION	27
7.1 Critères d'ergonomie mis en jeu par le patron	28
7.2 Choix des modalités préconisé par le patron	30
7.3 Niveaux d'abstraction du patron	30
8 CONCLUSION	30
CHAPITRE III MULTIMODALITÉ EN SORTIE	33
1 TERMINOLOGIE : MODALITÉ ET MULTIMODALITÉ EN SORTIE	35
1.1 Modalité	36
1.2 Multimodalité en sortie	38
2 CONCEPTION D'INTERFACES MULTIMODALES	42
2.1 Typologie de la multimodalité en sortie	42
2.2 Cadre de conception	46
2.3 Cadre de conception et patrons de conception	48
3 ESPACES DE CONCEPTION EXISTANTS	49
3.1 Espaces de composition de modalités	49
3.2 Espaces de caractérisation d'une modalité	55
3.3 Synthèse	59
4 NOUVEL ESPACE DE COMPOSITION DE MODALITÉS DE SORTIE	59
4.1 Aspect temporel d'une composition de modalité	62
4.2 Aspect spatial d'une composition de modalités	63
4.3 Aspect articulatoire d'une composition de modalités	65
4.4 Aspect syntaxique d'une composition de modalités	65
4.5 Aspect sémantique d'une composition de modalités	66
4.6 Aspects et schémas : un espace unificateur	67
5 CARACTÉRISATION DES MODALITÉS DE SORTIE	68
5.1 Caractérisation des dispositifs	68
5.2 Caractérisation des langages	69
5.3 Synthèse	72
6 PATRONS DE CONCEPTION	72
6.1 Patron N°1 : Test du dispositif	73
6.2 Patron N°2 : Préférence du canal le moins occupé	76
6.3 Patron N°3 : Adaptabilité des modalités de sortie	78

6.4	<i>Patron N°4 : Modalités complémentaires et simultanées</i>	83
7	CONCLUSION	85
CHAPITRE IV VISUALISATION D'UNE GRANDE QUANTITÉ D'INFORMATION		87
1	ESPACES DE CONCEPTION DES INTERFACES DE VISUALISATION	89
1.1	<i>Cadre d'étude de Ed. Chi</i>	90
1.2	<i>Taxonomie de Leung</i>	92
1.3	<i>Taxonomie de Shneiderman</i>	93
1.4	<i>Taxonomie de Bruley</i>	94
1.5	<i>Espace de conception de Stuart Card</i>	96
1.6	<i>Synthèse : un cadre d'étude fédérateur</i>	97
2	COMPOSITION DES MODALITÉS DE VISUALISATION	103
2.1	<i>Vues multiples</i>	103
2.2	<i>Composition du focus et du contexte</i>	105
3	CARACTÉRISATION DES TECHNIQUES DE VISUALISATION	106
3.1	<i>Caractérisation des points de vue sur les données</i>	106
3.2	<i>Caractérisation de l'espace</i>	110
3.3	<i>Caractérisation des points de vues sur l'espace</i>	113
4	PATRONS DE CONCEPTION POUR LES TECHNIQUES DE VISUALISATION	122
4.1	<i>Patron N°5 : Continuité spatiale des modalités</i>	123
4.2	<i>Patron N°6 : Continuité temporelle</i>	126
4.3	<i>Patron N°7 : Plaquage des données en traits graphiques</i>	129
4.4	<i>Patron N°8 : Compréhension de la déformation</i>	132
4.5	<i>Patron N°9 : Réutilisation des outils de navigation</i>	135
5	CONCLUSION	137
CHAPITRE V RÉALISATION LOGICIELLE		139
1	MOTIVATIONS	141
2	RÉALISATION DE VUES HYPERBOLIQUES	142
2.1	<i>Fonction de déformation</i>	143
2.2	<i>Application de la fonction aux systèmes de coordonnées</i>	145
2.3	<i>Manipulation du coefficient de déformation</i>	146
2.4	<i>Application de fonctions de déformation différentes sur chaque axe</i>	152
2.5	<i>Utilisation du paramètre R pour remplir des espaces rectangulaires</i>	152
3	ARCHITECTURE LOGICIELLE	153
4	DÉTAILS D'IMPLÉMENTATION	159
5	CONCLUSION	160
CHAPITRE VI VITESSE : VISUALISATION DES RÉSULTATS D'UN MOTEUR DE RECHERCHE DU WEB		163
1	INTRODUCTION	165
2	CONCEPTION DE L'INTERFACE	165
2.1	<i>Point de vue sur les données</i>	166
2.2	<i>Espace</i>	167
2.3	<i>Point de vue sur l'espace</i>	171
3	RÉALISATION LOGICIELLE	182
3.1	<i>Architecture logicielle</i>	182
3.2	<i>Implémentation</i>	186
4	CONCLUSION	188
4.1	<i>Bilan</i>	188
4.2	<i>Perspectives</i>	191
CHAPITRE VII MULTAB : VISUALISATION DE GRANDES TABLES		193
1	INTRODUCTION	195
2	CONCEPTION DE L'INTERFACE	196
2.1	<i>Point de vue sur les données</i>	197
2.2	<i>Espace</i>	197
2.3	<i>Point de vue sur l'espace</i>	198
3	RÉALISATION LOGICIELLE	205

3.1	<i>Architecture logicielle</i>	205
3.2	<i>Implémentation</i>	207
4	CONCLUSION	209
4.1	<i>Bilan</i>	209
4.2	<i>Perspectives</i>	210
CHAPITRE VIII PARENT : VISUALISATION D'UNE HIÉRARCHIE DE FICHIERS.....		213
1	INTRODUCTION	215
2	CONCEPTION DE L'INTERFACE	216
2.1	<i>Point de vue sur les données</i>	216
2.2	<i>Espace</i>	216
3	RÉALISATION LOGICIELLE	226
3.1	<i>Architecture logicielle</i>	226
3.2	<i>Implémentation</i>	227
4	CONCLUSION	229
4.1	<i>Bilan</i>	229
4.2	<i>Perspectives</i>	229
CHAPITRE IX CONCLUSION		231
1	CONTRIBUTION DE LA THÈSE	233
2	PERSPECTIVES	237
2.1	<i>Extensions</i>	237
2.2	<i>Prolongements</i>	238
RÉFÉRENCES		241
1	COMMUNICATIONS DE RECHERCHE RÉFÉRENCÉES	243
2	SYSTÈMES DE RÉFÉRENCE	254
ANNEXES		257
1	ANNEXE 1 : LISTE DES CRITÈRES ERGONOMIQUES DE L'ÉQUIPE IIHM [ABOWD 92][GRAM 94] ...	259
1.1	<i>Souplesse</i>	259
1.2	<i>Robustesse</i>	262
2	ANNEXE 2 : PRINCIPES ERGONOMIQUES DE DOMINIQUE SCAPIN ET CHRISTIAN BASTIEN [SCAPIN 93]	266
INDEX DES FIGURES		267

CHAPITRE I

INTRODUCTION

Chaque récit à ses règles. Isaac Asimov a introduit une collection entière d'ouvrages par ces trois phrases :

*"- **Première Loi** : un robot ne peut blesser un être humain ni, par son inaction, laisser un être humain exposé au danger.*

*- **Deuxième Loi** : un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la première loi.*

*- **Troisième Loi** : un robot doit protéger sa propre existence aussi longtemps qu'une telle protection n'est pas en contradiction avec la première ou la deuxième loi."*

1 ***Sujet***

Nos travaux s'inscrivent dans le domaine de l'Interaction Homme-Machine et ont trait à la conception d'interfaces multimodales en sortie pour manipuler de grandes quantités d'information.

Aujourd'hui, l'efficacité recherchée n'est plus celle de la technologie prise isolément mais celle du couple "machine-humain". Il ne s'agit plus seulement d'exploiter au mieux les ressources techniques, mais de développer une technologie qui soit conforme aux attributs de la communication humaine. Et l'être humain voit, entend, parle, gesticule, manipule. Dans ce contexte, la multimodalité en sortie (du système vers l'utilisateur) prend son sens aussi bien dans la variété des canaux d'interaction à destination de l'utilisateur que dans les diverses manières d'utiliser ces canaux. La mise en œuvre de la multimodalité en sortie se caractérise comme un processus de définition d'un énoncé multimodal qui doit être l'image des informations à présenter, adaptée au contexte d'interaction. La finalité de nos travaux est de contribuer à une meilleure prise en compte de la multimodalité dans la conception des interfaces en sortie dédiées à la manipulation de grandes quantités d'information.

La conception de techniques d'interaction qui permettent aux utilisateurs d'explorer, comprendre et appréhender une grande quantité d'information définit un sujet d'étude d'importance en réponse aux avancées des technologies de la communication. En particulier, les autoroutes de l'information permettent une diffusion instantanée d'une grande quantité d'information. Mais cette abondance à l'échelle planétaire nourrit en même temps le risque de détruire l'information : cette dernière existe, mais elle devient inaccessible. Le problème n'est plus tant la disponibilité de l'information mais la capacité de sélection d'une information qui réponde au besoin. Les travaux récents sur la télévision interactive illustrent parfaitement le problème : une grande quantité d'information doit être accessible à tout public et tenir compte de la variété des conditions d'usage à domicile.

Face à cette croissance continue des quantités d'information manipulées, il convient donc d'assister les utilisateurs dans leur quête d'information pour des situations chaque fois particulières. Plusieurs approches sont envisageables pour assister l'utilisateur : améliorer les moteurs de recherche dans un espace d'information, déléguer certaines tâches de recherche à des agents logiciels, permettre l'expression fine des requêtes en langue naturelle augmentée de dialogues de clarification, faire appel au savoir collectif, améliorer la qualité perceptuelle et interactionnelle de l'espace d'information. A l'évidence, toutes ces pistes sont valides et fonctionnent de manière complémentaire. Dans notre étude, nous adoptons une approche

centrée sur l'interaction et nous visons à concevoir de nouvelles formes d'interaction multimodale qui aident l'utilisateur à appréhender un grand espace d'information.

2 *Objectifs*

Nous visons trois objectifs dans notre étude de la conception d'interfaces multimodales en sortie pour manipuler de grandes quantités d'information.

Un cadre de conception de la multimodalité en sortie : ce cadre visé doit couvrir les deux angles d'étude de la conception, l'utilisateur et le système interactif. Il doit permettre d'identifier et d'organiser les éléments utiles à la conception d'interfaces multimodales. La généralité de ce cadre conceptuel est primordiale pour notre étude. En effet, s'appuyant sur les concepts de notre cadre de conception, nous visons la définition de patrons de conception d'interfaces multimodales en sortie. Les patrons définiront des solutions à des problèmes récurrents de conception d'interfaces, en s'appuyant sur la multimodalité. Ils constitueront une avancée importante en vue d'une meilleure prise en compte de la multimodalité dans la conception des interfaces en sortie.

Un cadre de conception de la visualisation de données : ce cadre à établir doit intégrer la particularité des étapes du processus de visualisation. Il devra aussi infléchir et affiner le cadre de conception de la multimodalité en sortie. Ce cadre contribuera donc à clarifier l'espace problème en le structurant d'éléments qu'il convient de considérer lors de la conception d'une technique de visualisation. Là encore, notre objectif est d'établir des patrons de conception d'interfaces de visualisation.

Une solution générique à la réalisation logicielle de techniques de visualisation : cette solution visera à prouver le caractère opérationnel de nos cadres de conception pour la réalisation logicielle.

Nous justifions ces trois objectifs par les constats suivants.

Dès 1980, la multimodalité est étudiée aux États-Unis : en particulier Richard Bolt propose le paradigme du "mets ça là" dans le cas de l'intégration de la parole et du geste. En 1983, Nicholas Negroponte fonde le MediaLab au MIT avec comme objectif l'étude de l'intégration de ces nouvelles technologies. En France, la prise de conscience a lieu plus tard avec la création en 1990 d'un groupe d'étude au sein du GDR-PRC CHM. Le concept d'interaction multimodale est né. Dès lors, les travaux se sont axés sur la multimodalité en entrée, mettant en œuvre plusieurs dispositifs physiques. Peu de travaux se sont intéressés à la

conception de la multimodalité en sortie. Les travaux existants n'adoptent pas un point de vue globale de l'interaction multimodale : en effet certains étudient la pertinence de l'information véhiculée par les modalités, d'autres l'accès aux documents et aux postes de travail pour les handicapés (en particulier visuels) et enfin certains étudient la génération automatique d'énoncés multimodaux du point de vue des supports de l'information, le terme multimédia y étant le plus souvent retenu. Ces travaux aboutissent à des cadres de conception dont la portée est nécessairement limitée au point de vue restrictif adopté. Face à des cadres conceptuels disparates et à la portée limitée, il convient d'organiser le savoir expérimental dans des cadres aussi rigoureux que possibles. De là, un espace de conception structurant permet de raisonner, comparer, évaluer les interfaces existantes ou à concevoir ainsi que d'organiser les concepts nécessaires à la mise en place de patrons de conception d'interfaces multimodales en sortie.

En réponse au problème actuel qui n'est plus tant la disponibilité de l'information mais la capacité de sélection d'une information qui réponde au besoin grâce à la mise en œuvre d'une interface de visualisation, nous constatons une profusion de techniques de visualisation. Cette multiplicité des techniques de visualisation témoigne du dynamisme certain de cet axe d'étude. Néanmoins dans ce contexte aux avancées brutales se traduisant par de nombreuses techniques de visualisation présentées isolément, notre contribution concerne l'établissement de référentiels applicables à la conception afin de mieux cerner et comprendre l'espace de conception des interfaces dédiées à de grandes quantités d'information. A ces fins, nous adoptons une démarche d'analyse nouvelle consistant à considérer une technique de visualisation comme une modalité d'interaction de sortie. Nous jetons donc un nouveau regard sur les techniques de visualisation en adoptant le point de vue de la multimodalité en sortie.

Montrer enfin le caractère opérationnel de nos éléments de conception de la multimodalité et de la visualisation est un dernier point contributif important, qui lui est utile à l'ingénierie des logiciels étudiés. Il se justifie par l'absence d'outils de réalisation (boîte à outils ou générateurs d'interfaces) qui permettraient un développement efficace de tels interfaces multimodales. En effet, les techniques de visualisation sont actuellement développées en s'appuyant sur les primitives graphiques de bas niveau d'abstraction de la plate-forme de développement.

3 *Structure du mémoire*

L'organisation du mémoire reflète notre démarche d'analyse, de l'abstrait, la multimodalité, au concret, la visualisation. Nous étudions donc d'abord la multimodalité en sortie au Chapitre III puis la visualisation au Chapitre IV. Les deux facettes de notre étude

conceptuelle emploient un outil de Génie Logiciel commun, les patrons de conception, que nous introduisons au Chapitre II. Les résultats conceptuels étant présentés, le Chapitre V étudie leur mise en pratique dans la réalisation logicielle. Ce chapitre expose les principes génériques de réalisation logicielle, qui ont donc été appliqués pour réaliser nos trois systèmes. Ce chapitre est donc introductif de nos trois réalisations logicielles présentées ensuite aux Chapitres VI, VII et VIII.

Le Chapitre II est consacré aux patrons de conception, outil de Génie Logiciel que nous employons dans ce mémoire. Il est important de noter que les patrons de conception sont utilisés dans nos travaux comme un outil et ne constituent pas un objet de recherche. En effet nous présentons des solutions de conception d'interfaces multimodales en sortie et de visualisation sous la forme de patrons de conception. Après avoir fait le point sur les approches existantes de définition de langages de patrons, nous définissons le format de nos patrons, puis nous organisons nos patrons de conception en tenant compte des particularités de notre domaine d'étude : la multimodalité en sortie et la visualisation de grandes quantités d'information. Ces principes d'organisation définissent notre langage de patrons.

Le Chapitre III explore la multimodalité en sortie. Après avoir assis la terminologie, nous détaillons notre cadre de conception de la multimodalité en sortie. Nous présentons ensuite les espaces de conception les plus pertinents par rapport à notre cadre de conception. Un bilan analytique nous conduit à présenter notre propre contribution motivée, pour l'essentiel, par les problèmes de conception : un espace de composition de modalités et un espace de caractérisation d'une modalité pure ou composée. S'appuyant sur les concepts de nos deux espaces, nous définissons quatre patrons de conception d'interfaces multimodales en sortie.

La généralité des résultats du Chapitre III nous permet au chapitre IV, d'infléchir et d'affiner ce cadre de conception au cas de la multimodalité graphique informationnelle pour de grandes quantités de données : les techniques de visualisation. Nous montrons que notre cadre de conception fédère les approches existantes, et organise les étapes du processus de visualisation. Reposant sur ces étapes de conception, deux espaces de caractérisation, l'un dédié aux techniques de visualisation, l'autre à la composition de ces techniques, sont ensuite présentés. Comme pour le cas plus général de la multimodalité en sortie du Chapitre III, nous validons enfin notre approche en présentant cinq patrons de conception d'interfaces de visualisation de grandes quantités d'information.

Avec le Chapitre V, nous voulons montrer comment rendre nos résultats conceptuels exposés aux Chapitres III et IV opérationnelles. Ce chapitre explore donc le support pratique de

l'activité de réalisation en procédant par l'exemple. Pour cela nous détaillons les éléments de réalisation communs à nos systèmes, en les présentant selon les étapes du processus de développement logiciel : conception d'algorithmes particuliers, architecture logicielle et implémentation. Parmi ces éléments de réalisation, nous accordons une place de choix à la réalisation des vues hyperboliques utilisées dans VITESSE (Chapitre VI) et MulTab (Chapitre VII). Nous expliquons ensuite l'architecture logicielle des trois systèmes selon le modèle d'architecture PAC-AMODEUS puis nous abordons les principes de programmation en JAVA. La mise en place d'une architecture logicielle nous a conduit à établir une correspondance entre les composants logiciels du modèle d'architecture PAC-AMODEUS et les étapes du processus de visualisation de notre cadre fédérateur (Chapitre IV). Ce chapitre fournit donc un cadre conceptuel générique à la réalisation logicielle du processus de visualisation.

Les Chapitres VI, VII et VIII présentent trois systèmes, respectivement VITESSE [Vitesse], MulTab [MulTab] et Parent [Parent]. VITESSE est un système de visualisation d'une grande quantité de réponses retournées par un moteur de recherche du Web. Ce système est basé sur une technique de vues multiples, les unes hyperboliques, les autres à différents facteurs de zoom. MulTab est un système de visualisation d'un très grand tableau de données. Contrairement à VITESSE, l'interface repose entièrement sur une seule vue hyperbolique. Néanmoins cette vue unique est composée de multiples foci. Enfin, Parent est un système de visualisation d'une arborescence de fichiers. L'espace d'information construit repose sur un nouvel algorithme, étendu de celui des Treemaps, que nous avons baptisé Fieldmaps. La composition de modalités en sortie permet l'édition des fichiers par manipulation directe.

Pour chacun des systèmes, nous adoptons la même structure de présentation : la conception du système est d'abord détaillée selon nos cadres conceptuels des Chapitres III et IV. Nous exposons ensuite leur réalisation logicielle en nous reposant sur les principes de réalisation présentés au Chapitre V. Ces trois systèmes forment un bon démonstrateur de nos patrons de conception.

Le Chapitre IX conclut nos travaux en dressant un bilan des points contributifs selon les deux facettes suivantes : interface multimodale et interface de visualisation. Nous développons ensuite les perspectives pour nos travaux futurs, à court et à long terme.

La bibliographie est décomposée en deux parties. Compte tenu du grand nombre d'applications graphiques utilisées comme exemples, nous les avons distingués des articles de recherche. Dans le manuscrit, les références aux systèmes ne sont pas datées (par exemple

[AltaVista]) tandis que celles aux articles finissent par les deux derniers chiffres de l'année de publication (par exemple [Abowd 92]).

Un index des figures et deux annexes sont inclus dans ce mémoire. L'Annexe A définit les critères d'ergonomie établis par notre équipe IIHM du laboratoire CLIPS-IMAG, tandis que l'Annexe B liste sans les détailler les principes ergonomiques identifiés et organisés par Dominique Scapin et Christian Bastien. Les annexes consignent donc un ensemble de critères servant à caractériser la notion d'utilisabilité.

CHAPITRE III

PATRONS DE CONCEPTION

“ ... More formally the Writing-a-thesis pattern may be stated:

Problem: *Writing a thesis that should pass.*

Context: *To make the thesis structural and comprehensible.*

Forces: *Time is running out and some concepts cause confusion.*

Solution: *Use the supervisors advice ”*

Thèse de Master soutenue par Magnus Kardell, Umeå University.

Ce chapitre est consacré aux patrons de conception, outil de Génie Logiciel que nous utilisons dans ce mémoire. Il est important de noter que les patrons de conception sont utilisés dans nos travaux comme un outil et ne constituent pas un objet de recherche. En effet nous présentons des solutions de conception d'interfaces multimodales en sortie et de visualisation sous la forme de patrons de conception.

Dans ce chapitre, après un historique des patrons de conception, nous asseyons la terminologie que nous employons dans le mémoire ; en particulier, nous détaillons le terme de langage de patrons. Nous exposons ensuite notre format des patrons de conception. Ces derniers seront ensuite présentés dans les Chapitres III et IV. Enfin, nous anticipons sur le contenu de ces deux chapitres en organisant nos patrons de conception selon trois axes, dessinant ainsi les contours de notre langage de patrons.

1 Historique

Un patron de conception décrit une solution réutilisable à un problème récurrent de conception. Il contient en particulier une description des aspects saillants du problème traité et de sa solution ainsi que des exemples illustratifs. Il est présenté dans un format précis pour faciliter sa compréhension. Nous relevons ici les étapes marquantes de l'histoire des patrons de conception.

Le concept de patron de conception a originellement été introduit dans le livre "A Pattern Language" [Alexander 77] dans lequel l'architecte Christopher Alexander a présenté des résultats en Architecture du bâtiment sous la forme de patrons de conception ("design rule" ou "design pattern" en anglais). Par exemple, un patron intitulé "Entrance Transition" traite du problème de la conception de l'entrée d'un bâtiment. La solution avancée consiste à concevoir un espace de transition entre la rue et la porte d'entrée :

"Make a transition space between the street and the front door. Bring the path which connects street and entrance through this transition space, and mark it with a change of light, a change of sound, a change of direction, a change of surface, a change of level, perhaps by gateways which make a change of enclosure, and above all with a change of view." [Alexander 77] page 552.

Kent Beck et Ward Cunningham [Beck 87] sont les premiers à définir des patrons de conception en Informatique. [Beck 87]. Ces précurseurs ont initialement développé cinq patrons pour la programmation en langage SmallTalk, qui sont recensés à l'adresse Internet suivante : <http://c2.com/cgi-bin/wiki?HistoryOfPatterns>. Il est intéressant de noter que ces patrons ont été

élaborés en collaboration avec les utilisateurs finaux d'un système en cours de conception, dans le cadre d'une conception centrée sur l'utilisateur. A titre illustratif, nous présentons le patron nommé "Window per Task" :

"Make a specific window for each task the user must perform. All of the information needed to complete a task should be available in the FewPanels of the window. Assume prerequisite tasks have been completed (if they haven't, the user will simply change windows.)..."

Ce patron préconise la création d'une fenêtre pour chaque tâche de l'utilisateur, principe maintenant bien connu dans notre domaine. Toute l'information nécessaire pour la réalisation d'une tâche doit être regroupée dans une fenêtre. Les éléments graphiques composant la fenêtre font l'objet d'un autre patron de conception, noté "FewPanels".

Bien que ces premiers patrons traitent de la conception des éléments graphiques d'une interface, c'est dans le domaine de la programmation orientée objet qu'ils connaissent leur essor. Une contribution marquante du domaine est sans nul doute le livre "Design Patterns, Elements of Reusable Object-Oriented Software" écrit par Gamma et al. [Gamma 95]. Ce livre contient 23 patrons traitant de classes et d'objets en programmation orientée objet. Par exemple un patron "Singleton" explique comment réaliser une classe qui n'autorise qu'une seule instance :

"Intent : Ensure a class only has one instance, and provide a global point of access of it."
[Gamma 95] page 127.

La mise en place de patrons pour la conception d'interfaces ou patrons d'interaction est plus récente. En effet, il semble intuitivement plus difficile d'élaborer des patrons d'interaction qui doivent prendre en compte les facteurs humains, que des patrons de programmation orientée objet dont le domaine d'application est plus rigide et cerné. Néanmoins, les patrons d'interaction constituent actuellement une approche reconnue dans le domaine de l'Interaction Homme-Machine (IHM), comme les patrons de conception d'hypermédias [Nanard 98].

Nous notons aussi l'atelier dédié aux patrons de conception en IHM, organisé lors la conférence CHI'97 [Bayle 97], qui a donné naissance à de nombreux travaux. Cette approche répond aux besoins de capitalisation d'expériences, de partage et de communication entre les concepteurs d'interfaces. Les patrons d'interaction proposés dans ces travaux s'adressent à un public plus large que les concepteurs d'hypermédias. Toutefois, plus un domaine applicatif est restreint (par exemple les hypermédias), plus les patrons proposés sont bien définis et s'adressent à un public cohérent de concepteurs.

Nous relevons malheureusement une utilisation différente du terme de patron de conception dans les différents travaux que nous venons de citer. Les auteurs de [Nanard 98] proposent quatre termes selon la nature (abstraite ou concrète) et la phase de développement (conception ou implémentation). Le terme de patron de conception est présenté comme un outil abstrait qui intervient lors de l'implémentation. Cette approche correspond aux patrons de Gamma [Gamma 95] mais pas à l'approche de Alexander [Alexander 77] et des travaux sur les patrons d'interaction issus de [Bayle 97].

Nous constatons que les auteurs de patrons d'interaction ont résolument pris le parti d'Internet car la validité d'un patron passe surtout par son usage, qui implique sa diffusion, mais aussi par sa mise en relation avec d'autres patrons pour former un langage de patrons. Nous fournissons quatre adresses Internet qui illustrent cette démarche :

- <http://webfuse.cqu.edu.au/Information/Resources/Readings/Patterns/>,
- http://www.pliant.org/personal/Tom_Erickson/InteractionPatterns.html,
- <http://www.cs.vu.nl/~martijn/patterns/>,
- <http://www.cs.ukc.ac.uk/people/staff/saf/patterns/gallery.html>.

Afin d'illustrer les patrons d'interaction, nous présentons le patron intitulé "Progress" qui appartient à "The Amsterdam Collection of Patterns in User Interface Design" (<http://www.cs.vu.nl/~martijn/patterns/>) :

"Problem: The user wants to know whether or not the operation is still being performed as well as how much longer the user will need to wait.

Solutions: Show that the application is still working and give an indication of the progress."

Ce patron est à mettre en relation avec un principe reconnu en IHM d'observabilité de l'état interne du système que nous décrivons dans notre liste de critères d'ergonomie en Annexe 1, page 259.

Un autre exemple de patron d'interaction, intitulé "Link destination announcement" [Nanard 98] est lui dédié à des systèmes interactifs particuliers (les hypermédias) :

- "Pattern name: Link destination announcement
- Motivation: To avoid unnecessary link firing by providing information about the destination

- Context: When rolling over an anchor
- Intent: Provide information about the link destination
- Solution: Detect roll over on anchor, enhance the anchor, display short abstract."

Ce patron de conception qui préconise l'affichage d'un résumé lorsque le curseur de la souris passe au-dessus d'une ancre, repose sur le principe de Prévisibilité dans les interfaces que nous définissons en Annexe 1.

Par souci de concision, cette section ne contient que des exemples partiels de patrons. Pour chaque patron, seule la description succincte du problème et de sa solution a été présentée. Bien d'autres éléments appartiennent à un patron. Comme nous l'avons annoncé, un patron adhère à un format. Avant de décrire le format que nous avons choisi pour nos travaux, nous fixons la terminologie liée à un patron.

2 *Terminologie*

Nous définissons dans cette section les termes clefs qui caractérisent un patron et un langage de patrons. Afin d'asseoir la terminologie adoptée dans ce mémoire, chaque terme est défini dans le contexte de nos travaux : la conception d'interfaces multimodales en sortie, et son application à la visualisation.

2.1 *Auteur et concepteur*

Celui qui écrit un patron et celui qui l'utilise sont tous deux des concepteurs d'interfaces. Afin de clarifier notre discours, nous désignons celui qui écrit un patron comme l'**auteur**, et celui qui applique les patrons de conception pour concevoir une interface comme le **concepteur**.

2.2 *Patron et langage de patrons*

Comme nous l'avons souligné à la section 1, plusieurs types de patrons existent et concernent des phases différentes du cycle de vie d'un logiciel. Nous rappelons que notre approche vise l'établissement de **patrons** d'interaction ou patrons de conception d'interfaces, utiles lors de la phase de Spécifications Externes dans un cycle en V [Boehm 81] par exemple.

Une caractéristique importante d'un patron est son appartenance à un ensemble structuré. Un ensemble cohérent et organisé de patrons définit un **langage de patrons**. (" pattern language " en anglais). La motivation première d'un langage de patrons est de rendre une collection de patrons plus utilisable : un ensemble cohérent de patrons avec différents points

d'entrée est plus utilisable au concepteur qu'une liste non ordonnée de patrons. De plus comme le souligne Elisabeth Bayle [Bayle 97], le langage de patrons fournit un langage commun à tous les acteurs de disciplines différentes qui interviennent dans la conception d'une interface (informaticiens, graphistes, ergonomes, etc.). Comme le souligne Gerald Meszaros [Meszaros 98], chaque patron doit résoudre un problème spécifique dans le contexte partagé du langage.

*" Each pattern should solve a specific **problem** within the shared context of the language. "* [Meszaros 98].

Comme nous l'avons annoncé à la section 1, un langage de patrons véhicule un format : chaque corps d'un patron est présenté selon un format précis. Dans ce mémoire, tous les patrons présentés pour la multimodalité et la visualisation adhèrent à un format que nous présentons à la section suivante. Outre le format, Sally Fincher [Fincher 99] caractérise un langage de patrons selon quatre facettes fondamentales : la capture de pratiques de conception, l'abstraction, le système de valeurs et le principe d'organisation. La capture de pratiques et l'abstraction dépeignent un patron considéré isolément tandis que le système de valeurs et le principe d'organisation caractérisent un ensemble de patrons et donc un langage.

Un patron traduit une pratique de conception qui démontre et illustre l'application d'un principe de conception. Par exemple un patron pourrait traduire la pratique suivante : lorsque le nombre d'entrées dans un menu augmente, le concepteur cherche à scinder le menu en deux, sauf si le nombre de menus est déjà jugé trop grand, auquel cas il met en œuvre un système de menus hiérarchiques. L'auteur est généralement aussi un concepteur. A ce titre, il peut capturer ses propres expériences et pratiques de conception.

L'abstraction consiste, sur la base de plusieurs exemples, au moins trois si l'on se réfère à la règle "Rule Of Three" [Tracz 95], à identifier les caractéristiques communes aux différentes solutions qui ont contribué à la résolution du problème. Il s'agit ici d'abstraire la pratique de conception pour qu'elle soit applicable dans des contextes différents, comme le souligne la définition d'un patron de Christopher Alexander :

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million of times over, whithout ever doing it the same way twice"

L'auteur d'un langage de patrons doit veiller à la cohérence du niveau d'abstraction des patrons. Cette étape d'abstraction, sans doute la plus difficile, est fondamentale dans le processus de création des patrons de conception. Par exemple, à partir de l'étude de la

conception de menus au sein de différentes interfaces, l'auteur va dégager les points saillants des solutions à faire figurer dans le patron : est-ce la taille du menu à l'écran qui est un facteur dominant ou le nombre d'entrées dans le menu ?

Le système de valeurs nécessaire à l'interprétation des patrons est implicite dans le langage et est défini par les utilisateurs du langage, les concepteurs. Par exemple les concepteurs de programmes orientés objet définissent extrinsèquement le système de valeurs du langage de patrons présentés dans [Gamma 95]. De même les concepteurs d'interfaces multimodales définissent le système de valeurs des patrons, présenté dans les chapitres suivants.

Un langage décrit un ensemble organisé de patrons. Le principe d'organisation définit le critère de structuration des patrons. Dans [Alexander 77], les patrons pour la conception de bâtiments sont organisés selon une échelle allant de la ville à la maison individuelle. Des relations entre patrons d'échelles différentes sont établies, un patron de conception à l'échelle de la ville s'appuyant sur des patrons de conception de maisons. Les patrons de programmation orientée objet [Gamma 95] sont eux structurés selon une décomposition fonctionnelle : les patrons de création, de structure et de comportement de classes ou d'objets. Nous organisons l'ensemble de nos patrons dans un espace cohérent défini selon trois dimensions qui font l'objet de la section 4. Ces dimensions définissent la structure des deux chapitres suivants, III et IV.

Ayant défini les termes de patron et de langage de patrons, nous pouvons maintenant définir le format de patrons choisi puis le principe d'organisation adopté pour notre langage de patrons.

3 *Format de patrons*

Les formats des patrons existants diffèrent peu les uns des autres. En effet, nous retrouvons le même contenu d'un patron à l'autre et les différences de format résident principalement dans le regroupement ou l'éclatement en rubriques distinctes de certaines parties de la description. Nous notons peu de variations dans l'ordre de description du contenu d'un patron. Nous présentons ici deux exemples de formats qui corroborent notre assertion, avant de présenter le format que nous avons adopté.

De nombreux auteurs s'appuient sur le format de Christopher Alexander [Alexander 77] décrit en ces termes :

► *" A picture, which shows an archetypical example of the pattern,*

Une image qui illustre le patron

- ▶ *An introductory paragraph, which sets the context for the pattern,*
Un paragraphe d'introduction qui décrit le contexte d'application du patron
- ▶ *Three delimiting diamonds,*
Trois losanges de séparation
- ▶ *A headline in bold type that gives the essence of the problem in one or two sentences,*
Un titre en gras décrivant l'essence du problème en une ou deux phrases
- ▶ *The body, the longest section: background, motivation, variations,*
Le corps du patron, la rubrique la plus longue : contexte, motivation, variation
- ▶ *The solution, in bold type: how to solve the problem,*
La solution, en gras : comment résoudre le problème
- ▶ *A diagram, that shows the solution as a labeled picture,*
Un diagramme qui présente la solution sous la forme d'une image commentée
- ▶ *Another three diamonds to terminate the main body,*
Trois autres losanges de séparation pour délimiter le corps du patron
- ▶ *A paragraph that ties the pattern to all the smaller related patterns that round out this one".*
Un paragraphe qui lie sémantiquement le patron à d'autres patrons de niveau d'abstraction inférieur.

Extrait de [Alexander 77], pages x-xi.

D'autres auteurs ont proposé des formats de patrons ayant des titres à chaque rubrique, comme l'illustre le patron "Toolbox" de Jenifer Tidwell [Tidwell 99].

- ▶ **Examples:** *Shape palettes in Powerpoint, Illustrator, Macdraw, etc.*
Toolbar buttons for font styles and text alignment in Word, Netscape Composer, etc.
Component libraries in Visual Cafe and other software development environments
Artist's set of pens, paintbrushes, pastels, etc.
A garage workbench.
- ▶ **Context:** *The artifact supports the creation of other artifacts, as in a WYSIWYG Editor.*
- ▶ **Problem:** *How should the artifact present the actions that the user may take?*
- ▶ **Forces:**
Tools used to create things are different in kind from ordinary actions performed upon existing objects. A user who is learning the artifact should know what tools are available. An experienced user needs the tools ready to hand. If the user brings in new tools, they need to know where they will go.
- ▶ **Solution:** *Keep tools together, put them physically close to the user's working surface, and make sure they are distinct from other actions that the user may take. If the tools can be*

represented well as icons and not words, use icons; they usually represent visual objects or modes anyway. Let the users position the tools where they want (Personal Object Space). Organize the tools according to how they are used, so that the most common ones are the most prominent (for instance), and localize them by task, so that the user doesn't have to jump around a lot as they're performing common complex tasks. Users can generally deal with seeing large sets of them -- no one minds a well-organized 20-item palette, and exposing them all means the user doesn't have to spend effort getting to them -- but break them up by function if necessary (Groups of Related Things). Allow the user to introduce other tools from outside the artifact, if possible. Put the foreign tools with the others, so that the user doesn't have to deal with an artificial separation between "ours" and "theirs."

- **Resulting Context:** *In a software artifact, use Short Description or Pointer Shows Affordance if there's any doubt that the user will recognize the tools as they are statically represented. Try not to depend upon them, though -- it's easy to get lazy and use poor icons, knowing that a user can just use the tooltip for something they don't understand. The problem, of course, is that a user then has to move the mouse over every single tool to figure out what they all are. This is far from ideal.*
- **Notes:** *The first force in this pattern is just my intuition, based on empirical evidence. I freely admit that don't have much of a basis for it. Unfortunately, the pattern's distinctiveness hinges on it, so any help on this front would be welcome...*

Avant de comparer ce format [Tidwell 99] avec le précédent [Alexander 77], nous expliquons ses rubriques, par l'exemple, en détaillant le contenu du patron "Toolbox".

Les illustrations sont mises en exergue et font l'objet de la première rubrique. Ce patron traitant de palettes d'outils, l'auteur cite des exemples variés à la fois logiciels et dans le monde réel comme un atelier de mécanique. L'auteur définit ensuite le contexte d'application du patron, c'est-à-dire la conception d'un artefact qui permet de créer d'autres artefacts. Le problème énoncé réside dans la conception d'un artefact qui explicite les actions que l'utilisateur peut entreprendre. Quatre motivations sont fournies par l'auteur :

- *Les outils utilisés pour créer des objets sont de nature différente des actions accomplies sur des objets existants.*
- *Un utilisateur qui apprend à utiliser un artefact doit savoir quels sont les outils disponibles.*
- *Un utilisateur expérimenté a besoin de tous les outils à portée de main.*
- *Si un utilisateur introduit de nouveaux outils, ces derniers doivent s'intégrer facilement dans la palette existante.*

La solution avancée consiste à regrouper les outils et à les placer à côté de la surface de travail. L'auteur préconise également une façon de représenter et d'organiser les outils. Dans la rubrique suivante intitulée "contexte résultant", l'auteur attire l'attention du concepteur sur la possibilité d'utiliser un menu flottant pour nommer un outil dont l'icône ne serait pas compréhensible. Toutefois, cette possibilité est présentée comme une solution non optimale. Enfin l'auteur termine son patron par une note explicative indiquant que la première motivation énoncée est une intuition de sa part.

Les deux formats présentés [Alexander 77] [Tidwell 99] sont proches : les quatre premières rubriques sont communes. La différence principale entre les deux formats concerne la description de la solution. Tandis que le premier format [Alexander 77] décrit la solution par un texte et un diagramme dans deux rubriques distinctes, le deuxième format [Tidwell 99] contient une seule rubrique, la plus longue, qui contient un énoncé textuel de la solution. De plus la rubrique "Contexte résultant" du deuxième format [Tidwell 99] n'est pas explicitement présente dans le premier format. Le contexte résultant de l'application de la solution pourrait être décrit, dans le premier format, au sein de la rubrique dédiée à la solution. Enfin, une dernière différence significative, de forme et non de structure, réside dans la présentation des exemples, le premier format préconisant une image et le deuxième une description textuelle. Comme il est souligné dans [Fincher 99], dans le cas de patrons de conception d'interfaces, une présentation graphique statique peut parfois être inadéquate car l'interaction est par définition dynamique.

De ces deux exemples de formats, nous définissons notre format que nous détaillons dans les paragraphes suivants. Comme Gamma [Gamma 95], nous utilisons une liste de sections qui décrivent les différents aspects du patron. Comme le note Dirk Riehle [Riehle 96], aucun format n'est a priori meilleur. Nos patrons sont composés des sections suivantes :

- ▶ Exemples illustratifs
- ▶ Description du problème et du contexte
- ▶ Motivations
- ▶ Contre-exemple
- ▶ Solution proposée

3.1 Exemples illustratifs

Nous illustrons nos patrons par trois exemples au minimum : au moins un des exemples est détaillé. Les exemples seront illustrés de copies d'écran lorsqu'une présentation statique est pertinente. Les exemples que nous avons choisis font référence à des systèmes très répandus ou téléchargeables par Internet. Ce choix est une condition nécessaire mais non suffisante à la

validité du patron. L'exemple est le premier attribut du patron car c'est à partir de l'exemple que sont produits les patrons.

3.2 Description du problème et du contexte

La description du problème traité par le patron nous semble indissociable de celle du contexte. Le contexte traduit tout ce qui a un impact sur le problème et est difficilement discernable de ce qui définit le problème. En l'absence d'une frontière marquée entre ces deux concepts, nous les regroupons en une seule rubrique. Le problème doit être récurrent et ne pas être le cas particulier d'une application. L'objectif de cette description est de permettre au concepteur d'identifier si son problème est traité par le patron. Aussi les patrons sous forme électronique doivent fournir une option de recherche par mots clés sur les descriptions de problèmes.

3.3 Motivations

Cette rubrique contient les motivations de l'auteur à écrire le patron. Elle correspond à celle intitulée "Forces" dans le format de [Tidwell 99]. Dédié à un domaine très évolutif, il est important que l'auteur justifie clairement le patron afin de pouvoir rapidement déterminer sa validité ou son obsolescence. Par exemple, si un patron se justifie par un manque ou une limite dans les périphériques disponibles, ce patron devient caduc lorsque les nouveautés technologiques répondent aux motivations avancées par l'auteur. Considérons un patron qui préconise des messages d'erreur graphiques car certaines plates-formes ne gèrent pas plusieurs sons en parallèle. Une telle limitation peut motiver un patron qui déconseille l'usage du son pour les messages d'erreurs graves car ces derniers risquent de ne pas être perçus, si l'utilisateur écoute de la musique par exemple. Cependant, cette limitation est actuellement vraie dans le système d'exploitation Windows NT4 [Windows] et non sur celle MacOS [MacOS]. Il est donc important que l'auteur fournisse une argumentation précise et évoque les futures versions de Windows qui rendront le patron obsolète.

3.4 Contre-exemple

L'introduction d'un contre-exemple dans le format définit une rubrique nouvelle par rapport aux deux formats que nous avons présentés précédemment. Cette rubrique ne correspond pas exactement aux anti-patrons introduits par Andrew Koenig [Koenig 95]. Alors qu'un anti-patron est un patron entier, un contre-exemple n'est qu'une illustration de l'anti-patron correspondant à ce patron. Comme le remarque James Coplien [Coplien 96], l'apprentissage par anti-patron n'est pas une méthode pédagogique.

Nous remarquons que, par rapport au domaine du Génie Logiciel, il est plus facile de disposer d'exemples et de contre-exemples en IHM, car il s'agit de la partie perçue et manipulée du système informatique. De plus, la conception d'interfaces se base fréquemment sur l'expérience acquise avec des systèmes interactifs peu ergonomiques. Réutiliser ce savoir dans un patron de conception nous semble primordial.

3.5 *Solution proposée*

La solution proposée dans un patron est le cœur même du patron. Sa clarté est essentielle car comme dans le format de [Tidwell 99], cette rubrique constitue la partie la plus importante du patron. Dans nos travaux, les patrons définissent une solution en termes d'éléments d'interfaces ou modalités d'interaction à mettre en œuvre sans traiter leurs réalisations logicielles. Ainsi l'architecture logicielle et la programmation ne sont pas considérées dans la solution de nos patrons.

4 *Principe d'organisation*

Le format de patrons étant fixé, nous organisons nos patrons de conception en tenant compte des particularités de notre domaine d'étude : la multimodalité en sortie et la visualisation de grandes quantités d'information. Les trois axes de la Figure 1 définissent la structure du langage de patrons présenté dans les chapitres suivants. Ces trois axes dessinent donc les contours de notre langage de patrons dédié à l'interaction homme-machine pour les interfaces multimodales en sortie.

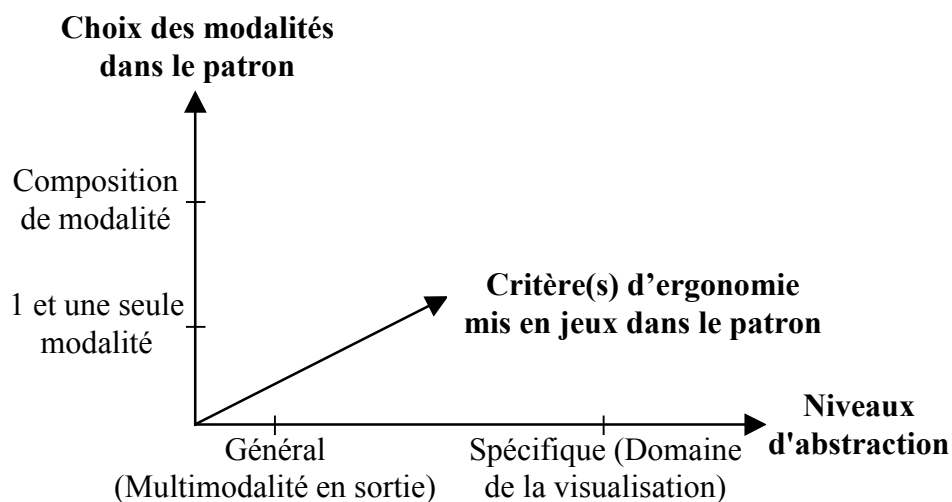
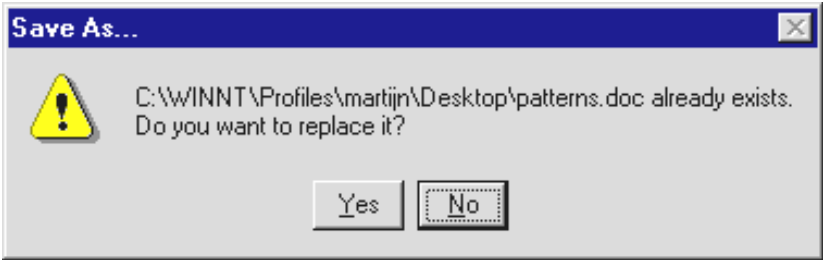


Figure 1 : Axes de classification des patrons de conception.

4.1 Critères d'ergonomie mis en jeu par le patron

La première caractéristique générale à un patron de conception d'IHM et non spécifique aux patrons de notre domaine d'étude, est le ou les critères d'ergonomie mis en jeu par le patron. Comme Martijn van Welie [Van Welie 00] nous utilisons les critères ergonomiques ("Usability principle" dans les travaux de van Welie) afin de classer les patrons de conception. En effet, un patron préconise une solution de conception, et celle-ci doit donc pouvoir être reliée à un ou plusieurs critères ergonomiques. Nous présentons ci-dessous l'exemple du patron appelé "le bouclier" [Van Welie 00]. L'auteur relie ce patron au principe ergonomique de gestion des erreurs, principe énoncé par Dominique Scapin et Christian Bastien [Scapin 93].

Name	The Shield
Problem	The user may accidentally select a function that has irreversible (side) effects.
Usability Principle	Error Management
Context	The user needs to be protected against unintended or accidental actions that have irreversible (side) effects. The (side) effects may lead to unsafe or highly undesired situations. For example the unintended deletion or overwriting of files. Do not use for actions that are reversible.
Forces	- The user is striving for speed while trying to avoid mistakes. - The severity of the (side) effects.
Solutions	Protect the user by inserting a shield. Add an extra protection layer to the function to protect the user from Making mistakes. The user is asked to confirm her intent with the default answer being the safe option.
Examples	 A copy of the file already exists at the specified location. Overwriting it will result in loss of the copy. The default is "No" so that the speedy user has to take the effort of saying "Yes".
Usability Impact	Increased safety, less errors and higher satisfaction. However, it requires extra user action which leads to lower performance time.
Rationale	The extra layer causes the user to require 2 repetitive mistakes instead of 1. The safe default decreases the chances for a second mistake.
Known Uses	Microsoft Explorer, Apple Finder

Ce patron traite de la prévention des erreurs. L'auteur souligne dans la rubrique "contexte" que seules les erreurs irréversibles sont concernées. Les motivations sont la gravité des effets de l'erreur et le compromis entre la vitesse d'interaction et la prévention des erreurs.

La solution préconisée consiste à rajouter une étape de confirmation avant la réalisation d'une action irréversible. L'utilisateur devra donc réaliser deux actions avant d'arriver dans un état critique comme l'illustre l'exemple. L'auteur souligne le compromis entre la prévention des erreurs et l'efficacité dans une rubrique intitulée "Impact sur l'utilisabilité". Enfin avant de citer deux exemples d'application du patron (Microsoft Explorer et Apple Finder), l'auteur explique la rationalité de son patron en expliquant que deux erreurs consécutives sont nécessaires avant d'aboutir à un état non désiré.

Nous avons présenté un patron en relation avec le principe ergonomique de [Scapin 93]. La liste de ces principes est présentée en Annexe 2 (page 266). Notre équipe a établi une autre liste de critères ergonomiques [Abowd 92][Gram 94] que nous utilisons pour classer nos patrons (Annexe 1, page 259). Chaque patron sera donc mis en relation avec un ou plusieurs critères, présentés en Annexe. Cette classification n'implique pas un ordre d'importance des patrons car les critères ne sont pas eux-mêmes ordonnés. En effet, notre liste est décomposée en deux classes de critères non ordonnés : les critères de souplesse et les critères de robustesse. La souplesse de l'interaction exprime l'éventail des choix pour l'utilisateur et le système tandis que la robustesse de l'interaction vise la prévention des erreurs, l'augmentation des chances de succès de l'utilisateur. Ainsi, l'exemple précédent, le patron intitulé "le bouclier", serait mis en relation avec le critère de robustesse nommé *curabilité* (Principe de l'effort commensurable : ce qui est difficile à défaire doit être difficile à faire). A titre illustratif, considérons deux critères de souplesse ainsi que de robustesse qui sont particulièrement pertinents pour notre étude.

Un premier critère de souplesse important pour notre étude est le critère d'*atteignabilité*. Ce critère caractérise la capacité du système à permettre à l'utilisateur de naviguer dans l'ensemble des états observables du système. Dans un système de visualisation de grandes quantités d'information, ce critère est crucial car il caractérise la navigation au sein de l'espace observable. Un autre critère remarquable de la même classe est celui de la multiplicité du rendu. Ce critère caractérise la capacité du système à fournir plusieurs représentations pour un même concept. Ce critère est central à de nombreux systèmes multimodaux en sortie.

Un critère de robustesse fondamental à notre étude de la visualisation de grandes quantités d'information est l'observabilité. Ce critère dénote la capacité du système à rendre perceptible l'état interne pertinent du système. L'observabilité est difficile à vérifier lorsque tout l'espace d'information doit être rendu perceptible afin que l'utilisateur puisse réaliser sa tâche. Enfin le critère de robustesse appelé *insistance* caractérise la capacité du système à forcer la perception de son état interne. Ce critère est une motivation à de nombreux systèmes

multimodaux en sortie, utilisant par exemple des modalités graphiques couplées à des modalités sonores, afin de favoriser la perception.

Dans les chapitres suivants, pour chaque patron, nous citerons le ou les critères d'ergonomie mis en jeu et leur classe (souplesse ou robustesse).

4.2 Choix des modalités préconisé par le patron

La seconde distinction entre les patrons repose sur le nombre de modalités mises en œuvre dans les solutions avancées. Notre approche consiste à différencier les patrons qui indiquent une modalité précise à mettre en œuvre, des patrons qui composent plusieurs modalités comme solutions aux problèmes. Nous rapprochons cet axe de classification de celui appliqué par Mahemoff [Mahemoff 99] qui distingue les patrons portant sur un élément d'une interface des patrons préconisant un arrangement d'éléments de l'interface.

4.3 Niveaux d'abstraction du patron

Le dernier axe de classification des patrons identifié est le niveau d'abstraction : nous différencions les patrons généraux, traitant de la multimodalité en sortie, des patrons spécifiques au domaine de la visualisation de grandes quantités d'information. Les patrons de conception généraux seront présentés au Chapitre III et les patrons de conception spécifiques à la visualisation de grandes quantités d'information seront exposés au Chapitre IV.

5 Conclusion

Nous avons présenté un outil conceptuel, les patrons de conception, que nous utiliserons dans les deux chapitres suivants afin de définir une aide aux concepteurs d'interfaces multimodales et de visualisation de grandes quantités d'information. Les patrons constituent un outil d'aide à la conception ; il est intéressant de le comparer avec des outils complémentaires comme QOC [MacLean 90]. QOC permet de rationaliser les choix de conception : Question (Q), Options (O), Critère (C). Un schéma QOC contient une liste d'options ou solutions qui répondent à une question. Le choix de la solution s'effectue selon plusieurs critères. Les solutions sont reliées par des traits aux critères qu'elles satisfont. La Figure 2 présente un exemple.

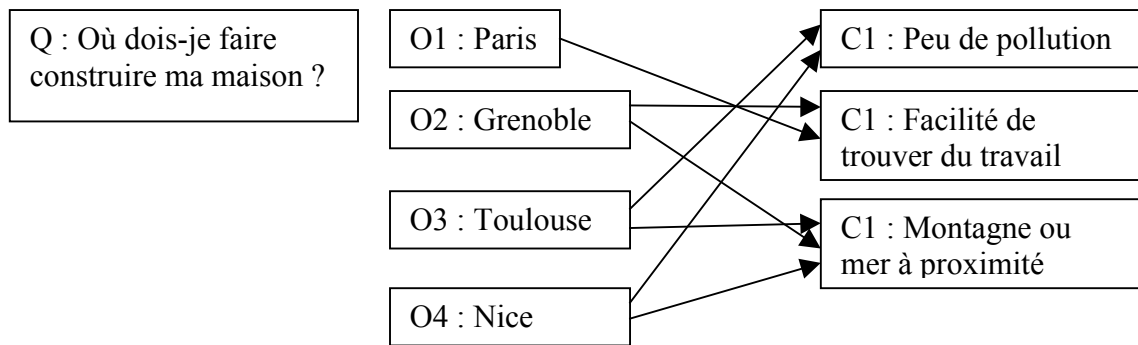


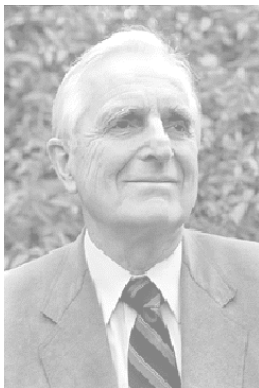
Figure 2 : Exemple de schéma QOC pour la question "où dois-je faire construire ma maison ?".

Nous constatons les correspondances entre la rubrique "Description du problème et du contexte" de notre format de patron et la Question de QOC, entre la rubrique "Solution" et les Options de QOC ainsi qu'entre la rubrique "Motivations" et les Critères de QOC. Les deux outils sont donc assez proches puisque chaque composant d'un schéma QOC se retrouve dans un patron. Cependant QOC sert à rationaliser les choix de conception tandis que les patrons fournissent une solution unique à un problème. Selon notre point de vue, un patron est une description détaillée d'un schéma QOC contenant une seule option.

Dans ce chapitre, outre le format des patrons présentés dans le mémoire, nous avons exposé les axes de classification définissant notre langage de patrons. Ces axes impliquent la structure du mémoire. En effet, le Chapitre III porte sur la multimodalité en sortie tandis que le Chapitre IV traite de la visualisation (axe "Niveau d'Abstraction"). De plus les patrons présentés dans ces deux chapitres sont organisés selon les axes "Critères d'ergonomie" et "Choix des modalités" (modalité atomique ou composée).

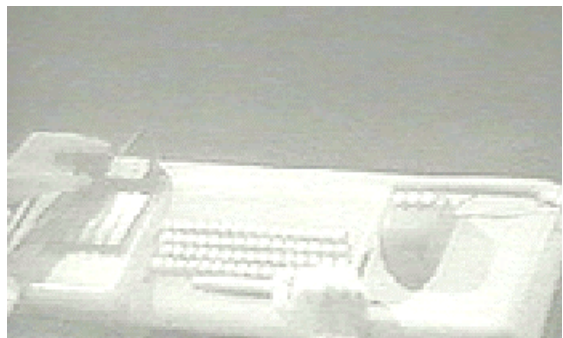
CHAPITRE III

MULTIMODALITE EN SORTIE



“ By our view, we do not have to wait until we learn how the human mental processes work, we do not have to wait until we learn how to make computers more intelligent or bigger or faster, we can begin developing powerful and economically feasible augmentation systems on the basis of what we now know and have. ”

Doug Engelbart,
Octobre 1962.
[Engelbart 62]



Ce chapitre est consacré à l'interaction multimodale en sortie, et jette les fondements du Chapitre IV dédié aux modalités graphiques informationnelles ou techniques de visualisation. Les interfaces multimodales sont objets d'études depuis que les concepteurs ont conçu des systèmes mettant en œuvre plusieurs dispositifs en entrée (permettant à l'utilisateur d'interagir avec le système). Les études en ingénierie de l'interaction Homme-Machine s'intéressent également à l'interaction en sortie (du système vers l'utilisateur) mais le terme multimodalité y est rarement appliqué. Le terme multimédia est plus souvent retenu, et désigne davantage les supports de l'information échangée (médias) plutôt que son contenu. Face à une terminologie mal assise, il convient d'organiser le savoir expérimental dans des cadres aussi rigoureux que possibles. De là, un espace de conception structurant permet de raisonner, comparer, évaluer les interfaces existantes ou à concevoir ainsi que d'organiser les concepts nécessaires à la mise en place de patrons de conception d'interfaces multimodales en sortie.

Après avoir assis la terminologie, nous présentons les espaces de conception les plus pertinents par rapport à notre cadre de conception. Un bilan analytique nous conduit à présenter notre propre contribution motivée, pour l'essentiel, par les problèmes de conception : un espace de composition de modalités et un espace de caractérisation d'une modalité pure ou composée. S'appuyant sur les concepts de nos deux espaces, nous définissons quatre patrons de conception d'interfaces multimodales en sortie.

1 Terminologie : modalité et multimodalité en sortie

La multimodalité en sortie caractérise la capacité d'un système interactif à exploiter plusieurs modalités de sortie, du système vers l'utilisateur, lors de l'interaction. La variété des définitions du terme modalité, laisse entrevoir la difficulté d'identifier une définition consensuelle pour la multimodalité en sortie. La diversité des points de vue justifie l'absence de consensus. A l'évidence, le terme modalité est tiraillé entre d'une part, les préoccupations des psychologues et des ergonomes qui visent à développer des théories explicatives ou prédictives sur les interfaces multimodales, et d'autre part, les considérations des concepteurs et informaticiens qui visent à définir des outils pour la conception et la réalisation de telles interfaces.

Face au flou d'une terminologie mal assise qui fléchit au gré des points de vue, il nous a semblé important de choisir et proposer un ensemble de termes qui couvrent raisonnablement notre domaine de réflexion et traduisent notre point de vue, la conception. Dans cette section, nous annonçons et justifions la définition du terme modalité, que nous affinons en trois niveaux d'abstraction, et du terme multimodalité en sortie que nous illustrons par des systèmes existants.

Pour une revue terminologique des termes modalité, média et mode nous encourageons le lecteur à se reporter à [Nigay 96].

1.1 Modalité

Parce que nos travaux sont dédiés à la conception d'interfaces multimodales, il convient d'adopter une définition de la modalité qui couvre les deux angles d'étude de la conception, l'utilisateur et le système interactif. A cette fin, nous adoptons la définition de Laurence Nigay [Nigay 96] : une modalité est définie comme un couple $\langle d, l \rangle$ où d désigne un dispositif physique et l , un langage d'interaction. Un langage d'interaction se définit par un vocabulaire d'éléments terminaux et une grammaire. Les éléments terminaux sont produits ou captés par les dispositifs d'entrée/sortie.

Cette définition constitue une façon de caractériser les échanges entre un système et son utilisateur car elle identifie et met en relation deux niveaux d'abstraction pertinents tant du côté humain que du côté système. Du côté humain, nous distinguons le niveau physique (ou matériel) qui fait appel à la perception et le langage d'interaction qui fait appel à la cognition. Du côté système, nous différencions le dispositif physique mis en jeu et le format des expressions échangées. Aussi une modalité définie comme le couple $\langle d, l \rangle$ nous permet d'envisager la définition d'un cadre de conception qui recouvre à la fois les perspectives utilisateur et système en identifiant deux niveaux d'abstraction, physique et logique. La Figure 1 montre que les deux niveaux d'abstraction cernés par les notions de dispositif et de langage d'interaction définissent deux points de contact entre le système et l'utilisateur.

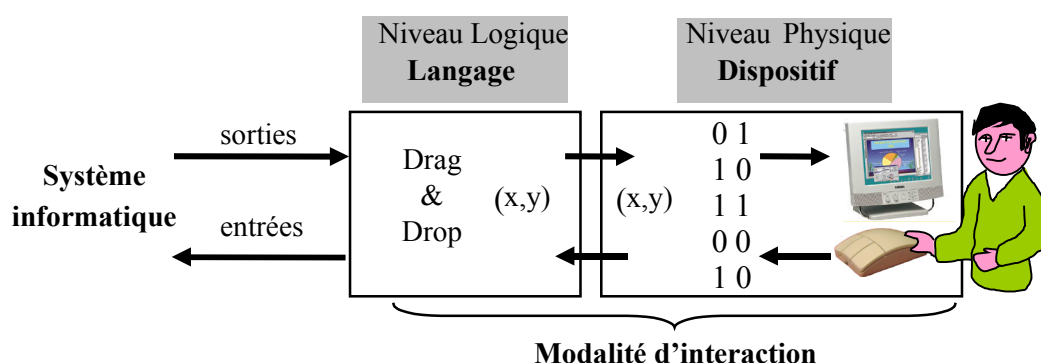


Figure 3 : Niveaux physique et logique d'une modalité d'interaction.

En sortie, le dispositif physique et le langage d'interaction forment les deux facettes complémentaires d'une expression échangée du système vers l'utilisateur : le dispositif définit la réalité physique observable tandis que le langage définit la structure c'est-à-dire les relations de

dépendance qu'entretiennent les constituants de l'expression. Dans l'échange d'expressions entre le système et l'utilisateur, nous distinguons donc le niveau articulatoire fixé par le dispositif, du niveau syntaxique défini par le langage. A plus haut niveau d'abstraction, le niveau sémantique est bien sûr à considérer, car il permet d'isoler l'information contenue dans l'expression, de la forme physique et logique de l'expression. Néanmoins la sémantique n'est pas un critère discriminant d'une modalité. En effet nous ne différencions pas deux modalités sur la base des informations véhiculées. Toutefois la composition de modalités ne peut s'affranchir de la notion fondamentale de complémentarité et de redondance du niveau sémantique, que nous développerons dans le paragraphe 4.5 (page 66) intitulé "Aspect sémantique de la composition de modalités". Nous justifions le classement de la Figure 4, en trois niveaux d'abstraction ordonnés, par le fait que la syntaxe du langage d'interaction structure les données brutes du niveau articulatoire et que la sémantique de l'expression repose sur la syntaxe du langage.

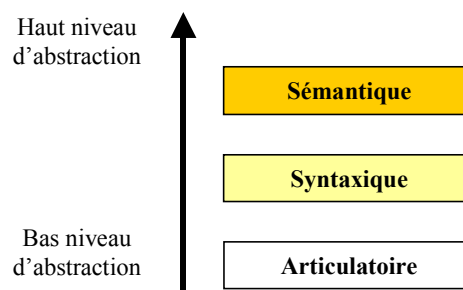


Figure 4 : Trois niveaux d'abstraction d'une modalité.

Nous pourrions considérer un niveau inférieur au niveau articulatoire que l'on baptiserait niveau électronique ou physique. Cependant ce niveau, matérialisation des données brutes en flux électriques, sort de notre cadre d'étude et de compétence.

Au niveau supérieur, nous pourrions ajouter le niveau pragmatique qui compléterait le niveau sémantique, en mettant la sémantique en situation. Pour certains, ce niveau est crucial pour la composition de modalités et en particulier la gestion des références comme la désignation d'un objet graphique (par clignotement de l'objet) complétée d'une phrase en langage naturel qui lui fait référence "Regarde là". Néanmoins les approches sont contradictoires, certains affirmant à l'opposé que le problème de la référence doit être traité au niveau précédent, celui de la sémantique qui livre le contenu propositionnel, clé en mains pour ainsi dire. De plus l'approche pragmatique ne se réduit pas à la détermination des actes de langage, mais porte davantage sur la récupération des intentions du locuteur : or notre locuteur est le système, dans l'étude des interfaces en sortie. Aussi pour l'étude de la multimodalité en sortie, nous ne considérons que le niveau sémantique, les références réduites aux co-références

étant traitées à ce niveau et détaillées selon ce que nous notons les aspects sémantiques de la composition de modalités (paragraphe 4.5, page 66).

1.2 Multimodalité en sortie

Pour appréhender la multimodalité en sortie, nous nous basons sur la définition d'une modalité de sortie (le couple <dispositif, langage>), le préfixe multi traduisant le nombre. L'étude de la multimodalité en sortie implique donc de considérer le nombre de dispositifs et de langages distincts constitutifs des passerelles logiques et physiques, du système vers l'utilisateur. Nous obtenons alors trois cas de multimodalité, comme le souligne l'espace M²LD [Nigay 94] appliqué aux interfaces en sortie et présenté à la Figure 5. Matérialisée par les trois cases grisées de la Figure 5, la multimodalité en sortie revêt les trois aspects suivants :

- ▶ Multidispositif et monolangage de sortie
- ▶ Multidispositif et multilangage de sortie
- ▶ Monodispositif et multilangage de sortie

Ces aspects de la multimodalité se retrouvent dans les travaux de Maybury [Maybury 99] sous des termes différents : ainsi pour Maybury les termes multimodal et multicode correspondent respectivement à multidispositif et multilangage.

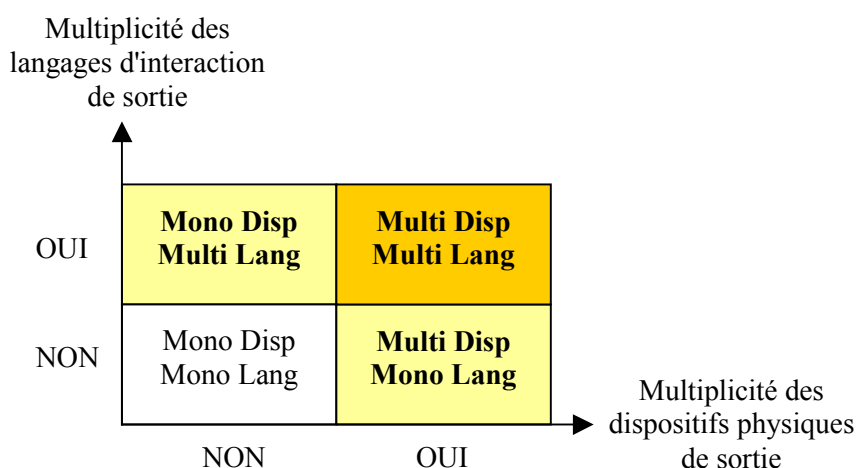


Figure 5 : M²LD, une classification des systèmes multimodaux en sortie selon la multiplicité des langages et des dispositifs de sortie. La valeur "non" sur un axe signifie que le système est "mono" pour cette dimension. La valeur "oui" correspond au cas "multi" [Nigay 94].

A l'opposé des interfaces en entrée, l'interaction multidispositif en sortie est un domaine assez pauvre à cause du manque de variété dans les dispositifs de sortie. Les manettes à retour d'effort ne sont utilisées que pour les simulateurs et les jeux. Les terminaux braille sont dédiés à

des interfaces pour des non-voyants. Les haut-parleurs constituent l'alternative la plus courante aux écrans (écran classique ou casque semi-transparent). Les modalités sonores, exploitant des haut-parleurs, soulèvent néanmoins des problèmes :

- ▶ Le volume des haut-parleurs peut être réglé (éventuellement à 0) par l'utilisateur, sans que le système en soit informé.
- ▶ Le volume de l'environnement sonore de l'utilisateur peut être plus élevé que celui de la modalité.
- ▶ Le contexte d'interaction de l'utilisateur (lieu de travail par exemple) peut lui imposer un volume très bas, voire nul.

Pour toutes ces raisons, les modalités sonores sont rarement exploitées dans l'interaction, excepté dans certains types de logiciels comme la création musicale où elles sont inévitables.

Dans les interfaces traditionnelles, les modalités sonores (parole ou son) sont généralement utilisées pour répéter l'information graphique (redondance et insistance). Nous notons néanmoins dans [André 98] l'usage de messages sonores en langage naturel qui complémente, explique et accompagne une information affichée à l'écran, selon une modalité graphique. Un autre exemple, le système CUBRICON [Neal 88], dédié au contrôle aérien tactique, affiche par exemple une table à l'écran accompagné d'un message vocal complémentaire : *"The corresponding table is being presented on the monochrome screen"*. CUBRICON, système précurseur de la multimodalité, est un exemple intéressant d'interaction multidispositif composé de haut-parleurs d'un écran monochrome et d'un deuxième écran en couleur. C'est l'un des rares systèmes multimodaux qui gèrent deux écrans.

Au contraire, dans les interfaces pour personnes handicapées de la vue, les modalités sonores constituent une solution lorsque les modalités graphiques ne sont pas utilisables [Truillet 99]. Par exemple les travaux de Thèse d'Elizabeth Mynatt [Mynatt 95] concernent la traduction temps réel de toutes interfaces graphiques sous X Window en interfaces sonores pour non-voyants : la traduction, dans ce projet intitulé Mercator, repose sur un modèle abstrait de l'interface graphique qui est élaboré dynamiquement. Il est intéressant de noter que Mercator intègre aussi un terminal braille, autre dispositif de sortie, pour la présentation de certaines informations textuelles comme des programmes. En effet, la synthèse de parole a été montrée non adaptée pour la lecture de programmes [Mynatt 95].

Les exemples ci-dessus d'interaction multidispositif en sortie, principalement composés de haut-parleurs, sont pour la plupart multilangage. Rares sont les interfaces multidispositifs et

monolangage. C'est le cas de systèmes qui n'autoriseraient par exemple que le langage naturel mais qui afficheraient le texte à l'écran ou émettraient une phrase sur le haut-parleur. Aussi, afin d'affiner le cas de l'interaction multidispositif et multilangage, il est intéressant d'étudier la relation existante entre un langage d'interaction donné et les dispositifs. Deux cas sont à considérer : soit le langage est assigné à un dispositif unique, soit le langage peut être utilisé avec des dispositifs différents. La relation entre un langage et les dispositifs est alors caractérisée comme suit :

$$\text{Langage d'interaction} \xrightarrow{1, n} \text{Dispositif(s) physique(s)}$$

Considérons le démonstrateur ICPplan, un éditeur de plans architecturaux qui intègre un système de reconnaissance et de synthèse de parole. ICPplan est l'un des premiers démonstrateurs multimodaux [Bourget 92]. En sortie, tout message oral en langage naturel, expliquant l'action qui vient d'être accomplie, est accompagné de sa représentation textuelle à l'écran. Le langage naturel est donc associé à deux haut-parleurs et l'écran (relation multivaluée). A l'opposé, la présentation graphique des plans est assignée à l'écran (relation monovaluée).

Tandis que l'interaction multidispositif implique fréquemment une interaction multilangage, le contraire est faux. En effet de nombreuses interfaces sont monodispositifs et multilangages. Plusieurs modalités graphiques et textuelles sont couramment mises en œuvre grâce au même dispositif, l'écran. Par exemple, nous distinguons une modalité différente pour chaque type d'affichage du contenu d'un répertoire proposé par l'explorateur de Windows [Windows] et reproduit à la Figure 6. Dans le système WIP [Walster 93], des explications pour utiliser une machine à café sont affichées sur l'écran selon deux modalités, l'une textuelle en langage naturel et l'autre schématique selon un langage graphique. WIP est un autre exemple de système monodispositif et multilangage.

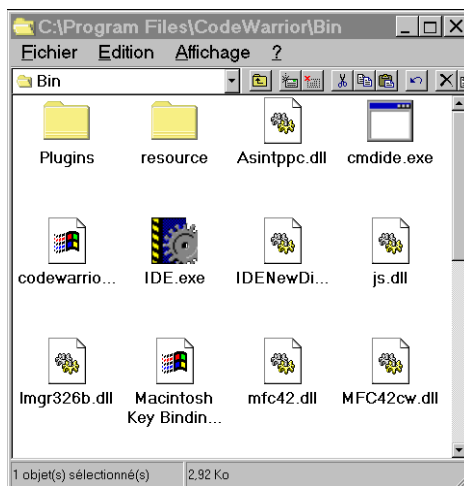


Figure 6a : Affichage d'une liste de grandes icônes dans Windows NT [Windows].

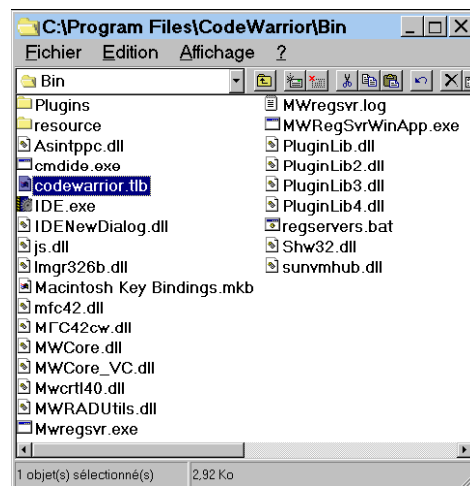


Figure 6b : Affichage d'une liste de petites icônes dans Windows NT [Windows].

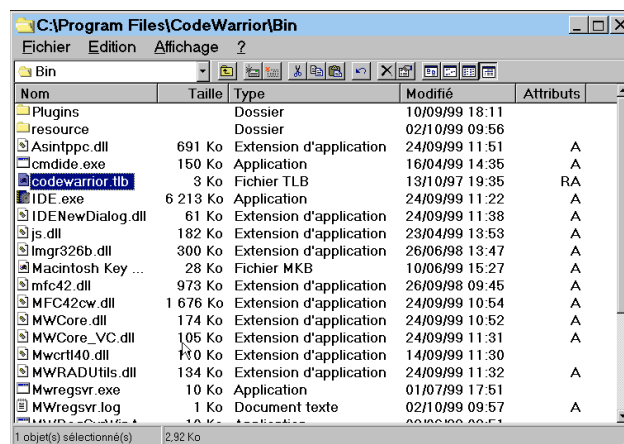


Figure 6c : Affichage d'une liste détaillée de petites icônes dans Windows NT [Windows].

En résumé, nous prenons comme point de départ la dualité substance/structure d'une expression véhiculée par une modalité : la substance dépend du dispositif physique support et la structure dépend d'un langage d'interaction. Ces notions nous permettent de mettre en exergue trois aspects de la multimodalité en sortie. Nous en retenons l'aspect le plus fréquent, l'interaction multilingage, qu'elle repose sur un ou plusieurs dispositifs physiques. Ce constat est à mettre en opposition avec l'interaction multimodale en entrée qui s'appuie le plus souvent sur l'interaction multidispositif [Nigay 96].

Nous venons de fixer la terminologie adoptée dans notre étude de la multimodalité en sortie. Il convient maintenant de relier ces termes en un espace utile à la conception d'interfaces

multimodales. Dans cet objectif, nous analysons les propositions de la littérature selon notre cadre de conception que nous précisons dans la section qui suit.

2 Conception d'interfaces multimodales

Nous abordons la conception d'interfaces multimodales en intégrant les deux niveaux d'abstraction d'une modalité de sortie : dispositif et langage. La mise en œuvre de la multimodalité en sortie se caractérise comme un processus de génération (langage) et de présentation (dispositif) d'un énoncé multimodal qui doit être l'image des données à présenter, adaptée au contexte d'interaction. Ce dernier peut par exemple être déterminé par l'état courant du dialogue, les capacités sensorielles et les préférences de l'utilisateur. Par rapport aux interfaces en entrée, nous constatons que la conception d'interfaces multimodales en sortie est un problème plus complexe à traiter. En effet dans le cas des entrées, le choix des modalités et de leurs usages revient couramment à l'utilisateur : les enjeux de conception résident alors dans la définition d'une panoplie de possibilités et dans la mise en place d'un mécanisme d'analyse des énoncés multimodaux d'entrée pour en déduire l'intention de l'utilisateur. A l'opposé, dans le cas de la multimodalité en sortie, le choix des modalités (génération) et leurs mises en œuvre (présentation) doivent être intégrés dans le système pour définir un énoncé multimodal en sortie. Ce processus de génération et de présentation, effectué par le système, revêt des caractéristiques différentes que nous développons dans le paragraphe suivant.

2.1 Typologie de la multimodalité en sortie

Nous discernons trois cas qui impliquent des processus de génération et de présentation distincts :

1. Le choix des modalités et de leurs usages a été figé par le concepteur. Le processus de génération et de présentation du système est conçu et développé selon ces choix de conception.
2. Le processus de génération et de présentation intègre un mécanisme de sélection des modalités et de leurs usages selon un ensemble de critères liés aux données collectées et maintenues dynamiquement par le système.
3. Le choix des modalités et de leurs usages est fixé par l'utilisateur. Le système offre alors un ensemble de possibilités quant aux choix des modalités et un mécanisme permettant à l'utilisateur d'effectuer ces choix.

Dans le cas 1, les choix des modalités et de leurs usages sont effectués lors de la phase de conception, tandis que dans le cas 2 ces choix sont réalisés à l'exécution pour tenir compte de

l'utilisateur en situation réelle. Le mécanisme de sélection repose alors sur des critères, liés à des données collectées dynamiquement comme la tâche courante entreprise par l'utilisateur. Dans le cas 3, les choix des modalités sont aussi accomplis à l'exécution, mais reviennent à l'utilisateur.

Les trois cas identifiés ne sont pas indépendants. En effet, comme l'explicite la Figure 7, les choix offerts à l'utilisateur ou effectués par le système sont limités par ceux figés par le concepteur. A la Figure 7, nous nommons ces trois types de multimodalité en sortie : aucune adaptation (choix des modalités figé par le concepteur), adaptativité (choix des modalités calculé par le système) et adaptabilité (choix effectué par l'utilisateur).

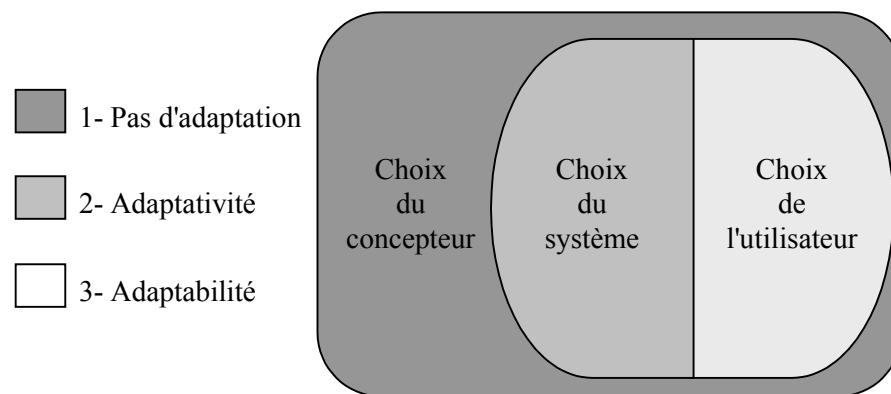


Figure 7 : Typologie de la multimodalité en sortie selon les acteurs du choix des modalités.

Nous illustrons les trois types de multimodalité en sélectionnant des systèmes multidispositif et/ou multilangage afin de souligner le fait que le choix des modalités peut concerner le processus de génération (choix du langage) et/ou le processus de présentation (choix du dispositif).

(1) Le système d'information sur les produits Sony, basé sur la métaphore conversationnelle [Takeushi 93], présente les réponses à des questions de l'utilisateur sous la forme de messages vocaux en langage naturel couplés avec des expressions faciales d'un assistant virtuel affiché à l'écran. Le système est donc multidispositif et multilangage et n'offre aucune adaptation. Le choix de la métaphore conversationnelle concrétisée par les haut-parleurs, langage naturel) et (écran, langage d'expressions faciales) ont été figés lors de la phase de conception.

(2) Les systèmes multimodaux en sortie adaptatifs diffèrent par les critères de sélection sur lesquels se base le choix des modalités et de leurs usages.

PDA-ITV [Robertson 96] est un système d'information sur des maisons à acquérir qui est multidispositif (écran d'un ordinateur de poche Newton, écran de la télévision interactive) et multilingage (description textuelle, carte, image haute et basse résolution). PDA-ITV constitue un exemple de système adaptatif à l'environnement physique. En effet, si l'utilisateur est chez lui et que la télévision est disponible, la description textuelle de la maison sélectionnée est affichée sur l'ordinateur de poche et une image haute résolution sur l'écran du téléviseur. Par contre, si l'utilisateur est en route pour visiter une maison, la description textuelle ainsi qu'une image basse résolution sont affichées sur l'écran de l'ordinateur de poche. Ainsi PDA-ITV est adaptatif à l'environnement physique, traduit ici par la disponibilité des dispositifs.

Un autre exemple de système adaptatif, le système COMET (Coordinated Multimedia Explanation Testbed) [Feiner 91], très proche du système WIP [Walster 93] présenté au paragraphe précédent, choisit dynamiquement le ou les langages d'interaction à employer pour fournir une réponse adaptée à l'utilisateur à propos de la réparation d'une radio. COMET est monodispositif (un écran) et multi-langage (langage naturel écrit et langage graphique). Comme dans le système PPP [Rist 97] (Personalized Plan-based Presenter) qui génère dynamiquement des présentations textuelles et graphiques incorporant un personnage de bande dessinée, la sélection des langages repose sur :

- ▶ les attributs sémantiques des données à présenter : par exemple des données en relation avec des caractéristiques physiques comme la forme ou la taille sont présentées graphiquement tandis que des données décrivant des propriétés abstraites sont présentées textuellement ;
- ▶ les actes de communication : l'énoncé multimodal est représenté par un ensemble d'actes de communication comme par exemple informer, illustrer, détailler ou encore mettre en évidence. Ces actes de communication sont ensuite associées à des langages d'interaction. Par exemple, "mettre en évidence" dans le système PPP [Rist 97] est associé à une représentation graphique d'un personnage de bande dessinée en train de montrer du doigt l'objet à mettre en évidence. A ce personnage est associé un langage de postures.

Dans les systèmes comme COMET, WIMP et PPP, les graphismes représentent l'objet manipulé et les actions prescrites (flèche ou animation). Lorsque les langages graphiques sont des graphes, courbes, tableaux ou des réseaux, c'est-à-dire des graphismes informationnels et non figuratifs comme précédemment, les critères de sélection sont différents. Ainsi dans le système SAGE, [Roth 90a] la sélection des langages graphiques et textuelles pour fournir une explication à l'utilisateur repose :

- sur le type de données à présenter : dans [Roth 90b] une liste de caractéristiques est fournie pour le cas de données relationnelles comme le type de relation d'ordre entre les données qui peut être quantitative ou encore nominale ;
- sur les tâches que l'utilisateur pourra effectuer efficacement sur l'énoncé multimodal généré afin d'atteindre son but : des exemples de tâches incluent "observer une valeur d'un élément", "comparer deux valeurs" ou encore "étudier la corrélation de l'évolution de données" [Roth 90b].

Ces systèmes multimodaux adaptatifs, générant des énoncés multimodaux graphiques et textuels, ont en commun l'exploitation des caractéristiques des données à présenter pour effectuer le choix des langages d'interaction. Le deuxième critère de choix est néanmoins différent et dessine les contours de deux approches de recherche qui cohabitent : la première consiste à baser la sélection sur les actes de communication, par exemple "mettre en évidence" tandis que la deuxième repose sur les tâches-utilisateur comme critères de sélection, par exemple "comparer deux valeurs". Dans des travaux récents [Kerpedjiev 00], nous trouvons une tentative de couplage de ces deux approches, en affinant les actes de communication en termes de tâches utilisateur dans le processus de génération c'est-à-dire de choix des langages.

(3) Comme pour les systèmes adaptatifs, les systèmes adaptables diffèrent selon les critères de sélection qui sont dans le cas de l'adaptabilité proposés à l'utilisateur.

Un premier exemple de système adaptable est le système AudioStreamer [Schmandt 95], système multidispositif avec des haut-parleurs placés autour de la tête de l'utilisateur. AudioStreamer permet d'entendre trois sources d'information en même temps. Une seule est le centre d'intérêt de l'utilisateur à un instant donné. Tout en écoutant une source sonore, l'utilisateur peut néanmoins entendre les deux autres sources (phénomène appelé "effet cocktail party"). Elle/il peut changer d'attention à tout moment en tournant la tête dans la direction de la source d'intérêt. Lors d'un changement d'attention, le volume de la source constituant le nouveau centre d'intérêt est augmentée tandis que le volume des autres sources diminue. Le volume du haut-parleur est un paramètre du dispositif physique et nous considérons donc AudioStream comme un système adaptable, l'utilisateur choisissant dynamiquement un dispositif en tournant la tête.

Les travaux consistant à rendre utilisable une application graphique classique par des non-voyants s'inscrivent dans l'étude de l'adaptabilité. Par exemple le générateur d'interfaces HOMER [Savidis 95] permet de générer une interface graphique ainsi qu'une version pour des non-voyants exploitant la synthèse de parole et un terminal braille. Plus qu'une traduction directe, les métaphores d'interaction sont différentes, et les deux versions de l'interface

emploient outre des dispositifs différents, des langages d'interaction distincts. Ainsi la même application peut être utilisée par des utilisateurs voyants et non-voyants.

L'objectif étant d'illustrer les trois types de multimodalité, chaque exemple illustre un type de multimodalité. Néanmoins les types de multimodalité ne sont pas exclusifs à plusieurs titres :

- Tout d'abord, et à des fins analytiques, nous avons réduit l'adaptation aux deux cas d'adaptativité et d'adaptabilité. Nous les considérons comme les deux extrêmes d'un continuum. Une analyse plus fine de l'adaptation nécessite de considérer en plus des deux acteurs, le système et l'utilisateur, les différentes étapes : initiative (qui déclenche l'adaptation), génération de propositions (qui génère la ou les solutions d'adaptation), décision (qui décide de l'adaptation). Ainsi une valeur intermédiaire entre l'adaptativité et l'adaptabilité serait une interface qui propose à l'utilisateur une adaptation automatique, l'utilisateur gardant l'initiative. Pour une étude détaillée de l'adaptation, nous encourageons le lecteur à se reporter à [Dieterich 93].
- De plus il est important de comprendre qu'un système peut à la fois ne pas offrir d'adaptation pour certaines données à présenter par exemple, être adaptatif pour d'autres cas et enfin adaptable par l'utilisateur.

Pour les trois types de multimodalité, la définition d'un énoncé multimodal repose sur des critères de choix des modalités et de leurs usages. Les critères de sélection, les modalités et leurs usages sont donc les éléments émergents dans la conception. L'objet du paragraphe suivant est d'organiser ces éléments dans un cadre de conception.

2.2 *Cadre de conception*

Les critères de choix des modalités constituent une facette de la conception importante. Ces critères sont à la fois utiles lors de la conception (modalité de type 1 à la Figure 7), exploités dans un processus dynamique de génération d'énoncés multimodaux (modalité de type 2 à la Figure 7) et présentés à l'utilisateur qui souhaite adapter l'interface (modalité de type 3 à la Figure 7). Ces critères sont de natures différentes comme nous l'avons illustré par des systèmes existants : critères liés aux caractéristiques des données à rendre perceptibles, à l'état du dialogue (acte de communication), à la tâche courante de l'utilisateur, et au modèle de l'utilisateur tel que ses capacités sensorielles.

Afin que le processus de définition d'énoncés multimodaux, basé sur les critères ci-dessus, ne soit pas dépendant des modalités, il est important que les choix n'aboutissent pas à

des modalités particulières mais à des classes de modalités. Pour effectuer un choix, les modalités doivent donc être caractérisées et donc classées.

Enfin l'usage de plusieurs modalités nécessite de caractériser la relation existante entre les modalités choisies. Les caractéristiques des relations entre plusieurs modalités définissent l'ensemble des compositions possibles de modalités.

En synthèse, nous organisons l'ensemble des éléments considérés dans la conception d'une interface multimodale dans notre cadre de conception de la Figure 8.

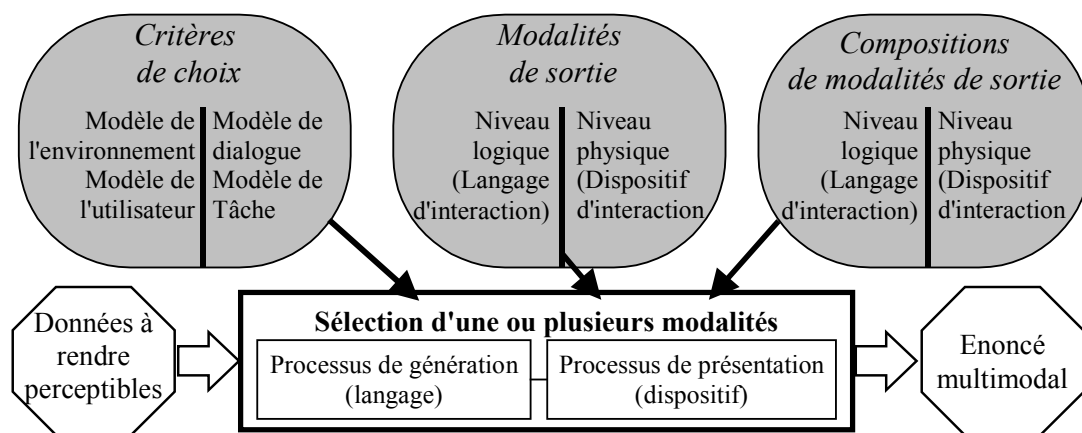


Figure 8 : Cadre de conception de la multimodalité en sortie.

Au sein de notre cadre de conception, la caractérisation des modalités de sortie et de leurs compositions définit de façon générique l'espace des possibilités. Les critères de choix identifiés de nature très différente ainsi que les caractéristiques des données à rendre perceptibles cernent le contexte d'interaction. La sélection d'une solution dans l'espace des possibilités pour un contexte d'interaction donné doit être motivée par l'utilisabilité générale de l'énoncé multimodal. A cette fin, nous déclinons l'utilisabilité en termes de critères d'ergonomie. Notre approche de conception repose sur la liste des critères établis par notre équipe et présentés en Annexe 1. Cette liste se décompose en deux sous-listes, les critères de souplesse et les critères de flexibilité. Dans le Chapitre II, des exemples de critères sont fournis au paragraphe 4.1, page 28.

En résumé nous cernons la conception d'une interface multimodale en sortie au processus de sélection d'une ou plusieurs modalités. Ce processus est motivé par les critères d'ergonomie pour un contexte d'interaction donné. Notre cadre de conception contribue donc à clarifier l'espace problème en le structurant d'éléments qu'il convient de considérer lors de la

conception d'une interface multimodale en sortie. Il identifie et organise aussi les éléments nécessaires à la définition de patrons de conception d'interfaces multimodales.

2.3 *Cadre de conception et patrons de conception*

Comme nous l'avons rappelé au chapitre II, les patrons de conception offrent des solutions réutilisables à des problèmes identifiés récurrents, afin d'aider le concepteur. Situés dans notre cadre de conception, les patrons définissent des solutions réutilisables en préconisant des choix de modalités. En faisant référence à notre format de patron, les éléments de conception identifiés dans ce chapitre permettent d'affiner le contenu des rubriques d'un patron pour la conception d'interfaces multimodales en sortie:

- **Description du problème et du contexte** : Cette rubrique décrit le problème traité et délimite le cadre dans lequel le patron peut être appliqué. Cette rubrique définit tous les facteurs liés au problème c'est-à-dire les critères qui vont influencer les choix des modalités, la solution au problème abordé. Cette rubrique s'appuie donc sur les critères de sélection qui forment ce que nous avons dénoté le contexte d'interaction.
- **Motivations** : Cette rubrique justifie le choix de la ou des modalités, préconisé dans la rubrique Solution. L'argumentation repose donc sur les critères ergonomiques associés au patron. Comme expliqué au Chapitre II, chaque patron est associé à un ou plusieurs critères d'ergonomie.
- **Solutions** : La solution consiste à définir les modalités et leurs usages choisis pour répondre au problème traité par le patron. Afin que la solution soit réutilisable, le choix ne concerne pas des modalités particulières mais des classes de modalités et de compositions de modalités.

Nous proposons donc une approche située dans notre cadre de conception pour définir les patrons de conception. Ces derniers définissent une aide pour réaliser le choix des modalités et de leurs usages afin de produire un énoncé multimodal. Nous démontrerons la validité de notre approche en définissant des patrons de conception à la section 6 (page 72). Pour la complétude de notre approche, il convient maintenant de caractériser les modalités et leurs usages. La généralité nécessaire des patrons repose sur cette caractérisation. Dans cet objectif, nous analysons les propositions de la littérature. Un bilan analytique nous conduit à présenter notre propre contribution, un espace de composition des modalités et un espace de caractérisation.

3 *Espaces de conception existants*

Dans les sections qui suivent, nous présentons les espaces de conception existants selon nos deux perspectives : la composition de modalités et la caractérisation d'une modalité simple ou composée.

3.1 *Espaces de composition de modalités*

3.1.1 *Propriétés CARE*

Les propriétés CARE sont au nombre de quatre et définissent des liens entre les modalités, dans le contexte d'une interaction multimodale. C, A, R et E sont les initiales de quatre propriétés : Complémentarité, Assignment, Redondance et Equivalence. Ces notions ont été initialement présentées par Jean Claude Martin [Martin 94], [Martin 99]. Dans [Coutaz 94], Joëlle Coutaz et Laurence Nigay leur donnent une facture formelle. Enfin dans [Coutaz 95] les auteurs montrent leur utilité pour classer et surtout évaluer les systèmes multimodaux, par l'analyse des propriétés CARE du point système (S-CARE, "System CARE") et du point de vue utilisateur (U-CARE, "User CARE").

A l'origine, ces propriétés ont été introduites pour caractériser l'interaction multimodale en entrée. Par exemple, les propriétés U-CARE [Coutaz 95] décrivent les choix des modalités et de leurs usages effectués par l'utilisateur pour communiquer avec un ordinateur. Les éléments de la définition formelle des propriétés sont néanmoins tout à fait applicables à l'interaction multimodale en sortie : état, but, modalité et relation temporelle.

- ▶ Un état est un vecteur de variables observables, un ensemble de propriétés mesurables à un instant donné qui caractérise une situation.
- ▶ Un but est un état qu'un agent veut atteindre. Un agent peut être l'utilisateur, le système ou seulement un composant du système. Généralement, un agent est une entité capable d'initialiser le déroulement d'une action. Dans notre étude, l'agent est le système.
- ▶ Une modalité est une méthode d'interaction utilisée par un agent pour atteindre un but. Afin d'estimer le pouvoir d'expression d'une modalité, dans [Coutaz 95] les auteurs définissent une fonction d'accessibilité appelée "Reach" qui prend la valeur vraie, pour deux états, si une modalité permet de passer directement de l'un à l'autre.
- ▶ Une relation temporelle caractérise l'utilisation dans le temps d'un ensemble de modalités. L'utilisation peut être simultanée ou en séquence.

Aussi, dans ce mémoire, nous ne détaillerons pas la définition formelle des propriétés CARE [Coutaz 95] qui reste valable pour la multimodalité en sortie. Dans les sections

suivantes, nous définissons informellement ces propriétés appliquées à l'interaction en sortie. L'assignation et l'équivalence caractérisent la portée des choix en matière de modalités tandis que la redondance et la complémentarité qualifient la combinaison des choix.

3.1.1.1 *Assignation*

Cette propriété exprime l'obligation pour un état de départ et d'arrivée fixés d'utiliser une modalité. Cette propriété exprime donc l'absence de choix pour réaliser une tâche, consistant à passer d'un état de départ à un état d'arrivée. Ceci n'est pas forcément un point négatif pour l'interaction. Tout d'abord l'assignation d'une modalité en sortie à une tâche permet de garantir une stabilité de la forme des expressions de sortie. L'assignation dénote cependant un manque évident de souplesse mais peut néanmoins contribuer à la robustesse en sortie. L'exemple le plus fréquent est le message d'erreur. Ce dernier doit pouvoir être produit quel que soit l'état d'instabilité du système, comme par exemple la déficience du système de gestion du son ou du graphisme. La solution adoptée par Apple [MacOS] est l'utilisation du "debugger" en mode texte pour les messages d'erreur.

3.1.1.2 *Equivalence*

L'équivalence de modalités se définit pour un état de départ et d'arrivée fixés : l'équivalence d'un ensemble de modalités est vérifiée si chaque modalité permet d'atteindre l'état d'arrivée depuis l'état de départ en une seule étape. L'équivalence et l'assignation soulèvent le problème du choix des modalités. Dans le cas de l'interface en sortie, ce problème va au delà de l'existence ou non d'un choix. La relation d'équivalence permet de créer des classes de modalités, qu'il est envisageable de mettre en œuvre pour un état de départ et d'arrivée fixés. Nous pouvons par exemple instancier cette notion générale de passage d'un état de départ à un état d'arrivée en termes d'actes de communication [Rist 97] ou de tâches [Roth 90b] : ce sont deux critères de choix dynamique de modalités. Ainsi deux modalités peuvent être équivalentes pour un acte de communication ou une tâche.

3.1.1.3 *Redondance*

La redondance dénote l'utilisation séquentielle ou parallèle de plusieurs modalités équivalentes. En entrée, la redondance d'information en provenance de l'utilisateur implique la prise en compte d'une seule des modalités par le système, l'autre pouvant éventuellement contribuer à désambiguïser l'expression obtenue. En sortie, la redondance est un choix de conception liée aux deux critères d'ergonomie d'observabilité et d'insistance. En effet si le concepteur a choisi de montrer de différentes manières une même information, cela a pour conséquence de renforcer son observabilité et constitue aussi un moyen pour attirer l'attention

de l'utilisateur. La saturation de tous les canaux sensoriels de l'utilisateur par exemple renforce les chances de perception de l'information : cet assertion peut être expliquée avec la théorie ICS [Barnard 93], qui modélise l'activité mentale de l'utilisateur en termes de flux de données entre des sous-systèmes dédiés à la perception (sous-systèmes perceptifs), la cognition (sous-systèmes centraux) et l'action (sous-systèmes effecteurs).

3.1.1.4 Complémentarité

La complémentarité entre les modalités d'un ensemble M exprime le fait que pour passer d'un état à un autre, il faille utiliser toutes les modalités de l'ensemble M. Cela signifie qu'aucune des modalités de l'ensemble M ne suffise à elle seule pour passer de l'état de départ à l'état d'arrivée. Toutefois, il n'est pas exclu qu'une modalité non contenue dans M puisse permettre cette transition. Les déictiques, caractérisés par des références entre modalités, constitue un exemple très étudié d'utilisation complémentaire de modalités. Comme pour la redondance, l'utilisation complémentaire de modalités peut être parallèle ou séquentielle. En sortie, la complémentarité implique que l'utilisateur établisse les liens entre les informations présentées selon des modalités différentes : il doit combiner les informations perçues pour interpréter l'expression de sortie. La théorie ICS [Barnard 93] permet de déterminer les conditions que les différentes sources d'information doivent respecter afin d'être interprétées comme un tout cohérent. ICS distingue plusieurs types de combinaisons d'informations perçues et donne des critères pour que ces combinaisons soient possibles. La combinaison d'informations distinctes est principalement effectuée par les sous-systèmes centraux (cognition). Les informations susceptibles d'être combinées peuvent provenir soit des sous-systèmes perceptifs (perception), soit des sous-systèmes centraux. Différentes sources d'informations perçues simultanément peuvent être combinées à différents niveaux cognitifs : au niveau capteur, dans les sous-systèmes perceptifs, dans les sous-systèmes centraux. Nous mettons en relation ces trois niveaux cognitifs de combinaison avec les trois niveaux, articulatoire, syntaxique et sémantique que nous avons identifiés pertinents à considérer dans une modalité (paragraphe 1.1). Les conditions de combinaison sont présentées en détail dans [Salber 95a]. Nous ne retenons ici que les règles essentielles :

- Combinaison au niveau capteur : Les limitations intrinsèques de nos sens ou certaines de leurs particularités permettent une perception globale d'informations discrètes. Par exemple, l'image sur un écran est perçue comme un tout alors qu'elle est composée d'un ensemble de points. La persistance rétinienne nous fait voir l'image cinématographique comme un mouvement continu alors qu'elle est constituée de 24 images par seconde.

- Combinaison dans les sous-systèmes perceptifs : Sur un même canal sensoriel, nous pouvons combiner différentes informations à condition qu'elles soient liées par une structure commune (similarité, symétrie, proximité) : on voit l'ensemble des oiseaux comme un tout, non pas chaque oiseau individuellement [Barnard 93]. Le rapprochement avec la théorie Gestalt est ici immédiat.
- Combinaison dans les sous-systèmes centraux : ICS prédit que la combinaison d'informations provenant d'un ou plusieurs canaux sensoriels dans les niveaux supérieurs centraux ne peut avoir lieu que si les flots d'information aux entrées du sous-système considéré sont cohérents. Un contre-exemple est donné par les chaînes de télévision dites "mosaïque" qui affichent simultanément plusieurs chaînes sur un même écran. Si le son correspondant à la mosaïque est celui d'une des chaînes affichées, il est très difficile, voire impossible, d'identifier à quelle image correspond le son entendu. L'attention du spectateur, passant d'une image à l'autre, définit un flot d'information visuelle incohérent avec le flot d'information auditive : par la suite, la combinaison entre les informations visuelles et auditives ne peut s'effectuer dans les sous-systèmes centraux et donne lieu à une interprétation erronée : c'est le cas de l'effet McGurk. De plus ICS prédit que les différents flots d'information doivent être présents simultanément aux entrées du sous-système, avec une marge de tolérance qui peut être calculée expérimentalement et qui dépend des informations considérées (exemple de la désynchronisation de la bande-son d'un film).

En appliquant ICS à la perception d'informations complexes, il est donc possible de déterminer des critères afin que des informations complémentaires puissent être perçues et interprétées correctement. ICS montre aussi que la combinaison d'informations est un phénomène complexe qui peut intervenir à différents niveaux d'abstraction. Nous aborderons ces niveaux d'abstraction dans notre espace de composition à la section 4.

En synthèse, les propriétés CARE définissent deux types de composition de modalités, la redondance et la complémentarité. Nous avons montré que nous pouvons les affiner en considérant différents niveaux d'abstraction issus de la théorie ICS de psychologie cognitive. Nous développerons ce point dans notre espace de composition. L'espace TYCOON ci-après complète CARE en définissant deux autres types de composition.

3.1.2 Espace TYCOON

Dans l'espace TYCOON [Martin 99], une modalité est modélisée par un processus informatique d'analyse (entrée) ou de synthèse (sortie). Nous avons détaillé le processus de synthèse dans le paragraphe 1.1. La multimodalité est alors définie par la coopération entre

plusieurs modalités ou processus. Le cadre d'étude TYCOON identifie six types de coopération entre processus. Les quatre coopérations désignées par Complémentarité, Spécialisation, Redondance et Equivalence définissent les fondements des propriétés CARE. La concurrence, cinquième type de coopération, caractérise des modalités utilisées de façon indépendante. Nous voyons dans la concurrence l'opposé de la complémentarité. Le sixième type de coopération est le transfert, caractérisant deux modalités, les expressions de la première modalité servant d'entrées au processus de génération et de présentation associé à la deuxième modalité. Dans [Martin 99], l'exemple du système de génération d'animations d'un agent virtuel à l'écran à partir d'une description textuelle en langage naturel d'une explication illustre la coopération par transfert. En appliquant notre définition d'une modalité (le couple (d, l)), nous décrivons la modalité obtenue par : ((écran, langage de postures), langage naturel). Par cet exemple, nous voyons que le transfert se décrit par une composition récursive. Aussi, au sein de l'équipe IIHM, nous désignons les propriétés CARE par CCARE pour inclure cette notion de composition récursive.

Plus qu'une typologie des relations entre les modalités, TYCOON établit des liens entre les types de coopération et leurs buts. Un exemple de but est la rapidité d'apprentissage. Il a été montré expérimentalement que l'usage redondant d'un texte affiché à l'écran et d'un message sonore accroît la rapidité d'apprentissage [Martin 99]. Dans notre cadre de conception, ces buts se traduisent par les critères d'ergonomie.

3.1.3 Espace MSM

L'espace MSM [Nigay 96] est plus général que CARE et TYCOON en définissant des axes pertinents pour la conception de systèmes multimodaux. Nous le présentons car il affine la notion de parallélisme et met en évidence des niveaux d'abstraction que nous avons souligné importants pour affiner les types de composition de CARE et TYCOON. MSM propose les six axes reproduits à la Figure 9, comme cadre d'étude de la multimodalité.

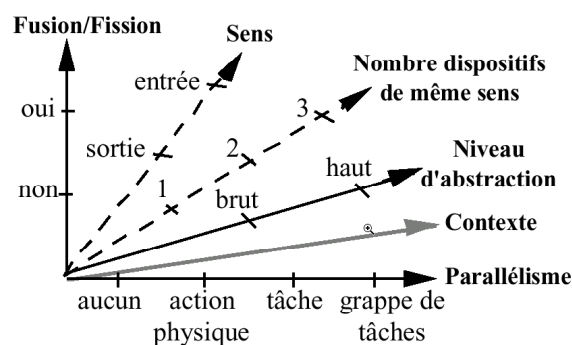


Figure 9 : Six axes de l'espace MSM [Nigay 96].

Cet espace adopte le point de vue du système et non de l'utilisateur. Il concerne l'interaction en entrée et en sortie. Nous ne détaillons ici que les axes qui traduisent des compositions de modalités de sortie. Sur l'axe Fusion/Fission, bien que la fusion s'applique aux interfaces en sortie, il convient pour l'étude de la composition de modalités en sortie de comprendre la fission d'information, chacune des parties étant véhiculée selon des modalités différentes. Les deux valeurs, oui et non de cet axe, traduisent respectivement la complémentarité (CARE) et la concurrence (TYCOON). L'axe Parallélisme décrit l'usage parallèle de plusieurs modalités à des niveaux de granularité différents. Cet axe affine la relation temporelle maintenue entre plusieurs modalités. Cette caractéristique temporelle est présente dans CARE et TYCOON, sans y être détaillée. Sur l'axe "Niveau d'abstraction", nous trouvons deux valeurs, brut et haut, qui correspondent aux niveaux articulatoire et sémantique de notre définition d'une modalité (paragraphe 1.1). Nous préconisons l'ajout d'une valeur intermédiaire, correspondant à notre niveau syntaxique. Cet axe "Niveau d'abstraction" traduit la volonté de considérer une fission à plusieurs niveaux d'abstraction.

3.1.4 Synthèse

Nous avons présenté trois contributions, les propriétés CARE et les espaces TYCOON et MSM, selon notre point de vue : la composition de modalités. Ces espaces sont complémentaires. TYCOON propose deux types de composition en plus de celles identifiées par les propriétés CARE. De ce bouillon conceptuel, nous retenons que l'assignation et l'équivalence traduisent l'existence de choix. Le transfert dans TYCOON ou composition récursive de modalités dans l'extension CCARE constitue un constructeur de modalités, le résultat étant une seule modalité construite et non plusieurs modalités composées. Trois types de composition de modalités sont identifiées, la redondance, la complémentarité et la concurrence. L'espace MSM met en évidence des niveaux d'abstraction pour affiner la complémentarité et la concurrence. Pour fixer ces niveaux d'abstraction, nous préconisons d'appliquer les trois niveaux, articulatoire, syntaxique et sémantique d'une modalité que nous avons définis au paragraphe 1.1. Enfin MSM met en évidence des niveaux de parallélisme qui affinent la relation temporelle que peuvent entretenir deux modalités composées. Nous organisons et étendons ces types de composition dans notre espace de composition à la section 4. Nous concluons d'abord notre revue des espaces de conception existants en étudiant ceux qui contribuent à caractériser une modalité. Nous avons résolument choisi d'étudier les compositions de modalités avant les caractéristiques d'une modalité puisque celles-ci peuvent concerner une modalité simple, construite ou composée.

3.2 Espaces de caractérisation d'une modalité

3.2.1 Systèmes humains de perception

Une caractéristique fréquemment utilisée dans l'étude de la multimodalité est le sens humain mis en jeu pour percevoir l'information véhiculée par la modalité : le sens visuel, auditif, tactile, olfactif et gustatif. Comme nous l'avons déjà illustré par des systèmes existants, les sens visuel et auditif sont les plus sollicités dans les interfaces actuels, le sens tactile étant moins exploité. Ce constat n'est pas surprenant puisque l'humain perçoit plus de 70% des informations sensorielles par le canal visuel [Ware 96]. Enfin les interfaces impliquant les sens olfactif et gustatif en sont à leurs balbutiements. Il existe par exemple le dispositif OLFACOM [Olfacom] qui est un générateur de senteurs piloté par un ordinateur. Ce type de dispositif peut trouver des domaines d'application variés comme les jeux ou le commerce électronique.

A titre illustratif, nous considérons l'espace de conception des interfaces en sortie de D. Frohlich. [Frohlich 91] qui met en exergue les systèmes humains de perception. L'espace de Frohlich de la Figure 10 décrit selon cinq facettes les modalités de sortie. L'une des facettes est le sens humain, désigné par le canal de communication de l'utilisateur. Les sens humains considérés sont l'ouïe, la vue et le toucher. Il y a correspondance biunivoque entre le Canal Humain et le Canal Système, deuxième facette d'une modalité. Par exemple à l'ouïe correspond le canal système audio. La facette Style désigne des techniques reconnues d'interaction comme le langage naturel ou les icônes. Les styles sont organisés en deux modes, quatrième facette d'une modalité. Le mode, langage ou action, désigne les deux métaphores d'interaction, respectivement la métaphore conversationnelle et la métaphore du monde réel (manipulation directe). La facette Media, cinquième facette d'une modalité, est définie comme un système représentationnel pour l'échange d'information, avec pour exemple la parole ou le graphisme. Nous constatons là encore une correspondance biunivoque entre le couple (Style, Canal) et Media.

Une analyse détaillée de cet espace est disponible dans [Nigay 94].

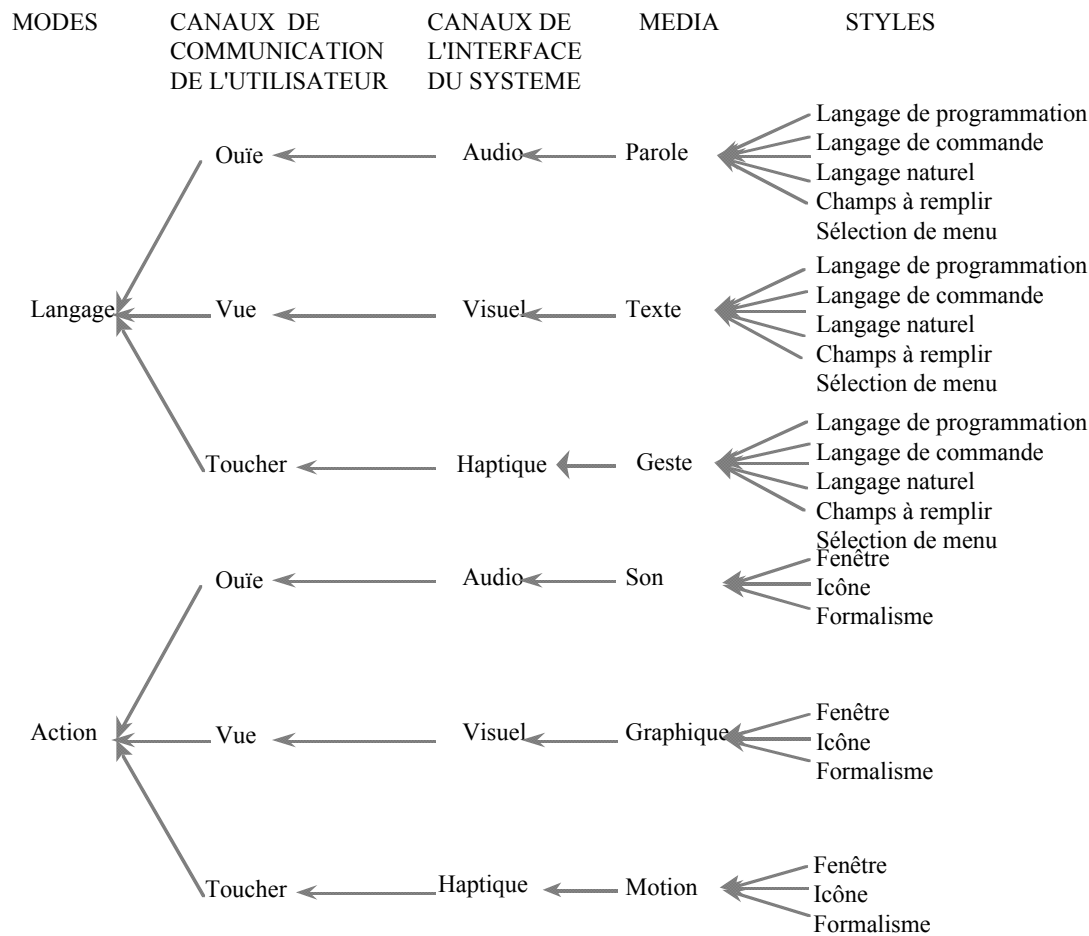


Figure 10 : L'espace de conception des interfaces en sortie de Frohlich [Frohlich 91].

De l'étude des liens entre les cinq facettes d'une modalité, nous déduisons de cet espace que deux facettes suffisent à caractériser une modalité :

- L'une décrite par le Style puisqu'il implique le Mode,
- L'autre décrite par le Canal Humain ou le Canal Système, puisqu'il existe une correspondance biunivoque entre ces trois facettes.

Ces deux facettes sont en accord avec notre définition de la modalité : le Canal Humain ou le Canal Système correspond au dispositif tandis que le Style définit le langage. La facette Style repose sur des exemples de technique d'interaction et manque de généralité. La complétude de ces styles est donc à mettre en cause. Nous retenons néanmoins pour notre étude la facette Canal comme une caractéristique des dispositifs physiques. Trois classes de dispositifs sont définies et liées aux sens humains. Un affinement de ces classes consisterait alors à exploiter les théories de psychologie de la perception. Par exemple, le toucher peut se décomposer en trois composantes, le tactile, la proprioception et la kinesthésie. Tandis que le

tactile lié à la perception de la pression nous permet de sentir des textures par exemple, la proprioception concerne l'acquisition des formes des objets par le tactile et la kinesthésie la sensation de poids et de résistance [Cadoz 94].

Le sens humain constitue donc une caractéristique des dispositifs. La revue des dispositifs existants établie par Colin Ware [Ware 96] réaffirme cette assertion, les dispositifs étant classés selon les cinq sens. La taxonomie suivante de Bernsen est complémentaire et caractérise le langage d'interaction d'une modalité.

3.2.2 *Taxonomie de Bernsen*

La taxonomie établie par Bernsen [Bernsen 93] est la seule qui est dédiée aux modalités de sortie. Pour cela, l'auteur identifie des propriétés des modalités. Ainsi dans son espace de conception, une modalité est définie par deux traits : le média et le profil. Le media fait référence au sens perceptif humain mis en œuvre, présenté au paragraphe précédent. Le profil se décline selon quatre propriétés. Chaque propriété a une valeur booléenne :

- ▶ Statique / dynamique,
- ▶ Linguistique / non linguistique,
- ▶ Analogique / non analogique,
- ▶ Arbitraire / non arbitraire.

La propriété "statique/dynamique" repose sur la présence ou non d'une dimension temporelle dans la communication. Nous notons que la dimension spatiale, souvent liée à la dimension temporelle, est absente de la taxonomie de Bernsen. Pourtant les modalités liées à un écran peuvent être mono, bi ou tridimensionnelles. De plus des recherches sont menées sur la spatialisation du son avec des dispositifs sonores dédiés [Tsingos 98] utilisant un ensemble de haut-parleurs placés dans l'espace autour de l'utilisateur. Enfin les dispositifs à retour d'effort se caractérisent aussi par un nombre de degrés de libertés [Cadoz 94] que nous pouvons décrire comme autant de dimensions spatiales.

Une modalité "linguistique" repose sur un système structuré de symboles remplissant une fonction de communication. Ce système définit le vocabulaire et la syntaxe du langage. Un langage possède aussi des règles de composition permettant de définir les énoncés corrects à partir du vocabulaire. Bernsen ne fait aucune distinction de complexité entre les différents langages possibles. Une telle distinction trouverait son utilité dans l'évaluation du temps d'apprentissage d'une interface. Le langage naturel (le Français par exemple) est un exemple de langage qui peut être utilisé par une modalité qualifiée de linguistique. En informatique

(reconnaissance ou génération) les concepteurs utilisent généralement un sous-langage plus simple du langage naturel noté langage pseudo naturel.

La propriété "analogique" repose sur la notion de ressemblance avec la réalité. Elle est à rapprocher du concept de métaphore du monde réel très utilisé en IHM. La propriété "analogique" et la précédente "linguistique" désignent les deux modes dans l'espace de Frohlich, respectivement, action et langage. Par exemple l'icône de la Figure 11a est analogique puisqu'elle représente un pinceau. Au contraire l'icône de la Figure 11b est non-analogique et contient des indices sémantiques abstraits comme les connecteurs et boîtes d'un schéma. Pour les modalités graphiques, cet propriété permet de distinguer respectivement les modalités figuratives (un dessin de maison) de celles informationnelles (un graphe). Une modalité auditive peut aussi être analogique : usage de sons du monde réel dans une interface.

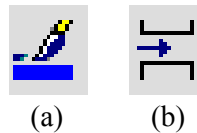


Figure 11 : Exemples de deux icônes (a) l'une analogique, (b) l'autre non-analogique [Word].

Intuitivement une modalité analogique semble une solution à favoriser. Nous distinguons deux cas où l'usage d'une modalité analogique risque d'engendrer des problèmes d'utilisabilité. Le premier cas réside dans l'usage d'une modalité dont l'analogie n'est pas connue de l'utilisateur. Par exemple, dans une palette d'outils d'un éditeur, une icône peut représenter un outil du monde réel que l'utilisateur ne connaît pas. Le deuxième cas consiste à utiliser une modalité dont l'analogie ne correspond pas à la sémantique de l'expression véhiculée. Ce dernier cas consiste à utiliser une modalité analogique mais dont la correspondance entre l'analogie et la sémantique véhiculée est arbitraire.

L'arbitraire fait référence à l'absence de lien entre le contenu et le contenant. Une modalité non-arbitraire repose donc sur une fonction de correspondance qui n'a pas été définie par le concepteur mais qui est conventionnelle et connue de l'utilisateur. Un exemple de modalité non arbitraire consiste à utiliser le code de couleur (rouge, orange et vert) pour signifier l'arrêt et le fonctionnement d'un processus supervisé par le système. Dans une page Web, l'utilisation des couleurs pour le texte des liens visités ou non encore visités repose par contre sur une modalité arbitraire. Dans le cas de modalités arbitraires, l'utilisateur risque de ne pas être capable d'interpréter l'information véhiculée. Il est difficile d'affirmer qu'une modalité est arbitraire sans tenir compte des connaissances de l'utilisateur. Un utilisateur expert du Web

pourra en effet trouver qu'un autre système qui utilise les mêmes couleurs que son navigateur Web est non arbitraire.

Grâce à ces quatre propriétés, combinées avec les trois sens (graphique, son, toucher), 48 classes de modalités $((2 \times 2 \times 2) \times 3)$ sont définies. Vingt huit d'entre elles sont à exclure car elles correspondent à des impossibilités dues au fait par exemple qu'une modalité linguistique ou analogique ne peut pas être arbitraire. Bernsen met donc en évidence 20 modalités de sortie. Ces modalités sont notées " modalités pures " qui sont à opposer aux modalités composées.

3.3 *Synthèse*

Les deux espaces de caractérisation d'une modalité sont complémentaires : le premier considère les cinq sens humains afin de caractériser les dispositifs physiques. Cette caractéristique est fréquemment utilisée et nous l'avons illustré avec l'espace de conception de Frolich. Caractériser une modalité selon les sens humains permet d'établir une passerelle avec les travaux de psychologie cognitive comme la théorie ICS. Les propriétés de Bernsen permettent de caractériser le langage d'interaction d'une modalité. Ainsi les deux espaces présentés couvrent les deux facettes d'une modalité : (dispositif, langage).

Les deux nouveaux espaces, l'un de composition de modalités, l'autre de caractérisation, que nous exposons dans les deux sections suivantes, complètent et unifient les espaces étudiés dans cette section.

4 *Nouvel espace de composition de modalités de sortie*

Notre étude des référentiels existants a mis en évidence l'existence de niveaux d'abstraction dans la composition de modalités. Ces niveaux se retrouvent aussi dans les activités mentales de l'utilisateur, comme nous l'avons expliqué avec la théorie ICS. Dans les espaces TYCOON et CARE, les concepts de complémentarité et de redondance sont centraux et relèvent d'une composition du niveau sémantique. Dans l'espace MSM, les aspects articulatoires et sémantiques sont identifiés par un axe "niveau d'abstraction". Enfin la distinction entre modalités séquentielles et parallèles est présente dans les espaces étudiés à la section précédente.

Notre approche repose sur cinq aspects de la composition : trois des aspects correspondent aux trois niveaux d'une modalité définis au paragraphe 1.1 (page 36). Nous considérons aussi les aspects temporels et spatiaux de la composition.

Les cinq aspects d'une composition sont donc les aspects temporel, spatial, articulatoire, syntaxique et sémantique. L'aspect temporel caractérise les enchaînements possibles de modalités. L'aspect spatial reflète la volonté du concepteur de placer les modalités à des endroits spécifiques de l'environnement de perception de l'utilisateur. La composition articulatoire des modalités dénote une composition à un niveau d'abstraction plus bas, entre les flux d'information transitant par les dispositifs associés aux modalités. La composition syntaxique considère les relations entre les langages d'interaction utilisés. Enfin, la composition sémantique concerne la composition des informations véhiculées par les modalités.

Les travaux d'Allen [Allen 83] sur l'aspect temporel nous ont incités à considérer un axe orthogonale à celui défini par les aspects. Ainsi, la seconde étape de notre approche consiste à considérer, les cinq aspects de la composition (temporel, spatial, articulatoire, syntaxique et sémantique) selon les cinq schémas issus des relations de Allen [Allen 83] et illustrés à la Figure 12. Ces schémas sont utilisés dans un cadre plus général que celui des travaux de Allen car nous les utilisons comme un paradigme pour l'étude de différents aspects de la composition de modalités.

Nous ne relevons pas les autres schémas de Allen qui se déclinent à partir des cinq schémas de la Figure 12 car nous ne désirons pas prendre en considération l'orientation de la composition. Seul l'aspect temporel est fortement orienté (du passé vers le futur).

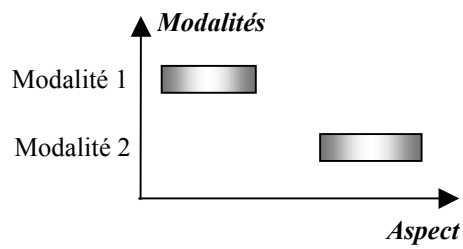


Figure 12a : Composition de modalités éloignées.

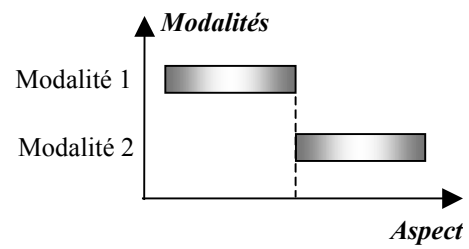


Figure 12b : Composition de modalités avec un point de contact.

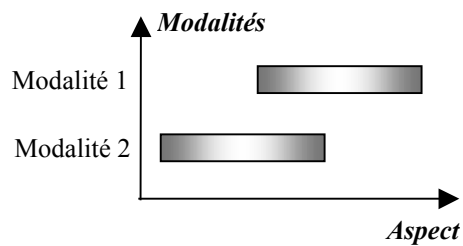


Figure 12c : Composition de modalités avec une intersection non vide.

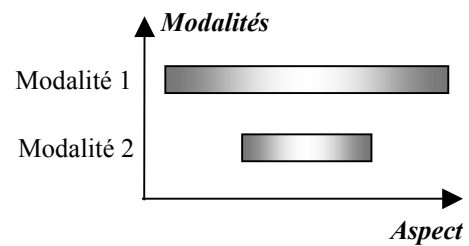


Figure 12d : Composition de modalités dont l'une est plus étendue et englobe l'autre.

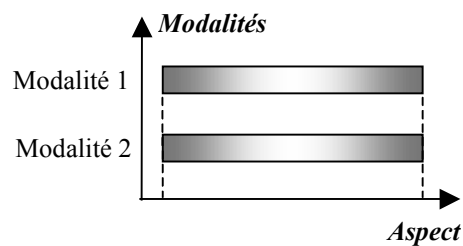


Figure 12e : Composition de modalités de même étendue.

L'application des cinq schémas de composition aux cinq aspects de composition définit 25 types de composition, présentés dans le Tableau 1. Nous nommons chaque composition, ainsi définie par un aspect et un schéma.

		Schémas de composition				
Aspects de composition	Composition					
	Temporelle	Anachronique	Séquentielle	Concomitante	Coïncidente	Parallèle / Simultanée
	Spatiale	Disjointe	Adjacente	Intersectée	Imbriquée	Recouvrance
	Articulatoire	Indépendance	Fissionnée	Fissionnée + Dupliquée	Partiellement Dupliquée	Dupliquée
	Syntaxique	Différente	Complétion	Divergence	Extension	Jumelage
	Sémantique	Concurrente	Complémentaire	Complémentaire + Redondante	Partiellement Redondante	Totalement Redondante

Tableau 1 : Application des schémas de composition aux cinq aspects proposés.

4.1 Aspect temporel d'une composition de modalité

L'étude des espaces existants ont souligné le fait que la séquentialité et le parallélisme dans l'usage des modalités ont été largement étudiés. Au contraire, les trois autres schémas de composition temporelle du Tableau 1 sont nouveaux pour la composition de modalités et constituent donc un affinement significatif.

La composition anachronique de modalités se distingue de la séquentialité par l'intervalle de temps entre les modalités. La durée minimale de l'intervalle pour passer d'une composition anachronique à une composition séquentielle est fixée par le concepteur en fonction de ses besoins. L'intervalle nécessaire pour séparer deux messages sonores sans rapport entre eux, et que le concepteur veut donc anachroniques, doit par exemple être de plusieurs secondes.

La composition par concomitance intervient quand une modalité en remplace une autre, avec un certain intervalle de temps pendant lequel les deux modalités sont présentes. Comme pour la composition anachronique, la durée de cet intervalle de transition dépend des besoins du concepteur. Une telle composition est importante pour aider et accompagner l'utilisateur dans la transition d'une modalité à une autre. Elle nécessite cependant que le ou les dispositifs puissent rendre en parallèle ces deux modalités pendant quelques instants.

Enfin, la composition coïncidente implique que l'une des deux modalités n'apparaisse que lorsque la seconde est déjà présente. Cette même modalité se termine avant que la seconde

soit achevée. Une telle relation est nécessaire lorsque le concepteur met en œuvre une modalité qui fait sens seule et une seconde modalité qui ne fait sens qu'au regard de la première. Le système PPP, présenté au paragraphe 2.1 (page 42), utilise cette composition : le message sonore n'est joué que lorsque l'animation pointe sur un objet de l'interface comme une image.

4.2 *Aspect spatial d'une composition de modalités*

L'aspect spatial de la composition est un trait plus particulier aux modalités de sortie. En effet si les aspects temporel et sémantique de la composition des modalités d'entrée sont importants, l'aspect spatial est souvent négligé car les dispositifs d'entrée sont souvent différents pour chaque modalité. En sortie, le problème est différent car les modalités ne partagent essentiellement que deux dispositifs : l'écran et les haut-parleurs. Si nous considérons que deux modalités peuvent recouvrir le même lieu (comme la surface d'un écran ou l'espace de propagation d'un son), c'est que les sens perceptifs humains (vision et ouïe) nous permettent de percevoir plusieurs éléments au même endroit. La transparence par exemple nous permet de percevoir deux informations affichées au même endroit d'un écran. De même, lorsque le son n'est pas spatialisé, tous les sons perçus au travers des haut-parleurs nous semblent provenir du même endroit.

Une composition disjointe signifie que l'utilisateur peut percevoir une distance entre les modalités. A contrario, une composition adjacente signifie que l'utilisateur perçoit les modalités en des lieux différents mais comme côte à côte. Les compositions intersectée, imbriquée et recouvrante dénotent trois types de transparence. La composition intersectée définit par exemple le fondement des lentilles magiques ("Magic lenses") [Bier 93]. Une lentille magique est translucide et recouvre une partie de l'espace d'information sur laquelle s'applique l'opération de la lentille. Comme le montre la Figure 13, lorsque deux lentilles s'intersectent, leurs effets s'accumulent.

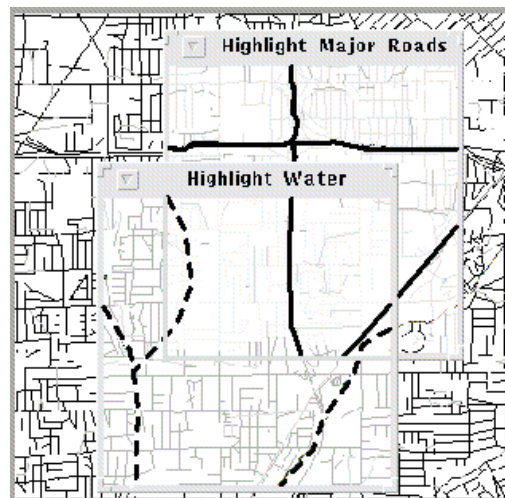


Figure 13 : Lentilles magiques [Bier 93] : cas de deux lentilles ("Highlight Major Roads" et "Highlight Water" mettant en valeur les routes principales et les voies fluviales sur un fond de carte d'une ville.

La composition imbriquée est illustrée par des exemples générés automatiquement par le processus décrit dans [Maybury 93] : un rectangle (modalité imbriquée) dénote à tout instant la zone du schéma (modalité englobante) qui est en cours d'explication orale.

Enfin notre système des "Pixels Miroir" [Vernier 99], illustré à la Figure 14, est un exemple de composition recouvrante où la totalité de l'écran est utilisée par deux modalités différentes : le document et le reflet de l'utilisateur.



Figure 14 : Système des pixels miroir [Vernier 99] où la totalité de l'écran est utilisée pour afficher une carte météorologique et une vidéo en temps réel de l'utilisateur.

Ces cinq types de composition sont importants pour le concepteur d'une interface graphique, généralement limitée par l'espace disponible à l'écran. Les compositions recouvrante et imbriquée utilisent moins d'espace, mais induisent généralement un effort de perception et d'interprétation de la part de l'utilisateur (problème de continuité visuelle).

4.3 Aspect articulatoire d'une composition de modalités

La composition articulatoire concerne la composition des flux d'information qui transitent par les dispositifs associés aux modalités. Directement liés à la composition articulatoire, nous trouvons la notion de fission définie dans l'espace MSM.

Les cinq cas de composition articulatoire affinent ce terme de fission.

Les compositions, nommées indépendante et dupliquée, aux deux extrêmes du Tableau 1 traduisent respectivement l'absence de lien entre les couches bas niveau des modalités ou au contraire une duplication de toute l'information entre les deux modalités. Par exemple la composition des deux modalités exploitant deux haut-parleurs d'un système monophonique est dupliquée.

La composition fissionnée repose sur une fission d'un flux d'information. Les systèmes tels que les murs d'images (un même film découpé sur plusieurs écrans) mettent en œuvre une fission à bas niveau d'abstraction d'un flux vidéo découpé en sous-parties contiguës. Chaque partie est jouée sur un dispositif différent donc une modalité distincte.

Les compositions dupliquées et fissionnées sont rarement mises en œuvre et nécessitent un contexte spécial. Nous considérons par exemple comme dupliquées et fissionnées deux vidéo issues d'une même source mais dont chacune est sous-titrée dans une langue différente.

La composition partiellement dupliquée est par contre plus fréquente. On constate en effet que certains dispositifs ont la capacité de présenter des informations identiques mais en quantité ou qualité différentes. L'écran d'une station de travail et d'un ordinateur de poche peuvent par exemple présenter la même image mais celle de l'ordinateur de poche sera tronquée par rapport à celle de la station de travail. Un document Postscript sorti sur imprimante peut également avoir une meilleure résolution que son double affiché à l'écran.

4.4 Aspect syntaxique d'une composition de modalités

Tandis que la composition articulatoire se réfère au dispositif de la modalité, la composition syntaxique a trait à son langage d'interaction. Nous distinguons cinq schémas de

composition syntaxique qui vont de la composition de modalités dont le langage d'interaction est le même (composition jumelle) jusqu'à la composition de modalités aux langages d'interaction différents, comme l'exemple de VITESSE décrit en introduction.

La composition de complétion dénote l'utilisation de modalités dont les langages ont des pouvoirs d'expression complémentaires. Le langage pseudo naturel complète par exemple celui des dates afin de produire des énoncés tels que "la réunion aura lieu le 10/03/2000". De même les "smileys" { :-) = sourire, ;-) = clin d'œil, etc. } constituent un langage qui complète le langage naturel dans les courriers électroniques.

La composition divergente indique l'utilisation de langages d'interaction ayant une partie du vocabulaire en commun. C'est le cas par exemple du langage écrit et du langage parlé.

Enfin, nous proposons la composition d'extension qui indique des langages aux pouvoirs d'expression imbriqués. Par exemple, le texte mis en forme avec des attributs tels que l'italique ou le soulignement est un langage plus étendu que le texte brut, sans formatage. De même le langage de requête des moteurs de recherche étend un langage pseudo naturel en y ajoutant des symboles liés à la logique booléenne.

4.5 Aspect sémantique d'une composition de modalités

Un aspect central à considérer dans une composition de modalités est la relation entre la sémantique des informations véhiculées par les deux modalités. Les formes de multimodalité qui ont été le plus étudiées sont sans nul doute la complémentarité et la redondance, définies dans les espaces CARE et TYCOON. La concurrence entre modalités, identifiée dans l'espace de conception TYCOON n'est généralement pas considérée comme un cas de multimodalité car une analyse séparée des deux modalités est préférable. La composition que nous appelons partiellement redondante et celle que nous appelons complémentaire et redondante nécessitent un complément d'explications.

La composition complémentaire et redondante véhicule des informations dont une partie seulement se répète. Une telle composition de modalités graphiques est souvent nécessaire dans les vues multiples. Chaque vue présente des informations différentes. Une partie des informations est cependant répétée dans chacune des vues afin de permettre à l'utilisateur d'établir un lien entre les vues. Par exemple le système "MagniFind"¹ [MagniFind], présenté à la Figure 15, permet à l'utilisateur d'interagir sur une vue listée et détaillée de ses fichiers (taille, date, etc.) (la partie droite de la Figure 15) et en même temps sur une vue en arbre hyperbolique

¹ Disponible sur Internet : <http://www.inxight.com/>.

(la partie gauche de la Figure 15) qui ne permet pas de visualiser les fichiers ni les détails sur les répertoires, mais qui permet de voir de manière synthétique la liste des sous-répertoires. Ainsi, ces deux modalités graphiques, à première vue complémentaires, présentent pourtant une partie redondante puisque la liste des répertoires de plus haut niveau apparaît dans les deux vues.

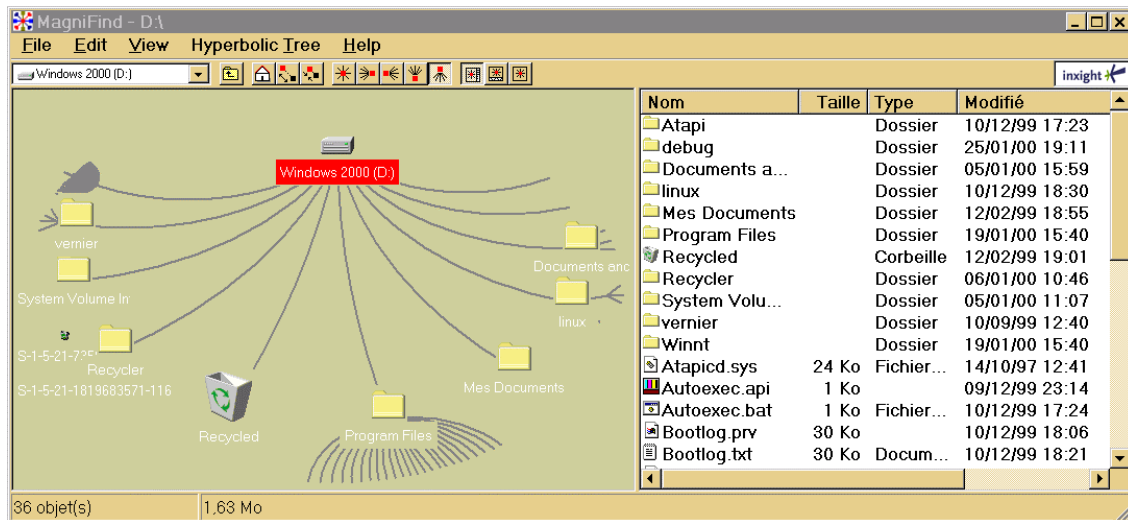


Figure 15 : Système MagniFind [MagniFind]: Exemple de composition complémentaire et redondante de modalités de sortie.

Nous appelons une composition des modalités M1 et M2 partiellement redondante lorsque M1 véhicule une partie du sens véhiculé par M2 sans en reprendre la totalité. C'est un aspect très courant dans le domaine de la visualisation d'une grande quantité d'information car les vues synthétiques de l'espace, proposées dans de nombreux systèmes, n'offrent en général qu'une redondance partielle avec la vue principale de l'espace.

4.6 Aspects et schémas : un espace unificateur

En synthèse, une composition de modalités de sortie fait intervenir l'une des cinq compositions pour les aspects temporels, spatiaux, syntaxiques et sémantiques. Ces quatre axes indépendants impliquent donc 5^4 relations différentes entre modalités. Nous notons la particularité de la troisième ligne du Tableau 1, composition articulatoire, qui n'est pas indépendante des deux dernières lignes (compositions syntaxique et sémantique). Certaines combinaisons ne sont pas possibles. Ainsi, si une composition repose sur un flux d'information dupliqué vers deux dispositifs de sortie, les deux modalités ne sont pas sémantiquement complémentaires. La composition fissionnée et dupliquée et la composition partiellement dupliquée sont également restrictives. A l'opposé, les deux compositions indépendantes et fissionnées autorisent toutes les compositions aux niveaux syntaxique et sémantique.

L'espace de conception que nous proposons complète et unifie les espaces existants. Notre espace est plus complet puisqu'il propose 25 types de compositions organisées selon cinq aspects fondamentaux que nous avons conjugués avec cinq schémas de composition. Ces compositions de modalités permettent de créer de nouvelles modalités dites composées. Cet espace cerne l'espace des possibilités quant à l'usage de plusieurs modalités. Il convient maintenant de caractériser les modalités qu'elles soient simples ou composées.

5 *Caractérisation des modalités de sortie*

Comme nous l'avons expliqué à la section 2, nous avons considéré la conception d'interfaces multimodales en sortie selon deux angles d'étude : la composition de plusieurs modalités et la caractérisation d'une modalité. La composition de modalités permet de construire de nouvelles modalités et donc d'étendre les possibilités de conception. La caractérisation d'une modalité simple ou composée est elle importante pour permettre le choix d'une modalité, que celui-ci soit effectué par le concepteur, le système ou l'utilisateur. Notre espace de caractérisation repose sur les deux facettes d'une modalité : dispositif et langage.

5.1 *Caractérisation des dispositifs*

Pour caractériser le dispositif physique, nous considérons les ressources sensorielles de la modalité de perception. Pour cela, nous caractérisons un dispositif physique par le sens humain mis en jeu. Afin de classer plus finement les dispositifs, nous considérons ensuite les composantes de chacun des sens. Par exemple, le toucher peut se décomposer en trois composantes, le tactile, la proprioception et la kinesthésie.

Cette caractéristique des modalités est importante car elle jette un pont avec les résultats issus de la psychologie cognitive : nous avons détaillé ce point au paragraphe 3.1.1.4 (page 51) avec la théorie ICS [Barnard 93] qui permet de déterminer les conditions que les différentes sources d'information doivent respecter afin d'être interprétées comme un tout cohérent. Ces résultats sont directement applicables pour la composition de modalités, en particulier pour les aspects articulatoire, spatial et temporel de la composition. Néanmoins l'analyse du canal sensoriel mis en jeu par la modalité peut aussi concerner les autres aspects de la composition. ICS explique par exemple que sur un même canal sensoriel, nous pouvons combiner différentes informations à condition qu'elles soient liées par une structure commune (similarité, symétrie, proximité). Ce résultat reposant sur une théorie de psychologie cognitive permet d'envisager la composition de deux modalités aboutissant à un seul flux d'information : l'aspect articulatoire de la composition n'est donc plus à considérer. Ce résultat se traduit par une composition de deux modalités exploitant le même dispositif que nous décrivons comme suit :

- ▶ Temporel : Parallèle.
- ▶ Spatiale : Recouvrante, puisqu'il n'y a qu'un seul flux d'information.
- ▶ Articulatoire : Non applicable puisqu'il n'y a qu'un seul flux d'information.
- ▶ Syntaxique : Non Différente car les deux modalités doivent reposer sur une structure commune. La structure commune est déterminée par le schéma appliqué à l'aspect syntaxique : Complétion, Divergence, Extension ou Jumelage.
- ▶ Sémantique : Complémentaire.

5.2 *Caractérisation des langages*

Nous considérons maintenant l'autre facette d'une modalité : le langage d'interaction. Dans notre espace, nous reprenons les 4 propriétés identifiées par Bernsen (paragraphe 3.2.2, page 57) :

- ▶ statique ou dynamique,
- ▶ linguistique ou non-linguistique,
- ▶ analogique ou non-analogique,
- ▶ arbitraire ou non-arbitraire.

Ces propriétés nous permettent de raisonner sur des classes de modalités et donc de donner une facture générique à l'étude. Par exemple, nous avons montré à la section 2.1 que de nombreux systèmes de génération dynamique d'énoncés multimodaux, comme WIP, PPP et COMET, intègrent la règle suivante : "Utiliser un texte pour présenter des propriétés abstraites comme, une relation sémantique ou encore temporelle entre deux concepts." Nous abstrayons cette règle en préconisant le choix d'une modalité dite linguistique. Le choix du dispositif, qui implique celui de l'un des deux sens, visuel ou auditif, est ensuite à considérer.

Ces propriétés définissent des classes de modalités sur lesquelles le concepteur ou le système effectuent des choix. Les propriétés n'impliquent pas un choix évident d'une classe de modalités particulière puisqu'elles ne véhiculent pas une échelle de valeur : une modalité analogique n'est pas meilleure qu'une modalité non-analogique ou linguistique.

La composition de modalités permet d'envisager des modalités composées qui vérifient et ne vérifient pas l'une des propriétés. Par exemple une modalité composée peut être arbitraire et non arbitraire.

Nous complétons les travaux de Bernsen [Bernsen 93] par deux nouvelles caractéristiques.

5.2.1 *Caractéristique passive/active/passive-active*

La première caractéristique, passive/active/passive-active, jette un pont entre les caractéristiques des modalités de sortie et de celles des modalités d'entrée. Elle permet de distinguer une modalité de sortie, caractérisée d'active, qui est utilisée dans l'interaction en entrée (comme la représentation d'un bouton à l'écran) d'une modalité purement informative, caractérisée de passive (comme une barre de progression). Nous pourrions analyser plus finement les modalités actives, en caractérisant ensuite la tâche que l'utilisateur peut entreprendre en manipulant l'expression de sortie. Nous exploitons alors les taxonomies des modalités d'entrée. Cette approche est à mettre en relation avec la taxonomie de Roth [Roth 90b] qui identifie un ensemble de tâches abstraites, comme la comparaison de deux valeurs, pour classer des modalités graphiques. Ces tâches concernent le besoin d'information de l'utilisateur et n'impliquent pas des actions de l'utilisateur. Notre approche est donc complémentaire à celle de Roth en considérant aussi les actions que l'utilisateur peut entreprendre en manipulant l'énoncé multimodal. Comme le caractère actif et passif peut être dynamique au cours de l'interaction, nous introduisons une troisième valeur, passif-actif. Nous illustrons ce cas à la Figure 16. La modalité composée (langage icônique, langage naturel) associée au menu est passive-active. A l'opposé des caractéristiques issues des travaux de Bernsen qui s'ajoutent lors de la composition de modalités, une modalité composée d'au moins une modalité active ou passive-active est caractérisée respectivement d'active ou de passive-active.

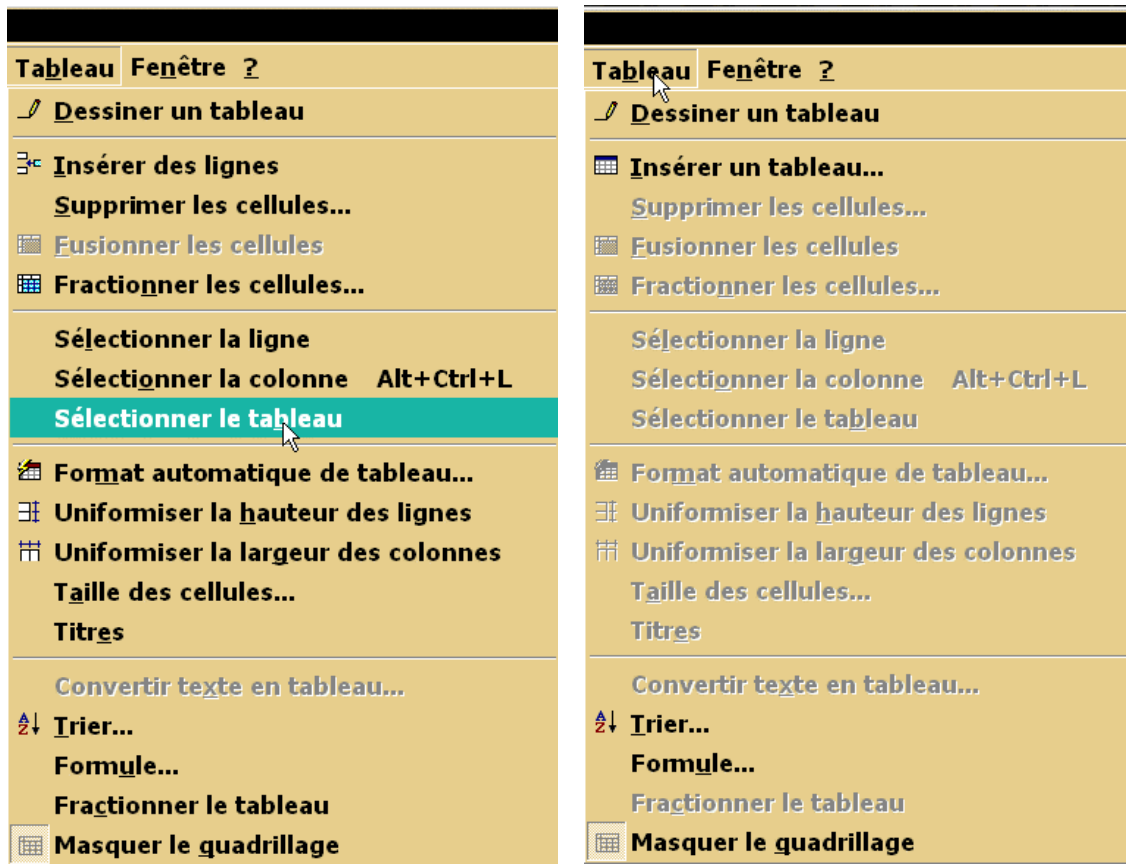


Figure 16 : Modalité passive-active composée (langage icônique, langage naturel).

5.2.2 Caractéristique Pure/Composée

Afin d'établir un lien explicite avec notre espace de composition de la section précédente, il convient de caractériser une modalité comme pure ou composée. Nous reprenons le terme "Pure" introduit par Bernsen et qui signifie atomique.

Le choix entre une modalité simple ou composée est en soi une décision de conception. En effet, dans le Chapitre II, nous avons expliqué que nos patrons de conception sont classés selon le caractère composé ou non de la solution préconisée. De plus des systèmes de génération automatique d'énoncés multimodaux comme le système COMET (paragraphe 2.1, page 42) encapsulent des règles qui préconisent l'usage d'une modalité composée. Par exemple dans COMET nous trouvons la règle suivante : "Utiliser du texte et du graphique pour expliquer une action".

Enfin la caractéristique Pure/Composée autorise différents niveaux de granularité dans la conception. A un niveau fin d'analyse, la modalité de la Figure 16 est composée de deux modalités (icône et texte) tandis qu'à plus haut niveau d'analyse une seule modalité composée

est utilisée. Cette caractéristique permet ainsi de raisonner sur la composition récursive de modalités : composition de modalités composées avec des modalités pures ou composées.

5.3 *Synthèse*

Nous avons proposé des traits de caractérisation d'une modalité afin de permettre un choix de conception qui ne soit pas dépendant d'une modalité particulière. Cet espace comprend les caractéristiques suivantes :

- ▶ statique ou dynamique ou statique-dynamique
- ▶ linguistique ou non-linguistique ou linguistique-nonlinguistique
- ▶ analogique ou non-analogique ou analogique-nonanalogique,
- ▶ arbitraire ou non-arbitraire ou arbitraire-nonarbitraire,
- ▶ passive ou active ou passive-active,
- ▶ pure ou composée.

Contrairement à Bernsen, nous ne compterons pas le nombre de modalités possible. Le nombre de possibilités est trop grand pour être significatif vis-à-vis du concepteur. Nous notons simplement que les combinaisons ne sont pas toutes possibles. En plus des combinaisons déjà éliminées par Bernsen, les modalités statique-dynamique, linguistique-nonlinguistique, analogique-nonanalogique arbitraire-nonarbitraire ne sont pas pures.

La dernière caractéristique "pure ou composée" établit un lien explicite avec notre espace de composition de la section 4. Cet espace de composition caractérise les relations possibles entre les modalités. Les deux espaces, de caractérisation et de composition, correspondent aux deux facettes de la conception d'interfaces multimodales. À la section 2, nous avons expliqué leur utilité dans la mise en place de patrons de conception. La généralité nécessaire des patrons de conception repose sur nos deux espaces qui cernent l'espace des possibilités de conception de façon générique.

6 *Patrons de conception*

Dans cette section, nous décrivons quatre patrons de conception d'interfaces multimodales. L'objet de cette section n'est pas de proposer un langage de patrons qui vise l'exhaustivité : ceci serait le travail d'une vie de recherche si ce n'est plus ! Notre objectif est de démontrer notre démarche de conception (section 2) et l'utilité de nos deux espaces conceptuels (sections 4 et 5).

Les quatre patrons suivants sont décrits selon le format présenté au Chapitre II :

- ▶ Patron du test du dispositif (paragraphe 6.1)
- ▶ Patron du canal sensoriel le moins occupé (paragraphe 6.2)
- ▶ Patron d'adaptabilité des modalités de sortie (paragraphe 6.3)
- ▶ Patron des modalités complémentaires et simultanées (paragraphe 6.4)

6.1 Patron N°1 : Test du dispositif

Le patron du test du dispositif est un patron lié au critère d'ergonomie d'adaptativité. Ce patron est général (patron de conception d'interface multimodale). Ce patron concerne une modalité.

Ce patron préconise de tester la disponibilité du dispositif avant la mise en œuvre d'une modalité l'exploitant.

6.1.1 Exemples illustratifs

1. L'application Winamp [Winamp] fonctionnant avec le système d'exploitation [Windows] teste le dispositif de rendu sonore (la carte son) afin de savoir si ce dernier est disponible pour jouer un fichier son sélectionné par l'utilisateur. Si la carte son n'est pas disponible (en train de jouer un autre fichier son), Winamp attendra la fin de l'occupation du dispositif pour jouer le fichier son.
2. Les systèmes d'impression des systèmes d'exploitation tels que [Windows] et [MacOS] testent la disponibilité de l'imprimante et affichent un message demandant à l'utilisateur d'allumer son imprimante si cette dernière est éteinte.
3. Avant d'afficher une fenêtre de grande taille, de nombreux programmes testent les résolutions possibles de l'écran. Les logiciels de jeux refusent de s'exécuter si l'écran ne peut pas afficher une résolution suffisante.

6.1.2 Description du problème et du contexte

Dans un système multimodal en sortie, il est possible que la forme physique de l'énoncé ne soit pas adaptée aux caractéristiques physiques du dispositif ou que le dispositif ne soit pas disponible (occupé ou défaillant). Les systèmes informatiques étant par nature très modulaires et très diversifiés, les dispositifs utilisés par l'utilisateur ne sont pas nécessairement identiques à ceux envisagés lors de la conception et utilisés lors de la programmation. De plus les caractéristiques physiques des dispositifs sont dynamiques et peuvent varier pendant l'interaction (taille du papier dans une imprimante par exemple).

6.1.3 Motivations

- ▶ Eviter les messages d'erreur du système (ou même l'arrêt du système) pour avoir tenté de mettre en œuvre un dispositif qui n'est pas disponible.
- ▶ Utiliser des modalités qui exploitent au mieux les capacités des dispositifs.
- ▶ Anticiper le mauvais paramétrage d'un dispositif et proposer à l'utilisateur une aide dans le paramétrage du dispositif.
- ▶ Ne pas retarder l'apparition d'une modalité lorsque l'aspect temporel est important (comme dans le cas d'une composition parallèle) en testant le dispositif.

6.1.4 Contre-exemple

L'utilitaire "magnétophone" de Windows NT (version 4) [Windows] permet de rejouer un son préalablement enregistré. Si un son est en train d'être joué, l'arrivée d'un second son provoque l'apparition du message d'erreur de la Figure 17. Cet utilitaire ne permet donc pas une composition de deux sons qui soit concomitante, coïncidente ou parallèle.

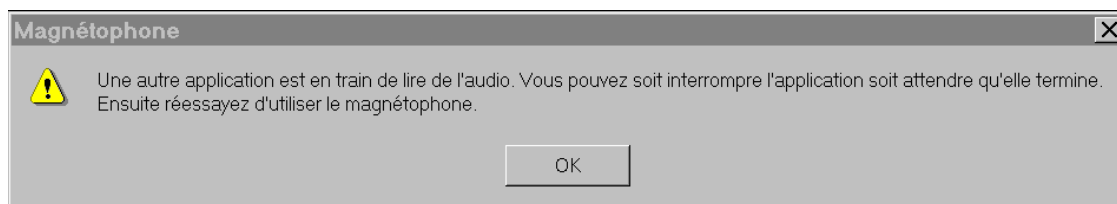


Figure 17 : Message d'erreur dû à un dispositif déjà occupé sous Windows NT.

Le logiciel MediaPlayer [MediaPlayer] comme le logiciel QuickTimePlayer [QuickTimePlayer] contreviennent au patron du test du dispositif car comme dans l'exemple précédent, deux sons ne peuvent être joués de façon concomitante, coïncidente ou parallèle ; de plus si un son est en train d'être joué, l'arrivée d'un second son provoque l'affichage d'une barre de défilement comme si le deuxième son est en train d'être joué alors que c'est toujours le premier son qui est joué. Nous constatons que le critère d'honnêteté (Annexe 1) est transgressé dans cet interface. A l'évidence ces applications ne testent pas la disponibilité du dispositif.

Un autre contre-exemple est celui de la modalité graphique associée à l'interacteur permettant de changer de mode d'affichage dans Windows NT [Windows]. Comme le montre la Figure 18, dans le mode d'affichage de 640 par 480 pixels, cette fenêtre est trop haute pour être affichée en entier. Le système d'exploitation n'a de toute évidence pas testé si le mode d'affichage courant permettait d'afficher l'expression entière selon la modalité graphique.

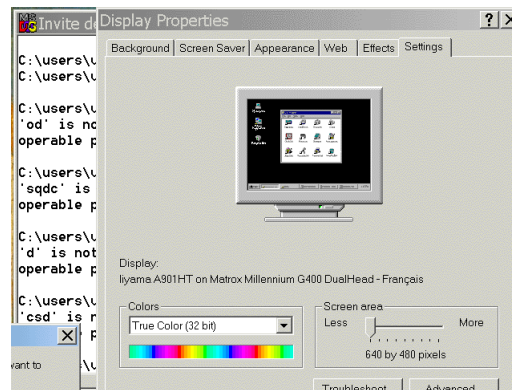


Figure 18 : Windows NT [Windows] en mode d'affichage minimal de 640 par 480 pixels. La modalité associée à l'interacteur, qui permet de changer de mode fait plus de 480 pixels de haut. Dans le mode 640 par 480, l'utilisateur n'a alors pas accès au bouton "OK" pour changer de mode et se trouve piégé(e) dans ce mode.

6.1.5 Solution proposée

En entrée, l'utilisateur a pris connaissance de la disponibilité des dispositifs et de leurs caractéristiques avant de les utiliser. En sortie, le système doit connaître l'état et les capacités des dispositifs. Pour cela le concepteur doit séparer la conception en deux étapes.

Concevoir un module de gestion du dispositif.

- Concevoir un module mettant en œuvre le langage d'interaction de la modalité.

Cette approche modulaire, expliquée par [Nigay 94], permet d'étudier chaque dispositif indépendamment des langages, et en particulier d'identifier l'ensemble des fonctions de tests pertinents pour chaque dispositif. Lors de la deuxième étape, le concepteur peut alors exploiter les fonctions de test identifiées lors de l'étape précédente pour son étude des langages d'interaction.

Si le test du dispositif n'est pas trop coûteux, il doit être refait avant chaque mise en œuvre de la modalité qui utilise le dispositif. Si le test d'un dispositif aboutit à un échec, le système tente de choisir un dispositif équivalent. Si aucun dispositif équivalent n'est possible, le système explore d'autres langages équivalents qui permettent de tester la disponibilité de d'autres dispositifs. Si toutes les tentatives échouent et qu'il n'est pas possible de produire une modalité équivalente, l'application doit garder le contrôle pour produire le message d'erreur adéquat. On évitera ainsi les messages provenant des couches de bas niveau souvent mal adaptés au contexte de l'interaction (message en Anglais par exemple). Lorsque la modalité considérée est redondante, il est préférable de ne pas afficher de message d'erreur, comme celui de la Figure 17. qui bloque l'interaction. Par exemple, le logiciel de gestion de courriers

électroniques Eudora [Eudora] produit un message graphique et un message sonore pour signifier l'arrivée d'un courrier ; la non disponibilité du dispositif graphique ou sonore n'entraîne pas un message d'erreur.

Le module de gestion du dispositif doit enfin gérer les accès concurrents (si plusieurs applications veulent connaître l'état d'un dispositif en même temps). Si ce module ne doit pas directement gérer les messages d'erreur, il peut par contre retourner une proposition de message d'erreur que l'application pourra ou non exploiter. Un message d'erreur de bas niveau est préférable au message d'erreur de type "erreur inconnue".

Si la mise en œuvre d'une modalité se fait dans une composition temporelle contraignante (contrainte de synchronisation par exemple), le test du dispositif peut n'être fait qu'une seule fois au démarrage du système.

6.2 Patron N°2 : Préférence du canal le moins occupé

Le patron de la préférence du canal le moins occupé est un patron lié au critère d'adaptativité. Ce patron est général (patron de conception d'interface multimodale). Ce patron concerne une modalité. Cependant il fait l'hypothèse que la modalité considérée est utilisée en parallèle de d'autres modalités.

Ce patron préconise l'usage du dispositif qui met en jeu le canal sensoriel humain le moins occupé.

6.2.1 Exemples illustratifs

1. Le système d'exploitation MacOS [MacOS] génère des messages oraux comme "le réseau est indisponible" et "le réseau est de nouveau disponible".
2. Le système de messagerie électronique Eudora [Eudora] informe l'utilisateur de l'arrivée de nouveaux messages par un son plutôt que par l'apparition d'une fenêtre affichant le traditionnel message "vous avez reçu du courrier".
3. Certains systèmes embarqués dans les voitures utilisent des messages oraux pour remplacer les voyants d'alerte (portière mal fermée, ceinture non attachée, etc.).

6.2.2 Description du problème et du contexte

Les systèmes actuels sollicitent principalement le canal sensoriel visuel. Les sons et messages oraux sont rarement utilisés. Nous considérons ici la composition concurrente et

parallèle (ou concomitante ou coïncidente) de modalités exploitant l'écran et donc le canal visuel. Le contexte de ce patron est plus généralement la présentation d'informations différentes et indépendantes selon des modalités qui partagent les ressources d'un même dispositif.

Une surcharge d'un canal sensoriel augmente la charge cognitive de l'utilisateur. Elle/il doit être capable de percevoir et discerner les différentes informations (fission du flux d'information sensorielle). Le risque de non perception d'informations est grand.

Bien que ce patron s'intitule "préférence du canal le moins utilisé", ce patron traite essentiellement de l'utilisation du canal auditif pour décharger le canal visuel. Nous n'excluons pas toutefois le cas inverse, d'une interface essentiellement sonore qui utiliserait le canal visuel pour décharger le canal auditif.

6.2.3 *Motivations*

- ▶ Eviter la saturation du canal visuel.
- ▶ Exploiter la capacité de traitements parallèles d'informations sensorielles par l'utilisateur.
- ▶ Adapter une interface avec une grande surface d'affichage au cas d'une surface plus petite.

6.2.4 *Contre-exemple*

Les systèmes actuels font essentiellement intervenir le canal sensoriel visuel. De nombreux contre-exemples sont donc envisageables. Par exemple, les agents d'aide des produits Microsoft sont graphiques et prodiguent leurs conseils dans des bulles d'aide. Des messages oraux pourraient être joués afin que l'aide n'empiète pas sur la zone d'édition.

6.2.5 *Solution proposée*

L'utilisation du canal sonore doit être envisagée, en plus de modalités graphiques (canal visuel). Le patron considère la composition de deux modalités, l'une visuelle et l'autre sonore, décrite comme suit : (Temporelle = concomitante, coïncidente ou parallèle) et (Sémantique = concurrente). Il s'agit donc de rendre perceptible des informations différentes et indépendantes les unes des autres. Le cas de la composition complémentaire (donc non concurrente) est traité par le patron N° 4.

L'usage d'un autre dispositif évite d'avoir à partager les ressources d'un seul dispositif. La bande passante du canal de communication entre le système et l'utilisateur est donc plus large. L'implication de différents canaux sensoriels aide à différencier les informations.

La mise en œuvre de ce patron nécessite de pouvoir évaluer la saturation d'un dispositif. Pour l'écran, le nombre de fenêtres peut être compté, la surface de chaque fenêtre additionnée, la fréquence d'utilisation considérée [Neal 88]. Pour le canal sonore la charge peut être évaluée par le nombre de sons joués à un instant donné. Les caractéristiques des sons (qui sont liés aux données à rendre perceptibles et au langage d'interaction des modalités exploitant les haut-parleurs) influencent aussi la charge. Par exemple, lorsqu'un son très long, comme une chanson, est interrompu par un son très court, comme un "beep" d'alerte, les deux expressions sont facilement perçues par l'utilisateur. Les deux modalités sont ici composées selon un schéma coïncident.

Le canal auditif est inévitable et donc intrusif. Il est important que l'utilisateur puisse choisir les sons joués. L'utilisateur doit pouvoir paramétrer la modalité sonore comme dans [Eudora] et [Windows]. L'adaptabilité d'une modalité fait l'objet du patron N°3.

Les expressions selon une modalité sonore sont éphémères. Dans le cas d'une information importante comme un message d'erreur, l'expression doit être persistante afin d'en assurer sa perception (critère d'insistance). Pour cela, le son ou la phrase peut être rejoué jusqu'à la prise en compte de l'erreur par l'utilisateur.

Le concepteur doit veiller à concevoir des modalités graphiques et sonores équivalentes. En particulier une modalité graphique implique un contexte d'interprétation des expressions défini par la localisation à l'écran. Ce n'est pas le cas de modalité sonore. Ainsi le contenu de l'expression selon une modalité sonore doit contenir le contexte. Par exemple le message sonore : "Eudora a besoin de votre attention".

6.3 Patron N°3 : Adaptabilité des modalités de sortie

Le patron d'adaptabilité des modalités de sortie est un patron lié au critère d'ergonomie d'adaptabilité. Ce patron est général (patron de conception d'interface multimodale). Ce patron concerne une modalité.

Ce patron décrit l'adaptation d'une modalité. Pour cela il préconise un sous-dialogue de configuration associé à la modalité.

6.3.1 Exemples illustratifs

1. Les options d'affichage concernant la modalité "vue hyperbolique" du logiciel MagniFind [MagniFind] sont regroupées dans un sous-menu "Hyperbolic Tree". Outre la configuration des couleurs, le système autorise de modifier deux paramètres importants

de l'algorithme des "vues hyperboliques" : la longueur des liens et leur répartition dans l'espace autour du centre d'intérêt. Une option du menu permet de réinitialiser ces paramètres avec les valeurs par défaut ("reset default setting").

2. Les options d'affichage du logiciel Word reproduites à la Figure 19.

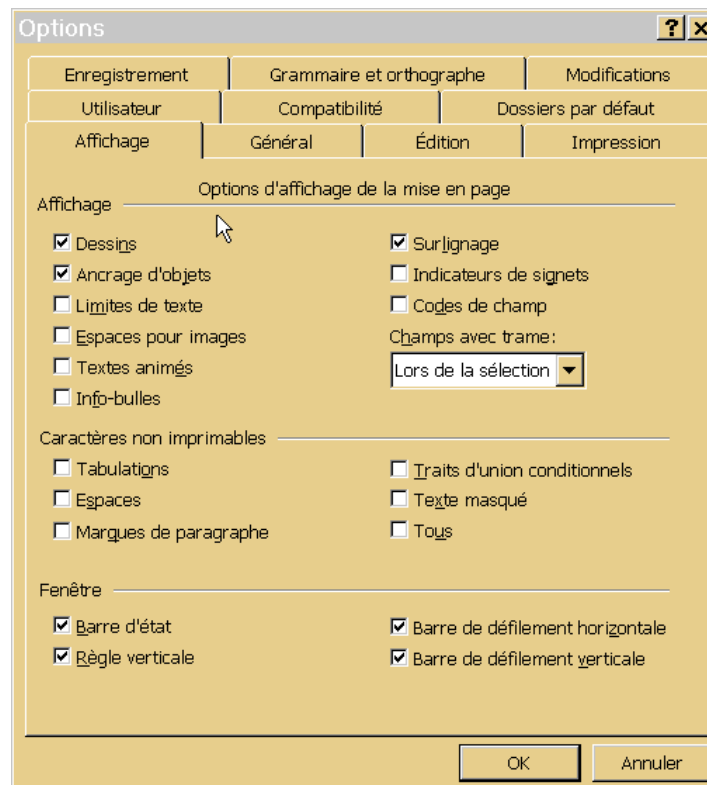


Figure 19 : Options d'affichage du système Word [Word].

3. Les options d'affichage de Codewarrior [Codewarrior] sont présentées à la Figure 20. Le bouton "Factory Settings" permet de réinitialiser les paramètres avec les valeurs par défaut.

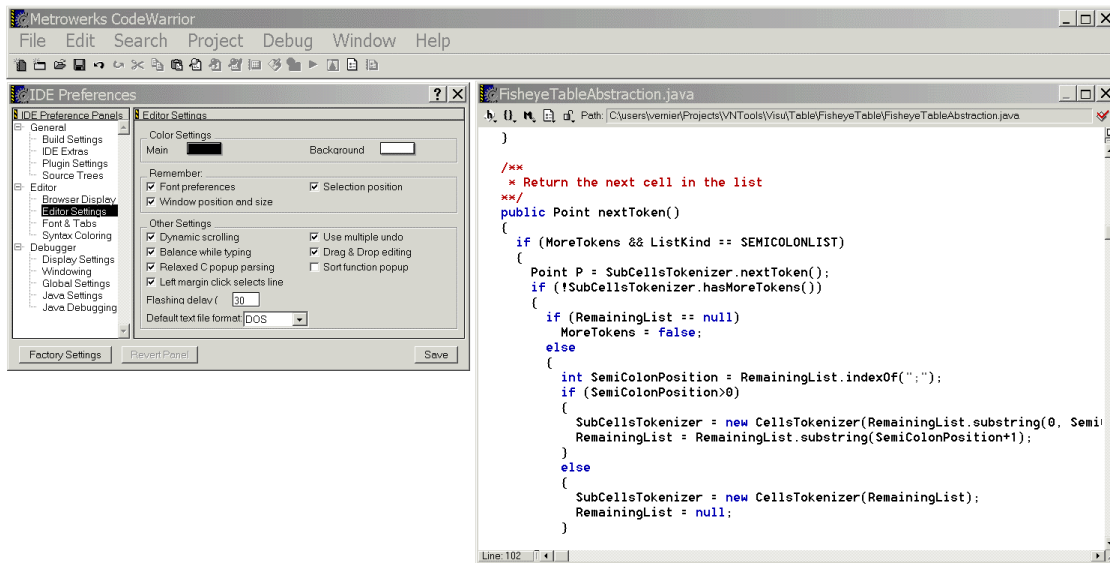


Figure 20 : L'interface de Codewarrior : à droite un programme visualisé selon une modalité graphique paramétrable grâce au panneau de configuration présenté à gauche.

6.3.2 Description du problème et du contexte

Bien que le concepteur puisse fixer une valeur jugée optimale pour chaque paramètre de la modalité de sortie conçue, ces choix ne sont pas nécessairement adéquats pour tous les utilisateurs et tous les contextes d'interaction. Les tests expérimentaux en condition d'utilisation permettent de souligner ces problèmes d'inadéquation. De plus certaines modalités peuvent avoir de nombreux paramètres associés, qui influencent fortement la forme de l'expression obtenue. Sans adaptation possible, l'utilisateur ne peut exploiter toutes les possibilités offertes par la modalité.

L'adaptation d'une modalité ne concerne pas le choix des modalités, cas d'adaptabilité expliqué à la section 2, mais la modalité elle-même. Il s'agit ici de modifier les paramètres du langage d'interaction lorsque cela est possible. La modification des paramètres d'une modalité peut impliquer des formes très différentes des expressions. La décision de considérer qu'il s'agit de deux modalités distinctes ou d'une seule modalité paramétrable revient au concepteur.

Dans ce patron, nous considérons l'adaptation des langages d'interaction. Les exemples et contre-exemples ont été choisis en conséquence. Néanmoins le patron peut aussi s'appliquer au cas des paramètres d'un dispositif, comme la luminosité de l'écran ou le volume sonore des haut-parleurs. A la section 2, nous avons présenté un système intégrant trois haut-parleurs dont les volumes sonores étaient modifiés par le mouvement de la tête de l'utilisateur. Dans le cas de l'adaptation des dispositifs, décider qu'il s'agit d'un dispositif paramétrable ou de deux dispositifs différents revient là encore au concepteur.

6.3.3 *Motivations*

- ▶ Observer les valeurs des paramètres d'une modalité.
- ▶ Permettre aux utilisateurs experts de modifier la forme logique des expressions selon une modalité.
- ▶ Permettre aux utilisateurs novices de trouver rapidement comment changer les paramètres principaux d'une modalité de sortie.
- ▶ Visualiser le résultat d'un changement.
- ▶ Distinguer les paramètres importants que les utilisateurs novices peuvent vouloir modifier et les paramètres rarement utilisés.

6.3.4 *Contre-exemple*

En procédant par comparaison entre systèmes nous pouvons mettre en évidence un paramètre qui est modifiable dans un système et absent dans l'autre, pour une modalité considérée.

Les systèmes graphiques de gestion des fichiers de Windows NT [Windows] et de MacOS [MacOS] illustrés à la Figure 21 diffèrent par exemple sur la modifiabilité de l'apparence des icônes représentant les répertoires. En effet, l'utilisateur expert de MacOS peut changer les icônes de ses répertoires afin d'en améliorer l'identification et donc la navigation. Sous Windows NT, le système modifie l'apparence de répertoires particuliers comme la corbeille ou le dossier système, mais ne permet pas à l'utilisateur de modifier lui-même l'apparence de ses répertoires.

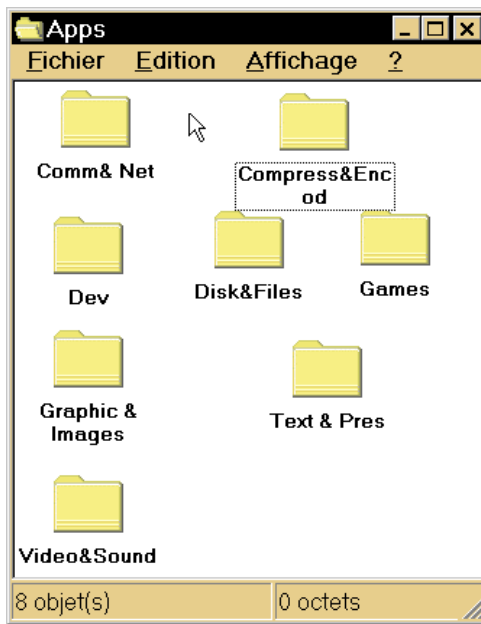


Figure 21a : Visualisation de répertoires dans Windows NT [Windows].

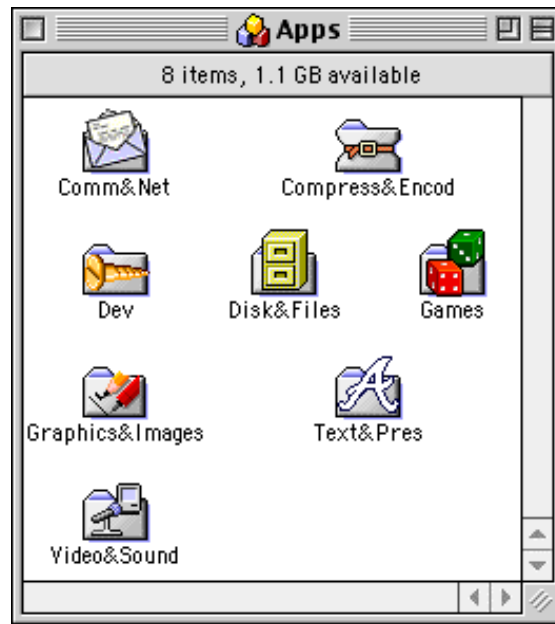


Figure 21b : Visualisation de répertoires dans MacOS [MacOS].

6.3.5 Solution proposée

La solution consiste à concevoir un sous-dialogue dédié à la configuration de chaque modalité de sortie. Si la modalité associée à des formulaires est la plupart du temps adaptée à cette tâche de configuration, il faut toutefois tenir compte d'environnements spécifiques comme les serveurs vocaux accessibles par téléphone. Il doit être possible d'éditer les paramètres d'une modalité sans que cette dernière soit utilisée dans l'interface.

Si la solution instanciée regroupe les paramètres de différentes modalités, ces derniers doivent être organisés par modalité. La Figure 19 montre un exemple où un onglet correspond à un ensemble de paramètres pour la modalité graphique utilisée pour afficher le texte édité tandis qu'un autre regroupe les paramètres de l'impression.

Si, pour une modalité, le nombre de paramètres est important, il convient de les regrouper en plusieurs catégories, selon leurs fréquences d'utilisation ou le niveau d'expertise requis. La Figure 19 présente en trois catégories les paramètres de la modalité graphique assignée à l'affichage du texte édité.

Lorsque les paramètres d'une modalité sont dépendants les uns des autres, il convient de présenter à l'utilisateur un exemple d'expression selon la modalité paramétrée (comme une pré-visualisation).

Pour chaque modalité, l'utilisateur doit pouvoir rétablir les valeurs par défaut des paramètres. Les valeurs des paramètres fixées par l'utilisateur doivent être persistantes.

6.4 Patron N°4 : Modalités complémentaires et simultanées

Le patron des modalités complémentaires et simultanées est un patron lié au critère d'observabilité. Ce patron est général (patron de conception d'interface multimodale). Ce patron concerne plusieurs modalités, c'est donc un patron de composition.

Ce patron préconise l'utilisation simultanée de modalités complémentaires.

6.4.1 Exemples illustratifs

1. Un agenda électronique tel StarSchedule [StarSchedule], illustré à la Figure 22, affiche l'agenda de l'utilisateur selon une modalité graphique et un calendrier selon une autre modalité graphique. La composition des deux modalités est caractérisée par : (Temporelle = parallèle), (Spatiale = disjointe), (Articulatoire = non applicable, un seul flux) (Syntaxique = différente), (Sémantique = complémentaire).

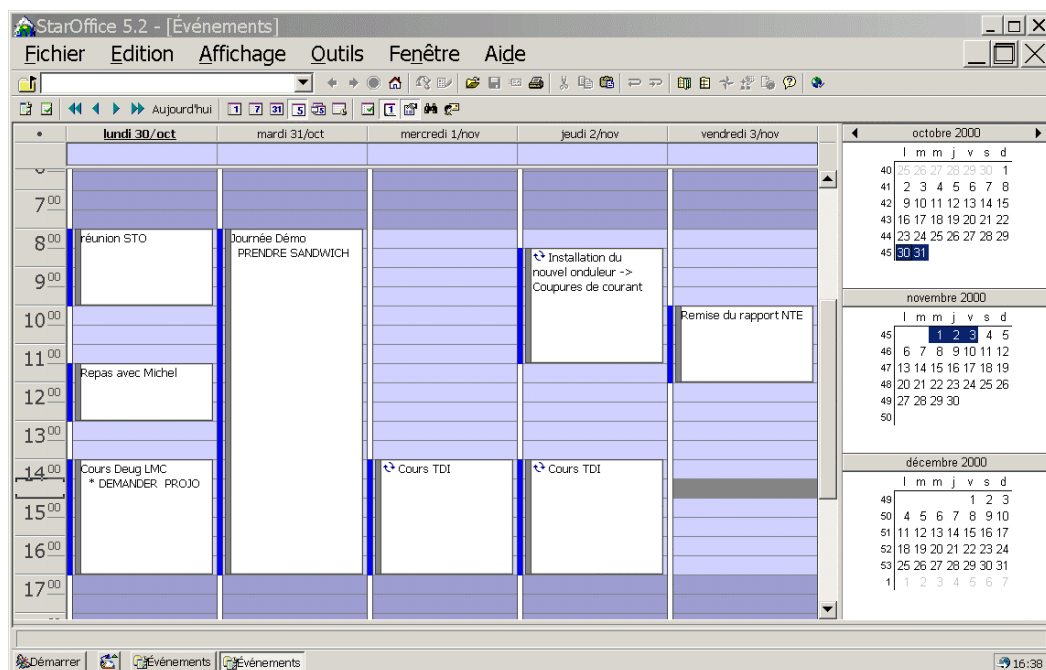


Figure 22 : Système StarSchedule [StarSchedule].

2. L'environnement de développement Codewarrior [Codewarrior] propose une vue graphique des interacteurs manipulés ainsi qu'un inspecteur sous la forme d'un formulaire contenant des informations complémentaires sur l'interacteur. Deux modalités sont

utilisées, l'une est (non-linguistique, non-analogique, non-arbitraire), l'autre est linguistique. Leur composition est caractérisée comme dans l'exemple 1. Lorsque l'utilisateur de Codewarrior édite un interacteur, l'inspecteur est automatiquement affiché.

3. La vue à grande échelle et celle à petite échelle d'un système tel que Illustrator [Illustrator] correspondent à deux modalités complémentaires et simultanées pour la navigation. Là encore la composition des deux modalités graphiques est caractérisée comme dans les deux exemples précédents.

6.4.2 *Description du problème et du contexte*

L'utilisation de modalités complémentaires est un aspect important des applications multimodales. Dans le cas de modalités complémentaires, l'utilisateur doit fusionner les informations véhiculées par les modalités. Lorsque les deux modalités ne sont pas disponibles au même instant, toute l'information nécessaire à la réalisation de la tâche n'est pas disponible (non observabilité). L'utilisateur doit donc mémoriser des informations et/ou passer d'une modalité à l'autre.

6.4.3 *Motivations*

- ▶ Minimiser l'effort de mémorisation.
- ▶ Pouvoir comprendre les deux modalités complémentaires dans n'importe quel ordre.
- ▶ Eviter les tâches articulatoires pour passer d'une modalité à l'autre.

6.4.4 *Contre-exemple*

Sur PalmPilot [PalmPilot], l'application Agenda équivalente à StarSchedule de la Figure 22, présente l'agenda et le calendrier séquentiellement. Bien sûr la justification d'une telle composition réside dans la petitesse de l'écran du PalmPilot.

6.4.5 *Solution proposée*

La solution consiste à composer des modalités complémentaires selon un schéma concomitant, coïncident ou parallèle. Le schéma parallèle est le plus souhaitable. Cependant l'ordre d'apparition des deux expressions selon des modalités distinctes peut traduire l'importance des informations véhiculées. Une composition concomitante ou coïncidente est alors possible.

7 Conclusion

Dans ce chapitre, nous avons présenté une revue de la terminologie et des espaces de conception existants pour la multimodalité en sortie. Les espaces sont limités par leur portée focalisant sur un aspect de la modalité ou de la multimodalité. Cette absence de cadre fédérateur empêche d'organiser et de capitaliser le savoir expérimental. La variété soulignée des systèmes existants témoigne pourtant de la multiplicité des possibilités.

Dans ce contexte, ce chapitre est consacré à la conception d'interfaces multimodales. Pour appréhender la multimodalité en sortie, nous nous basons sur le concept de modalité de sortie définie comme le couple (dispositif physique, langage d'interaction). Ces deux facettes d'une modalité nous permettent de mettre en exergue trois aspects de la multimodalité en sortie. Nous en retenons l'aspect le plus fréquent, l'interaction multilingue qu'elle repose sur un ou plusieurs dispositifs physiques. Dans notre typologie de la multimodalité, nous discernons les interfaces multimodales sans adaptation de celles qui sont adaptatives et adaptables.

Selon notre cadre de conception, nous cernons la conception d'une interface multimodale en sortie au choix motivé par des critères d'ergonomie d'une modalité simple ou composée pour un contexte d'interaction donné. Les critères de choix identifiés de nature très différente ainsi que les caractéristiques des données à rendre perceptibles cernent le contexte d'interaction. Pour la complétude de ce cadre de conception, nous avons présenté deux nouveaux espaces de conception, l'un pour caractériser une modalité, l'autre pour décrire la composition de modalités.

Notre cadre de conception contribue donc à clarifier l'espace problème en le structurant d'éléments qu'il convient de considérer lors de la conception d'une interface multimodale en sortie. Il identifie et organise aussi les éléments nécessaires à la définition de patrons de conception d'interfaces multimodales. Nous avons illustré notre approche en définissant quatre patrons de conception d'interfaces multimodales. Au Tableau 2, nous illustrons la place de ces quatre patrons dans l'espace de classification que nous avons présenté au chapitre II (Figure 1, page 27).

	Choix des modalités	Critère d'ergonomie	Niveau d'abstraction
<i>Patron N°1 : Test du dispositif</i>	1 modalité	Adaptativité	Interfaces multimodale
<i>Patron N°2 : Préférence du canal le moins occupé</i>	1 modalité + autres modalités	Adaptativité	Interfaces multimodale
<i>Patron N°3 : Adaptabilité des modalités de sortie</i>	1 modalité	Adaptabilité	Interfaces multimodale
<i>Patron N°4 : Modalités complémentaires et simultanées</i>	2 modalités ou plus	Observabilité	Interfaces multimodale

Tableau 2 : Caractéristiques de nos patrons de conception d'interfaces multimodales.

La généralité des résultats de ce chapitre est à souligner. Par exemple, les caractéristiques d'une modalité sont indépendantes de leur contexte d'utilisation (noté contexte d'interaction). Il en est de même pour les types de composition de modalités. La généralité des résultats nous permet ainsi au chapitre suivant, d'infléchir et d'affiner ce cadre de conception au cas de la multimodalité graphique informationnelle pour de grandes quantités de données : les techniques de visualisation. L'énorme variété des techniques de visualisation existantes, qui témoigne d'un grand dynamisme des travaux dans ce domaine, motive le choix de ce domaine d'application de la multimodalité.

CHAPITRE IV

VISUALISATION D'UNE GRANDE QUANTITE D'INFORMATION

***"A picture is worth
more than a
thousand words "***

Cette illustration est une vue depuis la 9^{ème} avenue de New York vers l'est des Etats-Unis considérée comme l'une des premières vue déformée (en œil de poisson). Manhattan apparaît plus large que le reste du continent américain.

Dessinée par Saul Steinberg (1976)



Ce chapitre est consacré à la visualisation d'une grande quantité d'information, un terrain d'étude où appliquer les principes conceptuels énoncés dans le chapitre précédent. La visualisation de grands espaces d'information est un domaine de recherche établi et dynamique auquel nous apportons notre point de vue issue de la multimodalité en sortie ainsi que l'outil de conception que constituent les patrons de conception. Aussi dans ce chapitre, nous reprenons la terminologie et le cadre de conception défini au chapitre précédent. En effet, nous ne désirons pas étudier la visualisation comme un nouveau domaine mais comme un approfondissement du domaine de la multimodalité, présenté au chapitre précédent.

Nous faisons l'hypothèse qu'une grande quantité d'information est constituée d'un grand nombre d'éléments où chaque élément possède des attributs variés : Ben Shneiderman [Shneiderman 96], *"I assume that users are viewing collections of items, where items have multiple attributes"*. Comme le souligne Lisa Tweedie [Tweedie 97] qui se base sur les travaux de Jacques Bertin [Bertin 77], les données à visualiser peuvent se décomposer en deux catégories : les valeurs et les structures. Nous considérons qu'une grande quantité d'information repose sur une structure générale où chaque élément est une valeur ou une petite structure contenant des valeurs. Cette dichotomie du problème de visualisation motive le premier et le dernier commandement du mantra énoncé par Ben Shneiderman : *"Overview first, zoom and filter, then details-on-demand"*. *"overview"* signifie avoir une vue d'ensemble de la structure générale de l'espace d'information, tandis que *"details-on-demands"* signifie détailler un élément à la demande de l'utilisateur. Les deux verbes, *"zoom and filter"* correspondent à zoomer et filtrer l'espace d'information. Ces actions de navigation permettent à l'utilisateur de définir la partie de l'espace d'information qui a besoin d'être détaillée pour l'accomplissement de sa tâche. Cette partie de l'espace d'information s'appelle le **focus**. A contrario, nous appelons la vue d'ensemble de la structure, le **contexte**.

Comme pour le cadre d'étude de la multimodalité, nous adoptons le point de vue de la conception. C'est dans ce sens que nous proposons, à la section 1, un état de l'art des espaces de conception des interfaces de visualisation. Nous en dégageons un cadre d'étude fédérateur pour ancrer notre étude de la composition et la caractérisation des modalités de visualisation aux sections 2 et 3. Nous présentons enfin, à la section 4, cinq patrons de conception qui viennent compléter la liste proposée au chapitre précédent.

1 Espaces de conception des interfaces de visualisation

Le processus qui transforme les données internes en réalisations visuelles perceptibles par l'utilisateur, noté **processus de visualisation**, est au cœur de la conception d'une technique

de visualisation. Dans le domaine de visualisation, nous constatons un foisonnement de taxonomies et cadres de conception qui sont soit limités par leur portée (une étape particulière du processus de visualisation) ou soit généraux (tout le processus de visualisation) mais bornés par le point de vue restrictif (visualisation déformante).

Aussi, comme point de départ à notre revue, nous présentons les travaux de Ed Chi [Chi 98] sur la décomposition en opérateurs du processus de visualisation au paragraphe 1.1. Les étapes et les opérateurs du processus de visualisation étant définis, nous pouvons ensuite présenter les autres cadres de conception de Ying Leung, Ben Shneiderman, Christophe Bruley et Stuart Card, respectivement aux paragraphes 1.2, 1.3, 1.4 et 1.5, en les ancrant dans le processus de visualisation. Nous synthétisons notre étude au paragraphe 1.6 par la définition d'un cadre d'étude fédérateur.

1.1 *Cadre d'étude de Ed. Chi*

Les travaux de Ed Chi [Chi 98] présentent deux contributions majeures que sont l'établissement d'un cadre d'étude¹ du processus de visualisation et une taxonomie en sept catégories des opérateurs intervenant dans ce processus. Quatre états et trois étapes décrivent le processus de visualisation, reproduit à la Figure 23. Ces sept éléments structurent le processus de transformation des données en éléments visibles par l'utilisateur. Au sein de cette structure, sont localisés sept opérateurs. Trois d'entre eux sont des opérateurs de transformation correspondant aux étapes de transformation (en gris clair dans la Figure 23). Les quatre autres sont des opérateurs internes aux états et sont représentés, à la Figure 23, par des flèches recourbées sur elles-mêmes.

Basé sur cette décomposition, Ed Chi propose de caractériser les systèmes de visualisation par les opérateurs qui sont mis en œuvre. Les acronymes anglais de ces opérateurs sont notés DSO, DTO, AASO, VTO, VASO, VMTO et VSO. Cette taxonomie permet de différencier les opérateurs sur les vues (rotation, zoom, etc.) et les opérateurs sur les données (tri, ajout de données, etc.), deux classes d'opérateurs largement utilisés en visualisation de grandes quantités d'information. Cette taxonomie met en évidence d'autres opérateurs intermédiaires qui modulent l'antagonisme Vue/Valeur identifié par Ed Chi et transcrit dans deux espaces :

- L'espace de la donnée (*Data Space*) contient les deux premiers états du processus (Donnée et Abstraction Analytique) et le premier opérateur (Transformation des données). L'espace

¹ "Framework" dans le texte original en anglais

de la donnée permet au système de contrôler le traitement (Extraction de meta-données) et le filtrage de l'information (Réduction de l'espace).

- L'espace de la vue (*View Space*) contient les deux derniers états du processus (Abstraction de visualisation et vue) et le dernier opérateur (Transformation de réalisation visuelle).

L'espace de la vue permet à l'utilisateur de contrôler

Il convient de comparer ce cadre d'étude avec le pipeline de visualisation proposé dans [Bender 00]. Ce pipeline identifie quatre niveaux d'abstraction pour les données (données en entrée, données multi-dimensionnelles, données de visualisation et données image) et trois étapes de transformation. Bien que ces travaux ne soient pas explicitement reliés aux travaux de Chi [Chi 98], la structure est proche et en accord avec celle établie par Chi.

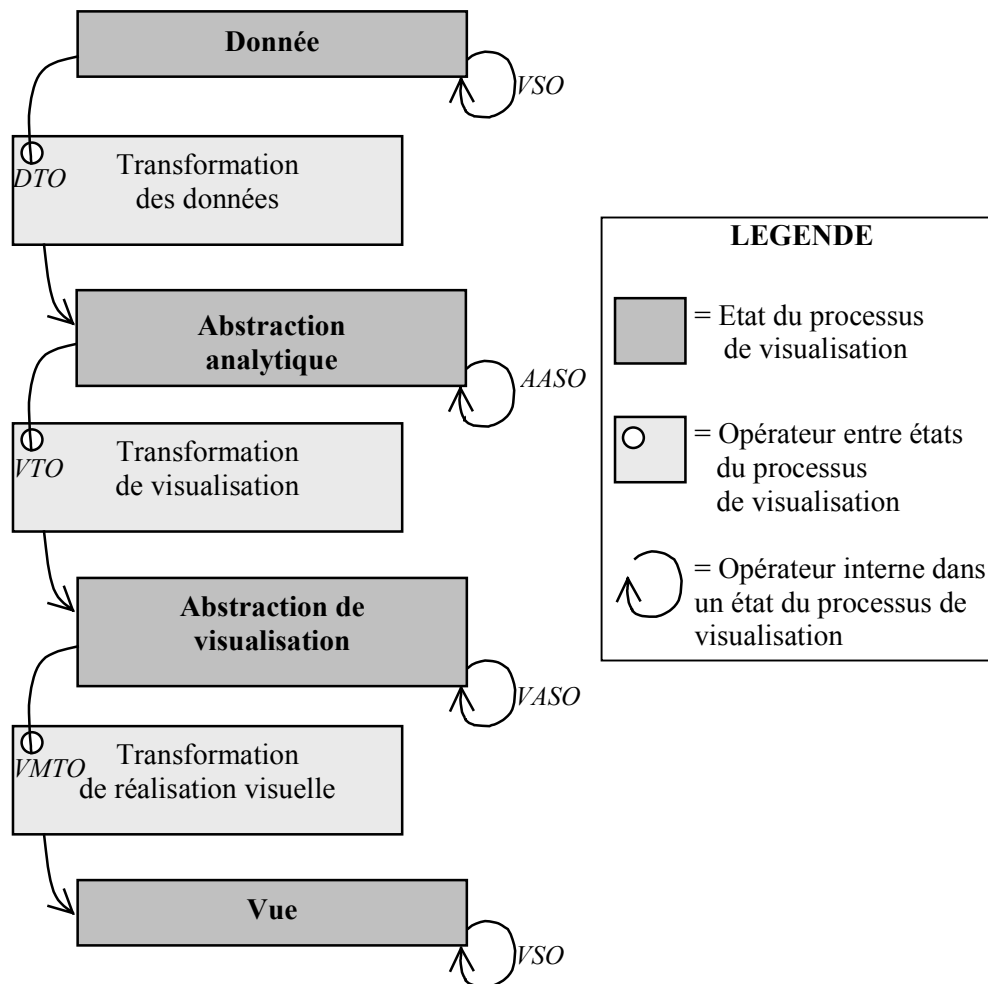


Figure 23 : Processus de visualisation de Ed Chi [Chi 98].

1.2 Taxonomie de Leung

Ying Leung [Leung 94] propose, une taxonomie générale couvrant toutes les étapes du processus de visualisation mais fortement influencée par son sujet d'étude, les visualisations déformantes.

Cette taxonomie s'appuie sur deux axes (Graphique/NonGraphique et Déformé/NonDéformé) qui permettent de distinguer quatre types de techniques de visualisation de grandes quantités d'information. Nous notons dans la présentation de cette taxonomie, reproduite à la Figure 24, la même intention que Ed Chi de caractériser le processus de transformation des données en vues.

Nous en retenons les deux points suivants : la distinction entre les données **intrinsèquement graphiques** et les données non-graphiques ainsi que l'utilisation d'une **déformation**, graphique ou non, pour restituer une grande quantité d'information. Leung définit des données comme étant intrinsèquement graphiques lorsqu'elles possèdent des relations spatiales :

"The data itself can also be classified according to whether it is inherently graphical in nature, with implicit spatial relationships ...".

Par exemple, les cartes et les plans (ville, région, bâtiments) sont intrinsèquement graphiques. A l'opposé, le graphe d'un site Internet et la hiérarchie des fichiers d'un ordinateur sont non-graphiques. Les données intrinsèquement graphiques sont naturellement posées dans un espace de visualisation. Les données non-graphiques devront faire l'objet d'une abstraction graphique (par exemple des noms de ville transformés en positions sur une carte de référence) ou bien être affichées par une technique de visualisation adaptée à la structure non graphique (une liste graphique des noms de ville). Nous illustrons, à la Figure 25, les quatre vues résultantes de notre interprétation des catégories de la taxonomie de Leung. Pour les exemples de la Figure 25a et de la Figure 25b nous avons considéré l'abstraction graphique qui fait correspondre au nom d'une ville française, sa position sur une carte de France. Cette abstraction graphique nous permet d'utiliser les mêmes données (une liste de 19 villes) pour illustrer les quatre catégories.

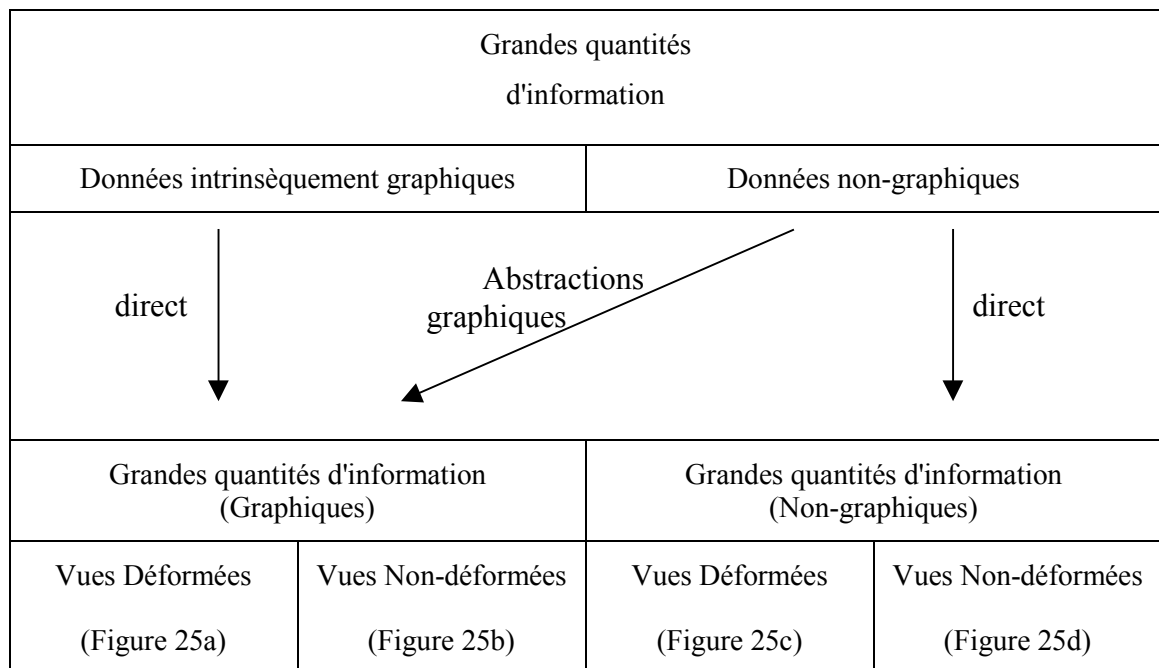


Figure 24 : Taxonomie de Leung [Leung 94].

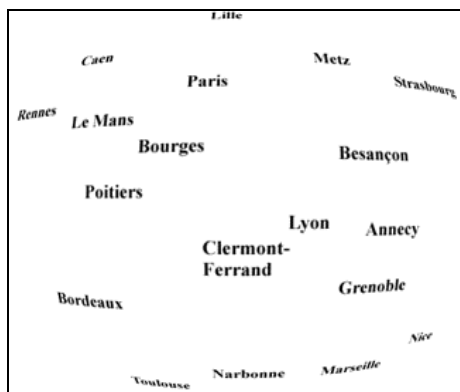


Figure 25a : Vue déformée de l'abstraction graphique des villes de France

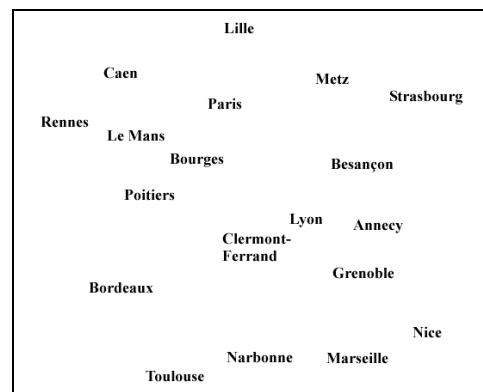


Figure 25b : Vue non déformée de l'abstraction graphique des villes de France

Ann. Besa. Bord. Bour. Caen Clermont-Ferrand Gren. Le Ma. Lill. Lyon Ma. Me. Na. Ni.

Figure 25c : Vue déformée de la liste des villes de France.

Annecy Besançon Bordeaux Bourges Caen Clermont-Ferrand Grenoble Le Mans Lille Lyon Marseille Metz Narbonne Nice Paris Poitiers Rennes Strasbourg Toulouse

Figure 25d : Vue non déformée de la liste des villes de France.

1.3 Taxonomie de Shneiderman

Dans ce paragraphe nous présentons les travaux de Ben Shneiderman. Les travaux auxquels nous nous référons [Shneiderman 96] proposent une taxonomie très pertinente pour

notre étude. Dans cette taxonomie sept types de données et sept types de tâches associées à la visualisation d'information sont identifiés.

Les 7 types de données sont :	Les 7 types de tâches sont :
► Données 1D (mono-dimensionnelles)	► Observer la globalité de l'espace
► Données 2D (bi-dimensionnelles)	► Zoomer (Agrandir et Réduire)
► Données 3D (tri-dimensionnelles)	► Filtrer
► Données temporels (périodiques)	► Obtenir des détails à la demande
► Données nD (multi-dimensionnelles)	► Relier des informations (trouver les relations existantes entre les informations)
► Données arborescente (hiérarchique)	► Accéder à l'historique des actions réalisées
► Données en graphe	► Extraire une partie des informations

Ces deux listes forment les graduations de deux axes d'un espace de conception simple mais très pertinent dans lequel le concepteur peut réfléchir aux tâches à offrir pour les différents types de données à mettre en œuvre. Le type de donnée utilisé par un système est généralement unique alors que plusieurs des tâches identifiées sont généralement offertes par un système.

1.4 *Taxonomie de Bruley*

Dans ses travaux, Christophe Bruley [Bruley 99] reprend ceux de Ben Shneiderman présentés ci-dessus. Il propose de considérer l'organisation des données induite par le point de vue plutôt que les types de données. Bruley ne considère donc pas l'organisation intrinsèque aux données mais plutôt l'organisation de l'espace présenté à l'utilisateur.

Comme Shneiderman, Bruley distingue les organisations temporelle, hiérarchique et relationnelle. Par contre, et contrairement aux travaux de Shneiderman, Bruley sépare les types 1D, 2D, 3D et nD dans deux catégories appelées spatiale et non-structurée. Cette séparation est à mettre en rapprochement avec les travaux de Leung présentés au paragraphe 1.2.. Bruley distingue le point de vue spatial qui est intrinsèquement graphique pour Leung, du point de vue multi-dimensionnelle non structuré qui est non-graphique pour Leung.

Catégories de point de vue		Description
Spatial	1D	Les entités sont organisées linéairement dans l'espace : les lignes de code d'un programme, les sections d'un document, etc.
	2D	Les entités sont positionnées ou possèdent une géométrie dans le plan : les villes d'un pays, la position des images dans un texte, etc.
	3D	Les entités sont positionnées ou possèdent une géométrie dans l'espace à trois dimensions.
Temporel		Les entités se succèdent ou se recouvrent dans le temps : historique médical d'un patient, débit d'un réseau, etc.
Hiérarchique		Les entités sont hiérarchiquement liées : un système de fichiers, une hiérarchie de classes d'objets, etc.
Relationnel		Les entités sont liées par une relation et forment une structure qui n'est pas hiérarchique : l'échange de messages entre objets d'un programme, les liens entre documents hypertextes, etc.
Non-structuré	1D	L'organisation séquentielle des entités est indépendante de toute considération temporelle ou spatiale des données. Un ensemble d'entités ordonnées selon un critère donné répond à cette définition : une liste de mots ordonnés alphabétiquement, une liste de fichiers, etc.
	2D	Le point de vue privilégie deux attributs qui ne sont pas liés à la position ou la géométrie des entités : la taille et le poids d'un ensemble de personnes par exemple.
	3D	Le point de vue privilégie trois attributs qui ne sont pas liés à la position ou la géométrie des entités.
	nD	Les entités possèdent un grand nombre d'attributs ($n > 3$) et le point de vue n'en privilégie aucun

Tableau 4 : Taxonomie de dix catégories de point de vue sur les données [Bruley 99].

Comme Chi, Bruley définit plusieurs étapes définissant quatre espaces différents. Ces espaces sont décrits comme des points de passage obligés de la conception et comme des éléments permettant de caractériser les représentations existantes. Bruley définit tout d'abord un espace des informations contenant les données à rendre perceptible. Il définit ensuite un point de vue sur cet espace qui est la perspective de l'ensemble des données, nécessaire à l'utilisateur. Bruley définit enfin deux espaces géométriques : l'espace de représentation et l'espace de construction. L'espace géométrique de représentation fixe la position des objets graphiques. L'espace géométrique de construction est la sous-partie de l'espace de représentation qui est réellement affiché. Par exemple un espace en trois dimensions peut au final n'être représenté que par une coupe en deux dimensions. Bien que ces travaux ne se réfèrent pas mutuellement, ces quatre espaces correspondent aux quatre étapes du processus de visualisation de Ed Chi.

1.5 Espace de conception de Stuart Card

La taxonomie de Stuart Card [Card 97] occupe une place à part dans le domaine de la visualisation. Ce travail est un essai original pour caractériser le processus de transformation d'une donnée informatique en trait graphique. Pour chaque système de visualisation, la taxonomie de Card propose un tableau qui le caractérise. Ce tableau qui présente en début de chaque ligne les variables du système à rendre observables est organisé en trois parties (trois ensembles de colonnes).

Dans la première partie, la taxonomie indique en titre de colonne une possible transformation (**F**) d'un ensemble de données (**D**) en un autre ensemble de données (**D'**). Les données à transformer et les données transformées sont nominales (**N**), ordonnées (**O**) ou quantitatives (**Q**). Cette distinction introduite dans [Stevens 46] a ensuite été utilisée par Jock Mackinlay [Mackinlay 86] pour caractériser les données informatiques. En plus Card distingue les variables quantitatives spatiales (**QX**) et non spatiales (**Q**). Cette distinction équivaut à celles de Leung (graphique / non graphique) ou Bruley (spatiale / non-structurée)

Dans la seconde partie du tableau, la taxonomie fait correspondre à chaque ligne représentant une variable du système un trait graphique indiqué en colonne.

Les traits graphiques sont des moyens de représenter graphiquement une variable et peuvent être une position dans les trois dimensions de l'espace (**X**, **Y** et **Z**), une fenêtre temporelle (**T**), une propriété rétinienne (**R**), une connexion (**---**) ou enfin une relation d'emboîtement (**[]**). Le terme peu évocateur de propriété rétinienne est un terme générique pour l'une des propriétés suivantes : la couleur, la taille, la forme, le niveau de gris, l'orientation ou la texture. Ces termes sont regroupés afin d'éviter l'explosion du nombre de colonnes.

Pour les trois dimensions de l'espace, Card raffine les associations possibles entre une variable et une dimension par une liste de marques : le point (**P**), la ligne (**L**), la surface (**S**), l'aire (**A**) et le volume (**V**).

Nous ne relevons pas la dernière partie de la taxonomie (**CP**) qui n'est pas assez détaillée pour être reporté dans cette thèse.

Armé de cette notation, Card fournit de nombreux exemples de caractérisations de systèmes parmi lesquels le FilmFinder. La Figure 26 présente sa caractérisation. Dans cet exemple nous notons que l'année de parution d'un film et sa note de popularité permettent de positionner un petit rectangle de couleur pour chaque film. La couleur du rectangle dépend du type de film. Cette variable ne possède pas de relation d'ordre particulière dans le système

FilmFinder, elle est donc notée **N**. L'année et la qualité sont par contre notées **Q** car ces variables sont quantitatives. L'année sert à positionner les rectangles sur l'axe des abscisses donc un **P** est inscrit dans la case (Année, X). De même la qualité sert à positionner les rectangles sur l'axe des ordonnées. Le **C** dans la case (Type, R) sert enfin à indiquer que la Couleur transcrit le type de film.

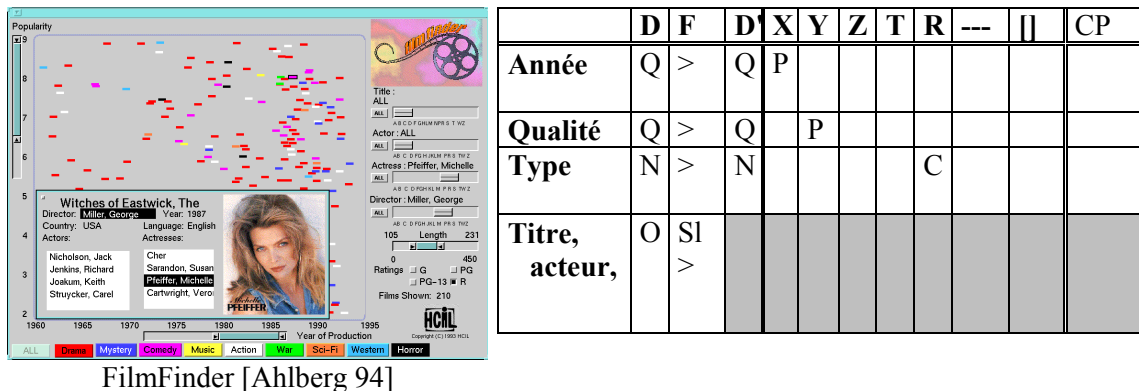


Figure 26 : Caractérisation du système FilmFinder [Ahlberg 94] par la taxonomie de Card [Card 97].

1.6 Synthèse : un cadre d'étude fédérateur

Les espaces de conception traitant des types de données, des traits graphiques et des tâches de visualisation sont complémentaires pour le travail du concepteur. Toutefois, la conception de techniques de visualisation requiert un point de vue global pour comprendre et mettre en relation les apports des différentes taxonomies. La décomposition du processus de visualisation en quatre étapes apparaît dans plusieurs travaux présentés. En nous appuyant sur ce processus, que nous attribuons à Ed Chi, nous proposons le cadre d'étude fédérateur de la Figure 27 dans lequel tous les éléments de conception que nous venons d'exposer sont localisés. Chaque espace ou taxonomie est décrit par ses points contributifs et son auteur entre parenthèses.

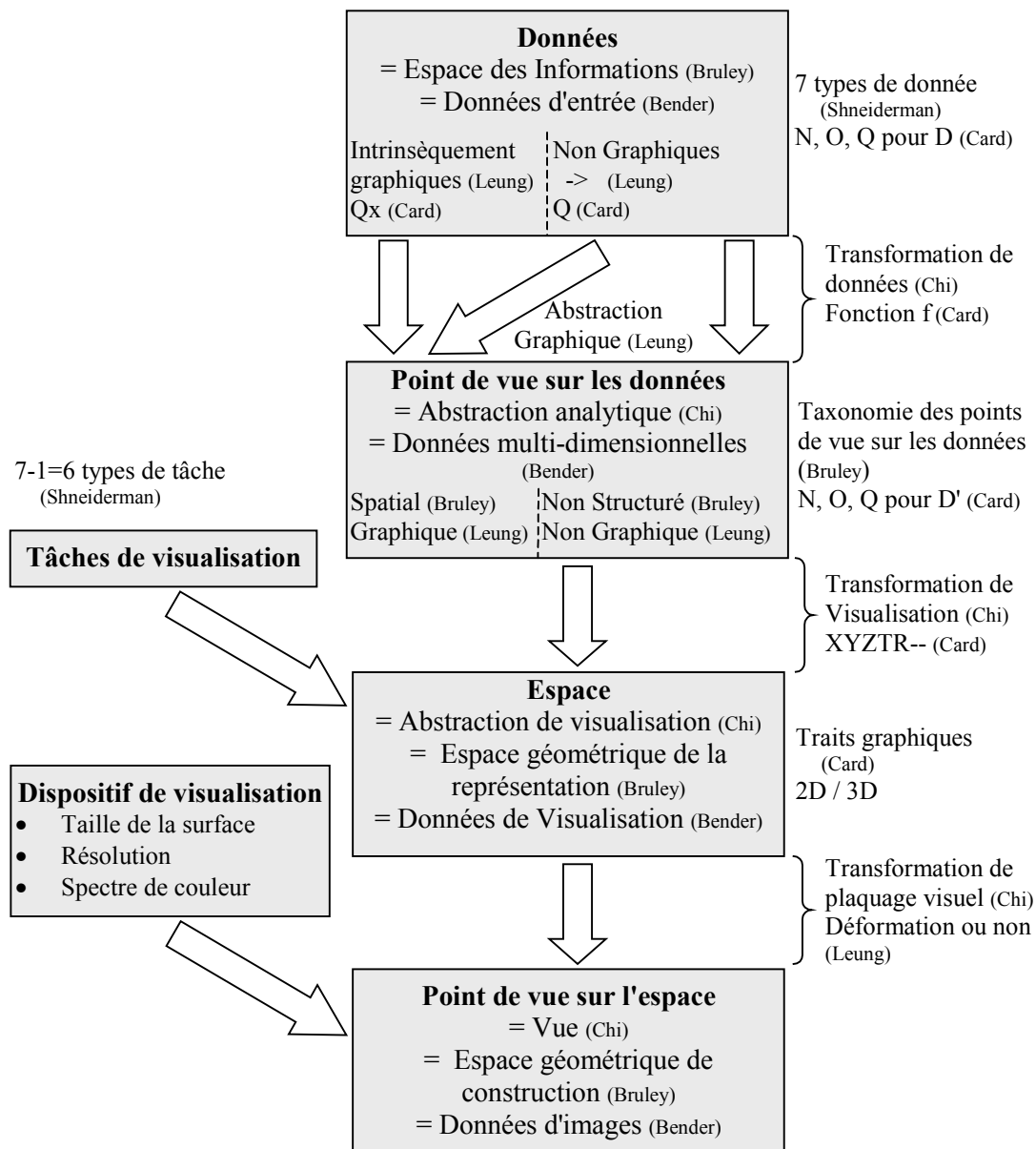


Figure 27 : Cadre d'étude fédérateur du processus de visualisation.

De cette synthèse, nous retenons les quatre étapes suivantes :

- Données
- Point de vue sur les données
- Espace
- Point de vue sur l'espace (ou vue)

La première et la troisième étapes (respectivement nommées Données et Espace) représentent des états incontournables du processus de visualisation. La seconde et la quatrième

étapes sont des points de vue sur l'état qui les précède, c'est-à-dire un choix d'interprétation du concepteur de cet état.

1.6.1 Données

L'étape intitulée "Données" se situe en amont de la conception de l'interface (regroupement des données, indexation, gestion des droits d'accès, etc.). Dans nos travaux, nous faisons l'hypothèse que les données sont disponibles et définies. Leurs caractéristiques ont néanmoins un impact évident sur la conception. Ainsi les sept types de données de Shneiderman, les caractéristiques Nominales, Ordonnées et Quantitatives (N, O et Q) de Card, les deux types (Intrinsèquement graphique et non graphique) de Leung constituent des approches pour caractériser les données à cette étape du processus de visualisation.

1.6.2 Point de vue sur les données

La seconde étape intitulée "Point de vue sur les données" correspond à une étape intermédiaire dans le processus de visualisation. Cette étape est relative à l'organisation des données (c'est-à-dire à la structure) que le concepteur met en œuvre dans la visualisation. Cette organisation consiste à prendre un point de vue sur les données parmi la quasi-infinité de possibilités d'organiser les données.

Par exemple, à partir d'un ensemble de pages Web, il est possible de construire un graphe, une liste ordonnée ou encore un historique des pages. Le graphe peut représenter tous les liens HTML, des liens de sémantique ou encore des liens correspondant aux parcours des utilisateurs précédents. La liste peut être ordonnée par le titre, la date ou encore la pertinence vis-à-vis d'une requête. L'historique peut correspondre à la date de création, de dernière modification ou de dernière visite de la page, etc.

Cette étape "Point de vue sur les données" doit intervenir dans une conception centrée sur l'utilisateur car le choix de la structure qui sera ultérieurement affichée, dépend de la tâche de l'utilisateur. En reprenant l'exemple de la Figure 25, nous pouvons argumenter que la structure spatiale 2D de la carte est mieux adaptée pour chercher un port sur la Méditerranée alors que la structure en liste est mieux adaptée pour chercher la capitale de la France, dont nous supposons que tout le monde connaît le nom.

Les catégories de point de vue de Bruley et les caractéristiques Nominales, Ordonnées et Quantitatives de Card (N, O, Q) sont des éléments contributifs essentiels à cette étape. Un point de vue peut être spatial (ou graphique selon les termes de Leung) pour les deux raisons identifiées par Leung :

- ▶ parce que les données sont intrinsèquement graphiques,
- ▶ parce qu'une abstraction graphique a permis de prendre un point de vue spatial sur l'espace d'information.

1.6.3 Espace

La différence entre l'étape Données et l'étape Espace est l'existence dans l'espace d'un modèle spatial des informations qui est utilisé pour l'affichage. L'espace calculé dans cette étape possède des attributs spatiaux que l'utilisateur percevra (quadrillages, alignements, pavages, recouvrement, etc.). Par exemple un graphe codé par une liste de nœuds et de transitions met en œuvre, dans cette étape, un algorithme de placement qui calcule des positions optimales pour les nœuds [Herman 00].

A cette étape, nous plaçons la taxonomie des traits graphiques proposée par Card. Nous distinguons également les espaces en deux et trois dimensions. Nous ne considérons pas la géométrie hyperbolique à ce niveau car nous ne voulons pas que le langage d'interaction repose sur un espace non-euclidien. Ces espaces sont difficiles à comprendre par un utilisateur et donc difficile à justifier pour le choix de l'espace. Une vue hyperbolique est toujours utilisée, en visualisation, pour représenter un espace euclidien déformé. Seuls de bons mathématiciens peuvent comprendre ces espaces, dont les propriétés particulières sont décrites par exemple dans [Herman 00] ("*...the sum of the internal angles of a triangle is no longer 180 degrees...*").

Dans le processus de visualisation, cette troisième étape du processus dépend de l'étape précédente (le placement d'un graphe n'est pas le même que le placement d'une liste ou d'un arbre). Elle dépend aussi des tâches que l'utilisateur effectue dans cet espace. Ces tâches sont beaucoup plus précises que celles évoquées à l'étape précédente. Les tâches proposées dans la taxonomie de Shneiderman correspondent à ce niveau d'abstraction sauf la tâche "Obtenir un historique". Cette dernière tâche s'applique à l'étape précédente où elle signifie choisir un point de vue temporel sur les données. Il s'agit d'une tâche générale. Pour illustrer notre discours, nous considérons l'exemple suivant : considérons un même point de vue sur les données (une structure en arbre) et considérerons plusieurs types d'espaces. Nous proposons de construire un espace avec l'algorithme des Treemaps [Shneiderman 83] si la tâche est d'observer la globalité de l'espace. Nous préférons construire un espace en graphe d'arbre si la tâche est de trouver les relations existantes entre les informations. Ces deux espaces sont à deux dimensions (2D) mais l'algorithme des Treemaps utilise le trait graphique de l'inclusion spatiale alors que le graphe d'arbre utilise le lien (taxonomie de Card).

1.6.4 Point de vue sur l'espace

La quatrième étape se nomme "point de vue sur l'espace". Elle correspond à la dernière étape de notre cadre d'étude avant la perception de l'information par l'utilisateur. Cette dernière transformation présente toujours à l'utilisateur un espace à deux dimensions, borné et quadrillé. Le dispositif de visualisation qui permet de rendre perceptible cet espace est soit un écran soit une imprimante. Le quadrillage que nous évoquons est donc une matrice de pixels ou de points d'encre.

Dans l'exemple d'un espace à trois dimensions, le point de vue correspond à celui de la caméra placée dans la scène 3D. Ce point de vue spécifique implique des projections, des faces cachées et des effets d'optique qui sont autant de particularités de cette quatrième étape du processus de visualisation. Dans un système de visualisation d'une table de données comme Excel [Excel], la quatrième étape consiste simplement à afficher un sous-ensemble des cellules du tableau. Ce système, dont nous reproduisons l'interface au chapitre VII (page 196), propose un point de vue partiel de l'espace d'information et rend le reste de l'espace accessible par des barres de défilement. C'est aussi dans cette étape que nous considérons une déformation en œil de poisson. La vue engendrée par cette déformation peut être hyperbolique ou non selon l'algorithme utilisé.

Une vue **hyperbolique** est une vue déformée d'un espace dont l'utilisateur perçoit l'infini depuis son point de vue. Nous continuons de parler de vue hyperbolique localement (loin de l'infini) lorsque la déformation perceptible est celle d'une vue hyperbolique. Cette définition n'est pas du tout celle du mathématicien mais celle du concepteur d'une technique de visualisation. Pour nous, une vue hyperbolique est équivalente à une vue en œil de poisson [Furnas 86].

Nous remarquons dans cette étape qu'il est nécessaire au concepteur de connaître les propriétés du dispositif d'affichage. Par rapport aux dispositifs très variés considérés dans les applications multimodales, les dispositifs d'affichage possèdent trois caractéristiques communes qui influencent énormément l'affichage d'un espace :

- la taille de la zone d'affichage (exprimée en centimètres ou en pouces),
- la résolution (exprimée en nombre de points par centimètre carré ou par pouce carré),
- le spectre de couleurs (exprimé en nombre de couleurs, entre 2 et 16 millions).

Ces caractéristiques ne sont pas forcément celles du point de vue sur l'espace car les environnements graphiques actuels utilisent des fenêtres. L'utilisateur choisit, en retaillant ses fenêtres, la sous-partie du dispositif d'affichage réellement utilisée par une technique de

visualisation. Lorsque la quantité d'information devient importante, utiliser toute la surface du dispositif devient une nécessité. En considérant la séparation dispositif / langage d'interaction, nous avons choisi de regrouper ces trois caractéristiques dans un élément à part de notre cadre d'étude.

1.6.5 Utilisation du cadre d'étude

Nous avons identifié de nombreux espaces de conception souvent bornés à un point de vue restrictif. Notre contribution réside dans une vue unificatrice des différents espaces exposés. La coexistence de nombreux espaces de conception dans un même cadre structurant permet de mieux cerner la conception d'une technique de conception.

Ce cadre d'étude identifie des étapes du processus de visualisation. Les choix effectués à chacune des étapes aboutissent à la définition de la technique d'interaction ou modalité graphique. L'étape Donnée, nous l'avons dit, sort de notre cadre d'étude. En effet, le choix et la collecte des données sont des activités à la frontière de la conception de l'interaction. Comme le souligne notre cadre de conception de la multimodalité en sortie du Chapitre III, les données sont une entrée au processus de définition de l'expression multimodale de sortie. Le type de ces données influence nécessairement le choix des modalités et constitue donc un facteur qui avec le contexte d'interaction va influencer sur le choix d'une modalité simple ou composée. Les types de données dans le processus de visualisation vont influencer le choix d'une technique de visualisation, mais ne peuvent en aucun cas caractériser la technique de visualisation elle-même, qui est un véhicule de l'information.

Nous retenons donc les trois niveaux suivants : "points de vue sur les données", "espace" et "point de vue sur l'espace". Placées dans notre cadre de conception de la multimodalité du Chapitre III, ces niveaux affinent notre définition (dispositif, langage) d'une modalité, en affinant la facette langage. Celle-ci dans le cas de la visualisation de données se caractérise à trois niveaux d'abstraction : "points de vue sur les données", "espace" et "point de vue sur l'espace". Aussi nous caractérisons une modalité selon les trois niveaux de notre cadre fédérateur de la visualisation. La caractérisation d'une modalité permet son choix. Il convient aussi de caractériser la composition de modalités (Chapitre III) afin de complètement cerner l'espace des possibilités. Là encore, nous considérons les trois dernières étapes de notre cadre fédérateur de la visualisation, comme expliqué à la section suivante.

2 *Composition des modalités de visualisation*

Pour cette étude, nous avons identifié deux types de compositions intéressantes et spécifiques à la visualisation de grandes quantités d'information. Nous abordons le premier cas, la composition de modalités pour les vues multiples, au paragraphe 2.1. Nous présentons ensuite les techniques de visualisation qui composent une modalité pour le focus et une modalité pour le contexte au paragraphe 2.2.

2.1 *Vues multiples*

Le premier cas de composition de modalités qui nous intéresse correspond aux vues multiples. En effet les vues multiples sont courantes dans les systèmes de visualisation comme en témoignent les travaux de Michelle Baldonado [Baldonado 00] qui proposent des règles de conception pour ces systèmes. Le lecteur remarquera que nous ne proposons pas de patron pour ce type de composition dans cette thèse. Nous vous invitons à vous reporter aux travaux de Baldonado qui proposent de bons exemples d'heuristiques de conception, malgré que la forme utilisée ne soit pas celle du patron.

Pour caractériser les systèmes de vues multiple, nous appuyons notre démarche sur le cadre fédérateur que nous venons de présenter. Comme l'illustrent les quatre cas de la Figure 28, nous définissons quatre classes de vues multiples en utilisant le principe de la fermeture Eclair [Patterson 94]. Ces classes définissent l'étape du processus de visualisation qui sépare la conception pour arriver à plusieurs vues.

- ▶ les vues multiples sur des données de types différents (par exemple le trafic autoroutier en temps réel et des statistiques d'accidents),
- ▶ les vues multiples sur les mêmes données mais selon un point de vue différent (par exemple une liste alphabétique des membres d'une famille et l'arbre généalogique de cette même famille),
- ▶ les vues multiples sur un même point de vue des données mais dans deux espaces (par exemple le même arbre généalogique rendu observable par des techniques différentes comme celles répertoriées dans [Herman 00]),
- ▶ les vues multiples sur un même espace (vue en fil de fer et vue solide d'une scène 3D).

Ces quatre types de vues multiples se conçoivent au regard du Chapitre III comme des systèmes multimodaux en sortie.

Le travail sur la caractérisation des compositions de modalités de sortie nous permet d'identifier la première classe de vues multiples (Figure 28a) comme une composition

complémentaire (niveau sémantique). Ce cas de vues multiple est celui qu'offre tous les systèmes d'exploitation et démarrant deux applications différentes sur des données différentes.

La seconde classe de vues multiples (Figure 28b) apparaît comme une composition complémentaire et redondante (niveau sémantique) puisque les données sont les mêmes (redondance) mais leur organisation diffère (complémentarité). Si l'organisation des données est la seule information pertinente alors la composition est strictement complémentaire.

La troisième classe de vues multiples (Figure 28c) comprend des vues sur des structures identiques mais cela ne signifie pas une composition de redondance. En effet, deux techniques d'interaction permettant de visualiser les mêmes structures (par exemple l'arborescence de fichiers) peuvent apporter des informations complémentaires dans l'organisation spatiale des données. Le simple alignement des données, verticalement ou horizontalement, peut énormément changer l'utilisation de la technique de visualisation. Comme le cas précédent, la composition reste donc complémentaire et redondante.

Enfin la quatrième classe de vues multiples (Figure 28d) est encore complémentaire et redondante au niveau sémantique mais nous nous remarquons surtout que c'est une composition de divergence au niveau syntaxique. En effet la construction commune de l'espace fait partie du langage d'interaction. Les points de vue sur l'espace différents font également partie du langage d'interaction.

Nous ne considérons pas une vue partielle de l'espace d'information comme nous considérons une vue hyperbolique. Dans cet exemple, le point de vue sur l'espace peut cacher ou mettre en valeur la structure. Dans le cas général, nous notons que certaines vues ne présentent qu'une partie de l'espace (surfaces cachées d'un monde 3D, zoom sémantique, etc.).

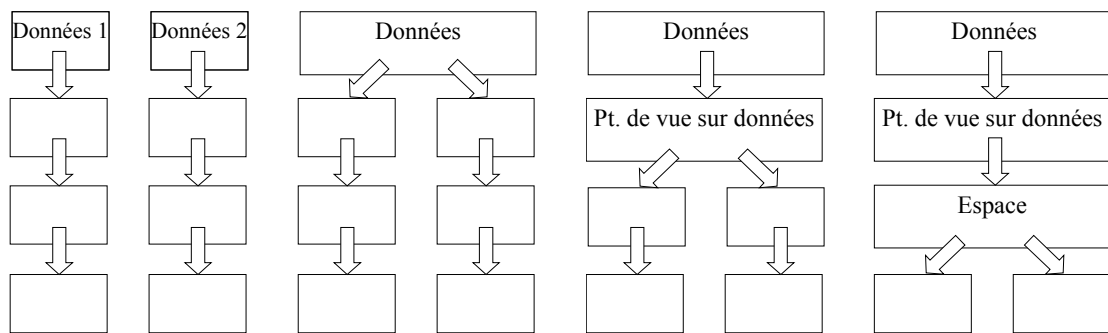


Figure 28a : Vues multiples sur des données différentes.

Figure 28b : Vues multiples selon un point de vue différent sur les données.

Figure 28c : Vues multiples du même point de vue sur les données.

Figure 28d : Vues multiples du même espace selon des points de vue différents.

2.2 Composition du focus et du contexte

Le second point qu'il est important d'aborder est la composition du focus dans le contexte. Comme nous l'avons annoncé en introduction de ce chapitre, la modalité de visualisation d'une grande quantité d'information peut souvent se décomposer en une modalité pour le focus et une modalité pour le contexte. Nous verrons à la section 3 la caractérisation de ces deux types de modalités. Au regard de l'étude que nous avons menée à la section 4 du Chapitre III (page 59), leur composition au niveau sémantique est l'une des suivantes :

- Complémentaire ou
- Complémentaire et Redondante.

En effet, l'information contenue dans le focus est forcément complémentaire de la structure globale présentée dans le contexte. Une partie de la structure peut par contre apparaître à la fois dans le focus et dans le contexte.

Un autre aspect qu'il est important de souligner est l'aspect spatial. En effet l'affichage du focus **dans** le contexte est un aspect important de la réussite de ce principe de visualisation. Nous faisons le rapprochement entre l'ordre déictique du "met ça là" et la composition spatiale du focus dans le contexte. La richesse de ce message ne se trouve pas uniquement dans l'affichage du focus et du contexte mais également par leurs positions respectives. Pour faire écho au message de Jock Mackinlay dans [Mackinlay 91], nous distinguerons les techniques dont le focus est spatialement intégré dans le contexte. Nous reviendrons sur cet aspect fondamental au paragraphe 3.3.1 (page 114) quand nous caractériserons la fonction de déformation qui fait apparaître un focus et un contexte.

Nous considérons enfin la composition du contexte et de plusieurs foci (pluriel latin de focus). En effet, plusieurs travaux ([Leung 94], [Carpendale 97] et [Sarkar 93]) évoquent la possibilité pour un même système de présenter plusieurs foci. La composition du contexte et d'un focus met en œuvre deux modalités alors que la présence de plusieurs foci peut entraîner la composition de trois modalités quand il n'est pas possible d'utiliser la même modalité pour chaque focus.

3 *Caractérisation des techniques de visualisation*

Comme nous l'avons fait pour les interfaces multimodales, après avoir caractérisé la composition, nous proposons des caractéristiques qui définissent le contexte d'interaction. Nous utiliserons ces caractéristiques à la section suivante pour écrire des patrons de conception.

Pour organiser ce travail nous utilisons encore une fois la décomposition en quatre étapes du processus de visualisation mais nous ne nous intéressons pas à la caractérisation des données. Nous proposons donc trois caractérisations correspondant aux trois dernières étapes du processus. Au paragraphe 3.1 nous proposons de rapprocher les propriétés N, O, Q et les dix classes de points de vue de Bruley. Ce rapprochement nous permet de proposer un espace de conception plus précis pour le choix d'un point de vue sur les données. Au paragraphe 3.2 nous regroupons les propriétés qui nous semblent pertinentes pour décrire la construction de l'espace. Enfin, au paragraphe 3.3, nous proposons une étude plus poussée des points de vue sur l'espace, organisée selon deux niveaux d'abstraction.

3.1 *Caractérisation des points de vue sur les données*

Comme nous venons de l'annoncer, nous proposons de rapprocher les travaux de Card et ceux de Bruley (donc par extension ceux de Shneiderman). Ce rapprochement a pour but de décrire les différents points de vue qu'il est possible d'adopter sur les données. Nous pensons que les travaux de Stevens (propriétés N, O et Q) permettent de décliner les points de vue sur les données proposés par Bruley.

En effet, les travaux de Card sont inspirés de ceux de Mackinlay [Mackinlay 86] et [Stevens 46] mais nous remarquons plusieurs exemples fournis par Card qui décrivent le point de vue par les attributs nominal, ordonné et quantitatif. Dans les exemples de Card, une structure spatiale est décrite comme quantitative (Q_x et Q_y) et une structure hiérarchique par le sigle $N*N$. Ces deux descriptions ne sont pas valables pour tous les systèmes. Une structure hiérarchique peut être ordonnée (par exemple la hiérarchie de titre de cette thèse est ordonnée). Un espace peut être également ordonné (par exemple la grille d'un tableur de donnée).

Une description nominale peut toujours être ordonnée, il suffit de trouver n'importe quelle relation d'ordre arbitraire (par exemple l'emplacement de chaque donnée dans la mémoire centrale de l'ordinateur). De même les données quantitatives peuvent toujours être ramenées à un point de vue ordonné puisque toutes les données numériques sont discrètes et finies. Le concepteur peut toujours choisir de ne pas utiliser les opérateurs algébriques qui caractérisent une donnée quantitative. Nous considérons donc que toute donnée peut faire l'objet d'un point de vue ordonné.

Le concepteur dispose de deux points de vue alternatifs (nominal et quantitatif) si les données le permettent. La distinction entre un point de vue ordonné et quantitatif est perceptible dans certains systèmes qui permettent de manipuler un entier à l'aide d'une glissière (point de vue quantitatif) ou par une liste à choix multiples de valeurs prédéfinies (point de vue ordonné). L'affichage des fichiers selon plusieurs vues comme celles de la Figure 6a à la Figure 6bc (page 41) relèvent d'un point de vue nominal (vue avec des icônes) et d'un point de vue ordonné (vue en liste) sur les mêmes données.

En conclusion nous considérons les quinze points de vue de la Figure 29. Les zones grisées de la figure indique les combinaisons qui nous paraissent impossibles. Nous appelons **granularité** du point de vue, l'aspect nominal, ordonnée ou quantitatif et nous appelons **structure** les 10 classes de points de vue identifiées par Shneiderman et Bruley.

Point de vue	Granularité	Nominale	Ordonnée	Quantitative
Structure				
Spatiale	1D		Echangeurs et zones de repos le long d'un axe autoroutier	Repères kilométriques le long d'un axe autoroutier
	2D		Grille de bataille navale	Outils de dessin vectoriel
	3D		Quadrillage de l'espace dans une fouille archéologique	Outils de CAO
Temporelle			Calendrier en jours ouvrables	Journal d'événements
Hierarchique		Arbre d'héritage de classes	Arbre généalogique (ordre de naissance)	
Relationnelle (sauf arbre)		Web	Treillis, Multitrees	
Non-structurée	1D	Liste des courses	Liste de choses à faire triés par priorité	Liste de notes
	2D	Chaque dimension doit être considérée séparément. On distinguera finalement les point de vue homogènes, où toutes les dimensions auront la même granularité, des points de vue hétérogènes.		
	3D			
	nD			

Figure 29 : Tableau récapitulatif des différents points de vue sur les données contenant un exemple dans chacune des quinze cases non grisées. Les données non-structurées à plus d'une dimension ne sont pas représentables dans cette taxonomie.

Un arbre ne peut pas être quantitatif mais les fils d'un nœud peuvent être ordonnés ou peuvent être simplement nominaux. Les exemples déjà cités pour motiver le besoin de raffiner la taxonomie de Card sont des exemples de points de vue nominaux et ordonnés d'un arbre.

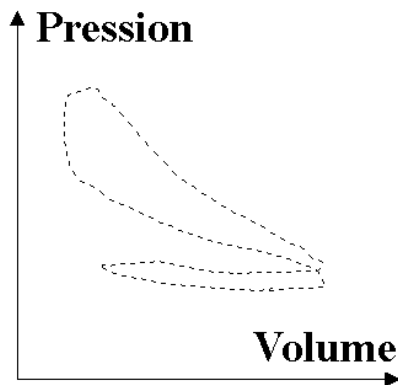
Nous considérons que les données temporelles ne peuvent pas être nominales. En effet le temps est une structure ordonnée par nature. Les données peuvent être quantitatives ou ordonnées. Si la structure temporelle est exprimée en seconde il est possible d'y appliquer des opérateurs arithmétiques alors que si la structure temporelle est exprimée en jours ouvrables pour une entreprise, de nombreux opérateurs arithmétiques ne sont plus applicables.

Les données linéaires (1D, 2D 3D et nD) non spatiales peuvent également être raffinées par les trois attributs de Card. Toutefois chaque dimension doit être considérée séparément. Le cycle thermodynamique Pression-Volume d'un moteur à explosion (Figure 30a) et la table périodique des éléments de Mendeleïev (Figure 30b) sont tous les deux des données 2D non spatiales mais se différencient par la granularité de leur dimensions. La table périodique utilise

les dimensions de l'espace de façon discrète et se présente donc comme une variable ordonnée (O). Le cycle thermodynamique est composé de points de passage caractérisés par une pression et un volume qui sont deux variables quantitatives. La structure est donc 2D quantitative (Q).

Un logiciel comme Illustrator [Illustrator] permet de manipuler des images vectorielles en deux dimensions réparties sur plusieurs calques superposés. Les deux dimensions du dessin sont quantitatives (Q) puisque le dessin est vectoriel. La troisième dimension, composée de calques, est simplement ordonnée (O). Cet exemple montre que les attributs de Card ne doivent pas s'appliquer globalement à l'ensemble des dimensions mais indépendamment à chacune des dimensions. Cette remarque nous amène quand même à distinguer deux types de point de vue sur les données 2D, 3D et nD non structurés :

- les points de vue homogènes où toutes les dimensions ont la même granularité,
- les points de vue hétérogènes qui mélangent les dimensions quantitatives, ordonnées et nominales.



	1 IA																	18 VIIIA 8A
1	H 1.008	2 IIA 2A									13 3A	14 4A	15 5A 5A	16 6A 6A	17 7A 7A			He 4.003
2	3 Li 6.94	4 Be 9.012									5 B 10.81	6 C 12.01	7 N 14.01	8 O 16.00	9 F 19.00	10 Ne 20.18		
3	11 Na 22.99	12 Mg 24.31	13 III 3B	14 IV 4B	15 V 5B	16 VI 6B	17 VII 7B	8 ----- VIII	9 ----- VIII	10 ----- VIII	11 IB	12 IIB 2B	13 Al 26.98	14 Si 28.09	15 P 30.97	16 S 32.07	17 Cl 35.45	18 Ar 39.95
4	19 K 39.10	20 Ca 40.08	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
5	Rb 85.47	Sr 87.62	Y 88.91	Zr 91.22	Nb 92.91	Mo 95.94	Tc 98.91	Ru 101.1	Rh 101.07	Pd 106.4	Ag 107.9	Cd 112.4	In 114.8	Sn 118.7	Sb 121.8	Te 127.6	I 126.9	Xe 131.3
6	55 Cs 132.9	56 Ba 137.3	La* 138.9	Hf 178.5	Ta 180.9	Ti 183.8	Re 186.2	Os 190.2	Ir 192.2	Pt 195.1	Au 197.0	Hg 200.5	Tl 204.4	Pb 207.2	Bi 208.9	Po 210	At 210	Rn 222
7	Fr (223)	Ra (226)	Ac~ (227)	Th (232)	Pa (231)	U (238)	Np (237)	Pu (244)	Am (243)	Cm (247)	Bk (247)	Cf (251)	Es (252)	Fm (257)	Md (258)	No (259)	Lr (262)	

Figure 30a : Exemple de donnée 2D quantitative : le cycle thermodynamique Pression-Volume d'un moteur à explosion.

Figure 30b : Exemple de donnée 2D ordonnée : la table périodique des éléments de Mendeleïev.

Contrairement aux données linéaires non spatiales, les données spatiales (1D, 2D et 3D), ne peuvent être que quantitatives ou ordonnées pour les mêmes raisons que les données temporelles. Nous ne considérons pas des noms de ville comme des données spatiales. De plus, pour des raisons d'homogénéité, nous ne considérons pas, comme dans le paragraphe précédent sur les données linéaires non spatiales, que les dimensions doivent être traitées séparément. Nous considérons une grille de bataille navale comme une donnée spatiale (2D) et ordonnée (O) mais il nous semble impossible d'envisager un espace composé de dimensions ordonnées et quantitatives.

Nous notons enfin que les graphes, comme les arbres, ne peuvent pas être quantitatifs. Les arbres qui sont un type spécial de graphe sont ordonnés par nature. Les graphes peuvent donc être ordonnés ou nominaux. Les graphes qui sont ordonnés ne se limitent pas aux arbres. Les graphes acycliques tels que les Multitrees [Furnas 94] le prouvent.

3.2 *Caractérisation de l'espace*

Le paragraphe précédent a présenté notre approche de la conception du point de vue sur les données. Dans le cadre d'étude que nous avons présenté au paragraphe 1.6, le niveau suivant est l'espace. Dans cette étape, le concepteur met en œuvre le **placement** des éléments de l'interface dans un **espace** qu'il construit.

Pour caractériser un espace, nous considérons tout d'abord le nombre de dimensions de l'espace : deux dimensions (2D) ou trois dimensions (3D). Cette caractéristique distingue les deux types de placements utilisés dans les techniques de visualisation. Un espace 3D offre plus de possibilités et correspond mieux à l'espace de notre monde. Ce type d'espace implique néanmoins deux principaux défauts :

- ▶ La projection implique que certains éléments peuvent se retrouver cachés aux yeux de l'utilisateur (contradiction avec le critère d'ergonomie de l'observabilité présenté en annexe 1, page 259).
- ▶ La navigation dans les mondes 3D est un problème difficile car nous ne disposons pas de dispositif efficace à six degrés de liberté (3 pour la position dans l'espace et trois pour la direction).

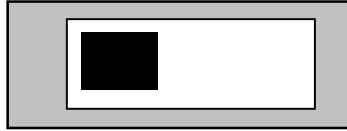
Le placement des données correspond à un espace qui peut avoir trois fonctions :

- ▶ Servir à transcrire certaines informations. Dans l'exemple de FilmFinder (voir page 97), chaque rectangle est placé dans l'espace 2D en fonction d'une donnée (année et popularité de chaque film).
- ▶ Correspondre à un espace existant. Dans l'exemple des systèmes d'information géographique tels que Sand [Sand] les informations sont placées sur une carte.
- ▶ Servir de support aux actions de l'utilisateur. Dans l'exemple de l'Explorer de Windows [Windows] et du Finder de MacOS [MacOS] (Figure 21, page 82) l'espace est utilisé par l'utilisateur pour placer lui-même ses fichiers.

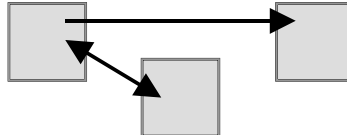
Nous considérons enfin, pour caractériser un espace, la présence des différents traits graphiques comme ceux présentés par Card [Card 97] et Mackinlay [Mackinlay 86]. Nous

avons choisi de caractériser la construction de l'espace par l'utilisation (éventuelle) des traits graphiques suivants :

- Les inclusions spatiales



- les liens



- les formes



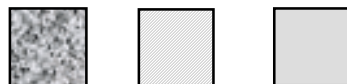
- les tailles



- les couleurs



- les textures



- les labels



L'**inclusion spatiale** intervient dans la construction d'un espace lorsqu'un élément, placé dans un autre, reflète une information pertinente pour l'utilisateur. Cette relation n'est pas symétrique et implique deux types de relation : *A contient B* et *B est contenu dans A*. Ces relations sont transitives. Ce trait graphique est lié à celui de la taille. La somme des tailles des éléments contenus dans un élément ne peut excéder la taille de ce dernier. L'algorithme des TreeMaps [Johnson 91][Shneiderman 91] illustré à la Figure 95 (page 215) utilise ce trait graphique pour représenter la structure hiérarchique des données. Les cartes géopolitiques qui présentent des départements dans leur région, les régions dans leur pays et les pays dans leur continent utilisent ce trait graphique également.

Nous considérons un **lien** lorsqu'une liaison est matérialisé dans l'espace entre deux éléments. Nous pourrions considérer ce lien comme un élément à part entière, d'un autre type. Toutefois la présence de lien dans des structures de base comme les graphes ou les hiérarchies nous incite, comme Card, à les considérer comme un trait graphique à part. Nous notons qu'il existe deux types de liens, les liens dirigés et les liens non dirigés. Les premiers sont souvent des flèches où la pointe indique l'élément d'arrivée du lien. Les liens non dirigés sont souvent des segments ou des courbes.

La **forme** des éléments est un trait graphique qui est très varié qui comprend le rectangle et le rond (2D) ou la sphère et la pyramide (3D). La forme peut être analogique si elle est assez compliquée et peut représenter une forme identifiable par l'utilisateur (bobine de film, volant de voiture, etc.). La forme peut parfois être décomposée en plusieurs aspects, chacun reflétant une facette différente des données. C'est le cas par exemple de la forme des pièces du jeu de Quarto (trouées ou non, carrées ou rondes et longues ou courtes).

La **taille** d'un élément de l'espace permet également de refléter une donnée. Ce trait graphique est difficilement utilisable dans un espace 3D car la taille est modulée par l'éloignement. Dans l'exemple des TreeMaps de la Figure 95 (page 215), la taille de chaque élément est proportionnelle à la taille du fichier représenté. Ce trait graphique, à priori sans limite, doit être utilisé avec prudence. Trop petit, un élément ne sera pas visible par l'utilisateur. Trop grands les éléments occuperont plus d'espace à l'écran qu'il n'est disponible. L'espace est une ressource critique dans le domaine de la visualisation de grands espaces d'information. A partir de la position et de la taille des éléments, le concepteur peut calculer l'espacement entre les éléments. La nécessité de laisser un espace entre les éléments pour la lisibilité de l'interface implique également des contraintes sur la taille des éléments.

Dans l'espace, la **couleur** d'un élément peut être uniforme ou bien représenter une **texture**. La texture ne peut s'appliquer que si la taille de l'élément est assez grande. L'utilisation d'une texture limite beaucoup le spectre des couleurs utilisables. La texture a toutefois pour avantage de pouvoir être analogique au sens de Bernsen (texture de bois, de marbre, de métal, etc.). Les couleurs unies peuvent être arbitraires ou non-arbitraires (rouge, orange, vert comme un feu de signalisation). Les couleurs peuvent très bien représenter une donnée quantitative (dégradé de couleur), ordonnée (plusieurs couleurs d'intensité plus ou moins grande) ou nominal (plusieurs couleurs de teintes différentes). La couleur peut être décomposée en sous-éléments

distinctifs comme les composantes rouge, verte et bleu (RGB¹) ou encore comme les composantes intensité, saturation et teinte (HSB¹).

Enfin les éléments de l'espace peuvent être un **label** ou contenir un label. Ce dernier permet, comme les autres traits graphiques, de refléter une propriété de l'élément représenté (par exemple le nom du fichier en dessous des l'icônes dans la Figure 21 à la page 82). Les espaces utilisant ces labels participent à la mise en œuvre d'une modalité linguistique au sens de Bernsen. Toutefois, comme la couleur, le label n'est pas un type simple mais peut être décomposé en plusieurs aspects graphiques. Si nous considérons le texte comme un élément graphique quelconque, tous les aspects que nous allons énumérer feraient partie de la forme. Notre travail en amont sur la multimodalité nous incite à considérer différemment cet aspect linguistique au sens de Bernsen. Nous différencions :

- ▶ la police de caractère utilisée dans le label (Arial, Helvetica, etc.),
- ▶ l'attribut gras ou non gras,
- ▶ l'attribut italique ou droit,
- ▶ l'attribut souligné, barré ou sans trait.

Ces aspects ne sont pas exhaustifs mais représentent ceux qui sont le plus utilisés.

En conclusion nous avons présenté un grand nombre de traits graphiques qui peuvent être utilisés pour rendre de manière graphique une information. Parmi ceux traditionnellement utilisés, certains sont **simples** (la taille, l'inclusion, et la texture) et d'autres peuvent être **décomposés** (le lien, le label, la forme et la couleur). Ces traits graphiques peuvent être combinés avec le placement (le fait de placer un élément à un endroit donné) pour rendre l'espace plus riche. Contrairement au placement, ces traits graphiques ne sont pas obligatoirement pris en compte par le concepteur.

3.3 *Caractérisation des points de vues sur l'espace*

La caractérisation des points de vues sur l'espace est le troisième et dernier point que nous abordons sur la caractérisation des techniques de visualisation. Ce niveau d'abstraction est extrêmement important dans cette thèse car c'est le niveau le plus proche de l'utilisateur. A contrario, le niveau donnée est le plus proche du système et nous en avons parlé.

Pour faire l'analogie avec les logiciels ludiques 3D, où l'utilisateur est matérialisé dans l'espace par un avatar, c'est à ce niveau que nous considérons l'emplacement de l'utilisateur.

¹ Nous utilisons des acronymes anglais, plus communément utilisés

Dans un système de visualisation l'utilisateur est rarement matérialisé dans l'espace alors nous parlerons de son point de vue que nous appelons aussi sa **vue** de l'espace. Dans cette vue, nous considérons également les actions de navigation dont l'étude est intimement liée à celle de la visualisation de grandes quantités d'information.

Comme le suggère la Figure 31, nous considérons deux niveaux d'abstraction dans l'activité de l'utilisateur. La visualisation décrit la lecture de la modalité en sortie par l'utilisateur. Intimement liés, mais à un plus haut niveau d'abstraction, les outils de navigation s'intéressent aux couples entrée-sortie de modalités mises en œuvre dans l'interaction avec la technique de visualisation. Les modalités en sortie utilisées par la navigation sont les modalités de visualisation. Nous considérons enfin la tâche de l'utilisateur, qui est centrale à la conception et qui peut se raffiner en sous-tâches de visualisation et en tâches articulatoires de navigation.

Nous remarquons à ce niveau que la navigation peut être prise en compte à l'étape précédente du processus de visualisation : la création de l'espace. L'activité de navigation pendant la création de l'espace correspond à des tâches de production d'un document à caractère spatial qui dépassent le cadre de notre étude.

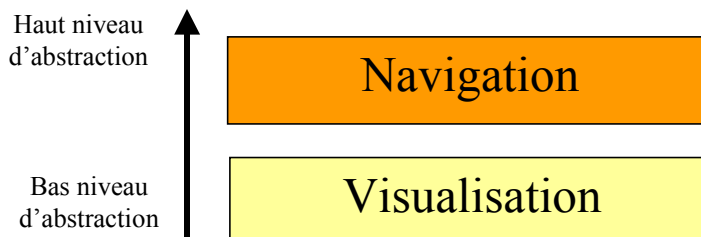


Figure 31 : Visualisation et navigation deux niveaux d'abstraction dans l'activité de l'utilisateur au niveau du point de vue sur l'espace.

Les deux niveaux d'abstraction considérés définissent également deux types de patrons :

- les patrons de visualisation indiquent comment construire le point de vue sur l'espace,
- les patrons de navigation préconisent tel ou tel outil de navigation selon le point de vue sur l'espace.

3.3.1 *Caractérisation de la visualisation*

Le premier niveau d'abstraction qui nous importe est la visualisation, c'est-à-dire l'image résultante du point de vue sur l'espace d'information. Quel que soit le point de vue de l'utilisateur sur une grande quantité d'information, nous pouvons le décomposer selon les concepts de "focus" et "contexte". Le focus est la modalité en sortie que l'utilisateur regarde pour réaliser sa tâche (par exemple lire des valeurs). A contrario, l'utilisateur perçoit le contexte

afin de situer son focus dans la grande quantité d'information. Certains points de vue ne présentent pas de contexte mais nous supposons que ces points de vue sont réservés à des tâches pour lesquelles l'utilisateur n'a pas besoin de situer l'information qu'il manipule.

Dans la visualisation de grande quantité d'information, le contexte à rendre perceptible est la structure de l'espace. Par sa taille cet espace ne peut pas être affiché en entier et en détail. Afin de réaliser une tâche qui nécessite la perception du contexte, l'utilisateur peut utiliser l'image mentale de la structure de l'espace qu'il a construit en déplaçant son focus à travers l'espace. Cette solution n'est pas réaliste quand la quantité d'information dépasse les limites de la mémoire humaine.

Puisque le contexte ne peut pas être affiché en détail, la solution consiste à afficher, dans l'espace disponible, le maximum d'informations utiles à l'utilisateur, à savoir :

- ▶ la structure de l'espace d'information,
- ▶ le voisinage direct autour du focus,
- ▶ un ou plusieurs foci.

Afin de cerner l'espace des possibilités pour ces différentes modalités de visualisation nous proposons l'espace de conception de la Figure 32 qui distingue une modalité selon qu'elle est précise ou vague, locale ou globale et déformée ou pas.

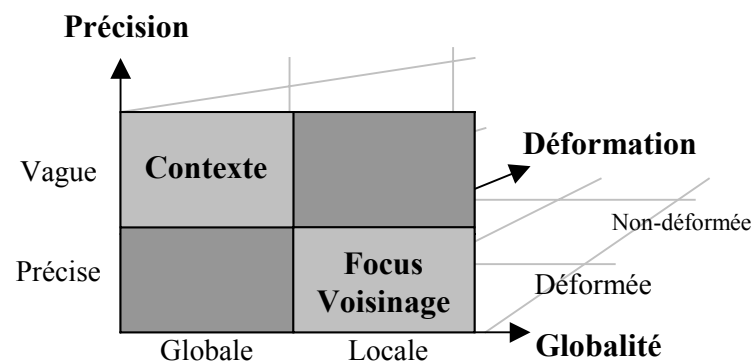


Figure 32 : Espace de conception des modalités représentant un point de vue sur l'espace d'information.

La propriété de précision pour un ensemble d'éléments d'information signifie que l'utilisateur trouve toutes les informations nécessaires à sa tâche dans le rendu de ces informations. Dans de nombreux systèmes, les représentations graphiques des objets manipulés ne sont pas assez précises. Le système fournit, dans une autre modalité, les informations précises sur un élément sélectionné par l'utilisateur. Cette modalité est souvent appelée "inspecteur d'objet". Cette modalité est précise pour les tâches d'édition et de consultation. La

modalités graphique qui présentent la totalité des éléments est vague. Le système Codewarrior [Codewarrior] utilise par exemple un inspecteur d'objets pour les interacteurs. Dans ces systèmes, le voisinage autour du focus n'est pas mieux représenté que le reste du contexte.

Nous distinguons, par la propriété de globalité, une modalité en sortie qui affiche la totalité de la structure de la totalité de l'espace (modalité globale) d'une modalité qui n'affiche qu'une sous-partie (modalité locale).

La Figure 33 illustre l'interface du jeu Warcraft [Warcraft]. Cet interface met en œuvre deux types de modalités dans une technique de visualisation appelée "vue radar". Cette technique met en œuvre une vue locale où le joueur manipule les unités (Focus) et une vue globale lui permettant de situer son focus dans l'espace de jeu (Contexte). Le système FilmFinder [Ahlberg 94], qui sera présenté à la Figure 39 (page 121), utilise un champs de points (modalité globale mais vague) et un descriptif de film (vue précise mais locale).



Figure 33 : Système Warcraft [Warcraft] utilisant une vue radar.

La Figure 34 illustre un second exemple d'utilisation conjointe d'une vue locale et précise et d'une vue globale et vague. Ce système, appelé DragMag [Ware 95] présente une vue globale et vague (en haut de la Figure 34) et plusieurs vues locales et précises (au nombre de deux au bas de la Figure 34).

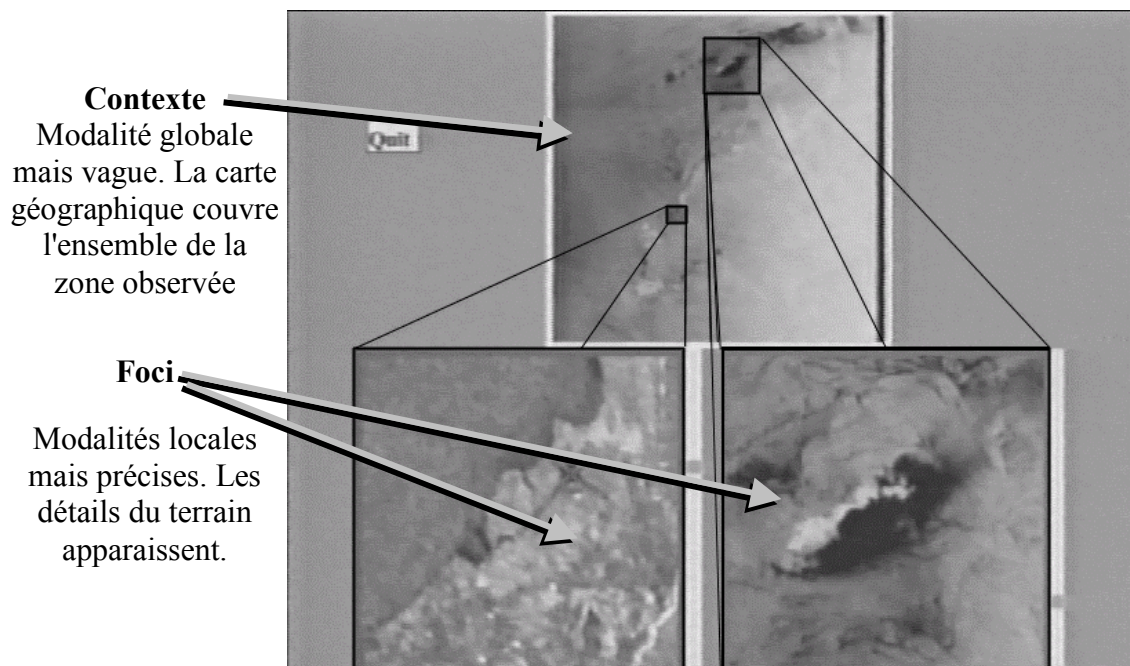


Figure 34 : Système DragMag [Ware 95] (Image aux contours retravaillés extraite de la vidéo de présentation du système DragMag).

Les propriétés de précision (vague ou précis) et de portée (globale ou locale) nous permettent de décrire correctement certaines techniques de visualisation mais pas celles qui reposent sur une fonction de déformation comme [Hovestadt 95]. Nous distinguons donc les techniques de visualisation qui utilisent une déformation de l'espace. Nous pensons que cette distinction est importante afin de séparer les deux familles de techniques identifiées depuis les travaux de Leung [Leung 94]. Les déformations que nous considérons sont celles qui permettent d'agrandir une partie de l'espace et de réduire le reste de l'espace afin de la faire rentrer dans les limites du dispositif. Pour rester cohérent avec le reste de notre discours, nous découpons en focus et contexte ces deux parties d'une modalité en considérant que les parties agrandies et réduites ne mettent pas en œuvre le même langage d'interaction. Dans les exemples de systèmes de ce paragraphe, nous ne manquerons pas de mettre en valeur ces deux parties.

Toutefois, les techniques à base de déformation sont aujourd'hui nombreuses et nous pensons que la propriété "déformée" nécessite d'être raffinée. Nous proposons donc trois axes supplémentaires de classification pour les effets de déformation que nous illustrons à la Figure 35.

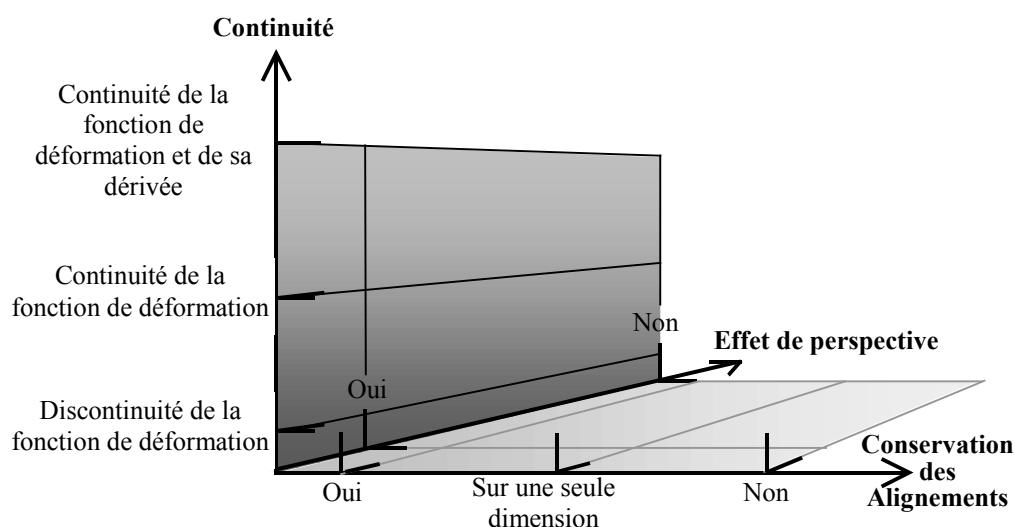


Figure 35 : Taxonomie en trois axes des effets de déformation.

Le premier axe que nous proposons prend en compte la continuité au sens mathématique de la fonction de déformation. Un effet de déformation est décrit par une fonction de transformation des points de l'espace. Cette fonction transforme des points non déformés (espace) en points de l'espace déformé (point de vue sur l'espace). Lorsque cette fonction est continue elle garantit la conservation du voisinage entre les éléments de l'espace mais également une continuité de la taille des éléments. La discontinuité de cette fonction est évoquée par Manojit Sarkar et ses coauteurs [Sarkar 93] pour établir les inconvénients du "stretching orthogonal".

Lorsque la fonction est continue, un élément fortement grossi (focus) ne se retrouvera pas immédiatement à côté d'un élément de petite taille (contexte). La continuité de la fonction de déformation et de sa dérivée implique en plus de cela que l'utilisateur ne perçoive pas plusieurs zones dans l'espace. En effet une fonction dont la dérivée est discontinue peut être décrite par l'enchaînement de plusieurs fonctions. Le passage d'une fonction à une autre sera perçu comme un changement de zone par l'utilisateur.

Les déformations proposées dans les travaux de Carpendale [Carpendale 97], Hovestadt [Hovestadt 95] et Keahey [Keahey 96] proposent toutes une partie avant aplani qui correspond à une continuité de la fonction de déformation. Le lecteur trouvera à la Figure 40 (page 123) une reproduction de la déformation de Keahey.

Les mêmes auteurs proposent des formes non aplanies qui correspondent à une fonction de déformation dont la dérivée est en plus continue. Enfin la fonction de déformation du "Bifocal

Display" [Apperley 82] est discontinue. La Figure 36 illustre ce passage brusque du focus au contexte.

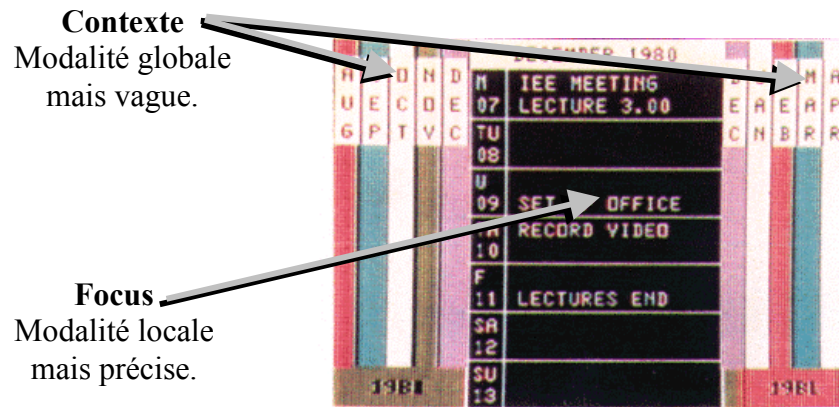


Figure 36 : Système du Bifocal Display [Apperley 82].

Le second axe différencie les effets de déformation qui conservent ou non l'alignement des données. Les vues hyperboliques comme la vue en sphère de [Carpendale 97], [Hovestadt 95] et [Keahey 96] ne conservent pas l'alignement des données. Le système du "Bifocal Display" et sa généralisation en deux dimensions identifiée par Leung conservent les alignements de données. Enfin le mur en perspective de Jock Mackinlay [Mackinlay 91], illustré à la Figure 37, conserve l'alignement des colonnes mais pas celui des lignes.

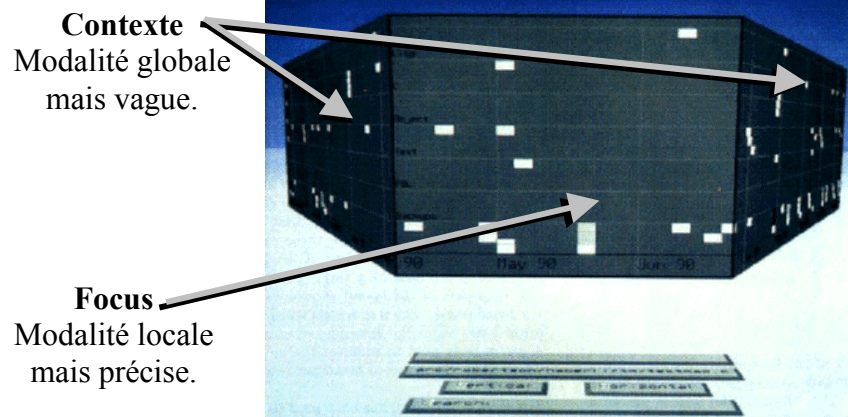


Figure 37 : Système du mur en perspective ("Perspective Wall") [Mackinlay 91].

Enfin le troisième axe décrit l'effet de perspective qui peut résulter de la fonction de déformation. Cette propriété de trompe l'œil peut paraître anecdotique d'un point de vue mathématique mais elle aide beaucoup à la compréhension de l'utilisateur. Elle est par exemple

présente dans la du mur en perspective (Figure 37) mais pas dans le système du "Bifocal Display" (Figure 36).

3.3.2 Caractérisation de la navigation

Comme nous l'avons annoncé, après avoir caractérisé l'aspect purement visuel au paragraphe précédent nous proposons dans ce paragraphe de caractériser les outils de navigation. Ces outils sont indispensables dans les techniques de visualisation qui utilisent un focus et un contexte car elles utilisent l'interaction pour changer l'information présentée dans le focus.

Un outil de navigation est un couple de modalités (une en sortie pour rendre l'espace et une en entrée pour manipuler l'un des paramètres de la visualisation). Nous pensons que les outils qui doivent être distingués lors de la conception sont :

- les outils de navigation grossiers et les outils de navigation au pas à pas,
- les outils fortement couplés et les outils au couplage relâché,
- les outils à manipulation directe et les outils à manipulation indirecte.

La navigation pas à pas permet d'atteindre n'importe quel état de la visualisation car le pas de navigation est le plus fin possible pour l'espace d'information. Au contraire nous appelons "grossier" un outil de navigation qui ne permet pas d'atteindre tous les états possibles de la visualisation. Dans un éditeur de documents textuels comme Word [Word], illustré à la Figure 38, les flèches de la glissière représentent un outil de navigation pas à pas alors que les flèches "page suivante" et "page précédente" ne permettent qu'une navigation page par page. Ce dernier outils ne permet pas de visualiser tout le document (navigation grossière). Cet outil deviendra pas à pas quand les écrans permettront d'afficher entièrement un document de 30 centimètres de haut (c'est-à-dire une page A4 complète).

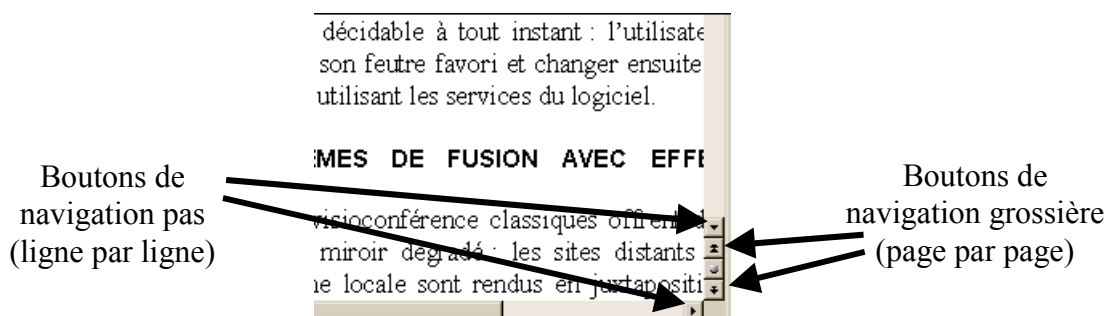


Figure 38 : Outils de navigation grossière et de navigation pas-à-pas dans le logiciel Word [Word].

En nous basant sur les travaux de François Bérard [Bérard 99], nous définissons ensuite un outil de navigation fortement couplé lorsque tous les événements en entrée sont répercutés sur la visualisation et que l'utilisateur utilise ces modifications dans la visualisation pour produire les événements en entrée.

"Une interaction est fortement couplée sur un intervalle de temps donné lorsque les systèmes humain et artificiel sont engagés de manière continue dans l'accomplissement d'actions physiques mutuellement perceptibles et dépendantes sur cet intervalle."
Extrait de [Bérard 99].

Ce type de navigation génère de très nombreuses modifications dans la visualisation. Un outil de navigation fortement couplé signifie que le concepteur doit mettre en œuvre une visualisation très rapidement calculée. Il n'est pas nécessaire de chercher un couplage plus rapide que 25 actions par seconde à cause des capacités sensorielles humaines. Le calcul et le dessin 25 fois par seconde d'un grand espace d'information est un problème compliqué qui explique souvent que ce type de navigation n'est pas utilisé. Nous relevons par exemple le système " FilmFinder " [Ahlberg 94], présenté à la Figure 39, qui met en œuvre un couplage fort entre les glissières et la visualisation d'un champ de points de couleur (dans la figure, le champs de point est le contexte et l'inspecteur d'objet est le focus).

Au contraire, dans une navigation relâchée, le concepteur considère un temps d'attente entre la modalité d'entrée et sa répercussion sur la visualisation. Un filtre sur les noms de fichier d'un disque dur (*.gif par exemple) est un exemple d'outil de navigation relâché dans lequel le temps d'attente peut être très long.

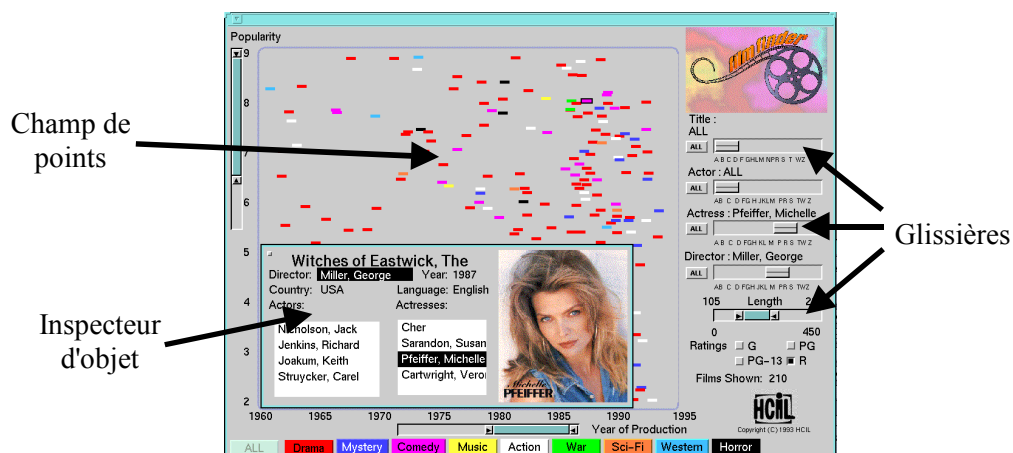


Figure 39 : Le système FilmFinder [Ahlberg 94] avec un couplage fort entre les glissières et le champs de points.

Enfin nous considérons les outils à manipulation directe et les outils à manipulation indirecte. Dans le premier cas, l'outil est transparent car il correspond à la manipulation d'objets de la visualisation. L'utilisateur manipule en entrée l'espace de la visualisation qu'il perçoit en sortie. Un tel outil de navigation nécessite une réflexion du concepteur afin de choisir une métaphore adaptée lors de la création de l'espace. Par exemple de petits éléments de couleur rouge peuvent très bien se voir (niveau d'abstraction de la visualisation) mais peuvent poser des difficultés à être sélectionnés et manipulés (niveau d'abstraction de la navigation).

Nous notons que le concept de manipulation directe, proposé par Ben Shneiderman dans [Shneiderman 83], n'est pas spécifique à la navigation mais s'adapte à tous types de tâches.

Le système HiNote [HiNote], basé sur la boîte à outil JAZZ, propose un exemple et un contre-exemple de navigation par manipulation directe. Dans ce système l'utilisateur peut déplacer les éléments graphiques de l'espace en les sélectionnant puis en opérant une action de glisser-déposer (drag & drop en anglais). Il s'agit d'un cas classique de manipulation directe, mais qui dans ce cas est un outil de navigation.

Dans ce système, la navigation d'un niveau de zoom à un autre ne correspond pas à un outil à manipulation directe. Par opposition à la première classe d'outil, nous qualifions ces outils de manipulation indirecte. Pour situer complètement cet outil dans notre espace nous notons qu'il s'agit d'un outil de navigation fortement couplé utilisant le glisser-déposer de la souris comme modalité en entrée.

4 *Patrons de conception pour les techniques de visualisation*

Comme dans le chapitre précédent, nous validons cette étude par un ensemble de patrons de conception. La visualisation de grandes quantités d'informations nous permet d'être plus précis vis-à-vis de la conception des modalités en sortie car nous avons pu affiner notre espace de conception. Les cinq patrons que nous présentons dans cette section sont les suivants :

- ▶ Patron de la continuité spatiale des modalités de visualisation
- ▶ Patron de la continuité temporelle des modalités de visualisation
- ▶ Patron du plaquage des données en traits graphiques
- ▶ Patron de la déformation de l'espace d'information
- ▶ Patron de réutilisation des outils de navigation

4.1 Patron N°5 : Continuité spatiale des modalités

Ce patron de visualisation présente l'aspect de continuité très recherché dans les modalités de type "Focus+Context". Ce patron explique la conception d'un focus intégré dans le contexte qui respecte la continuité spatiale. Il fait intervenir les critères d'ergonomie de l'observabilité et de l'honnêteté (équipe IIHM) et du groupement par la localisation (Scapin).

4.1.1 Exemples illustratifs

- Le système du mur en perspective [Mackinlay 91], que nous avons illustré à la Figure 37 (page 119), offre selon ses concepteurs, un focus et un contexte "intégrés de façon continue"¹
- Le système du "BiFocal Display" [Apperley 82] que nous avons également déjà illustré à la Figure 36 (page 119)
- La déformation des cartes géographiques proposé par Carpendale [Carpendale 97] que nous illustrerons à la Figure 44 pour un autre patron
- Le plan du métro de Washington déformé par l'une des techniques de [Keahey 96] et reproduit à la Figure 40

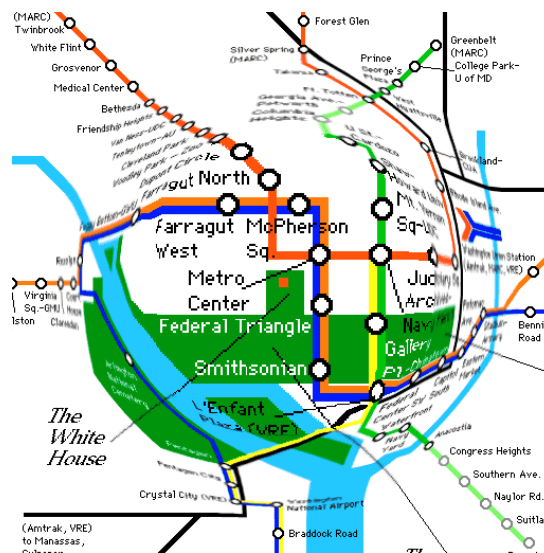


Figure 40 : Plan du métro de Washington, déformé autour de la maison blanche [Keahey 96].

4.1.2 Description du problème et du contexte

Le problème de la discontinuité spatiale est intimement lié à la capacité de l'utilisateur de n'avoir, à un moment donné et pour un sens perceptif donné, un seul focus d'attention. Pour

¹ "Detail and context smoothly integrated" dans le texte original en anglais

réaliser sa tâche l'utilisateur porte son attention sur une partie de l'interface que l'on a appelé le focus elle présente toutes les informations nécessaires (précision). Pour changer de focus, l'utilisateur opère une tâche articulatoire de navigation qui nécessite de situer le focus dans le contexte. Pour mener à bien cette tâche, l'utilisateur porte son attention sur le contexte dans lequel il peut percevoir la structure de l'espace d'information. L'utilisateur porte à nouveau son attention sur le focus quand la tâche de navigation est terminée. Le changement d'endroit où l'utilisateur porte son attention devient très fréquent quand la quantité d'information grandit.

Nous supposons également que les deux modalités n'utilisent pas le même langage d'interaction puisque l'une est vague et l'autre précise. Nous pouvons facilement anticiper un problème lorsque la visualisation présente un focus et un contexte dont la composition est la suivante :

- ▶ différente pour la syntaxe,
- ▶ complémentaire pour la sémantique,
- ▶ disjointes dans l'espace.

Nous voyons que l'utilisateur est obligé de changer de langage d'interaction en plus de changer l'emplacement de son focus d'attention. A cause de l'aspect complémentaire des informations, ces changements sont inévitables. Comme ce changement ne peut pas être instantané, l'utilisateur doit mémoriser l'information dont il a besoin.

4.1.3 Motivations

- ▶ Afficher un focus précis et un contexte global
- ▶ Eviter à l'utilisateur de perdre le fil de sa pensée quand il regarde successivement le focus et le contexte.
- ▶ Eviter les allers-retours entre focus et contexte pour les tâches de comparaison.
- ▶ Ne pas déformer le focus afin de ne pas ralentir les tâches habituelles qui s'effectuent dans celui-ci (Edition par exemple)

4.1.4 Contre-exemple

La technique de vue radar, précédemment illustrée à la Figure 33 (page 116), ne propose aucune continuité spatiale entre le focus et le contexte. Le seul lien entre ces deux modalités est un rectangle dans le contexte qui délimite les bords du focus. Si l'utilisateur cherche à identifier une rivière qui traverse le focus, il doit se reporter à la vue radar et chercher une rivière qui traverse le rectangle. Si un élément au bord de cette rivière attire son attention

dans la vue globale, il doit revenir au focus et chercher à nouveau cet élément pour le manipuler.

Le système SAND [Sand], reproduit à la Figure 41, repose sur une interaction similaire mais utilise la transparence pour rapprocher le focus de son contexte. Cette technique utilise moins de place à l'écran que la précédente mais présente les mêmes défauts.

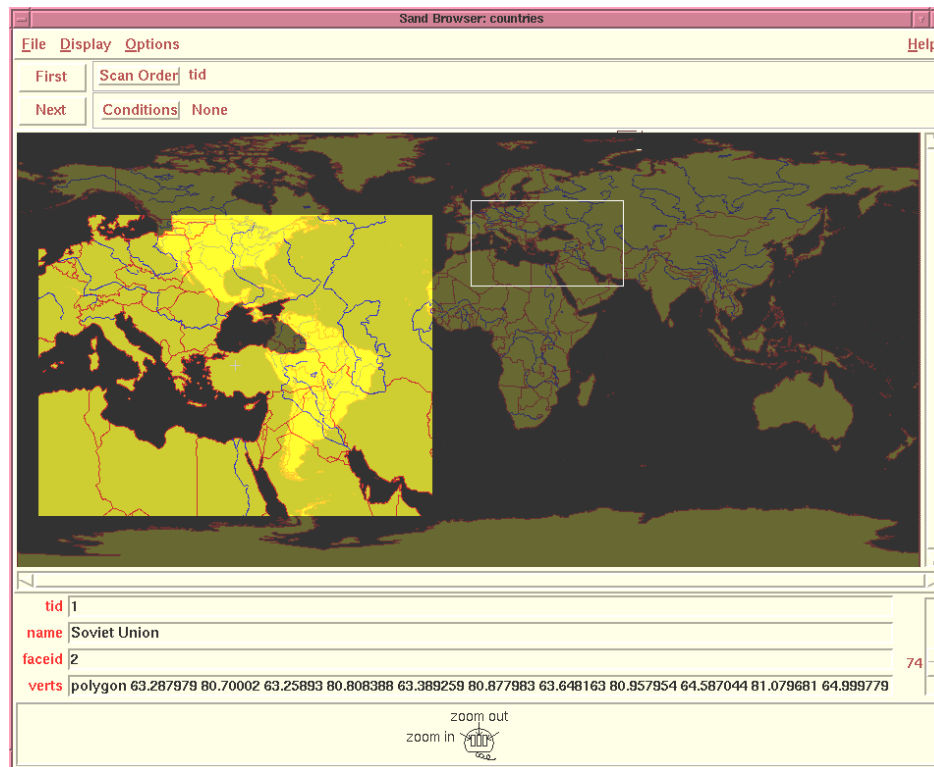


Figure 41 : Système Sand [Sand].

4.1.5 Solution proposée

La solution que nous proposons consiste à intégrer spatialement le focus dans son contexte lors de la conception de l'espace. Si le concepteur considère la modalité supportant le focus et la modalité supportant le contexte, la continuité spatiale entre ces modalités permettra à l'utilisateur de ne percevoir qu'une seule modalité. Pour le concepteur, cette modalité sera composée et la composition spatiale sera imbriquée.

Cette continuité nécessite d'être réajustée lorsque l'utilisateur déplace son focus d'attention dans l'espace. Dans l'interface, soit le focus, soit le contexte est déplacé. Si le concepteur décide de déplacer le contexte, il choisira généralement une position centrale pour le focus. Si le focus se déplace par rapport à un contexte stable, le concepteur doit prévoir les

effets de bord impliqués par le focus qui atteint les bords du contexte (éviter les occlusions et le gaspillage de place à l'écran).

Cette solution doit s'accompagner d'un outil adéquat de navigation permettant de déplacer le focus. Nous conseillons un outil de navigation par manipulation directe car la continuité spatiale permet à l'utilisateur de considérer un espace unique dans lequel les actions de glisser-déposer sont naturelles et rapides.

Afin de minimiser au plus la discontinuité le concepteur va tenter de faire correspondre tous les objets du focus à leurs voisins du contexte. Lorsqu'un compromis doit être fait, le concepteur doit favoriser les points saillants de l'espace relatifs à la structure de donnée. Ces points sont utilisés comme points de repère pour la navigation. Ce patron nécessite donc parfois d'identifier quels sont ces points saillants. A l'inverse, connaissant les points continus entre les modalités, le concepteur choisira parmi eux les points à mettre en saillance, lorsque ce choix est possible.

4.2 *Patron N°6 : Continuité temporelle*

Le second patron que nous proposons dans ce chapitre concerne la continuité temporelle. Il s'agit d'un patron de navigation qui s'applique aux outils de navigation grossiers et aux changements de vues. Les techniques de vues multiples concernées par ce patron sont :

- ▶ les vues multiples du même point de vue sur les données,
- ▶ les vues multiples du même espace selon des points de vue différents,

Les critères d'ergonomie concernés sont la tolérance au rythme, l'observabilité et l'honnêteté (équipe IIHM). Ce patron met en jeu une ou plusieurs modalités basées sur un dispositif graphique et une modalité sonore.

4.2.1 *Exemples illustratifs*

- ▶ Les techniques "Angular Detail Method", "Detail Outside Method" et "Detail Inside Method" proposée par John Stasko et Eugene Zhang [Stasko 00] jouent une animation chaque fois que l'utilisateur change de focus.
- ▶ Le système d'exploitation MasOs [MacOS] propose également une technique intéressante de continuité temporelle. Une animation s'opère entre le moment où une fenêtre est icônifiée en bas de l'écran et l'instant suivant lorsqu'elle a été rappelée par l'utilisateur. Selon les versions de ce système d'exploitation, plusieurs animations ont été utilisées : une

succession de rectangles blancs ou un effet de déformation appelé "génie" illustré à la Figure 42.

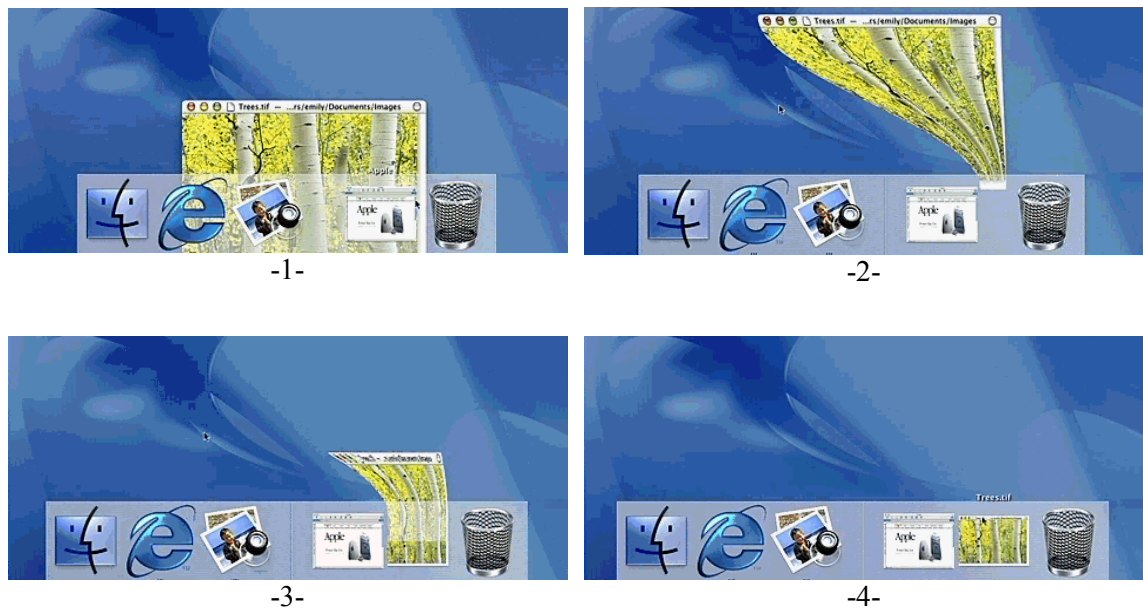


Figure 42 : Illustration de l'effet d'animation appelé "génie" dans MacOS [MacOS] lorsqu'une fenêtre est minimisée dans la barre des applications, en bas de l'écran.

- Le logiciel Internet Explorer livré avec Windows [Windows] utilise une animation qui fait défiler doucement un document Web d'une page à l'autre. Cette animation ralentit la navigation grossière page par page mais permet de mieux se rendre compte du déplacement dans le document.

4.2.2 Description du problème et du contexte

Le problème de la présentation séquentielle de modalités est de faire le lien entre le focus d'attention dans la première modalité et le focus dans la seconde modalité. Le changement de vue peut être dérangeant si l'utilisateur perd son focus pendant le basculement d'une modalité à l'autre. De même si la modalité représentant le focus ne change pas, mais que son contenu change cela signifie que le focus a été déplacé. Même si ce déplacement obéit à une requête de l'utilisateur, nous devons envisager l'hypothèse qu'il ne sache pas à l'avance le résultat de son action de navigation.

4.2.3 Motivations

- Evaluer les changements entre les vues.
- Garder la stabilité visuelle de l'écran.
- Evaluer le pas des outils de navigation grossiers.

- Permettre à l'utilisateur de suivre son focus d'attention.

4.2.4 *Contre-exemple*

Les contre-exemples à ce patron sont très nombreux car la programmation d'une animation est souvent très coûteuse en temps de développement pour un résultat parfois lent (information à afficher plus de vingt cinq fois par seconde pour un résultat fluide). Les systèmes graphiques de gestion de fichier de MacOS [MacOS] et Windows [Windows] proposent par exemple plusieurs vues du contenu d'un répertoire (voir Figure 6, page 41 pour une illustration de Windows). Le passage d'une modalité à l'autre est brutal et sans animation. Si l'utilisateur regarde un fichier et change de vue, il doit le retrouver dans la nouvelle vue qui a remplacé subitement la précédente.

Nous relevons également le système Word [Word] qui ne fait pas défiler les pages d'un document comme le système Internet Explorer (3^{ème} exemple).

4.2.5 *Solution proposée*

La solution est de créer, lorsque c'est possible, un ensemble de modalités intermédiaires qui seront perçues comme une animation par l'utilisateur. Quel que soit le focus d'attention dans la première modalité présentée, l'utilisateur pourra le suivre dans l'animation vers sa position dans la seconde modalité.

Le concepteur peut mettre en œuvre un mécanisme de fondu permettant de voir pendant quelques secondes les deux modalités en transparence l'une par dessus l'autre. Les éventuelles parties communes entre deux modalités ne seront pas modifiées par cette technique et seront mises en valeur.

Une solution plus appropriée consiste à mettre en œuvre une animation de l'emplacement des éléments et des différents traits graphiques. Une telle animation consiste à calculer les emplacements intermédiaires entre la position d'un élément dans la modalité de départ et dans la modalité d'arrivée. Un dégradé peut être calculé entre deux couleurs, des tailles intermédiaires peuvent être calculées entre deux tailles, etc.

Comme l'interaction avec une modalité dynamique (problème de la cible mouvante), la solution consiste à rendre passives les modalités pendant la phase d'animation. Afin de rendre perceptible ce changement d'état nous conseillons d'accompagner chaque animation par un son lorsque c'est possible. Pour ne pas gêner l'utilisateur, ce son peut rester faible et sourd à l'opposé des sons forts et stridents utilisés pour attirer l'attention. Nous considérons que chaque

animation doit avoir sa propre "musique" et faire ainsi participer le sens de l'ouïe à la connaissance de l'état interne du système.

La mise en œuvre de ce patron dans les outils de navigation grossiers permet de visualiser les éléments de l'espace entre le point de départ et le point d'arrivée. Si la technique de visualisation a pour ambition de supporter des tâches vagues comme l'exploration, un survol de l'espace permet de découvrir des endroits intéressants de l'espace. La technique d'animation est la même que lors du passage d'une modalité à l'autre sauf que nous ne considérons qu'une seule modalité.

Quel que soit le cas, l'animation doit respecter le critère d'ergonomie de la tolérance au rythme. Pour cela le concepteur prévoit le nombre d'étapes de l'animation en fonction de la quantité d'information mise en jeu. Un déplacement par un outil de navigation grossier doit prendre en compte la distance parcourue. L'animation d'une modalité à l'autre doit prendre en compte les différences entre les syntaxes des modalités.

4.3 Patron N°7 : *Plaquage des données en traits graphiques*

Ce troisième patron motive la préférence en visualisation de certains traits graphiques (l'inclusion spatiale, le lien, la forme, la taille, la couleur et la texture) et du placement dans l'espace plutôt que le trait graphique du label. Ce patron traite à la fois des critères d'ergonomie d'adaptativité et d'adaptabilité (équipe IIHM) et de Groupement / Distinction entre items (Scapin). Ce patron ne met en jeu qu'une seule modalité.

4.3.1 Exemples illustratifs

- ▶ Le logiciel FilmFinder présenté à la Figure 26, page 97, utilise la couleur pour caractériser les différents types de film et la position dans l'espace pour la popularité et l'année de sortie du film. Ce choix paraît dicté par le fait que la couleur comme le type de film ne sont pas ordonnés. Par contre la popularité et l'année sont ordonnées et les dimensions de l'espace 2D sont quantitatives, donc également ordonnées.
- ▶ Les TreeMaps [Shneiderman 91], illustrés à la Figure 95, page 215, utilisent la taille des rectangles pour transcrire la taille des fichiers (donnée ordonnée) et la couleur pour transcrire le type de fichier (donnée nominale : fichiers images, fichiers de programmation, etc.).
- ▶ Dans le système TableLens [Rao 94], illustré à la Figure 43, plusieurs représentations sont utilisées selon le type des données. Un segment d'histogramme de taille proportionnelle transcrit les données sur lesquelles le concepteur a choisi un point de vue quantitatif,

comme un nombre d'unité ("Units" en anglais dans la figure). Un code de couleur est utilisé pour représenter des données sur lesquelles le concepteur a choisi un point de vue ordonné ou nominal, comme le trimestre ("Quarter" ou "Channel" en anglais dans la figure).

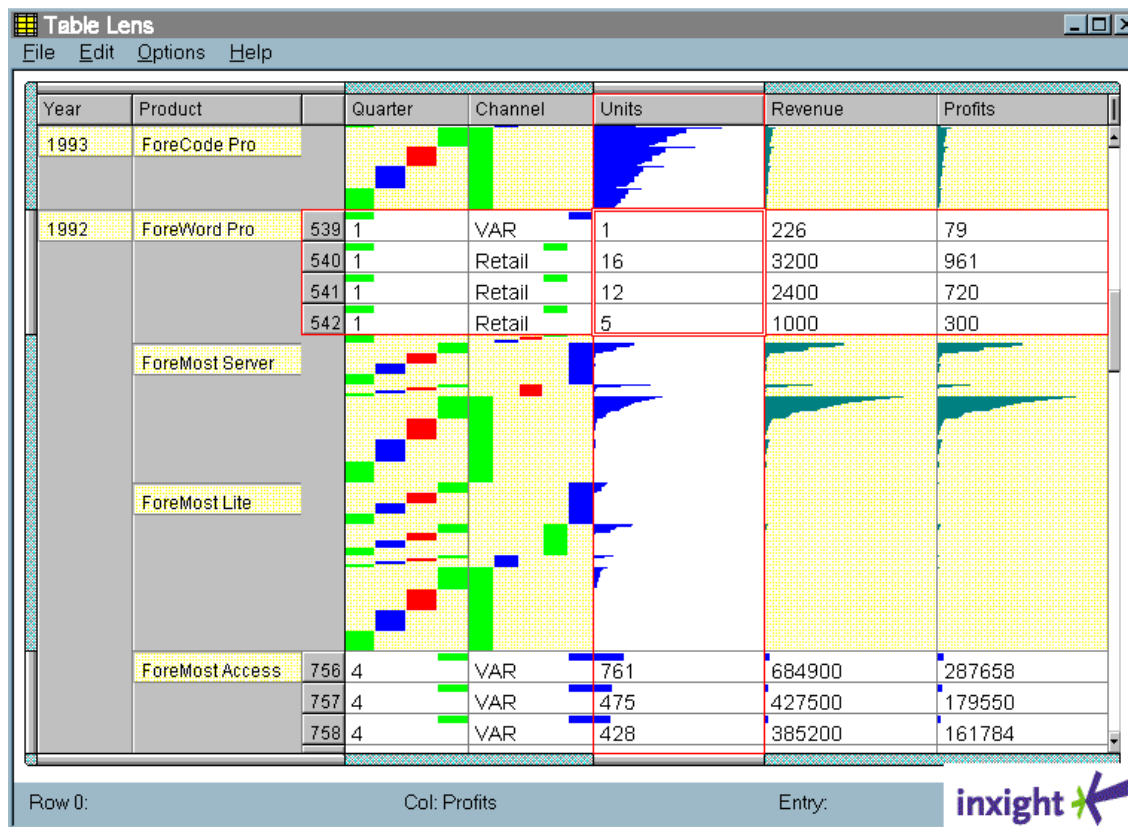


Figure 43 : Système des TableLens [Rao 94].

4.3.2 Description du problème et du contexte

Nous avons fait l'hypothèse que les données à représenter pouvaient se décomposer en éléments et que chaque élément pouvait avoir différentes facettes. Chacune de ces facettes doit être représentée par un trait graphique pour que l'utilisateur perçoive les données. Une méthode systématique consiste à utiliser un label pour chaque type de donnée. La langue naturelle véhicule un pouvoir d'expression très vaste qui permet de décrire tous type de donnée. Nous notons trois principaux défauts à cette approche. Tout d'abord elle est très coûteuse en espace à l'écran. Elle nécessite ensuite une charge cognitive importante pour la lecture. Un mot est plus difficile à distinguer d'un autre qu'une forme ou une couleur.

L'utilisation pour toutes les données de labels rend enfin difficile la recherche visuelle de tendances ou de motifs.

4.3.3 *Motivations*

- ▶ Concevoir une technique de visualisation concise qui permette d'afficher plus d'informations sur une même surface
- ▶ Permettre à l'utilisateur de séparer rapidement les facettes d'un élément afin de concentrer son attention sur les informations nécessaires à sa tâche (et éviter les confusions)
- ▶ Faire apparaître des tendances et des motifs permettant une meilleure compréhension d'un grand espace d'information
- ▶ Croiser les facettes d'un élément pour répondre à des questions utilisant les opérateurs logiques sur les facettes (Qu'est ce qui est vert et petit pour les traits de la couleur et de la taille par exemple)

4.3.4 *Contre-exemple*

Les systèmes à base d'affichage entièrement textuel sont un héritage que nous jugeons mal adapté mais encore très répandu. Le système d'exploitation DOS est encore inclus dans Windows [Windows] et permet d'organiser le disque dur avec une interface composée presque uniquement de texte (certaines versions de DOS utilisent également la couleur comme trait graphique). Nous relevons également que de nombreuses bases de données ne proposent pas de techniques de visualisation pour afficher leur contenu.

4.3.5 *Solution proposée*

La solution consiste à faire correspondre le plus possible les types de données à rendre perceptible avec les traits graphiques proposés au paragraphe 3.2.

La première donnée à considérer est la structure de l'espace. Cette donnée est souvent traduite par la position dans l'espace.

Nous proposons de considérer ensuite les traits graphiques qui permettent de traduire une facette des données dans une modalité analogique (la forme, la texture et la couleur). Si les éléments de l'espace d'information sont des fichiers, la forme d'une icône permet par exemple de traduire le type de fichier (une forme de bobine pour un film, une feuille cornée pour un document, un cadre de tableau pour une image, etc).

Parmi les traits graphiques restant, nous préconisons la prise en compte des traits qui attirent le plus l'attention de l'utilisateur (la couleur, la position dans l'espace, la taille puis la forme) pour les facettes des données les plus discriminantes. Ces traits graphiques formeront en effet les repères visuels auxquels l'utilisateur se raccrochera pour naviguer.

Si certains traits graphiques restent inutilisés plusieurs solutions sont envisageables. Le système peut présenter automatiquement (adaptativité), de façon redondante, l'une des facettes qui n'a pu être traduite que par un label (par exemple la plus discriminante). La seconde solution (adaptabilité) consiste à laisser l'utilisateur choisir l'utilisation du trait graphique. Si la couleur est inutilisée, le système peut proposer à l'utilisateur de colorier les éléments de l'espace en utilisant une palette de couleur.

4.4 Patron N°8 : Compréhension de la déformation

Ce quatrième patron sur la visualisation a trait à l'utilisation d'une déformation non linéaire de l'espace des données. Nous appelons ces vues hyperboliques ou en œil de poisson selon les termes consacrés dans le domaine. Ces modalités graphiques qui mettent en œuvre une déformation doivent faire l'objet d'une attention particulière lors de la conception pour satisfaire le critère d'ergonomie de l'honnêteté (équipe IIHM) et de protection contre les erreurs (Scapin). Ce patron, comme le patron N°5, distingue la modalité graphique du focus et la modalité du contexte. Elle met donc en jeu plusieurs modalités.

4.4.1 Exemples illustratifs

- Les travaux sur la déformation hyperbolique d'une interface [Hovestadt 95] présentent un effet de perspective car les icônes et les images considérées dans l'interface sont rectangulaires. Au centre des déformations proposées, nous remarquons également une partie tronquée. Cet artifice permet de présenter un focus non déformé qui reste continu avec le contexte.
- Les travaux de Sheelagh Carpendale [Carpendale 97] sur la déformation d'une carte géographique incluent l'affichage d'une grille et des effets d'ombre associés à la déformation. Ces deux artifices permettent de rendre compréhensible la déformation. Dans une autre partie de ces travaux, non illustrés à la Figure 44, cet auteur propose aussi des déformations tronquées au centre.

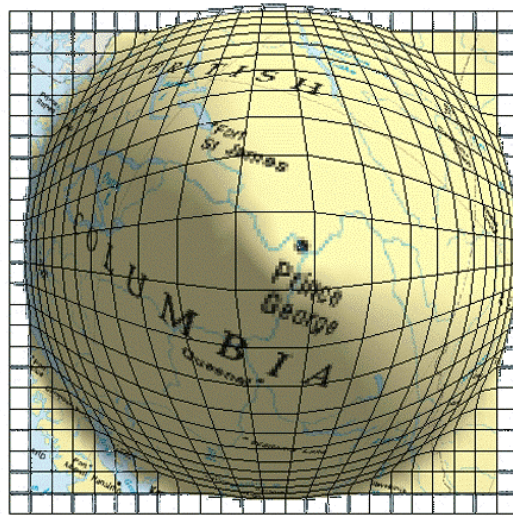


Figure 44 : Déformation d'une carte géographique [Carpendale 97] avec affichage d'une grille et d'un effet d'ombre.

- La carte de Washington, présentée à la Figure 40, page 123 déforme les pastilles (arrêt de métro) et les labels (nom des arrêts de métro et des lieux) avec un effet de perspective. La partie tronquée au centre permet de lire les labels des arrêts de métro.

4.4.2 Description du problème et du contexte

Si la déformation de l'espace permet d'améliorer l'observabilité du focus dans le contexte (Patron N°5), la perception de la déformation pose un problème. L'utilisateur doit comprendre l'espace à deux dimensions qui est déformé pour utiliser les informations qui y sont affichées. La déformation peut ralentir la lecture des informations ou entraîner des erreurs d'interprétation. L'utilisation du trait graphique de la taille doit être traitée avec beaucoup de précaution. Déformer le texte pour lui donner un effet de perspective peut rendre le texte illisible ou difficile à lire. Ne pas le déformer peut laisser croire à l'utilisateur qu'il n'y a pas de déformation.

4.4.3 Motivations

- Respecter le patron N°5 sur la continuité spatiale en utilisant une vue hyperbolique.
- Eviter les erreurs d'interprétation et notamment éviter qu'un utilisateur puisse ne pas s'apercevoir de la présence d'une fonction de déformation.
- Eviter de rendre un texte important illisible en le déformant.
- Donner à l'utilisateur des repères visuels pour l'aider dans la navigation (notamment des repères relatifs à la déformation).

4.4.4 Contre-exemple

La déformation de la carte des états unis [Sarkar 93] par une vue hyperbolique, illustrée à la Figure 45, nous laisse penser qu'un utilisateur non américain peut ne pas reconnaître le pays et les villes. La déformation s'applique sur les frontières de la carte (la péninsule de Floride par exemple). Un utilisateur non expert de la géographie américaine peut perdre des repères fondamentaux pour la navigation.

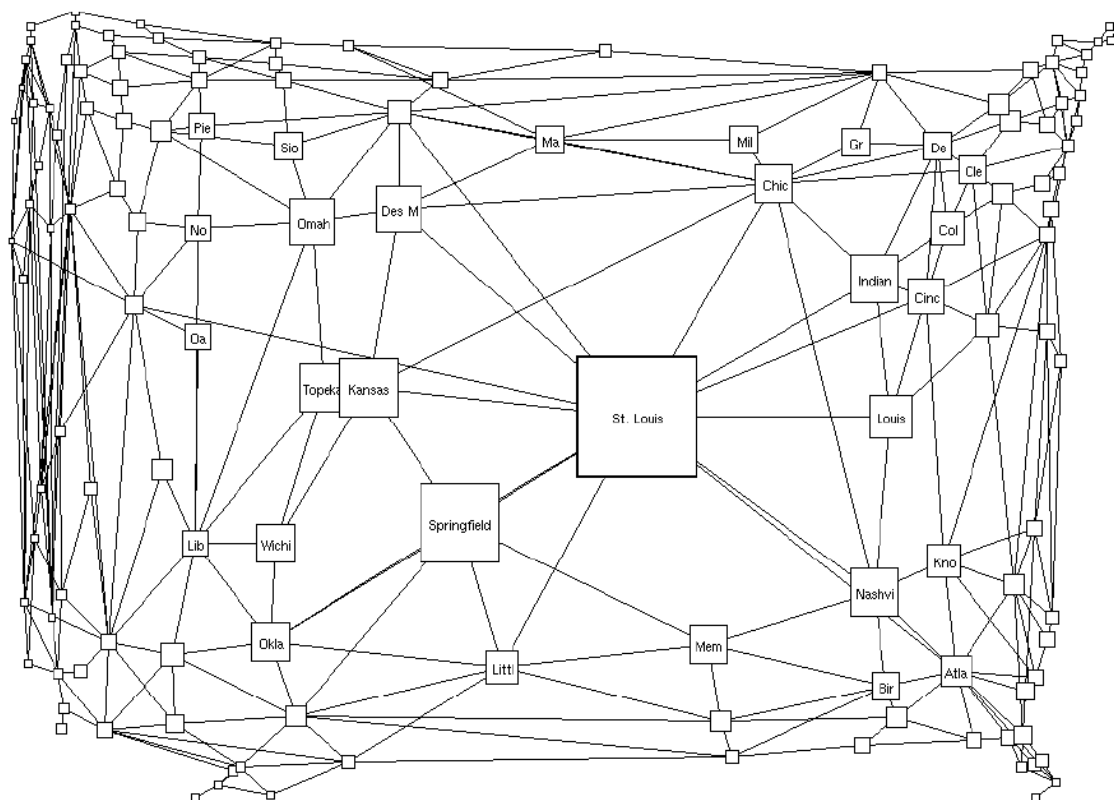


Figure 45 : Carte des USA déformée par une vue en œil de poisson (ou hyperbolique) [Sarkar 92][Sarkar 93].

4.4.5 Solution proposée

Lorsque le type de donnée est spatial à deux dimensions ou lorsque la position dans l'espace est utilisée pour rendre perceptible une donnée pertinente, l'utilisation d'une déformation avec un effet de perspective est préférable. Puisque nous considérons que l'espace est important pour la tâche de l'utilisateur il faut que la technique de visualisation soit honnête et permette une interprétation correcte des attributs spatiaux. Déformer l'espace sans donner d'effet de perspective peut entraîner des erreurs d'évaluation de distances ou de positions à l'intérieur de l'espace. L'effet de perspective ne permet pas de corriger les erreurs mais de les éviter en mettant à contribution les mécanismes cognitifs de perception de l'espace en trois dimensions.

La seconde partie de la solution consiste à éviter la déformation du focus. En effet le focus est sans doute la partie de l'interface la plus utilisée. La présence d'une zone non déformée et précise permet à l'utilisateur d'effectuer ses tâches locales plus confortablement. Le texte éventuellement déformé dans le contexte ne le sera pas dans le focus. La limite entre focus et contexte sera d'autant plus nette que le focus sera non déformé et le contexte sera déformé.

4.5 Patron N°9 : Réutilisation des outils de navigation

Ce neuvième patron est un patron de navigation qui explique la réutilisation des outils de navigation. Cette réutilisation n'est pas seulement un gain de temps pour le concepteur mais aussi une nécessité pour l'utilisateur de nombreux systèmes de visualisation de grande quantités d'information. La factorisation des outils de navigation lui permet d'apprendre plus vite à naviguer dans les espaces d'information. Ce patron met en jeu plusieurs modalités de sorties pour répondre aux critères d'homogénéité / cohérence (Scapin) et de prise en compte de l'expérience de l'utilisateur (Scapin). Il n'y a pas vraiment de critères de l'équipe IIHM qui correspondent à ce patron car ces ils prennent en compte l'utilisabilité des systèmes et non l'apprentissage. La réutilisation des entrées et des sorties est le seul critère de l'équipe IIHM qui soit pertinent mais il ne couvre pas l'essentiel du patron.

4.5.1 Exemples illustratifs

- L'outil de recherche par mot clé est présent dans de très nombreux systèmes qui utilisent des modalités de visualisation différentes. Les outils activés par la même commande "Ctrl+F" dans les systèmes Internet Explorer [Windows], Codewarrior [Codewarrior] et Netscape [Netscape] sont illustrés à la Figure 46. Ces trois interfaces de recherche par mot clé sont très ressemblantes (champs de saisie, sens de la recherche, mot entier et respect de la casse).

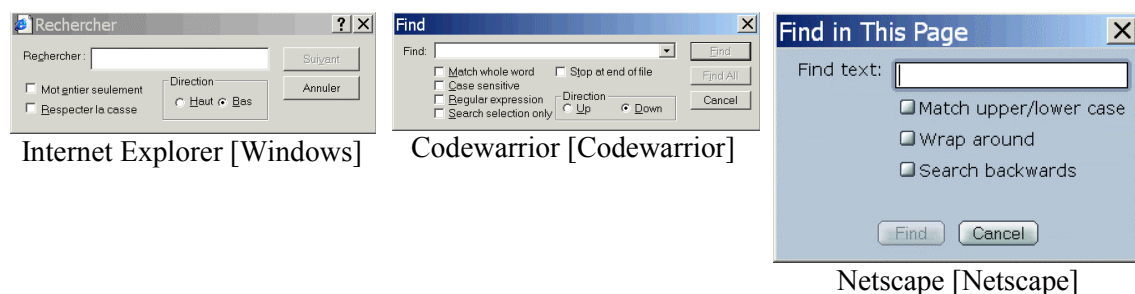


Figure 46 : Trois outils de navigation très proches dans trois systèmes différents.

- Le déplacement par les touches "flèches du clavier" et les touches " home ", " end ", " page up " et " page down " est également un outil de navigation commun à de nombreux systèmes. La navigation dans des documents avec cet outil est devenu presque universelle.
- Les barres de défilements (Scrollbar) sont également utilisées très largement pour naviguer dans de grands espaces 2D (une sur la droite, verticale et une en bas, horizontale).

4.5.2 Description du problème et du contexte

La navigation dans l'espace d'information est un problème récurrent qui a engendré le développement de multiples outils de navigation. Parce que le concepteur juge souvent que l'outil de navigation doit être adapté à son espace d'information, il conçoit un nouvel outil sans se préoccuper du comportement des outils qui existent déjà. Les différents systèmes d'une station de travail peuvent offrir des comportements de navigation incohérents. L'utilisateur a tendance à considérer la navigation comme un domaine de compétence indépendant des espaces d'information et cherche à réutiliser ses connaissances d'un système à un autre.

4.5.3 Motivations

- Faciliter l'apprentissage des outils de navigation en réduisant le nombre des interfaces qui proposent les mêmes fonctionnalités de navigation
- Permettre une meilleure interopérabilité des outils de navigation en améliorant leur homogénéité (Par exemple permettre un copier coller de requêtes entre deux interfaces de recherche par mot clef)
- Permettre à l'utilisateur de reproduire la même action de navigation dans plusieurs vues afin de chercher des différences entre les espaces d'information

4.5.4 Contre-exemple

La première version de la souris à molette générait des événements spécifiques au système Word [Word]. Les possesseurs de souris à molette ne pouvaient pas réutiliser leur molette dans d'autres systèmes. Nous notons avec satisfaction que le patron a fini par être appliqué suite à la pression exercée par les utilisateurs et des systèmes concurrents.

L'outil de recherche de Word, illustré à la , est également très différent des autres et nécessite une étape supplémentaire (Bouton "Moins" et "Plus" pour agrandir la fenêtre) pour paramétrer les options de la recherche, évoquées dans premier exemple de ce patron. De plus l'interacteur pour choisir le sens de la recherche (vers le haut ou vers le bas du document) n'est pas une case à cocher comme dans les autres outils mais un menu déroulant.

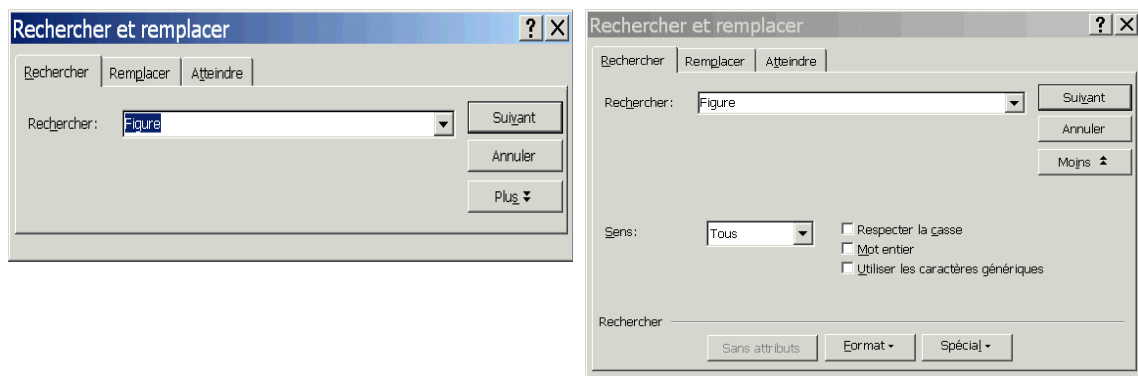


Figure 47 : Outil de recherche par mot clé dans le système Word [Word].

4.5.5 *Solution proposée*

La solution qui est utilisée consiste à réutiliser les outils de navigation ou reproduire l'interface des outils les plus fréquemment utilisés. Nous proposons de lier la conception d'un outil de navigation à un outil "témoin" issu d'un autre système ou d'une autre partie du système. L'outil témoin doit être très répandu pour vérifier le plus largement possible le critère de réutilisabilité des entrées et des sorties. Lorsque la conception d'un outil de navigation ne suit pas ce patron, nous conseillons de le composer avec un outil redondant mais plus classique.

5 *Conclusion*

Dans ce chapitre, nous avons présenté des espaces de conception pour le domaine de la visualisation de grandes quantités d'information. Cette approche est basée sur une démarche que nous avons fixée au chapitre précédent. Nous analysons donc un système de visualisation comme un système multimodal en sortie. Cette analyse nous permet de décomposer une technique de visualisation en plusieurs modalités. Comme au chapitre précédent nous caractérisons et composons des modalités mais cette fois dans le but d'obtenir une modalité graphique informationnelle.

L'étude des espaces de conception existants a débouché sur un cadre d'étude fédérateur sur lequel nous avons appliqué notre démarche. L'étude de ce chapitre est donc fortement influencée par les quatre étapes du cadre fédérateur intitulées "Données", "Point de vue sur les données", "Espace" et "Point de vue sur l'espace". Nous proposons un espace de conception différent pour chacune des trois dernières étapes. Dans notre approche centrée sur l'utilisateur, ces étapes correspondent à trois niveaux d'abstraction dans la communication des données de l'ordinateur vers l'utilisateur.

Afin de d'illustrer comment ces espaces de conception sont utiles dans les choix de conception nous proposons cinq nouveaux patrons spécifiques au domaine de la visualisation de grandes quantités d'information. Dans ces patrons nous utilisons à la fois les termes propres à la multimodalité et les termes propres à la visualisation. Au Tableau 5, nous situons ces cinq patrons par rapport aux trois axes de classification de patron que nous avons présenté au chapitre II (Figure 1, page 27).

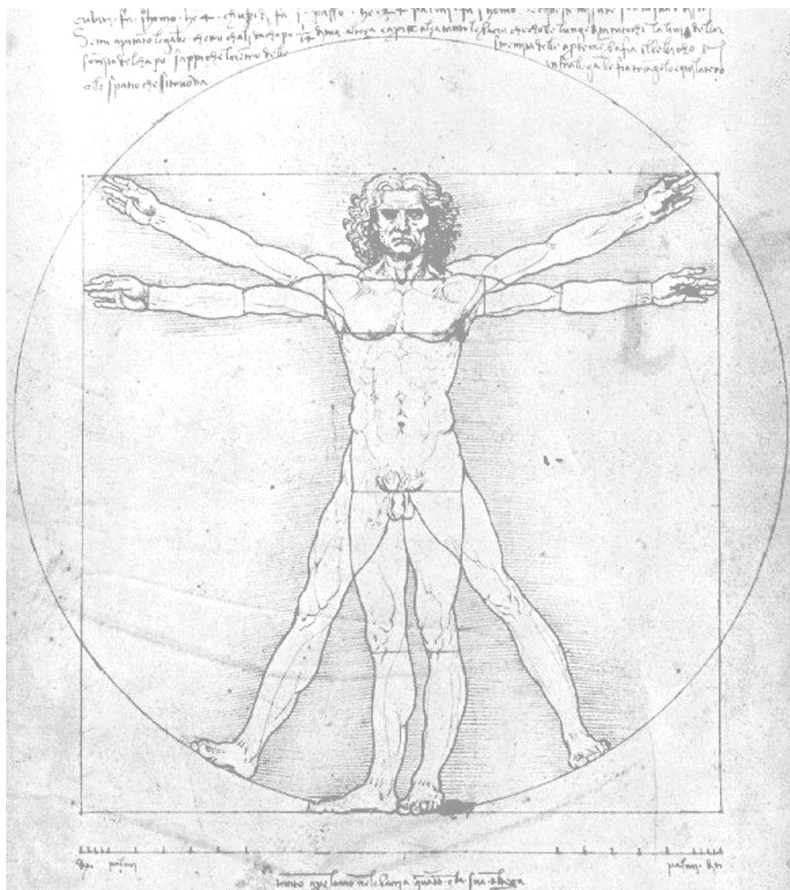
	Choix des modalités	Critère d'ergonomie	Niveau d'abstraction
<i>Patron N°5 : Continuité spatiale des modalités</i>	2 modalités	Observabilité Honnêteté (IIHM) Groupement par la localisation (Scapin)	Interfaces de visualisation
<i>Patron N°6 : Continuité temporelle</i>	1 modalité ou plus	Tolérance au rythme Observabilité Honnêteté	Interfaces de visualisation
<i>Patron N°7 : Plaquage des données en traits graphiques</i>	1 modalité	Adaptativité Adaptabilité Groupement / Distinction entre items (Scapin)	Interfaces de visualisation
<i>Patron N°8 : Compréhension de la déformation</i>	2 modalités	Honnêteté (IIHM) Protection contre les erreurs (Scapin)	Interfaces de visualisation
<i>Patron N°9 : Réutilisation des outils de navigation</i>	1 modalité ou plus	Homogénéité / cohérence (Scapin) Prise en compte de l'expérience de l'utilisateur (Scapin)	Interfaces de visualisation

Tableau 5 : Caractéristiques de nos patrons de conception d'interfaces de visualisation.

Dans les chapitres suivants, nous présentons les réalisations logicielles que nous avons réalisées pour étayer notre thèse. Ces réalisations portent notre discours sur le terrain concret du concepteur et illustre notre approche.

CHAPITRE V

RÉALISATION LOGICIELLE



*If anything can go wrong,
... it will*

Loi de Murphy

Dans les chapitres précédents, l'exploration de l'espace de conception des interfaces multimodales et de visualisation nous a permis d'établir des cadres conceptuels. Nous avons concrétisé ces concepts avec trois réalisations logicielles : VITESSE, MulTab et Parent. Ces systèmes font l'objet des trois chapitres suivants. Ce chapitre constitue une introduction à ces trois chapitres en définissant les supports pratiques communs à l'activité de réalisation logicielle.

Ce chapitre est organisé ainsi : nous motivons d'abord la réalisation de nos trois systèmes en les positionnant par rapport à nos travaux conceptuels. Nous développons ensuite les éléments de réalisation communs aux systèmes en les présentant selon les étapes du processus de développement logiciel : conception d'algorithmes particuliers, architecture logicielle et implémentation. Parmi ces éléments de réalisation, nous accordons une place de choix à la réalisation des vues hyperboliques utilisées dans VITESSE et MulTab. Nous expliquons ensuite l'architecture logicielle des trois systèmes selon le modèle d'architecture PAC-AMODEUS puis nous abordons les principes de programmation en JAVA.

1 Motivations

Nous identifions trois finalités complémentaires à la réalisation logicielle des systèmes VITESSE, MulTab et Parent.

En réalisant plusieurs systèmes (trois en tout), nous nous sommes placés au cœur du travail de conception. Cette situation privilégiée nous a permis d'identifier et de mieux comprendre quels étaient les enjeux de conception. Des itérations successives dans la conception des systèmes, ont émergé des points de conception récurrents, que nous avons ensuite analysés, abstraits et organisés dans des espaces cohérents.

La seconde motivation a été de pouvoir tester l'application des patrons. En effet, nous les avons rédigés avant que les trois systèmes n'aient atteint leur état actuel. Nous avons souvent bénéficié de situations où un patron a guidé la conception. Nous avons aussi constaté que les patrons complètement rédigés, et en particulier la rubrique contenant les exemples, ont contribué à identifier des solutions nouvelles. De plus cette expérience pratique nous a également servi à reconnaître dans d'autres logiciels, des exemples d'application d'un patron.

Enfin, ces systèmes forment un bon démonstrateur de nos patrons. Nous n'avons résolument pas utilisé nos systèmes comme des exemples dans nos patrons (Chapitres III et IV). En effet, jusqu'à la fin de nos travaux, ces systèmes étaient en cours de développement. Ils ne constituaient donc pas de bons candidats pour illustrer un patron. VITESSE, MulTab et Parent font maintenant l'objet de communications de recherche dans trois conférences internationales

(respectivement [Vernier 97], [Vernier 00a] et [Vernier 00b]). Ces systèmes sont également diffusés sur Internet par le mécanisme d'applet. Ils pourront donc à l'avenir illustrer les patrons que nous proposons dans ce manuscrit. En conclusion de ce chapitre, nous synthétisons dans un tableau les patrons illustrés par chaque système.

Les finalités de la réalisation de nos systèmes étant exposées, nous détaillons dans la section suivant la réalisation des vues hyperboliques, constituant une base commune à VITESSE et Multab, puis nous expliquons l'architecture logicielle des systèmes à la section 3 et enfin les principes de programmation à la section 4. Reposant sur ces éléments communs de réalisation, les trois systèmes sont subséquemment détaillés dans les chapitres suivants.

2 *Réalisation de vues hyperboliques*

Ce paragraphe est consacré aux outils mathématiques utilisés pour la réalisation des vues hyperboliques. Avant d'expliquer les différentes vues hyperboliques, nous posons les hypothèses suivantes, communes aux deux systèmes VITESSE et MulTab :

- ▶ L'espace est organisé en deux dimensions (espace 2D).
- ▶ Il n'y a qu'un seul focus.

La conception d'une vue hyperbolique se situe à l'étape "point de vue sur l'espace" présenté au paragraphe 1.6 du chapitre IV. Il est donc légitime de nous appuyer sur la première hypothèse qui se situe effectivement au niveau de la conception de l'espace, étape précédente dans notre cadre fédérateur. L'hypothèse sur le nombre de foci est valide dans ce chapitre. C'est une hypothèse minimale commune aux deux systèmes VITESSE et MulTab. Nous détaillons au Chapitre VII une modalité à plusieurs foci, spécifique à MulTab.

Pour présenter les vues hyperboliques, nous introduisons en premier lieu au paragraphe 2.1 la fonction de déformation nécessaire à la construction d'une vue hyperbolique. Nous expliquons, ensuite au paragraphe 2.2, l'application de cette fonction à deux systèmes de coordonnées pour obtenir deux vues hyperboliques différentes. Au paragraphe 2.3, la manipulation du coefficient de déformation permet de définir des variantes des deux vues obtenues. En utilisant deux formules différentes sur les deux axes de l'espace 2D, nous obtenons une vue complètement différente des précédentes, que nous exposons au paragraphe 2.4. Enfin, le paragraphe 2.5 décrit l'utilisation de l'un des paramètres de la fonction de déformation pour obtenir une zone d'affichage rectangulaire.

2.1 Fonction de déformation

Nous appliquons une fonction de déformation aux points de l'espace 2D. La fonction de déformation $FDeform(x)$, proposée par Volkmar Hovestadt [Hovestadt 95] et reproduite à la Figure 48, constitue le point de départ de notre réalisation. Nous présentons également la fonction inverse $FDeform^{-1}(x)$.

$$\blacktriangleright FDeform(x) = x \times \frac{\sqrt{(x^2 + z^2)(r^2 - (z - o)^2) + z^2(z - o)^2 + z(z - o)}}{x^2 + z^2}$$

$$\blacktriangleright FDeform^{-1} = \frac{z \times x}{\sqrt{r^2 - x^2 + (z - o)}}$$

Les auteurs de cette fonction définissent deux zones : une zone agrandie et une zone rétrécie. A la Figure 48, le tracé de la fonction de déformation $FDeform(x)$ est comparée à celui de la fonction identité pour r fixé à 300 pixels. La fonction identité $y=x$ est la fonction qui ne déforme pas l'espace. Lorsque la fonction $FDeform(x)$ croît

plus vite que la fonction identité, l'espace initial est agrandi. Lorsque la fonction $FDeform(x)$ croît moins vite que la fonction identité, l'espace initial est rétréci. Sur le tracé de la fonction $FDeform(x)$, la tangente à $y=x$ définit donc la limite entre la zone agrandie et la zone rétrécie.

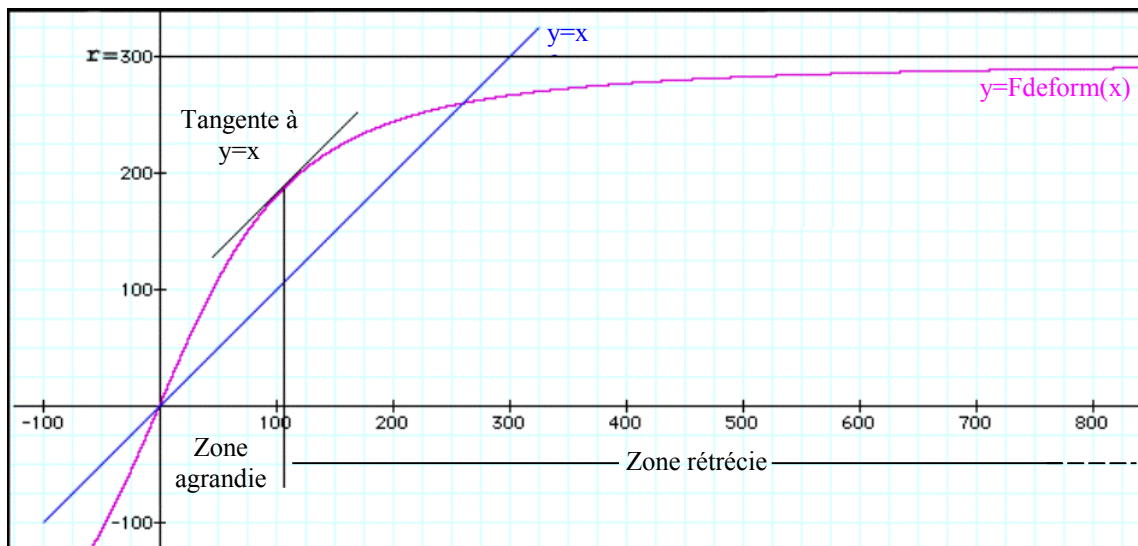


Figure 48 : Tracé de la fonction de déformation $FDeform(x)$ [Hovestadt 95] pour les valeurs : $r=300$, $o=150$ et $z=112$.

Cette fonction est basée sur le modèle géométrique reproduit à la Figure 49. Dans ce modèle, un point P fait d'abord l'objet d'une projection centrale depuis un plan sur une sphère ①. Dans un second temps la projetée de ce point est à nouveau projeté orthogonalement depuis la sphère sur le plan de départ en ②. Dans ce modèle x représente la distance de P à l'axe central (en pointillés verticaux) et $FDeform(x)$ représente la distance de P' à cet axe.

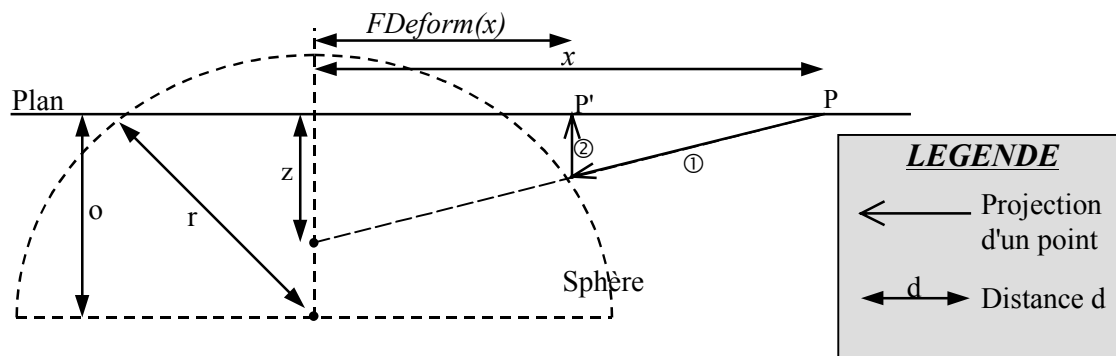


Figure 49 : Modèle géométrique de projection en deux étapes correspondant à la fonction de déformation $FDeform(x)$ [Hovestadt 95].

Dans les équations de [Hovestadt 95], x désigne la distance en pixels que l'on cherche à déformer. Les paramètres r , o et z permettent d'adapter la fonction aux besoins de l'interaction. Le paramètre r fixe la valeur limite de la fonction. Il fixe donc la surface de visualisation à l'écran : ainsi dans la visualisation sphérique de la Figure 51c, r désignera le rayon de la forme de sphère. Les paramètres o et z fixent les proportions de la zone agrandie et de la zone rétrécie par rapport à la surface totale de visualisation. Nous utilisons le terme de "puissance de l'effet de loupe" pour désigner l'impact de o et z dans la déformation de l'espace.

Pour illustrer cette fonction de déformation, imaginons une rangée de carrés répartie sur la dimension du plan présentée dans le modèle de [Hovestadt 95]. La Figure 50 illustre la déformation de ces carrés par la fonction $FDeform(x)$ sur une seule dimension.

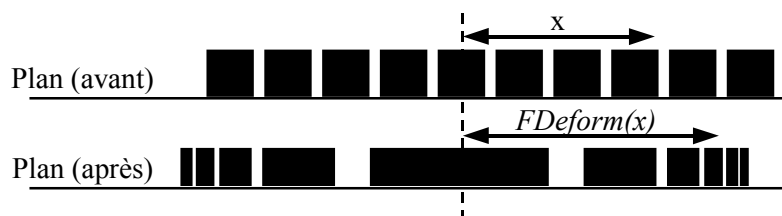


Figure 50 : Déformation d'une rangée de carré.

A partir du modèle géométrique, nous calculons la fonction inverse $FDeform^{-1}$. Cette fonction est nécessaire à l'interaction puisqu'elle permet de calculer la position des clics souris dans l'espace non déformé. Dans le reste de notre discours, nous ne présenterons pas l'utilisation de la fonction inverse qui se déduit systématiquement de l'utilisation de la fonction de déformation.

Nous remarquons que la formule de [Hovestadt 95] est compliquée à cause du modèle géométrique composé de deux projections. Nous pouvons interpréter cette formule et en déduire que seules les quatre propriétés suivantes nous intéressent :

- $FDeform(0)=0$. La fonction ne translate pas le centre de l'espace.
- $FDeform'(x)>1$ pour $x \rightarrow 0$. La fonction agrandit la zone près du centre.
- $FDeform(x) \rightarrow R$ pour $x \rightarrow \infty$. La fonction ramène tous les points d'un espace infini dans un espace fini, borné par R .
- $FDeform'(x)>0$ pour $x>0$. La fonction strictement croissante ne replie pas l'espace sur lui-même.

Dans [Nigay 98], nous avons donc remplacé les fonctions $FDeform(x)$ et $FDeform^{-1}(x)$ par des fonctions d'approximation plus simples et qui vérifient les mêmes propriétés. Dans tous nos travaux, nous avons utilisé ces fonctions afin de réduire le temps de calcul par l'ordinateur :

- $$FDeform(x) = \frac{R \times Z}{-Z - x} + R$$
- $$FDeform^{-1}(x) = \frac{R \times Z}{R - x} - Z$$

Bien que le paramètre R n'est pas exactement le paramètre r expliqué précédemment, nous l'utilisons de la même manière que r . De plus ces fonctions ne font apparaître qu'un seul coefficient relatif à la puissance de l'effet de loupe : Z . La manipulation par l'utilisateur de l'effet de loupe en est simplifiée. Il suffit de borner les valeurs possibles de Z entre deux valeurs proportionnelles à R . Comme pour R et r , nous notons que Z et z ne sont pas identiques.

2.2 Application de la fonction aux systèmes de coordonnées

La fonction $FDeform(x)$ que nous venons de présenter peut être appliquée à deux systèmes de coordonnées : le système de coordonnées cartésiennes (x, y) ou le système de coordonnées polaires (α, d) . Dans le deuxième cas, l'angle α est conservé et la distance d est déformée. Nous illustrons ces deux types d'application de la fonction de déformation à la Figure

51. L'espace choisi, pour illustrer ces deux points de vue sur l'espace, est une matrice de cent carrés noirs (dix en hauteur par dix en largeur) présentée à la Figure 51a. Cet espace sera utilisé pour toutes les illustrations de ce chapitre.

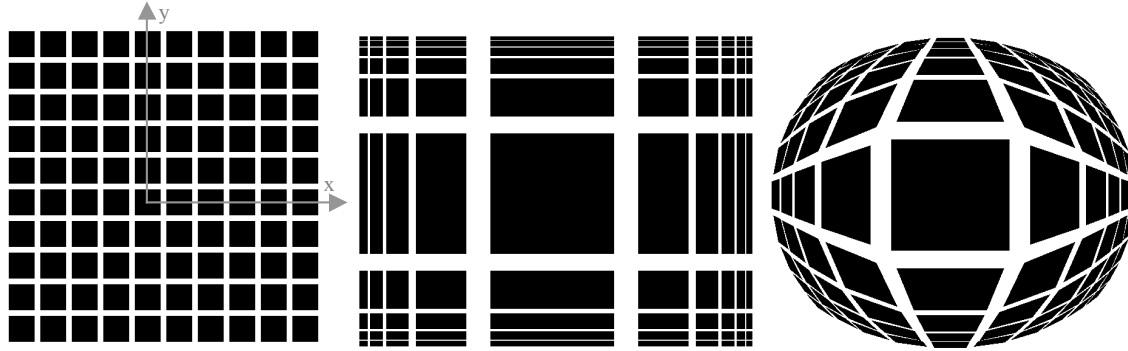


Figure 51a :
 $(x,y) \rightarrow (x,y)$
 Espace de départ.

Figure 51b :
 $(x,y) \rightarrow (FDeform(x), FDeform(y))$
 Déformation des coordonnées
 cartésiennes.

Figure 51c :
 $(\alpha, d) \rightarrow (\alpha, FDeform(d))$
 Déformation des coordonnées
 polaires.

Comparons les deux applications de cette fonction en utilisant notre taxonomie des effets de déformation, présentée à la Figure 35 (page 118). La fonction de déformation et sa dérivée sont continues. L'effet de perspective est présent dans la vue de la Figure 51c mais pas dans celle de la Figure 51b. Néanmoins la vue de la Figure 51b conserve l'alignement des lignes et des colonnes alors que la vue de la Figure 51c ne conserve aucun alignement.

2.3 Manipulation du coefficient de déformation

Ce paragraphe est dédié à un autre élément de réalisation : le coefficient de déformation. Ce coefficient est basé sur la fonction de déformation. Il facilite la manipulation de vues hyperboliques. En effet la fonction de déformation est une abstraction mathématique très utile, mais un concepteur non expert en Mathématiques ne peut pas facilement l'adapter à ses besoins. Nous proposons donc de considérer le coefficient de déformation *CoeffDeform* associé à l'application d'une fonction de déformation :

$$CoeffDeform = \frac{FDeform(v)}{v}$$

Si (x,y) et (x', y') représentent respectivement le point non-déformé et le point déformé, les étapes d'application de la déformation aux coordonnées cartésiennes et polaires sont alors les suivantes :

Application de la déformation aux coordonnées cartésiennes	Application de la déformation aux coordonnées polaires
-1- $Coeffx = \frac{FDeform(x)}{x}$	-1- $d = \sqrt{x^2 + y^2}$
-2- $Coeffy = \frac{FDeform(y)}{y}$	-2- $Coeff = \frac{FDeform(d)}{d}$
-3- $x' = x \times Coeffx$	-3- $x' = x \times Coeff$
-4- $y' = y \times Coeffy$	-4- $y' = y \times Coeff$

Pour la déformation cartésienne, nous multiplions deux coefficients différents à x et à y, alors que pour la déformation polaire, un coefficient unique est utilisé pour les deux coordonnées.

L'enjeu lors de la réalisation d'une vue hyperbolique réside alors dans le calcul de ce coefficient avant de l'appliquer. Dans les paragraphes suivants, différentes méthodes de calcul permettent de :

- obtenir une forme de pyramide,
- aplanir la partie centrale de l'espace,
- aplanir la partie extérieure de l'espace,
- calculer une forme hybride à partir de deux autres formes.

2.3.1 Déformation en forme de pyramide

Une première méthode de calcul du coefficient de déformation permet d'obtenir la forme de pyramide illustrée à la Figure 52. Cette forme est comparable à celle du "Document Lens" de Georges Robertson [Robertson 93], illustrée à la Figure 53, aussi proposée par [Leung 94].

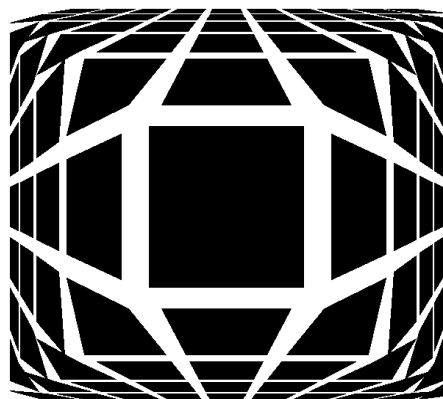


Figure 52 : Déformation en forme de pyramide à base carrée, vue du sommet.

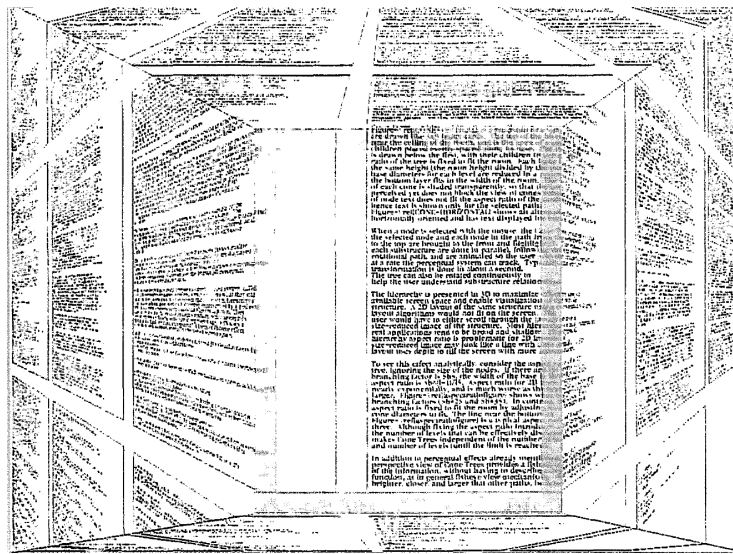


Figure 53 : Document Lens [Robertson 93].

Pour obtenir cette forme, nous reconsidérons la déformation des coordonnées cartésiennes en lui apportant la modification suivante : au lieu d'utiliser des coefficients de déformation différents pour l'axe des abscisses et des ordonnées, nous calculons un coefficient unique comme suit :

$$CoeffDeformCommun = \min(CoeffDeformx, CoeffDeformy)$$

Cette nouvelle forme offre un effet de perspective et ressemble à une pyramide à base carrée, vue du sommet. Toutefois, si nous considérons la déformation des polygones, nous constatons une discontinuité visuelle gênante au niveau des arêtes de la pyramide. Les polygones qui coupent une arête de la pyramide sont mal déformés. Cet effet indésirable est dû à la discontinuité de la dérivée de la fonction "min". Par conséquent, la dérivée de la fonction de déformation n'est pas continue. Pour obtenir une vue semblable à celle du "Document Lens" qui n'a pas ce défaut, il est nécessaire de considérer de nouveaux points sur les polygones. Ces points sont à l'intersection des quatre arêtes de la pyramide et des arêtes des polygones, comme expliqué à la Figure 54. Nous n'illustrons pas le résultat de cette amélioration que nous avons présentée uniquement pour permettre la comparaison de nos systèmes avec le "Document Lens". De plus nous n'avons pas programmé cette technique dans nos systèmes.

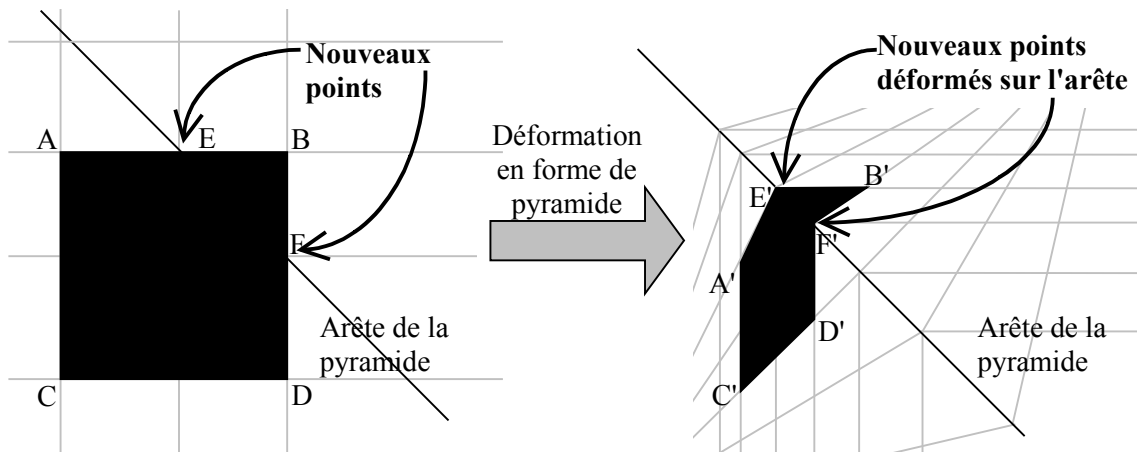


Figure 54 : Déformation d'un polygone à l'intersection de l'arête de la pyramide.

Nous n'illustrons pas le résultat de cette amélioration que nous présentons uniquement pour permettre la comparaison de nos systèmes avec le "Document Lens". Nous n'avons pas implémenté cette technique dans nos travaux.

2.3.2 *Aplanissement de la partie centrale*

Une deuxième méthode complémentaire de calcul du coefficient de déformation consiste à le borner en fixant une valeur maximale. En limitant la valeur maximale du coefficient *CoeffDeform* à *CoeffMax*, nous limitons la taille maximale des éléments. Cette méthode de calcul appliquée aux trois vues précédemment présentées (Figure 51b, Figure 51c et Figure 52) implique les formes de la Figure 55. La partie centrale est alors un simple agrandissement d'une partie de l'espace d'information. La modalité correspondant à cet agrandissement est précise, locale et non déformée. Il s'agit du **Focus**. La partie extérieure, autour de cette zone non déformée, présente le reste de l'espace selon une modalité vague mais globale et déformée. Il s'agit du **Contexte**.

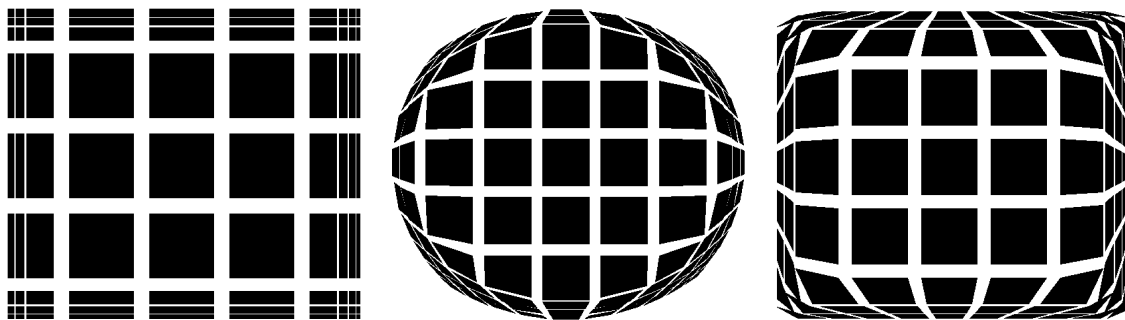


Figure 55 : Aplanissement de la partie centrale de la déformation en limitant la valeur maximale de *CoeffDeform*.

Le facteur d'agrandissement associé à la modalité "précise" dépend de la valeur de *CoeffMax*. Nous remarquons que si nous appliquons cette méthode de calcul sur les deux axes (abscisses et ordonnées), le facteur d'agrandissement est proportionnel au carré de *CoeffMax*. Un élément est agrandi quatre fois si *CoeffMax* vaut 2 car il apparaît alors deux fois plus haut et deux fois plus large.

2.3.3 *Aplanissement de la partie extérieure*

Une autre méthode de calcul du coefficient de déformation similaire à la précédente, consiste à fixer sa valeur minimale. Si nous limitons la valeur minimale du coefficient par *CoeffMin*, nous obtenons les deux vues de la Figure 56. La vue obtenue en appliquant cette méthode de calcul aux coordonnées cartésiennes avec deux coefficients distincts pour les abscisses et les ordonnées ne ressemble à aucune forme connue (non analogique) et n'est donc pas facilement compréhensible. Aussi nous ne présentons que deux vues à la Figure 56 (méthode de calcul appliquée aux vues de la Figure 51c et de la Figure 52).

Dans les deux vues de la Figure 56, la partie extérieure (c'est-à-dire le contexte) présente une vue globale beaucoup moins déformée. Pour s'en convaincre, posons-nous la question de savoir dans quelle figure (Figure 55 ou Figure 56) nous percevons mieux la matrice de 100 carrés (10*10) constituant l'espace. Cependant, et contrairement aux autres vues présentées, la zone de l'écran utilisée par ces vues n'est plus bornée par le paramètre R de la fonction de déformation. En effet, la place nécessaire de la vue obtenue selon la modalité qui présente le contexte, dépend de la taille de l'espace et du coefficient *CoeffMin*. L'emplacement du focus dans l'espace est aussi à considérer, car lorsque ce dernier se rapproche du bord, la partie déformée dépasse le cadre défini par la partie non-déformée.

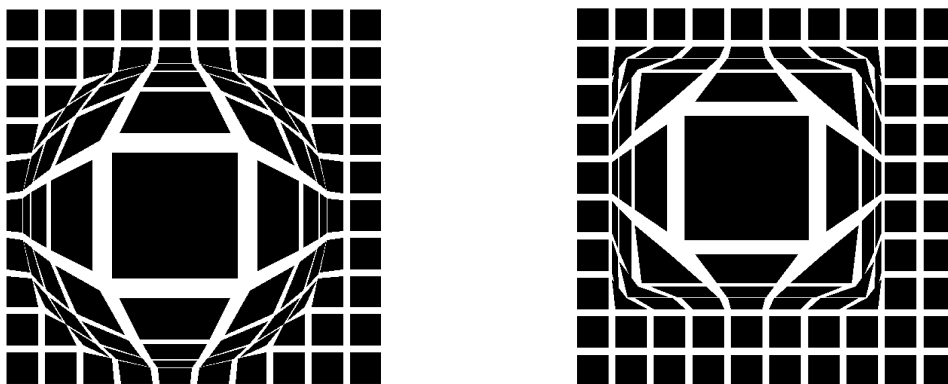


Figure 56 : Aplanissement de la partie extérieure de la déformation en limitant la valeur minimale de *CoeffDeform*.

La déformation obtenue en appliquant cette technique de conception aux coordonnées cartésiennes ne produit pas un espace très compréhensible. Nous avons pris le parti de ne pas la présenter à la Figure 56.

2.3.4 *Forme Hybride*

Une forme hybride est intermédiaire entre deux formes différentes. Nous voulons, par exemple, caractériser de forme hybride celle de la Figure 57 à mi-chemin entre la sphère et la pyramide. Une forme hybride est définie par un coefficient de déformation calculé par la moyenne de deux coefficients issus de méthode de calcul différent. En effet, pour obtenir une forme hybride, nous calculons un nouveau coefficient égal à la moyenne des coefficients des formes que nous voulons hybrider. Cette moyenne est ensuite utilisée comme coefficient appliqué aux axes des abscisses et des ordonnées. Une forme hybride donc est décrite par :

- les deux formes dont elle est issue,
- la proportion de chaque forme.

La forme hybride entre la sphère (50%) et la pyramide (50%) est illustrée à la Figure 57. La formule de moyenne fait intervenir le coefficient *CoeffDeform*, calculé pour la forme de sphère, et le coefficient *CoeffDeformCommun*, calculé pour la forme de pyramide. Notons que d'autres poids que 50%-50% peuvent être appliqués.

-1- $CoeffDeformHybrid =$

$$\left(\frac{Coeff}{2} + \frac{CoeffDeformCommun}{2} \right)$$

-2- $x' = x \times CoeffDeformHybrid$

-3- $y' = y \times CoeffDeformHybrid$

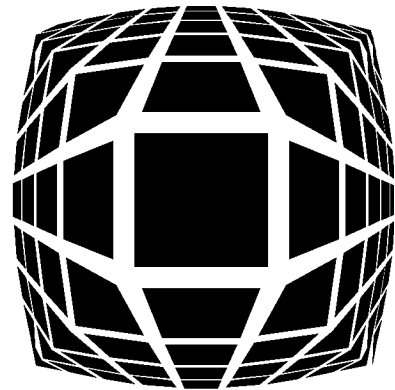


Figure 57 : Forme hybride entre la sphère (50%) et la pyramide (50%).

La forme hybride de la Figure 57, issue de deux formes analogiques, est encore analogique, car elle propose des effets de perspectives cohérents dans chaque direction. La conservation de l'analogie est due aux perspectives identiques dans les deux formes hybridées (toutes les deux, des formes hyperboliques avec une partie centrale plate et des effets de perspective selon chaque direction).

Nous remarquons enfin que cette forme remplit mieux les angles que la forme de sphère et gomme en partie les effets indésirables observés sur les arêtes de la pyramide. Il s'agit d'un compromis entre ces deux formes.

2.4 Application de fonctions de déformation différentes sur chaque axe

Un autre élément de réalisation de vues hyperboliques consiste à appliquer des fonctions de déformation différentes sur l'axe des abscisses et des ordonnées. La solution que nous avons retenue consiste à utiliser le coefficient issu de la déformation des coordonnées polaires pour l'axe des abscisses et le coefficient issu de la déformation des coordonnées cartésiennes pour l'axe des ordonnées. La forme produite est illustrée à la Figure 58. Selon notre taxonomie des effets de déformation, présentée à la Figure 35 (page 118), cette vue continue propose un effet de perspective et conserve l'alignement des lignes (valeur "sur une seule dimension" sur l'axe "Conservation de l'alignement"). Comme nous notons l'analogie avec un cylindre placé à l'horizontale, cette vue occupe une place unique dans notre espace de solution.



Figure 58 : Application de deux fonctions de déformation différentes sur chaque axe, dont résulte une forme de cylindre.

2.5 Utilisation du paramètre R pour remplir des espaces rectangulaires

Nous avons expliqué à la Figure 48 que le paramètre R est la valeur limite de la fonction de déformation. En utilisant deux valeurs différentes du paramètre R au moment de calculer la déformation sur chaque axe, nous obtenons les formes allongées illustrées à la Figure 60. L'avantage évident de ces formes est de mieux remplir un espace rectangulaire. En effet, les zones carrées ne correspondent ni aux écrans actuels (proportions 4/3 ou 16/9) ni aux fenêtres des systèmes d'exploitation graphiques que l'utilisateur retaille.

Nous notons que le cylindre occupe une place particulière. En effet, sa version allongée apparaît plus analogique (ressemblant plus à un cylindre) que celle de la Figure 58 qui occupe

un espace carré. Nous préconisons, pour favoriser l'analogie, d'allonger le cylindre uniquement sur l'axe où les alignements sont conservés (ici l'axe horizontal). Si la surface disponible est plus haute que large, il convient alors d'inverser l'utilisation des coefficients sur chaque axe.

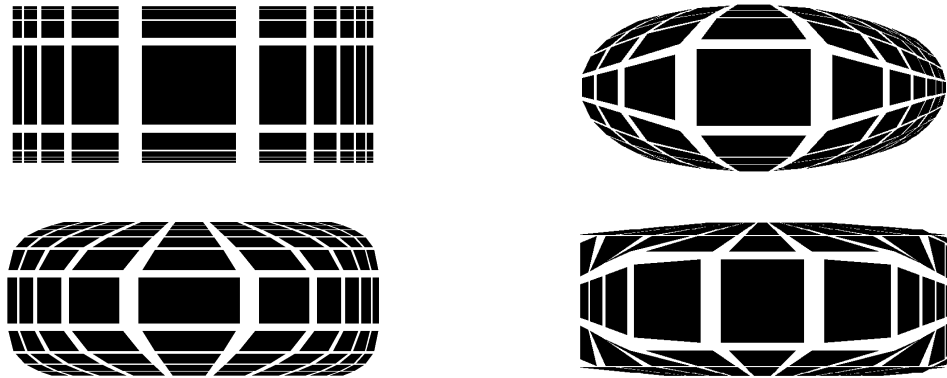


Figure 60 : Formes allongées obtenues en utilisant des valeurs du paramètre R différentes sur chaque axe.

En synthèse, la fonction de déformation et son coefficient de déformation définissent les concepts de base à la réalisation de vues hyperboliques de VITESSE et MulTab. La méthode de calcul du coefficient influence les vues obtenues. Le choix des systèmes de coordonnées, cartésien ou polaire, est aussi un élément de réalisation déterminant. Enfin nous avons souligné l'intérêt de méthodes mixtes reposant sur plusieurs fonctions de déformation et/ou sur les deux systèmes de coordonnées.

La réalisation des vues hyperboliques concerne VITESSE et MulTab. La vue particulière à Parent sera décrite dans le Chapitre VIII. Après avoir décrit les détails de réalisation des vues, nous considérons l'étape suivante dans le processus de développement logiciel : l'architecture logicielle. Notre discours est maintenant plus général et concerne les trois systèmes.

3 *Architecture logicielle*

Pour la conception logicielle de nos systèmes, nous avons appliqué le modèle d'architecture PAC-AMODEUS [Nigay 94], présenté à la Figure 58. Dans [Nigay 95], ce modèle est présenté comme un cadre conceptuel adapté à l'interaction multimodale en entrée. Nous montrons ici qu'il est aussi adéquat pour l'interaction multimodale en sortie.

Dans ses grandes lignes, ce modèle préconise cinq composants logiciels [UIMS 92] et peuple l'un deux, le Contrôleur de Dialogue (CD), d'agents PAC [Coutaz 87]. D'un côté de l'arche, le Noyau Fonctionnel (NF) maintient les informations à rendre perceptibles. De l'autre

côté, le Composant Interaction de Bas Niveau (CIBN) correspond à la plate-forme logicielle et matérielle (pilotes de périphériques).

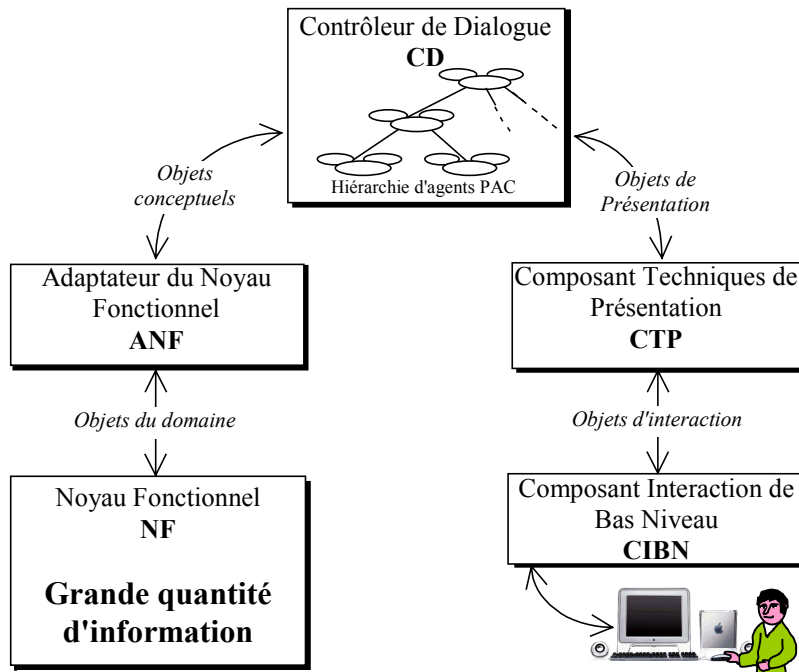


Figure 61 : Le modèle PAC-Amodeus [Nigay 94].

Ayant fixé les deux extrémités de l'arche, l'enjeu de conception logicielle réside dans la localisation des étapes du processus de visualisation de notre cadre fédérateur du paragraphe 1.6 (Chapitre IV, page 97) au sein des composants de l'architecture.

Comme nous l'avons dit, le NF est adapté au niveau Données de notre cadre fédérateur. L'Adaptateur de Noyau Fonctionnel (ANF) sert d'adaptateur entre le monde des objets du domaine et les objets conceptuels exportés vers l'utilisateur. Il filtre, fusionne, fissionne les données à rendre perceptibles. Il construit un point de vue sur les données qui sera ensuite concrétisé à l'utilisateur. L'ANF permet donc de passer du niveau Données au niveau Espace du Contrôleur de Dialogue en définissant un ou des points de vue sur les données (Espace de composition, paragraphe 2.1 du Chapitre IV, page 103). L'espace sur les données est maintenu dans la facette Abstraction d'un agent du Contrôleur de Dialogue. En effet, cette facette permet la communication entre le CD et l'ANF. La facette Présentation d'un agent définit le comportement perceptible de l'agent. Il maintient donc le point de vue sur les données. Ce point de vue présente l'information que l'utilisateur a besoin comme le focus à l'endroit désiré. Comme l'explicite la Figure 61, la facette Contrôle de l'agent maintient les liens entre l'espace et le point de vue sur l'espace. Cette facette définit donc l'espace qui sera perçu par l'utilisateur. La vérification des critères d'ergonomie d'observabilité et d'honnêteté réside donc dans la

réalisation du Contrôle de chaque agent. Nous comparons la fonction de la facette Contrôle à l'opérateur de transformation de réalisation visuelle dans le processus de Ed Chi décrit au paragraphe 1.1 du Chapitre IV (page 90). Nous justifions aussi le fait de localiser les espaces et les points de vue au niveau du CD par le fait qu'ils sont déterminants dans le dialogue. En effet, l'espace créé (niveau Espace) forme une interface décomposable dont la décomposition dépend du dialogue. Cet espace est rendu perceptible à l'utilisateur au niveau du point de vue sur l'Espace et ce point de vue peut également dépendre de l'état du dialogue, en particulier l'emplacement du focus. De plus dans le cas où l'utilisateur peut contrôler le placement des éléments graphiques, le CD garde le contrôle du dialogue en réalisant l'algorithme de placement des éléments. Le CTP quant à lui fournit les primitives pour restituer les éléments de l'espace avec les traits graphiques que le concepteur a choisis.

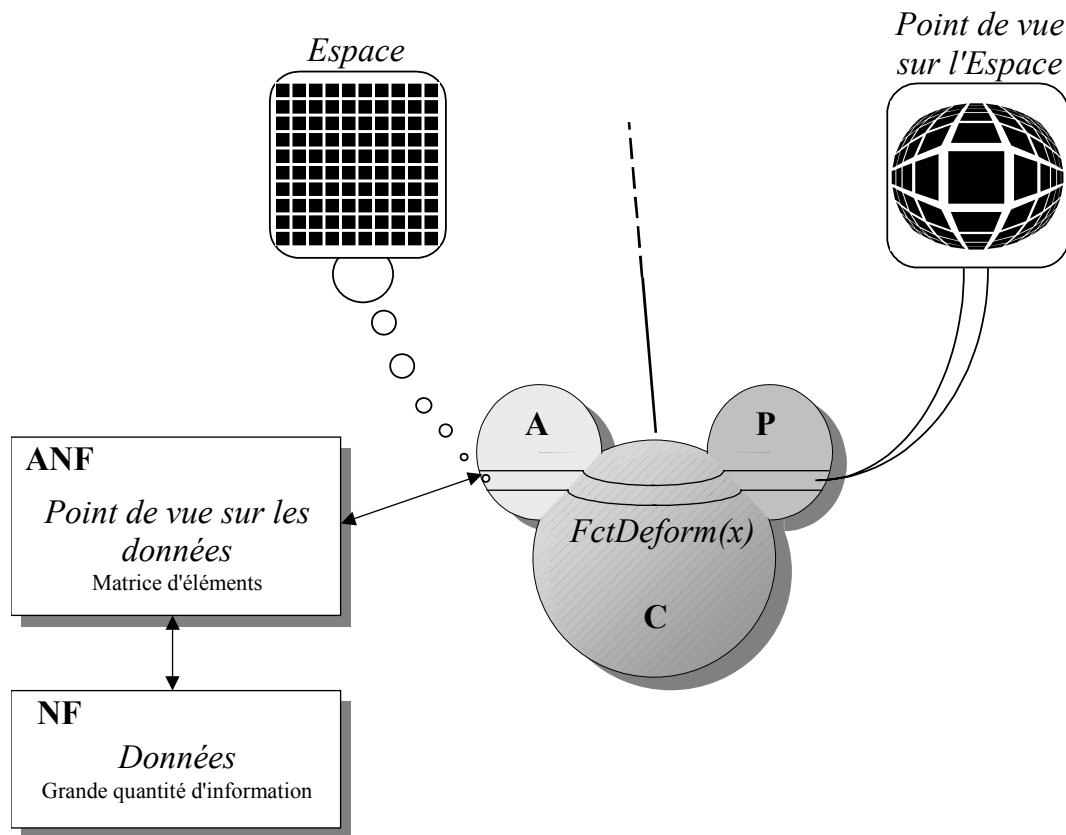


Figure 62 : Proposition de correspondance entre les étapes du processus de visualisation (*Italique*) et les composants du modèle PAC-AMODEUS (**Gras**).

Cette mise en correspondance entre les étapes de notre cadre fédérateur et les composants de PAC-AMODEUS fournit une solution générique à la conception logicielle du processus de visualisation. De plus l'intérêt d'avoir une décomposition logicielle fine du CD en agents permet de localiser les compositions de modalités. Comme expliqué dans notre espace

de composition du Chapitre IV, nous distinguons deux cas : la composition du focus et contexte et la vue multiple du même espace selon des points de vue différents.

Composition du focus et du contexte :

La décomposition en agents constitue un support logiciel pour décomposer l'espace en sous-espaces, selon les modalités exploitées pour les concrétiser. Nous préconisons un agent par sous-espaces. Un agent est dédié au focus tandis qu'un autre au contexte. Ainsi nous explicitons la composition de modalités au niveau du Contrôle d'un agent, père des agents maintenant les sous-espaces. Ils peuvent donc définir des points de vue différents sur ces sous-espaces et donc correspondent à des modalités différentes. Typiquement la modalité véhiculant le focus est précise tandis que celle véhiculant le contexte est vague. Un agent, père de ces deux agents, assure le passage d'une modalité à l'autre et la continuité entre les modalités. Nous montrons cette hiérarchie à la Figure 63a. Cet agent ciment définit donc le type de composition entre les deux modalités.

Vue multiple du même espace selon des points de vue différents :

De façon similaire au cas précédent, nous dédions un agent par point de vue sur l'espace. Néanmoins, l'espace n'est pas dupliqué dans chacun des agents, mais un exemplaire unique est maintenu par la facette Abstraction de l'agent père, ciment des agents définissant un point de vue sur l'espace. Dans [Vernier 97], nous avons exposé cette solution utilisant un agent ciment pour assurer la cohérence entre les vues. Cet agent appelé "Vue multiple" assure aussi le dialogue avec l'utilisateur pour passer d'une vue à l'autre et garantit la continuité du passage de l'une à l'autre.

Nous entrevoyons donc des classes d'agents selon les modalités et selon les compositions. A cet endroit du discours, il est important de comprendre qu'une hiérarchie d'agents PAC ne représente pas une hiérarchie de classes dans les termes de la programmation orientée-objet, mais des instances qui communiquent entre elles. Aussi si une modalité est exploitée pour visualiser deux sous-espaces distincts, à l'exécution, deux agents PAC de la classe correspondante à la modalité seront créés dynamiquement.

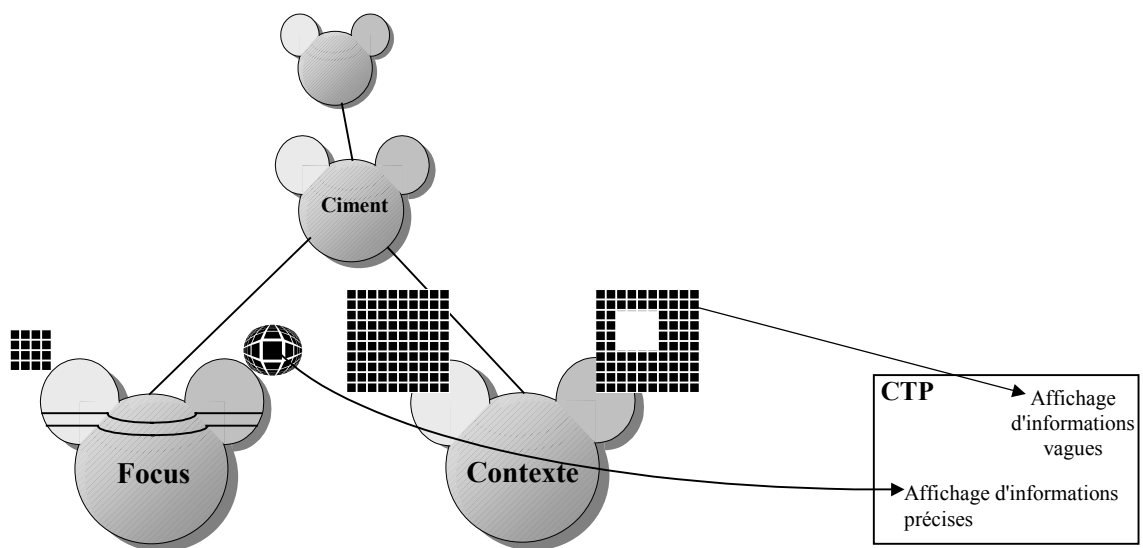


Figure 63a : Réalisation de la composition du focus et du contexte.

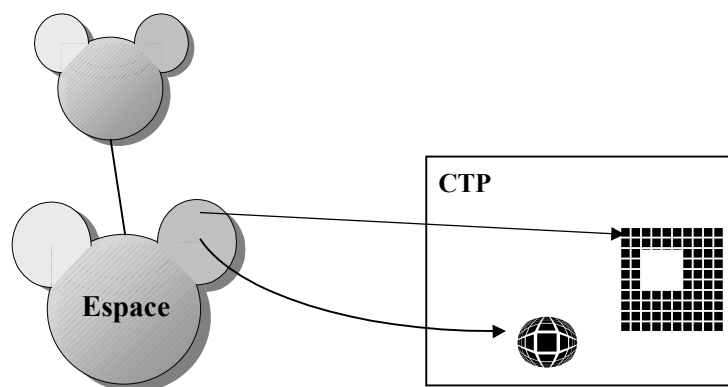


Figure 63b : Application du mécanisme Slinky à la définition du point de vue.

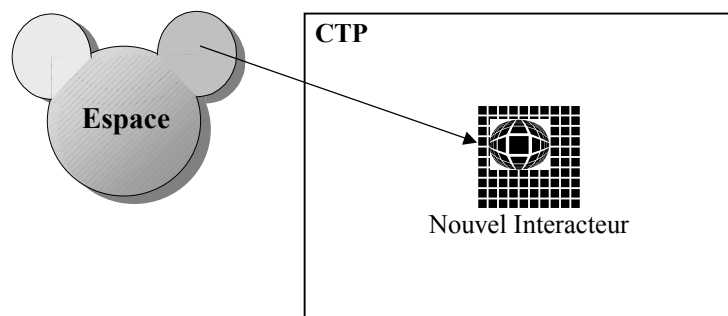


Figure 63c : Application du mécanisme Slinky à la composition du point de vue.

Considérer la notion de modalité dans le CD est en désaccord avec les résultats présentés dans [Nigay 95] qui prône un CD indépendant des modalités : pour cela, le CTP est présenté comme dépendant du langage d'interaction tandis que le CIBN est dépendant des

dispositifs physiques. Selon ces résultats, la correspondance entre l'espace et le point de vue sur l'espace devrait donc être réalisée dans le CTP, le niveau langage de la modalité. Dans le cas de modalités graphiques complexes en sortie, nous justifions notre conception logicielle par l'absence d'outils prêts à l'emploi qui réaliseraient ces modalités. Un principe de base dans la définition de la hiérarchie d'agents est que si un agent correspond à un objet d'interaction prédéfini dans l'environnement, alors l'agent est supprimé et sa compétence reléguée à son père. Or les boîtes à outils actuels n'offrent pas des modalités graphiques complexes.

Dans ce contexte, nous aurions pu alors adopter une approche différente, illustrée à la Figure 63, et décider d'étendre les boîtes à outils par les modalités graphiques définies. Pour cela et à partir de notre solution actuelle, nous devons nous reposer sur le mécanisme Slinky du modèle Arch sur lequel se base PAC-AMODEUS. Le modèle Arch [UIMS 92] constitue une solution canonique en cinq composants. Arch est un métamodèle, qui couplé avec le mécanisme Slinky de migration de fonctions d'un composant à l'autre, définit des instances d'Arch. Nous appliquons ce mécanisme Slinky à la définition du point de vue, que nous avons ci-dessus localisée dans la facette Contrôle (la facette Présentation maintenant ce point de vue) des agents PAC du CD. La modalité est alors réalisée dans le CTP et l'agent correspondant dans le CD est supprimé. La Figure 63b montre la hiérarchie résultante. La modalité définit un interacteur de haut niveau, qui pourra utiliser les spécificités du CIBN pour une restitution plus rapide.

Dans les cas de composition de modalités présentés ci-dessus, nous obtenons donc plus qu'un seul agent (ex agent ciment), car les agents fils sont supprimés. La composition est effectuée par cet agent comme suit :

- Au niveau de la facette Contrôle, l'agent fissionne ou duplique l'espace total alors maintenu dans la facette Abstraction.
- La facette Présentation est alors responsable de la continuité dans le changement de modalités définies dans le CTP.

Les modalités graphiques étant alors réalisées dans le CTP comme des interacteurs de haut niveau étendant la boîte à outils, nous pourrions aussi considérer l'application du mécanisme Slinky aux compositions. Ces dernières définiraient alors des mécanismes pour combiner des interacteurs de la boîte à outils. Comme le montre la Figure 63c, l'agent PAC définissant la composition n'a alors plus lieu d'être, sa compétence est reléguée à son agent Père.

4 *Détails d'implémentation*

L'architecture logicielle expliquée, nous nous intéressons maintenant à son implémentation. Pour réaliser nos techniques de visualisation, nous avons choisi l'environnement du JDK [JDK], créé par SUN. Cet environnement fournit un langage de programmation orientée objet, JAVA, et des bibliothèques de composants d'interaction (AWT, Swing, JavaSound, etc.) indépendants de chaque plate-forme. Ces bibliothèques forment un CTP acceptable qui permet au concepteur de s'abstraire des différences entre les plates-formes Windows, MacOS et Unix, ces dernières formant le CIBN.

Le langage JAVA est un langage à objet parfaitement adapté à la réalisation logicielle des agents du CD. Le mécanisme de classe et d'instance permet de créer un prototype d'agent et de l'instancier plusieurs fois dans la hiérarchie d'agents. Nous avons choisi de construire trois objets par agent PAC, un pour chaque facette. Chaque agent possède un nom unique qui permet aux autres agents de lui adresser un message. Chaque facette Contrôle possède une référence sur les facettes Contrôle de :

- ▶ son père dans la hiérarchie d'agents,
- ▶ ses fils dans la hiérarchie d'agents.

Le mécanisme d'abonnement aux messages du langage JAVA permet à plusieurs objets de recevoir des copies du même événement en provenance du CTP. Cette particularité permet une indépendance totale entre les interacteurs du CTP et les agents du CD. Ainsi, un agent peut gérer plusieurs interacteurs du CTP et un interacteur peut être géré par plusieurs agents du CD.

Nous avons enfin exploité le mécanisme d'applet (application graphique placée dans une page Web) de l'environnement du JDK, pour deux raisons distinctes :

- ▶ Une applet permet de délocaliser le NF et le CD sur deux ordinateurs différents.
L'architecture client-serveur que nous mettons ainsi en place (client=CD, serveur=NF) permet de séparer les environnements nécessaires pour le NF et le CD. Le NF stocke et traite de grandes quantités de données (périphériques de stockage rapides, mécanismes de sauvegarde, gestion des accès concurrents, etc.). Le CD gère le dialogue avec l'utilisateur dans son contexte (périphériques nomades ou fixes, différentes tailles d'écran, différentes qualités du rendu sonore, etc.).
- ▶ Le système d'applet permet également de montrer, à tous les concepteurs intéressés, les mécanismes d'interaction qui sont des exemples des patrons de conception. Un patron de conception, qui est par nature un document, peut être au format HTML et ainsi être illustré

par des exemples réels. Nous rappelons qu'un document HTML (par exemple une page Web) peut être composé de texte, d'images et aussi d'applets JAVA.

5 *Conclusion*

Dans ce chapitre, nous avons introduit les réalisations logicielles qui concrétisent les résultats conceptuels des chapitres précédents. Nous avons présenté les éléments de réalisation communs à nos trois systèmes. Organisés selon le processus de développement logiciel, nous avons d'abord détaillé la réalisation des vues hyperboliques de VITESSE et MulTab. Nous avons ensuite détaillé l'architecture logicielle des systèmes selon le modèle PAC-AMODEUS pour finir par son implémentation dans l'environnement du JDK de SUN.

La mise en place d'une architecture logicielle nous a conduit à établir une correspondance entre les composants logiciels du modèle d'architecture PAC-AMODEUS et les étapes du processus de visualisation de notre cadre fédérateur (Chapitre IV). Nous définissons ainsi un cadre conceptuel générique à la réalisation logicielle du processus de visualisation. De plus en établissant des liens explicites entre nos résultats conceptuels et notre architecture logicielle, nous dessinons les contours d'une démarche de traduction de nos patrons de conception d'interface en patrons de conception logicielle. Nous développerons ce point dans les perspectives à notre travail.

Dans les chapitres suivants, nous concrétisons notre cadre conceptuel de réalisation logicielle en détaillant les trois systèmes, VITESSE, puis MulTab et enfin Parent. Le Tableau 7 ci-dessous permettra au lecteur d'anticiper sur le contenu des chapitres suivants, en présentant, pour chaque système, les patrons de conception qui ont été utilisés ou illustrés.

	VITESSE	MulTab	Parent
<i>Patron N°1 : Test du dispositif</i>		Résolution de l'écran et test du dispositif sonore	Résolution de l'écran
<i>Patron N°2 : Préférence du canal le moins occupé</i>		Messages sonores pour indiquer l'apparition ou la disparition de focus	Musique de piano pour renforcer la perception de distance d'un glisser-déposer
<i>Patron N°3 : Adaptabilité des modalités de sortie</i>	Taille de la fenêtre et paramètres de la vue hyperbolique	Paramètres de la vue hyperbolique	Paramètres de l'algorithme des Fieldmaps
<i>Patron N°4 : Modalités complémentaires et simultanées</i>	Focus+Contexte	Focus+Contexte, foci multiples, table coloriée et légende	Fieldmaps et légendes des couleurs
<i>Patron N°5 : Continuité spatiale des modalités</i>	Vue hyperbolique et histogramme d'opinion	Vue hyperbolique et foci multiples	Bulles d'explication et vue alternative
<i>Patron N°6 : Continuité temporelle</i>	Passage d'une vue à l'autre	Apparition et disparition progressives des foci	
<i>Patron N°7 : Plaquage des données en traits graphiques</i>	Couleurs des éléments et tailles des barres d'un histogramme	Couleurs des cellules	Couleurs et tailles des éléments
<i>Patron N°8 : Compréhension de la déformation</i>	Formes analogiques à facettes	Quadrillage de la table de données	Non
<i>Patron N°9 : Réutilisation des outils de navigation</i>	Mêmes outils entre les vues	Même outil de légende des couleurs que Parent	Même outil de légende des couleurs que MulTab

Tableau 7 : Illustration des patrons de conception dans nos trois prototypes.

CHAPITRE VI

VITESSE : VISUALISATION DES RESULTATS D'UN MOTEUR DE RECHERCHE DU WEB



Photo prise par un appareil photographique
équipé d'une lentille fisheye.

Le chapitre précédent a posé les fondements communs de conception et de réalisation logicielle aux trois systèmes développés. L'objectif de nos réalisations logicielles est l'application et la vérification de nos résultats conceptuels. Chaque système est décrit dans un chapitre dédié. Aussi ce chapitre est consacré à notre premier système : VITESSE.

Après avoir expliqué la finalité du système VITESSE à la section 1, nous présentons la conception de son interface à la section 2 selon les étapes du processus de visualisation. La section 3 expose ensuite les aspects notables de la réalisation logicielle. La section 4 clôt ce chapitre par un bilan de la conception et l'énoncé de perspectives à ce système.

1 Introduction

VITESSE est un acronyme qui signifie "Visualization & Interaction Techniques to Enhance Several Search Engines". Ce système visualise les réponses à une requête soumise à un moteur de recherche sur le World Wide Web. Traditionnellement, ces moteurs de recherche proposent comme interface de sortie un ensemble de pages Web, contenant dix ou vingt paragraphes par page. Chaque paragraphe décrit une page Web sélectionnée par rapport aux mots-clés de la requête. Les pages Web sélectionnées sont classées par ordre de pertinence, la pertinence étant calculée par le système à partir des mots-clés de la requête. Dans chaque paragraphe, l'utilisateur peut lire le titre de la page, un résumé de quelques lignes, la langue ou encore la taille de la page. Le contenu du paragraphe résumant une page Web varie d'un moteur de recherche à l'autre. Néanmoins, ce paragraphe contient nécessairement une balise qui permet de naviguer vers la page décrite.

Avec l'essor du World Wide Web, le besoin de nouveaux outils de navigation est plus important chaque jour. Les outils actuels ne satisfont plus tous les utilisateurs qui naviguent sur le World Wide Web. En effet, bien que la plupart des utilisateurs accèdent aux moteurs de recherche, peu d'entre eux maîtrisent l'interface de ces outils. Les utilisateurs non experts formulent mal leurs requêtes et les réponses qui les intéressent peuvent se trouver au-delà des 50 réponses les plus pertinentes. Notre système VITESSE propose une interface de visualisation permettant à un utilisateur novice d'appréhender et de manipuler de grandes listes de réponses. Notre système peut également concerner un utilisateur expert qui cherche une grande quantité de réponses (par exemple pour établir un état de l'art dans un domaine donné).

2 Conception de l'interface

Nous décrivons la conception de l'interface de VITESSE selon les étapes du processus de visualisation exposés au Chapitre IV. Les données de départ sont définies par le moteur de

recherche. Nous détaillons maintenant la conception selon les trois étapes suivantes du processus de visualisation : le point de vue sur les données est présenté au paragraphe 2.1, l'espace construit est expliqué au paragraphe 2.2 et enfin le point de vue sur l'espace est exposé au paragraphe 2.3. Pour chacune des étapes, nous nous basons sur les espaces de caractérisation établis au Chapitre IV.

2.1 *Point de vue sur les données*

Avec VITESSE, nous avons adopté un point de vue différent sur les données que ceux des systèmes existants. Le moteur de recherche AltaVista [AltaVista], trie par ordre de pertinence les réponses à la requête. Le point de vue sur les données est donc de type <Non structuré 1D, Ordonné>. Notre approche consiste à considérer d'autres liens entre les réponses que ceux définis par la relation d'ordre de pertinence. Comme chaque réponse est un document HTML, il est lié aux autres documents HTML du Web par des liens de référence. Puisque les réponses présentées abordent un sujet commun, les probabilités de référence entre les pages sont élevées. En choisissant de considérer les liens entre les réponses (ceux qui lient deux réponses de la liste), nous adoptons un autre point de vue sur les données de type <Graphe, Ordonné>. Dans certains cas (requête vague ou sur un sujet vaste) aucun lien n'est présent entre les réponses et le point de vue adopté correspond alors à celui des interfaces des moteurs de recherche traditionnels.

Dans VITESSE, nous avons également choisi d'enrichir les réponses par de nouveaux attributs, autres que les liens. Ces ajouts ne modifient pas le point de vue sur les données, mais augmentent la quantité de données à rendre perceptibles par l'utilisateur. Comme l'interface des moteurs de recherche, VITESSE présente pour chaque page :

- ▶ son numéro dans la liste triée par pertinence,
- ▶ son adresse URL (<http://...>),
- ▶ son titre (les pages Web ayant presque toujours un titre),
- ▶ un court extrait du contenu (généralement une ligne qui contient un ou plusieurs des mots-clés).

VITESSE présente en plus les informations suivantes :

- ▶ sa disponibilité sur le Web (c'est-à-dire si la page est disponible au jour de la requête),
- ▶ sa date de dernière modification,
- ▶ sa taille en KiloOctets (KO¹),

¹ "KiloBytes (KB) en anglais dans le système"

- le nombre d'apparition de chaque mot clé dans la page,
- une estimation du temps de chargement (c'est-à-dire le temps pour charger la page depuis le serveur de VITESSE).

2.2 *Espace*

Parce que le point de vue sur les données est différent, l'espace dans lequel VITESSE présente les informations est également différent de celui des interfaces des moteurs de recherche traditionnels. Ces systèmes utilisent le langage HTML (Hyper Text Markup Language) pour construire des pages structurées de dix ou vingt paragraphes. Chaque paragraphe correspondant à une réponse, de nombreuses pages sont nécessaires pour rendre accessible une grande liste de réponses. Le moteur de recherche trie les réponses par ordre de pertinence. Une de ces pages est illustrée à la Figure 64, pour le moteur de recherche AltaVista [AltaVista]. Nous pensons en effet qu'elle est typique de beaucoup de moteurs de recherche. La liste de nombres soulignés (1 [2](#) [3](#) [4](#) [5](#) [6](#) etc.), en bas de la page, permet d'accéder aux autres pages par navigation dans l'hypertexte décrit en HTML.

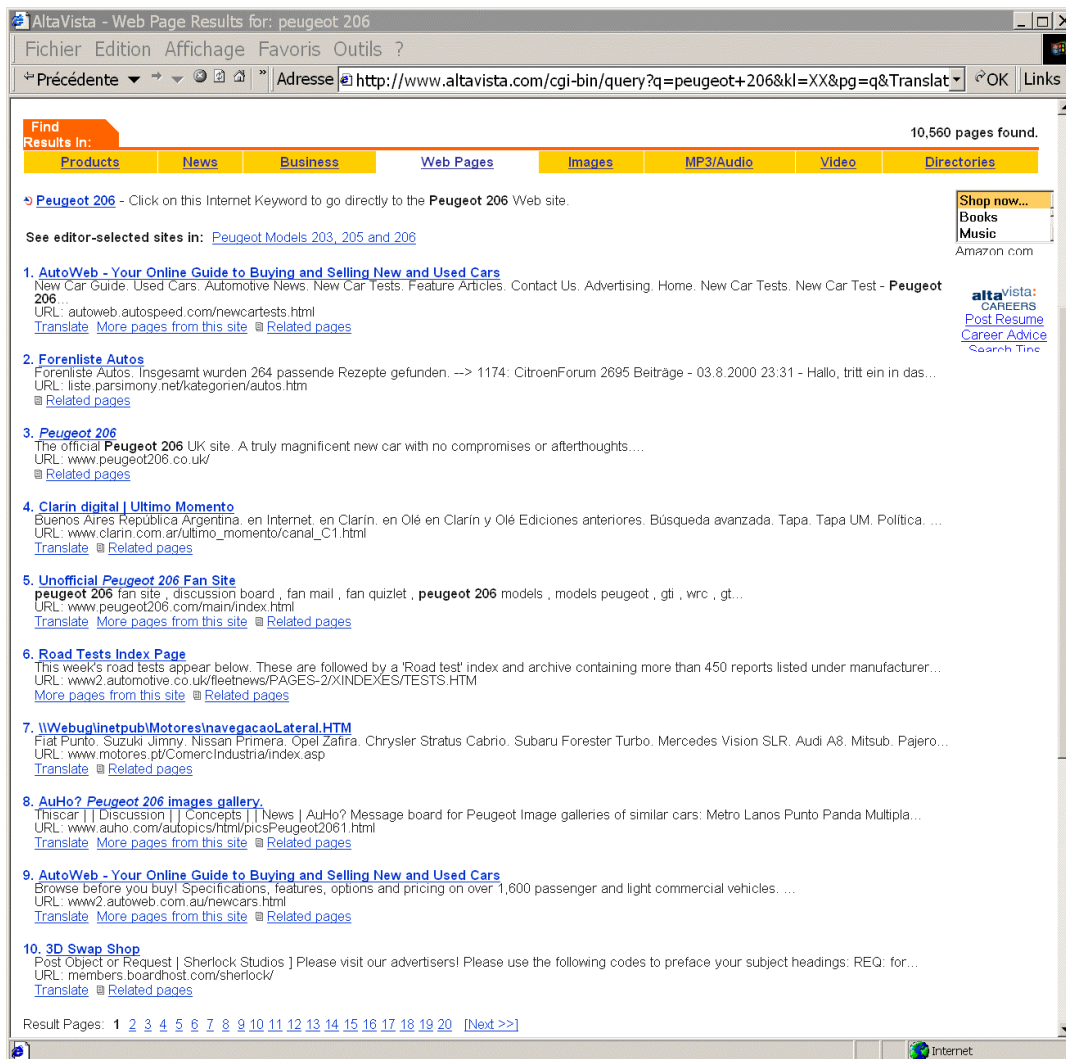


Figure 64 : Interface du moteur de recherche AltaVista [AltaVista].

L'espace de VITESSE, présenté à la Figure 65, diffère de celui d'AltaVista (Figure 64), car toutes les réponses sont regroupées dans un seul espace. Chaque page du moteur de recherche correspond à une ligne de rectangles dans l'espace de VITESSE. Ainsi chaque rectangle contenu dans une ligne correspond à un élément de réponse. Le texte contenu dans chaque rectangle correspond au paragraphe dans AltaVista enrichi des informations présentés au paragraphe précédent. Les liens entre les réponses sont matérialisés par des triangles bleus. Ces triangles apparaissent derrière les rectangles, sauf si les derniers appartiennent au focus de l'utilisateur. A la Figure 65, aucun élément n'est dans le focus. Nous définirons précisément le focus au paragraphe suivant qui traite du point de vue sur l'espace.

Selon les caractérisations de l'espace présentées au paragraphe 3.2 du chapitre IV (page 110), l'espace de VITESSE est décrit par les deux dimensions de l'espace (2D) et l'utilisation des traits graphiques suivants :

- Position : la position dans l'espace offre une représentation de la pertinence calculée par le moteur de recherche.
- Lien : les liens représentés par des triangles bleus rendent observables les liens HTML entre les réponses.
- Couleur : le blanc correspond à une réponse non encore analysée, le noir à une réponse indisponible sur le Web et le vert à une réponse analysée. La couleur verte est vive pour une réponse récente et pastel pour une réponse ancienne.

L'utilisation de ces traits graphiques respecte le patron de conception N° 7 sur le plaquage des données en traits graphiques car de nombreuses informations sont déjà présentées par une modalité linguistique. Nous remarquons à la Figure 65 que la partie droite est surtout composée de réponses anciennes. Ce motif peut inciter l'utilisateur à commencer sa recherche dans la partie gauche.



Figure 65 : Espace des réponses dans VITESSE.

L'espace de VITESSE peut être enrichi d'une information supplémentaire que constitue les opinions exprimées par d'autres utilisateurs à propos d'un résultat. Comme l'illustre la Figure 67, le fond vert de chaque élément peut être décoré d'un histogramme présentant les opinions favorables et défavorables qui ont été exprimées par d'autres utilisateurs à propos de cet élément. Cet histogramme apparaît sous l'effet d'une lentille magique identique à celles décrites dans [Bier 93] (voir Figure 13, page 64). Afin de comprendre l'histogramme, l'utilisateur dispose d'une légende rappelant que les opinions favorables sont montrées à droite de l'histogramme et les opinions défavorables à gauche. L'opinion des autres utilisateurs est donc

plutôt favorable pour l'élément N°1 de la Figure 67. Les opinions des utilisateurs sont présentées selon une modalité non linguistique (histogramme). Le trait graphique utilisé pour rendre observable les opinions est la taille (plus une barre de l'histogramme est grande et plus il y a eu d'utilisateurs qui partagent cette opinion). L'utilisation de ce trait graphique évite une ligne supplémentaire de texte pour chaque élément. Le patron N°7 nous a donc guidé pour ce choix de modalité. La composition entre cette modalité non linguistique et la modalité linguistique qui véhicule les informations sur un élément (texte) est caractérisée comme suit :

- imbriquée pour l'aspect spatial,
- différente pour l'aspect syntaxique,
- complémentaire pour l'aspect sémantique.

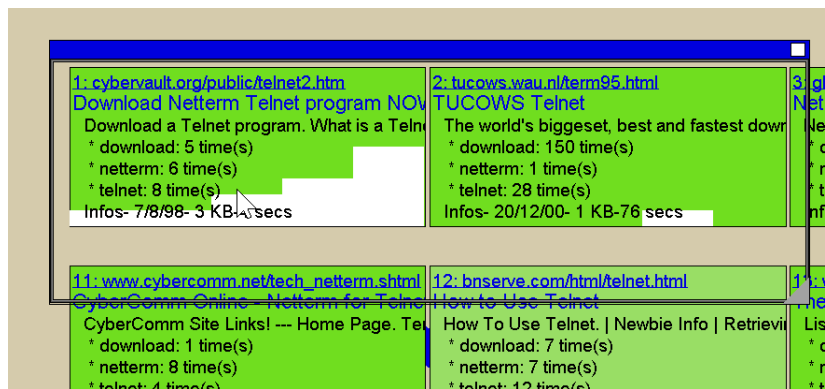


Figure 67 : Histogramme d'opinion en blanc, apparaissant sous l'effet d'une lentille magique.

La continuité spatiale (patron N°5) est respectée entre la modalité graphique non linguistique responsable de la présentation des éléments en rectangles de couleur, M1, la modalité linguistique véhiculant les informations sur un élément, M2, et la modalité graphique non linguistique assignée à la présentation des opinions, M3. Tandis que M2 et M3 sont complémentaires, il est intéressant de noter une composition partiellement redondante entre M1 et M2. En effet, la position du rectangle traduit sa pertinence, qui est aussi présentée par le chiffre au début du texte contenu dans chaque rectangle. Enfin les deux modalités M2 et M3 sont passives tandis que M1 est active : le déplacement de l'espace se fait en cliquant-glissant sur un rectangle. Une action de navigation dans l'espace se termine par un recalage automatique sur une grille. Le système replace au centre de l'interface le rectangle le plus proche du centre. De plus l'utilisateur peut double-cliquer dans un rectangle pour visualiser la page Web.

2.3 *Point de vue sur l'espace*

Comme l'illustre la petitesse du texte dans la Figure 65, l'espace construit à l'étape précédente ne peut pas être affiché dans sa totalité avec une taille de texte suffisante, comme celle d'AltaVista à la Figure 64 ou celle présentée à la Figure 67. Le point de vue sur l'espace ne peut pas être global et précis pour un écran d'ordinateur, même de très haute résolution. Aussi, VITESSE intègre un système de vues multiples du même espace selon des points de vue différents (Chapitre IV, paragraphe 2.1, page 103). VITESSE utilise le paramètre R de la fonction de déformation (technique décrite au paragraphe 2.5 du chapitre V, page 152).

Chaque vue occupe toute la place que l'utilisateur lui réserve en retaillant la fenêtre de VITESSE. La Figure 69 illustre l'une de ces vues avec le navigateur Web en arrière plan pour visualiser les pages Web. L'utilisateur peut changer de vue en sélectionnant l'un des onglets, mis en évidence à la Figure 69. La composition de ces vues, qui ne fait pas intervenir une composition de dispositifs (c'est-à-dire l'aspect articulatoire), est la suivante :

- ▶ séquentielle pour l'aspect temporel,
- ▶ recouvrante pour l'aspect spatial,
- ▶ divergente pour l'aspect syntaxique,
- ▶ totalement redondante pour l'aspect sémantique.

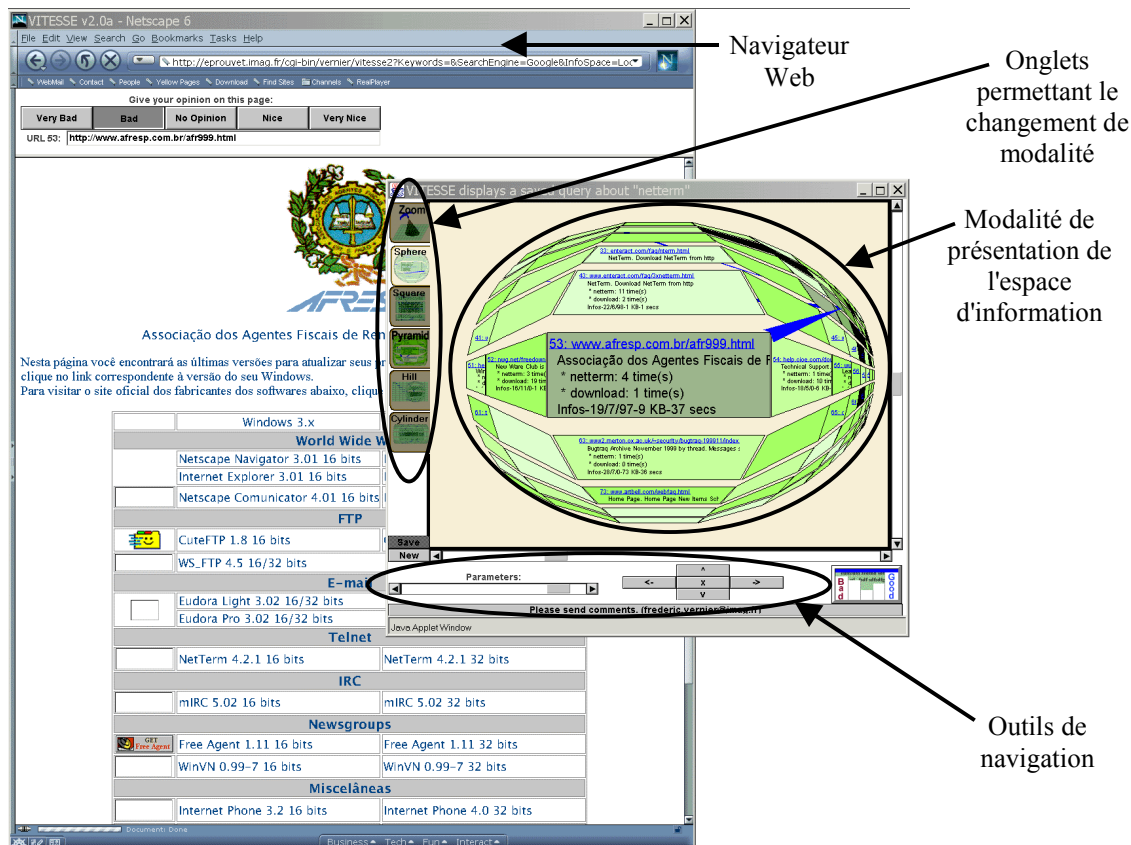


Figure 69 : Interface de VITESSE affichant un des points de vue sur l'espace (Sph  re). Les onglets permettent de changer ce point de vue. Les   l  ments verts sombres correspondent    des pages Web d  j visit  es.

La gliss  re qui se trouve en bas,    gauche de l'interface, est un outil de navigation commun    toutes les vues. Il permet de choisir la taille des informations dans le focus, quelle que soit la vue. Cet outil de navigation respecte donc le patron N  3 sur l'adaptabilit   des modalit  s de sortie et le patron N  9 sur la r  utilisation des outils de navigation.

Dans les paragraphes suivants (paragraphes 2.3.1    2.3.6), nous pr  sentons les six vues propos  es dans VITESSE. Les vues   tant d  crites, nous exposons ensuite la technique de fondu, qui permet un passage continu d'une vue    l'autre, au paragraphe 2.3.7. Enfin, nous terminons avec les outils de navigation dans l'espace au paragraphe 2.3.8.

2.3.1 Vue    diff  rents facteurs de zoom

La vue    diff  rents facteurs de zoom (ou vue en zoom) est un point de vue tronqu   et agrandi de l'espace selon un facteur d'  chelle d  fini par l'utilisateur. Comme l'illustre la Figure 70, ce type d'interface, appel   Interface Utilisateur Zoomable (ZUI en anglais), permet d'obtenir tour    tour une vue globale mais vague de l'espace, puis une vue pr  cise mais locale. La

composition temporelle est séquentielle et requiert un outil de navigation. Les modalités d'entrée pour changer le facteur de zoom nécessitent le clavier ou la souris. L'utilisateur a deux solutions : soit les touches "+" et "-" du clavier soit la glissière en bas à gauche de l'interface. Dans les deux cas, il s'agit d'un outil de navigation pas à pas : le passage d'un point de vue global à un point de vue local ne se fait pas directement mais par des facteurs de zoom intermédiaires.

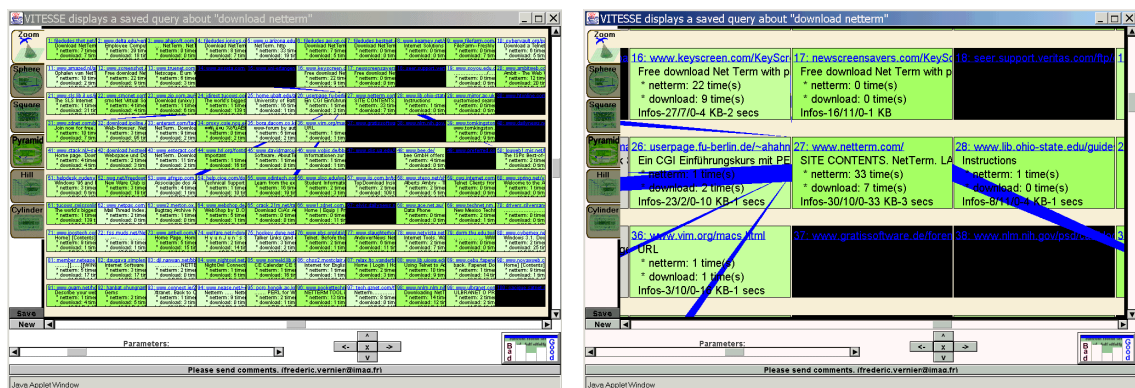


Figure 70 : Vue à différents facteurs de zoom, globale et vague (à gauche), puis précise et globale (à droite).

2.3.2 Vue en Sphère

La vue en sphère est une vue déformée qui utilise la déformation des coordonnées polaires que nous avons décrite au paragraphe 2.2 du chapitre V (page 145).

Comme nous l'avons évoqué au paragraphe 2.2, cette vue s'ajuste automatiquement sur l'élément le plus proche du centre afin de vérifier la non-déformation du focus. Par exemple, à la Figure 71, l'élément 27 est centré afin d'être présenté sans déformation de perspective. Cette fonction est présente pour toutes les vues autres que la vue à différents facteurs de zoom.

Comme l'illustre la Figure 71, la vue en sphère utilise le paramètre R pour mieux occuper une surface rectangulaire. Cette utilisation se fait automatiquement par le système en fonction de la taille de la fenêtre choisie par l'utilisateur.

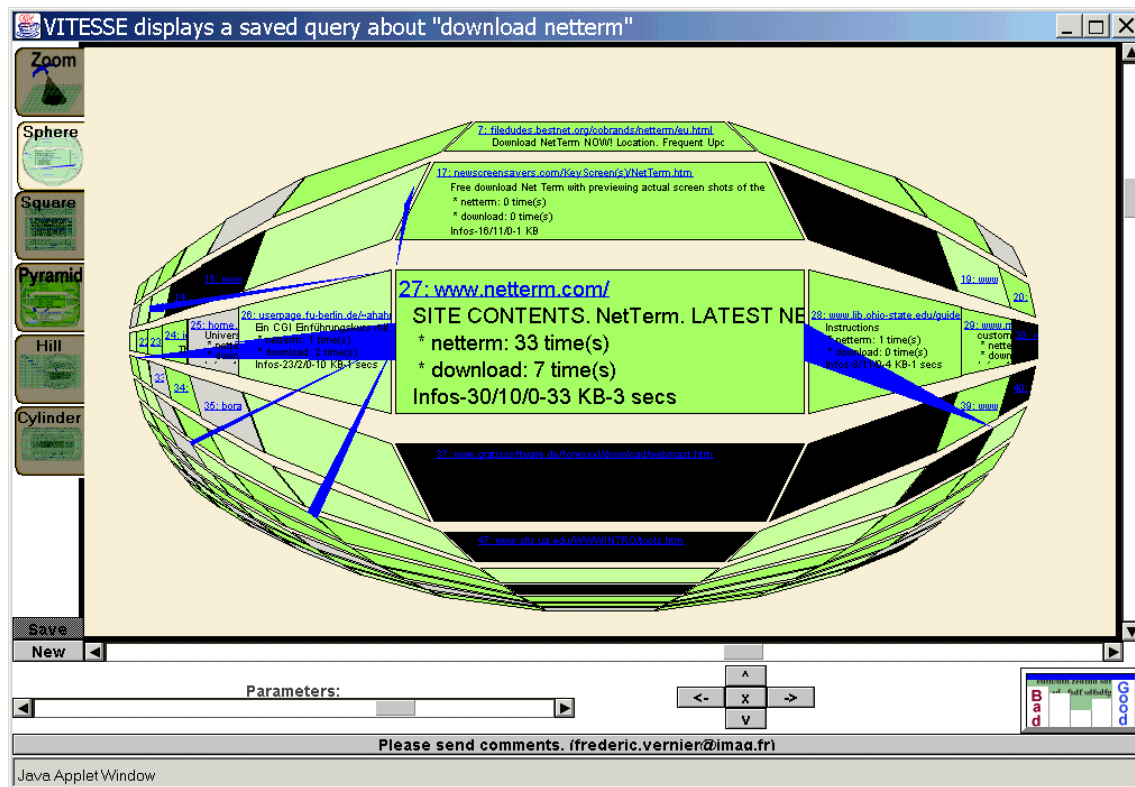


Figure 71 : Vue en sphère.

Enfin, nous avons lié le coefficient d'aplanissement de la partie centrale (c'est-à-dire le focus), présenté au paragraphe 2.3.2. du chapitre V (page 149), avec la glissière en bas à gauche, afin que cette dernière permette à l'utilisateur de grossir ou diminuer la taille de l'information présente dans la partie centrale.

En considérant notre espace de conception présenté au paragraphe 3.3.1 du chapitre IV (page 114), nous décrivons cette vue comme une composition des modalités suivantes :

- une modalité précise, locale et non déformée au centre de la vue,
- une modalité vague, globale et déformée autour de cette partie centrale.

La partie déformée est décrite par les aspects suivants :

- La déformation est continue, et sa dérivée aussi.
- Les alignements ne sont pas conservés.
- L'effet de perspective est présent.

La composition de la modalité du contexte (forme de sphère) et de la partie centrale aplanie crée une forme de sphère tronquée. Nous considérons cette sphère tronquée comme une

modalité composée, sa déformation n'étant pas continue. En effet, la dérivée de la fonction de déformation de cette modalité n'est pas continue à cause de la composition. L'utilisateur peut percevoir (selon la valeur du coefficient d'aplanissement) deux zones dans son interface de visualisation.

Nous remarquons que le patron N°5 sur la continuité spatiale est vérifié dans cette vue, ce patron ne préconisant une dérivée de la fonction de déformation continue. Le patron N°8, sur la compréhension de la déformation, est vérifié, car la déformation des rectangles se traduit par une forme composée de facettes facile à comprendre et à manipuler comme une sphère.

Deux points négatifs sont à noter dans la vue en sphère. Tout d'abord une part non négligeable de l'espace est perdue dans chaque angle. Nous constatons aussi que lorsque la sphère n'est pas tronquée, les voisins directs du focus sont très déformés et se prêtent mal à accueillir une information selon une modalité aussi contraignante que du texte.

2.3.3 *Vue cartésienne*

La vue cartésienne de VITESSE présentée à la Figure 72, correspond à la déformation cartésienne présentée au paragraphe 2.2 du chapitre précédent (page 145). Comme pour la vue en sphère, le paramètre manipulé par la glissière permet d'aplanir la partie centrale de la déformation. Il permet donc également de réduire ou d'agrandir les informations dans le focus.

Toujours en considérant notre espace de conception du chapitre IV (paragraphe 3.3.1, chapitre IV, page 114), nous décrivons cette vue comme une composition des modalités suivantes :

- ▶ une modalité précise, locale et non déformée au centre de la vue,
- ▶ une modalité vague, globale et déformée autour de cette partie centrale.

La partie déformée est décrite par les aspects suivants :

- ▶ La déformation est continue et sa dérivée aussi.
- ▶ Les alignements des lignes et des colonnes sont conservés.
- ▶ Il n'y a pas d'effet de perspective.

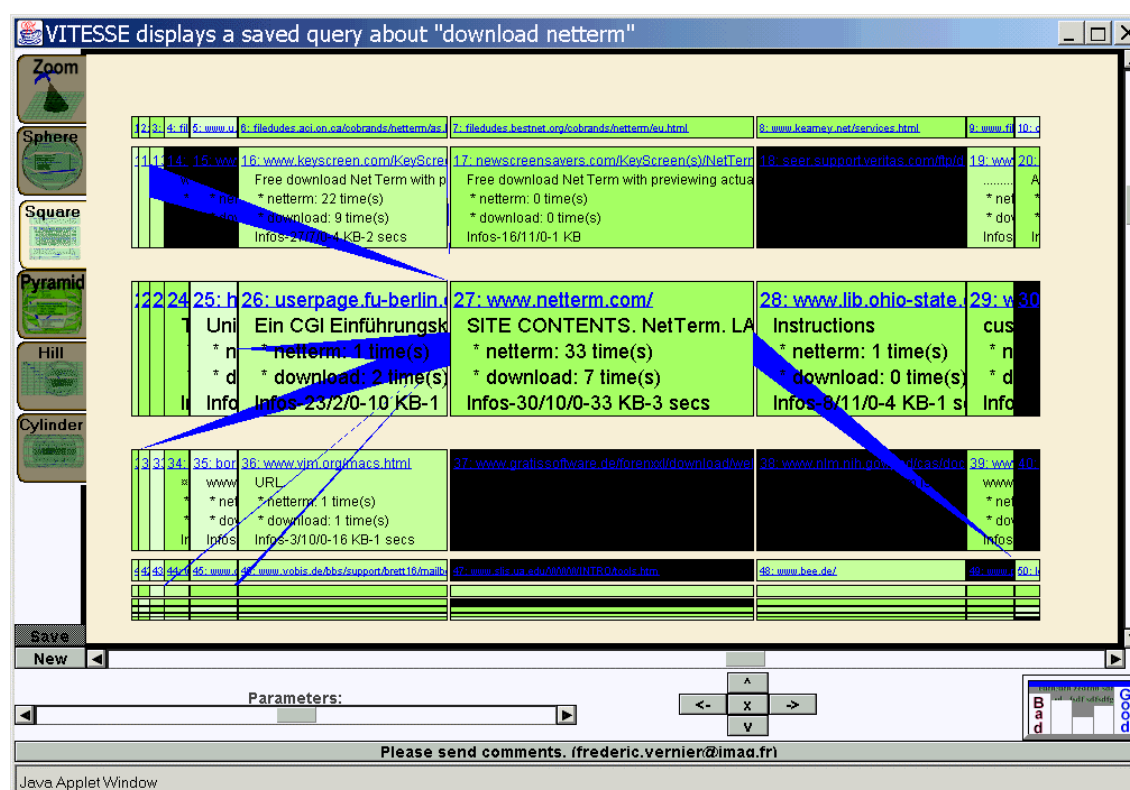


Figure 72 : Vue cartésienne.

Pour les mêmes raisons que la vue précédente, le patron N°5 sur la continuité spatiale est respecté. Cependant, le patron N°8 sur la compréhension de la déformation n'est absolument pas vérifié, néanmoins la vue cartésienne conserve tous les alignements.

2.3.4 Vue en pyramide

La vue en pyramide de VITESSE présentée à la Figure 74 correspond à la déformation détaillée au paragraphe 2.3.1. du chapitre précédent (page 147). Comme pour les deux vues précédentes (vue en sphère et cartésienne), le paramètre manipulé par la glissière permet de réduire ou d'agrandir les informations dans le focus. Le paramètre manipulé est là encore le coefficient permettant d'aplanir la partie centrale de la déformation.

Selon notre espace de conception (paragraphe 3.3.1, chapitre IV, page 114), cette vue est décrite comme une composition des modalités suivantes :

- une modalité précise, locale et non déformée au centre de la vue,
- une modalité vague, globale et déformée autour de cette partie centrale.

La partie déformée est décrite par les aspects suivants :

- La déformation est continue mais la dérivée est discontinue sur les arêtes de la pyramide.
- Les alignements des lignes et des colonnes sont conservés.
- L'effet de perspective est présent.

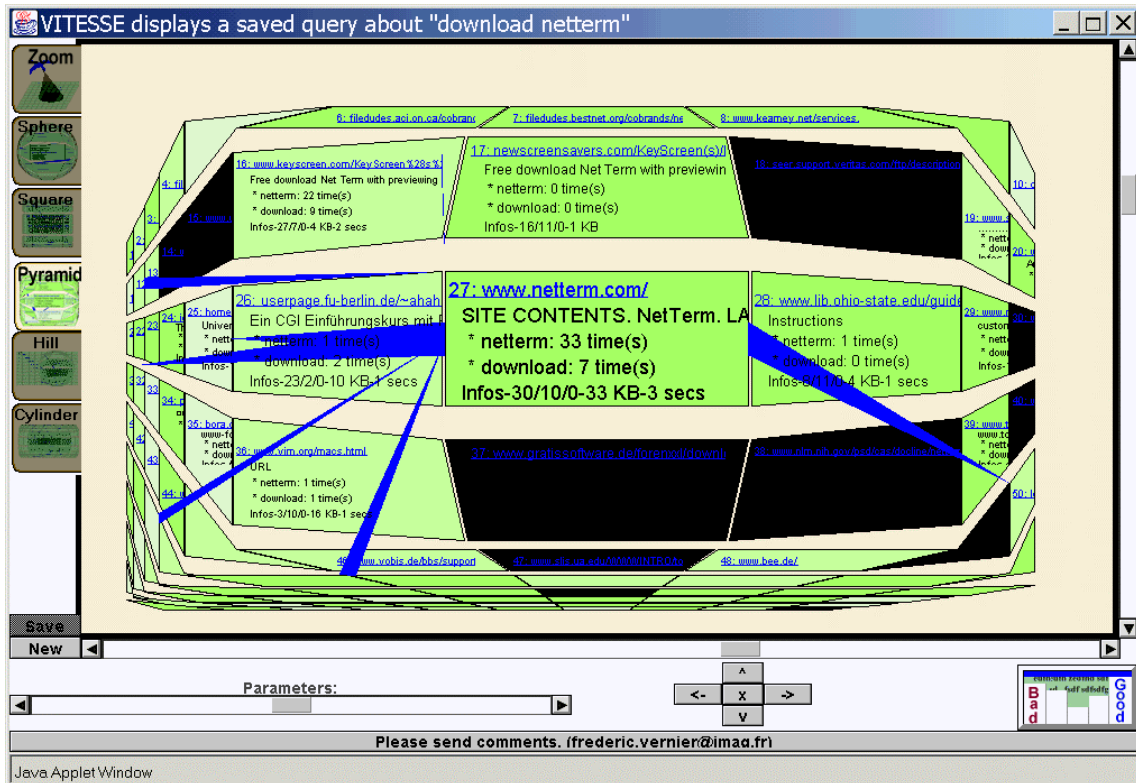


Figure 74 : Vue en pyramide.

2.3.5 Vue en colline

La vue en colline de VITESSE, présentée à la Figure 76, correspond à l'aplanissement de la partie extérieure d'une vue en sphère présentée au paragraphe 2.3.3 du chapitre V (page 150). Comme pour les vues précédentes, le paramètre manipulé par la glissière permet de réduire ou d'agrandir les informations dans le focus. Pour arriver à ce résultat, et contrairement aux autres vues, le paramètre manipulé est le rayon de la sphère (R). Manipuler le coefficient d'aplanissement ne correspondrait pas au but recherché et impliquerait un agrandissement ou une réduction des informations du contexte.

Selon notre espace de conception (paragraphe 3.3.1, chapitre IV, page 114), nous décrivons cette vue comme une composition des modalités suivantes :

- une modalité précise, locale et déformée au centre de la vue,
- une modalité vague, globale et non-déformée autour de cette partie centrale.

La partie déformée est décrite par les aspects suivants :

- La déformation est continue et sa dérivée aussi.
- Les alignements des lignes et des colonnes ne sont pas conservés.
- L'effet de perspective est présent.

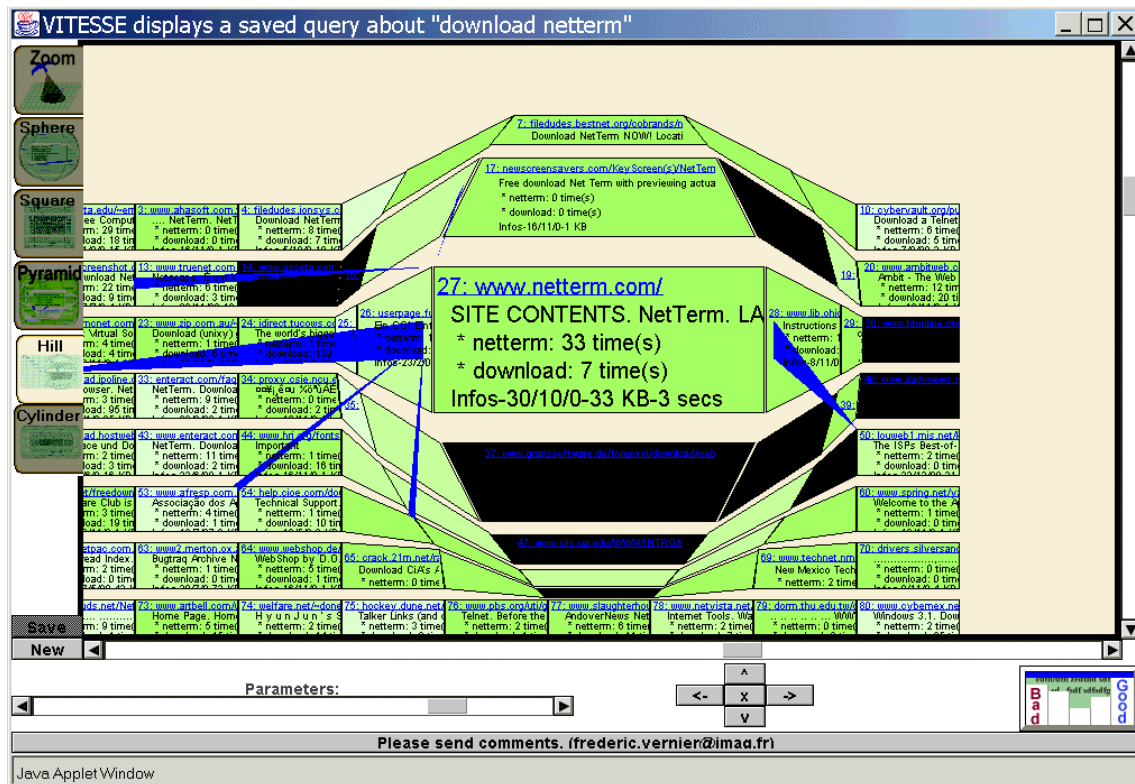


Figure 76 : Vue en colline.

Nous soulignons le fait que, pour cette vue, le contexte n'est pas visible en entier. Le patron N°4 lié à l'observabilité n'est donc pas vérifié. Toutefois, la partie manquante est la plus éloignée du focus et le contexte affiché en partie est le moins déformé de toutes les vues de VITESSE. De plus, cette vue offre une meilleure compréhension des liens que les autres vues où il est difficile de déterminer l'extrémité du lien dans le contexte déformé.

2.3.6 Vue en cylindre

La vue en cylindre de VITESSE, présentée à la Figure 78, correspond à l'application de fonctions de déformation différentes sur chaque axe, que nous avons présentée au paragraphe 2.4 du chapitre précédent (page 152). Comme dans les vues précédentes, le paramètre manipulé par la glissière permet de réduire ou d'agrandir les informations dans le focus. Le paramètre

manipulé est une nouvelle fois le coefficient permettant d'aplanir la partie centrale de la déformation.

Selon notre espace de conception (paragraphe 3.3.1, chapitre IV, page 114), nous décrivons cette vue comme une composition des modalités suivantes :

- une modalité précise, locale et non déformée au centre de la vue,
- une modalité vague, globale et déformée autour de cette partie centrale.

La partie déformée est décrite par les aspects suivants :

- La déformation est continue et sa dérivée aussi.
- L'alignement des lignes est conservé, mais pas celui des colonnes.
- L'effet de perspective est présent.

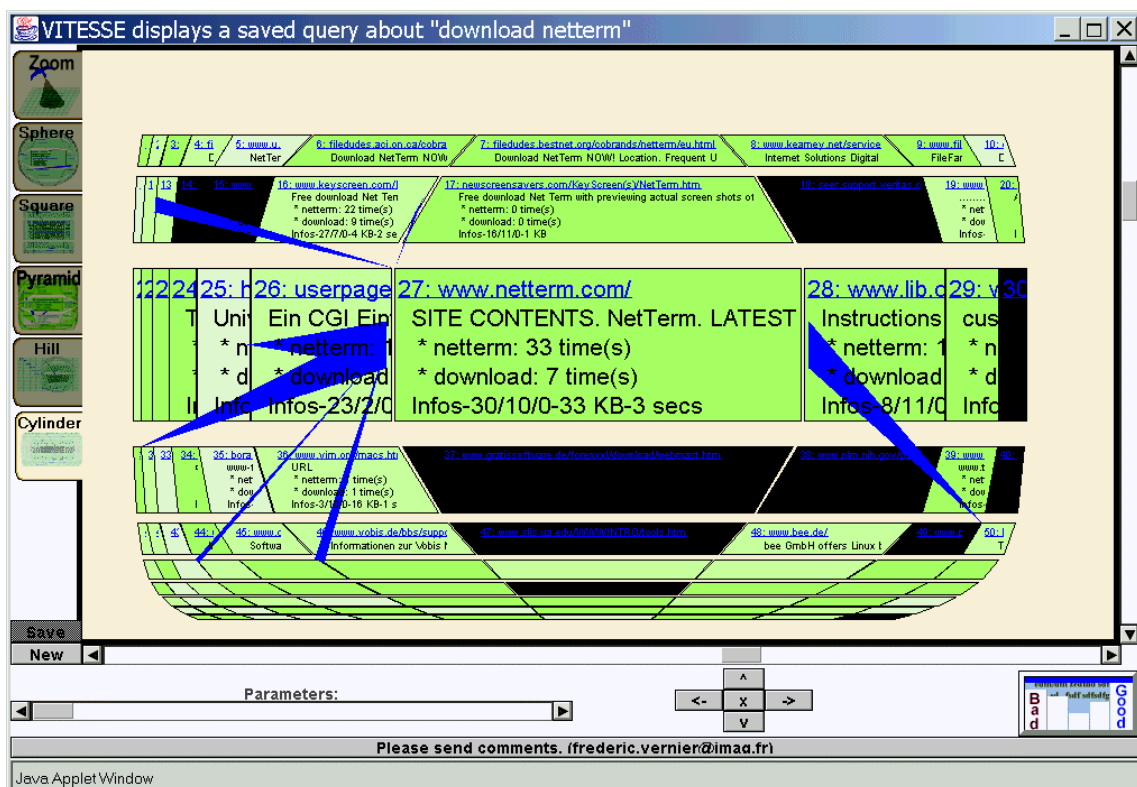


Figure 78 : Vue en cylindre de l'interface de visualisation de VITESSE.

Nous remarquons que les déformations sont moindres dans cette vue car les bords inférieurs et supérieurs de chaque polygone sont maintenus horizontaux.

2.3.7 *Passage d'une vue à l'autre*

Le passage d'une vue à l'autre dans VITESSE utilise une technique de déformation progressive de l'espace par le calcul de formes hybrides entre la vue de départ et la vue d'arrivée. Les formes hybrides calculées correspondent à celle décrite au paragraphe 2.3.4 du chapitre précédent (page 151).

Les couples de poids employés, pour une animation en neuf étapes, sont : 100%-0%, 90%-10%, 80%-20%, 70%-30%, 60%-40%, 50%-50%, 40%-60%, 30%-70%, 20%-80%, 10%, 90% et 0%-100%. Le premier et le dernier couple de poids ne correspondent pas à une forme hybride. Ce mécanisme de transition vérifie le patron N°6 sur la continuité temporelle.

2.3.8 *Outils de navigation pour le déplacement du focus dans l'espace*

Outre la glissière permettant de changer la taille des informations dans le focus (outil de navigation commun à chaque vue), VITESSE propose des outils de navigation pour déplacer le focus. Comme l'illustre la Figure 80, le focus peut être déplacé par manipulation directe en sélectionnant un endroit de l'espace et en le déplaçant. Le déplacement de la souris fait exactement coïncider le pointeur de la souris avec un point de l'espace. Dans la Figure 80, c'est le bas du rectangle 25 qui est sélectionné et déplacé. Pour les vues déformées, nous utilisons la fonction inverse de déformation pour calculer le mouvement du appuyer-glisser-relâcher dans l'espace plan. Les repères de la souris et de l'espace d'information sont donc liés par cette action de navigation. L'utilisateur perçoit ses actions comme une manipulation directe de tout l'espace d'information.

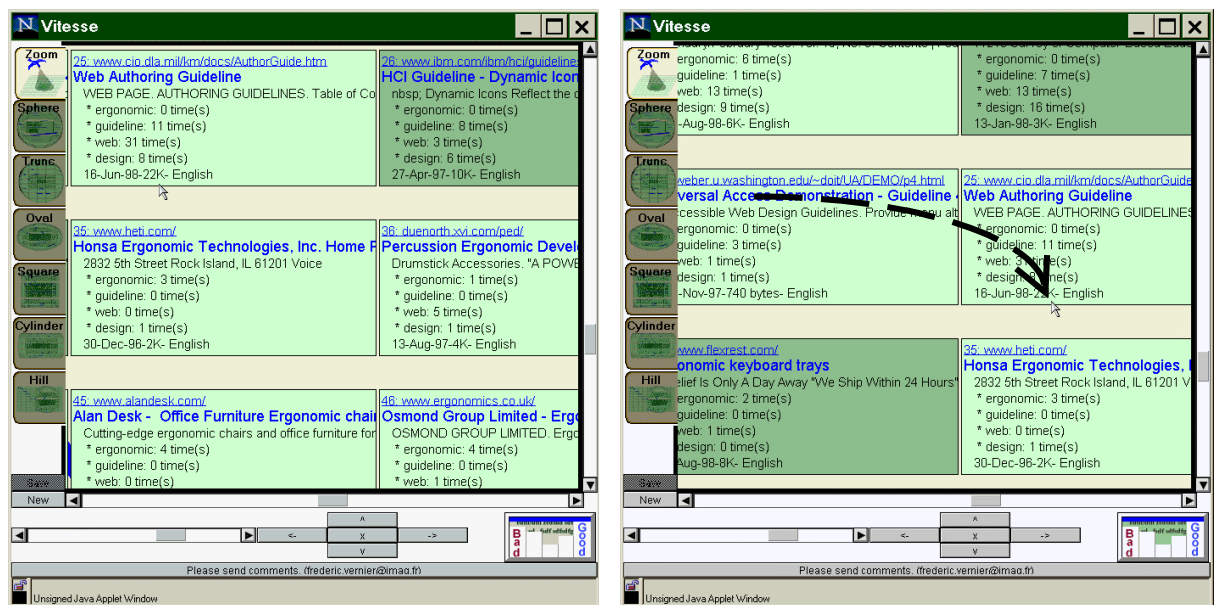


Figure 80 : Le système VITESSE avant et après un glisser-déposer (matérialisé par une flèche en pointillés).

La navigation peut également se faire à l'aide du clavier ou des barres de défilement. Pour ces outils, nous avons respecté le patron N°9 sur la réutilisation des outils de navigation. Ils sont communs à toutes les vues. En effet, la navigation avec les barres de défilement ou avec les flèches du clavier nous semble être conforme à la navigation dans de nombreux autres systèmes. Ces outils de navigation fonctionnent pas à pas, sont fortement couplés avec l'espace mais la manipulation est indirecte.

Nous proposons également un autre outil de navigation, plus original. Le dispositif physique de la modalité d'entrée associée à cet outil de navigation est le clavier. Le langage d'interaction de cette modalité est une analogie entre l'espace 2D des informations et l'espace physique du clavier. En effet, un clavier est un dispositif occupant un espace rectangulaire (d'environ 6cm sur 30cm). Dans cet espace l'emplacement des touches est fixe pour le clavier d'un pays donné. Pour la France on parle de clavier azerty et pour les USA de clavier qwerty. Prenons l'exemple du clavier français pour remarquer que la touche "Esc" est toujours en haut à gauche du clavier, la touche "q" est à gauche au milieu, la touche espace en bas au milieu. La touche "h" est à peu près au milieu du clavier et ainsi de suite. La navigation proposée consiste donc à utiliser les touches du clavier comme un raccourci de navigation vers une cinquantaine de points particuliers de l'espace. Chaque touche est associée au point de l'espace qui correspond à l'emplacement de la touche sur le clavier. La touche "Esc" permet ainsi à l'utilisateur de revenir à n'importe quel moment et en une seule action au coin en haut à gauche de l'espace. La touche "h" permet de se replacer à tout moment au centre l'espace et ainsi de

suite. Cet outil ne permet pas d'atteindre tous les éléments de l'espace car il y a plus d'éléments que de touches. Toutefois, la navigation à l'aide des flèches du clavier permet d'affiner cette navigation sans avoir à changer de dispositif (opérateur "Homing" dans la méthode d'évaluation prédictive et quantitative GOMS du niveau KEYSTROKE). Si un élément se trouve entre celui associé à la touche "e" et celui associé à la touche "r", il est possible d'y accéder par la combinaison de touche "e"+"→", "r"+"←".

3 *Réalisation logicielle*

Dans les deux paragraphes suivants, nous présentons les aspects de réalisation logicielle de l'interface présentée à la section précédente. En nous reposant sur notre solution générique de réalisation logicielle basée sur le modèle d'architecture et présentée à la section 3 du chapitre précédent (page 153), nous décrivons dans un premier temps, l'architecture logicielle (paragraphe 3.1) avant de présenter dans un second temps, les détails d'implémentation spécifiques à VITESSE (paragraphe 3.2).

3.1 *Architecture logicielle*

Nous détaillons chaque composant logiciel de VITESSE. Ces composants sont définis par le modèle PAC-AMODEUS.

Aucune programmation n'a été nécessaire pour le composant CIBN qui contient les composants de bas niveau de l'environnement d'applet JAVA.

Le CTP de VITESSE constitue une interface logicielle qui rend le CD indépendant des interacteurs de l'environnement d'applet JAVA. Il définit une collection d'interacteurs abstraits dans un ensemble de modules séparés. Les interacteurs implémentés pour VITESSE sont classiques : boutons, barre de défilement, etc.

Les composants NF et ANF de VITESSE permettent l'éclatement du système en deux parties : le côté client et le côté serveur. L'ANF, le CD et le PTC sont du côté client et le NF du côté serveur. Les échanges entre le client et le serveur s'effectuent donc entre le NF et l'ANF. Cette distribution est donc transparente pour le CD. L'ANF fonctionne sur un fil d'activité (Thread en anglais) différent de celui du CD. Les échanges sont donc asynchrones entre le CD et l'ANF. Cet asynchronisme permet de rendre indépendante la réactivité du CD, donc de l'interaction de celle de l'ANF. C'est un aspect important car la réactivité de l'ANF dépend du réseau : interruption de service, bande passante variable, latence perceptible, etc.

Le NF de VITESSE est composé de nombreux composants. En effet, le NF de VITESSE n'est pas uniquement un moteur de recherche du Web mais un ensemble de composants dont certains communiquent avec les moteurs de recherche. Les services fournis par le NF vont au-delà de ceux proposés par un moteur de recherche traditionnel. Le NF maintient l'espace résultat d'une requête pour sa restitution ultérieure sans avoir à formuler de nouveau la requête. Cette méthode permet un accès plus rapide à l'information quand il s'agit de retrouver une page Web déjà visitée lors d'une utilisation précédente de VITESSE. Le NF gère aussi le stockage des opinions exprimées sur les pages visitées. L'utilisateur donne son opinion grâce à la palette d'opinions qui sera présentée à la Figure 83 (page 186). Comme le montre la Figure 69 (page 172), cette palette est positionnée au dessus des pages visitées. Ces opinions sont stockées anonymement sur le serveur afin de produire les histogrammes d'opinions, présentés à la Figure 67 (page 170). Enfin le NF visite toutes les réponses retournées par le moteur de recherche afin de calculer certaines des informations présentées dans l'interface de VITESSE :

- ▶ l'occurrence des mots-clés dans la page Web,
- ▶ la date de dernière modification du fichier (qui peut-être plus récente que celle connue par le moteur de recherche),
- ▶ la taille de la page Web,
- ▶ le temps de chargement de la page Web au moment du démarrage de VITESSE.

Composant principal de l'interface, le CD est structuré en une hiérarchie d'agents PAC que nous présentons à la Figure 74 et à la Figure 75. Les noms des fichiers correspondants sont notés à côté de chaque facette. La première lettre du fichier indique le type de facette (Abstraction, Contrôle, ou Présentation).

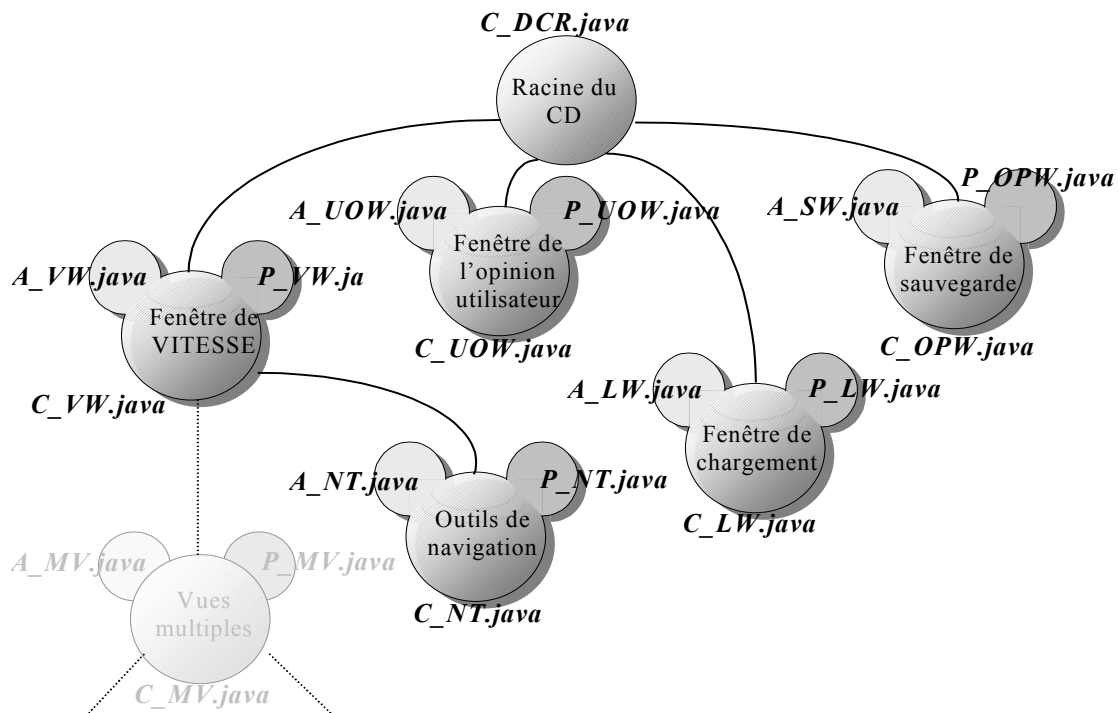


Figure 74 : Partie supérieure de l'architecture PAC du Contrôleur de Dialogue de VITESSE.

* **Attention**, Un seul des agents est présent en même temps.

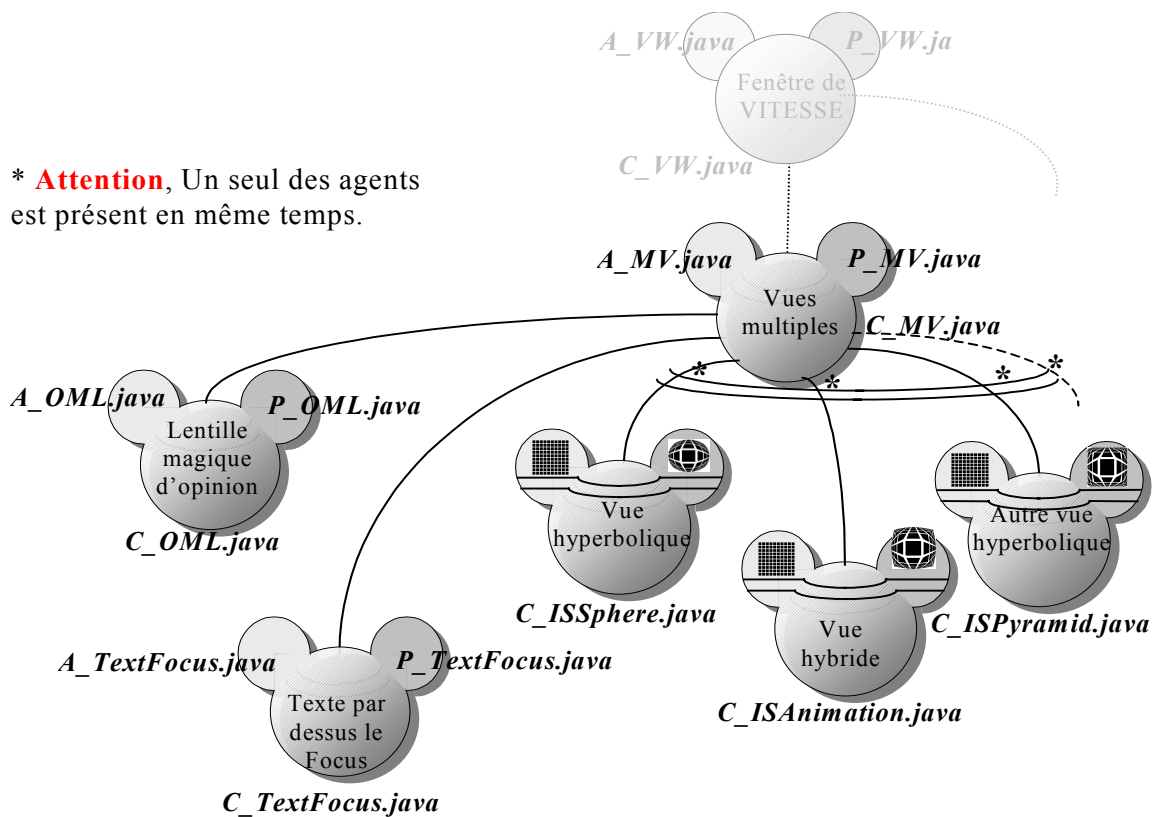


Figure 75 : Partie inférieure de la hiérarchie d'agents PAC du Contrôleur de Dialogue de VITESSE. Seule est illustrée la présence des agents "vue en sphère" et "vue en pyramide".

Nous présentons à la Figure 83 à Figure 82 la facette Présentation des agents, excepté les différentes vues hyperboliques et à facteurs de zoom déjà présentées à la section précédente.

Nous avons conçu un agent par vue (hyperbolique ou à facteurs de zoom) avec un agent père ciment noté "Vues multiples" comme expliqué dans notre solution générique de réalisation logicielle au Chapitre V. Il est à noter que les agents dédiés à une des vues ne sont pas présents en même temps dans la hiérarchie puisque leur composition est séquentielle. Aussi, contrairement à la solution décrite au Chapitre V pour des vues multiples parallèles, l'espace est maintenu par la facette Abstraction de l'agent "Vue" et non dans l'abstraction de l'agent "Vues multiples". Lorsqu'un de ces agents est dynamiquement détruit pour faire place à un autre (changement de vue), ses facettes Présentation et Abstraction sont conservées et attachées au nouvel agent. Ces facettes sont décrites dans les deux fichiers respectivement intitulés *P_ISDeformable* et *A_ISDeformable*. Seule la facette Contrôle est modifiée. Nous avons défini une facette Contrôle par vue. Les facettes décrites dans les fichiers *P_ISDeformable* et *A_ISDeformable* sont capables de communiquer avec n'importe laquelle des facettes Contrôle écrites dans les fichiers : *C_ISZoom*, *C_ISPyramid*, *C_ISOval*, *C_ISDeformable*, *C_ISCartesian*, *C_ISCylinder*, *C_ISHill* et *C_ISAnimation*. Cette genericité est rendue possible car chacun des objets JAVA qui implémente ces facettes Contrôle hérite d'un objet commun. Cet objet est décrit dans le fichier *C_ISDeformable*.

En résumé, les différentes vues utilisent la même facette Abstraction et présente donc le même espace. Elles utilisent la même facette Présentation et partagent donc le même emplacement du focus. La traduction de l'information depuis la facette Abstraction vers la facette Présentation change et subséquemment la forme du point de vue sur l'espace.



Figure 83 : Réalisation graphique de l'agent "Fenêtre de l'opinion de l'utilisateur".



Figure 77 : Réalisation graphique de l'agent "Outils de navigation".

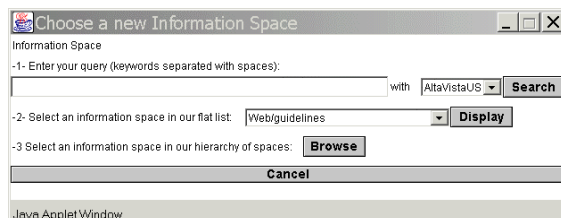


Figure 78 : Réalisation graphique de l'agent "Fenêtre de chargement d'un nouvel espace".

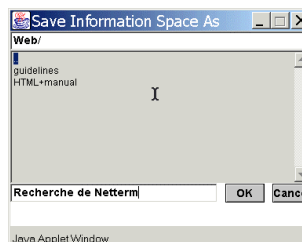


Figure 79 : Réalisation graphique de l'agent "Fenêtre de sauvegarde de l'espace affiché".



Figure 80 : Réalisation graphique de l'agent "Vues multiples".

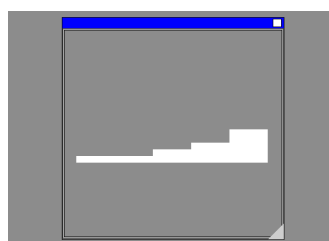


Figure 81 : Réalisation graphique de l'agent "Lentille magique d'opinions".

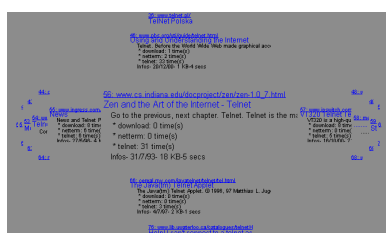


Figure 82 : Réalisation graphique de l'agent "Texte dans le focus".

3.2 Implémentation

Dans VITESSE, nous nous sommes appuyés sur le mécanisme d'applet de JAVA pour créer un système qui s'intègre dans le Web et qui effectue des recherches sur le Web. L'environnement de l'utilisateur est uniforme. Le NF (coté serveur) est également programmé en JAVA afin d'uniformiser la plate-forme de développement. Néanmoins le NF n'utilise pas le mécanisme d'applet de JAVA. Il fonctionne comme un ensemble de scripts CGI invoqués par le

serveur Web Apache [Apache]. Le principal défaut de cette solution est la lenteur lors du démarrage. En effet, les communications séquentielles entre le serveur et le client sont nombreuses. Nous l'illustrons à la Figure 83, en ajoutant une flèche en zigzag pour indiquer les moments d'attente liés à l'exécution des programmes. Outre le lancement du navigateur Web, les autres temps d'exécution correspondent à l'exécution d'un programme en JAVA. Ces exécutions nécessitent le démarrage d'un interpréteur JAVA, même pour l'exécution d'un script simple.

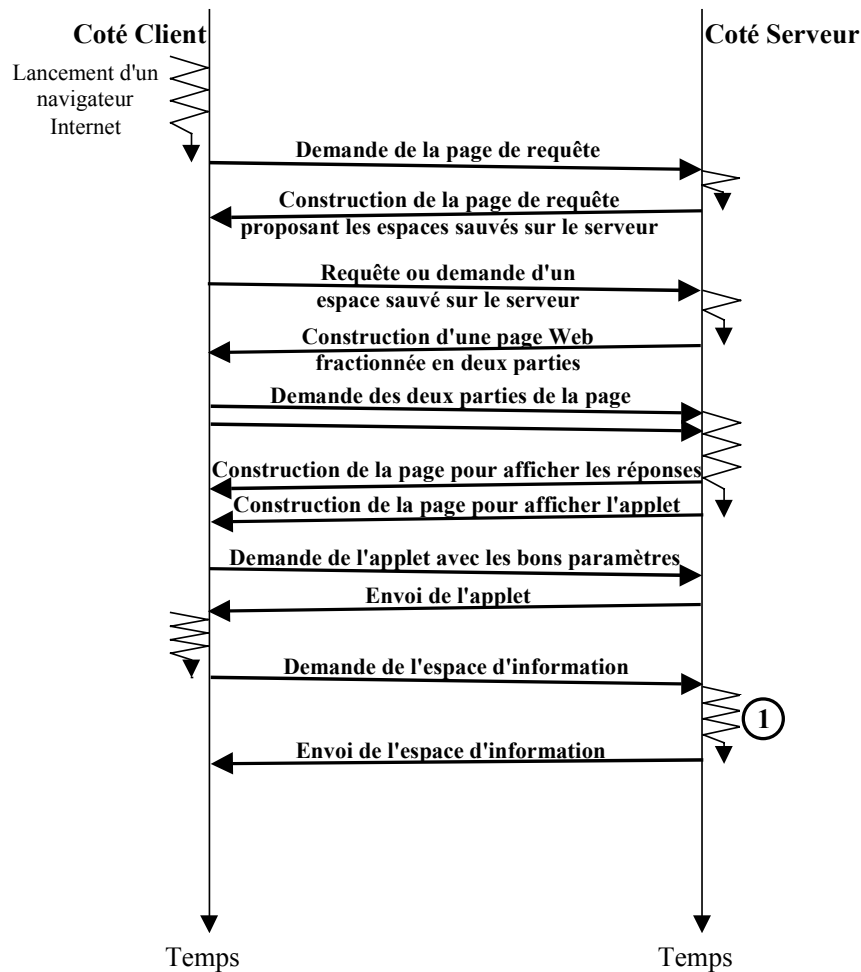


Figure 83 : Communication entre le coté serveur et le coté client de VITESSE.

La pastille ① à la Figure 83 indique que la durée d'exécution du code est très variable. Si l'espace d'information demandé est un espace sauvé sur le serveur, ce temps est assez court, de l'ordre de quelques secondes. Si au contraire la requête doit être sous-traitée à un moteur de recherche et les réponses visitées pour être analysées, le temps d'exécution de ce code peut prendre plusieurs minutes. Néanmoins les données retournées par cette partie du serveur ne sont pas stockées avant d'être envoyées au client, mais envoyées au fur et à mesure qu'elles sont prêtes. L'utilisateur voit donc apparaître progressivement l'espace d'information. Outre le fil

d'activité qui interroge le moteur de recherche, notre serveur lance une dizaine de fils d'activité en parallèle pour visiter les pages proposées par le moteur de recherche.

Les programmes du côté serveur passent par l'interface CGI (Common Gateway Interface). Cette interface permet l'exécution sur le serveur de programmes qui retournent des informations compatibles avec la norme WWW. Pour éviter l'accumulation des données avant l'envoi au client, nous utilisons la norme de script NPH (Non-Parseable Headers) du serveur Apache [Apache].

4 Conclusion

4.1 Bilan

Le système VITESSE visualise les réponses à une requête soumise à un moteur de recherche sur le World Wide Web. La valeur ajoutée par rapport à l'interface d'un moteur de recherche classique réside dans la visualisation de toutes les pages Web retournées dans un seul espace ainsi que dans les techniques de navigation. Notre système a beaucoup évolué au gré des remarques informelles des utilisateurs et suite à deux évaluations ergonomiques qui ont été réalisées.

La première évaluation a été réalisée avec Lisette Garrido, du DESS Ergonomie de l'Université René Descartes. Cette évaluation a permis de tester les différentes vues et leurs appréciations par les utilisateurs. L'expérimentation n'a pas été réalisée avec VITESSE mais avec un autre système appelé VITRIHOL, illustré à la Figure 84a. L'espace d'information était fixé par l'expérimentateur : une liste de description d'hôtels classés par ville et par catégorie. Suite à l'expérimentation, Nous avons remanié les vues proposées et modifié certains outils de navigation. L'ensemble des modifications effectuées sont consignées dans le rapport de Lisette Garrido [Garido 97].

La seconde évaluation a été réalisée avec Julie Bzostek, visiteur scientifique spécialiste en ergonomie. Cette étude a consisté à questionner les utilisateurs du Web sur le Web lui-même, pour mieux connaître leur besoin d'information et leur appréciation des icônes présentes dans VITESSE. A la suite de cette étude, nous avons changé l'ordre des lignes de texte dans chaque élément d'information et reconstruit entièrement les icônes. Nous présentons les résultats de l'enquête sur les informations pertinentes à propos d'une page Web à la Figure 85.

Ces deux évaluations ne prouvent en aucun cas le bien-fondé des patrons de conception qui sont basés sur une approche empirique. Néanmoins elles attestent que l'utilisation de patrons

de conception ne suffit pas pour produire une interface exempte de défaut. Les patrons de conception en eux-mêmes sont très difficilement évaluables, car ils définissent une solution générique à un problème récurrent et les évaluations sont toujours menées sur des cas spécifiques.

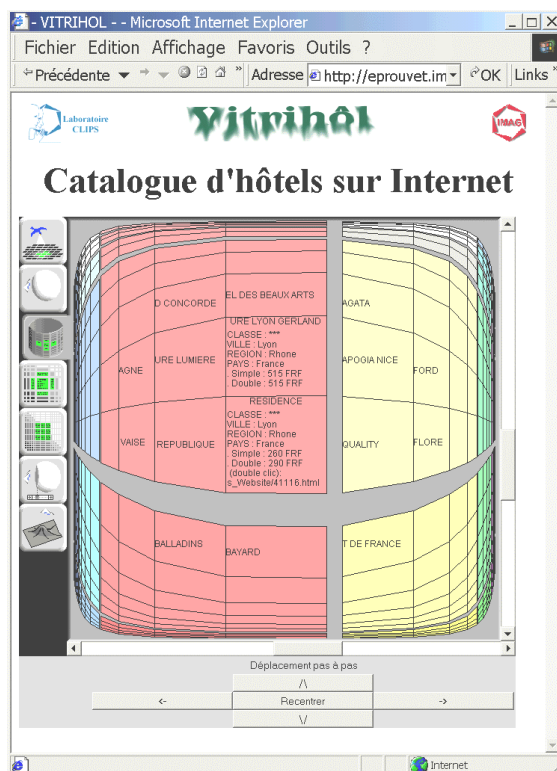


Figure 84a : Le système VITRIHOL.

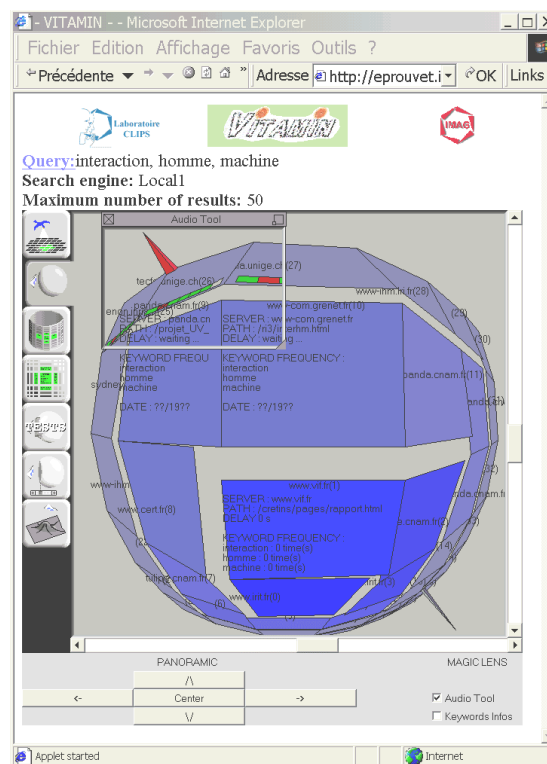


Figure 84b : Le système VITAMIN.

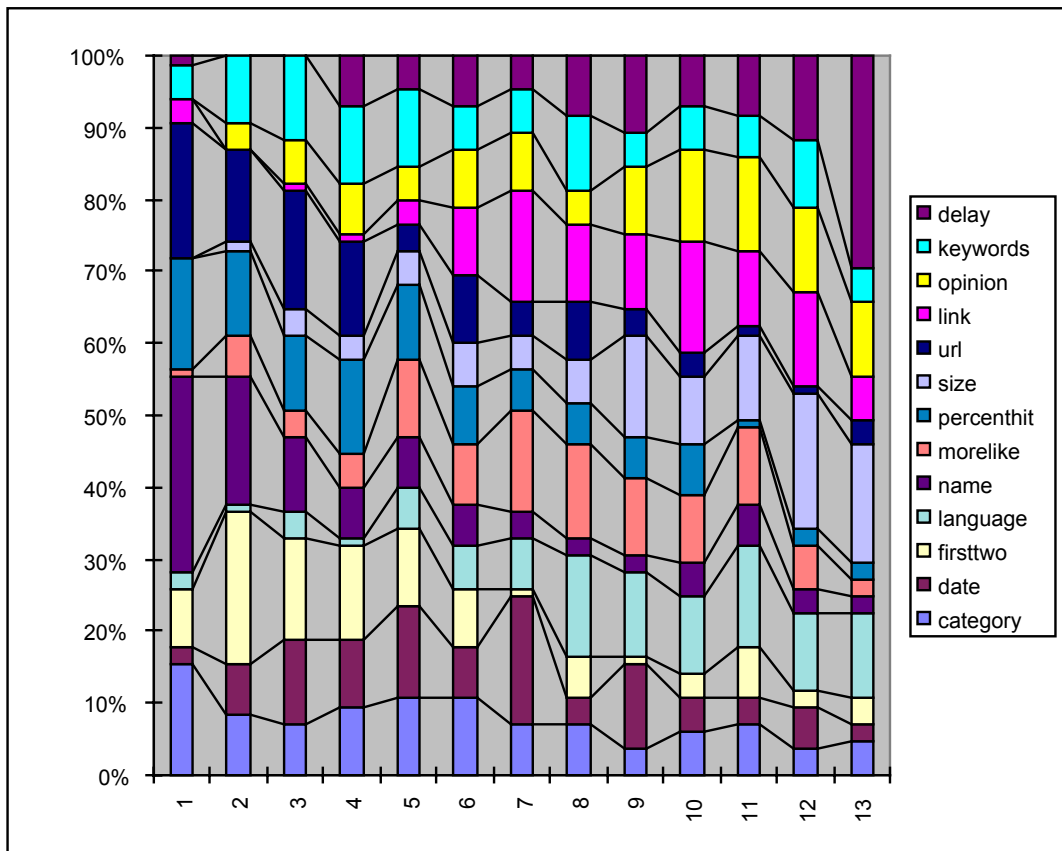


Figure 85 : Résultats du questionnaire pour identifier les informations à afficher à propos d'une page Web. Le questionnaire demandait de classer par ordre d'importance (de 1 à 13) treize informations différentes possibles à propos d'une page Web. Les colonnes sont ordonnées par ordre d'importance de l'information (de la plus importante = 1 à la moins importante = 13). Nous voyons par exemple dans la dernière colonne de droite, que presque 30% des utilisateurs questionnés considèrent que le temps de chargement est l'information la moins importante. Nous constatons à la colonne de gauche que les informations les plus importantes sont le titre et l'adresse URL de la page.

Avec la participation d'Emmanuel Dubois, nous avons réalisé le prototype appelé VITAMIN, également illustré à la Figure 84b. VITAMIN est une extension de VITESSE qui permet de poser des balises sonores et graphiques sur des pages Web d'intérêt. Le pique rouge de la Figure 84b correspond à une balise sonore : une balise clignote et engendre un son répétitif. La fréquence de répétition du son associé à une balise est proportionnelle à sa distance au focus (effet Doppler). L'ajout de balises se fait grâce à une "ToolGlass" [Bier 93].

Enfin, l'expérience de la programmation de VITESSE a également démontré l'utilité d'une architecture modulaire qui respecte le modèle PAC-AMODEUS. La forte modularité a guidé dans le développement et a permis un gain de temps considérable dans les modifications

correspondant aux différentes versions de VITESSE (1.0, 2.0 puis 2.5). De plus la structure induite par le modèle a permis le développement efficace de la version collaborative de VITESSE, par Yann Laurillau [CoVITESSE].

4.2 Perspectives

Les perspectives à donner au système VITESSE sont nombreuses. Les fonctionnalités du NF restent limitées et pourraient être fortement étendues. En effet, nous avons concentré notre effort sur le développement de l'interface. D'autre part, le noyau fonctionnel minimal que nous avons mis en place manque de robustesse et de rapidité. Ces deux problèmes empêchent la création d'une réelle communauté d'utilisateurs. L'intérêt d'une telle communauté est évident dans une approche centrée sur l'utilisateur.

Au niveau de l'interface, l'affichage de la page Web dans le navigateur peut être amélioré. En effet, puisque la page Web est maintenue par le serveur VITESSE, qui est aussi un serveur Web, la partie client de VITESSE pourrait consulter une version "améliorée" de la page Web (par exemple les mots-clés peuvent être mis en gras). Cette fonctionnalité requiert un traitement assez compliqué sur la page Web qui est récupérée et stockée sur notre serveur.

Enfin, nous proposons d'améliorer la construction de l'espace. VITESSE affiche un seul type d'espace. Nous pouvons envisager d'autres espaces, comme un espace où l'axe des abscisses et l'axe des ordonnées sont utilisés indépendamment pour traduire la pertinence de la réponse par rapport à deux mots clés différents.

CHAPITRE VIII

MULTAB : VISUALISATION DE GRANDES TABLES



Partie d'échec sur un damier en grandeur nature extraite de la série "le prisonnier"

Ce chapitre est consacré à notre deuxième réalisation, MulTab (MULTimodal TABLE). Le lecteur remarquera, dès les premières illustrations, la ressemblance avec la vue en colline proposée dans VITESSE et présentée au chapitre précédent.

Pour présenter MulTab, nous reprenons la même structure que celle du chapitre précédent, VITESSE. Nous décrivons MulTab selon les deux facettes suivantes : conception de l'interface et réalisation logicielle. Nous concluons le chapitre par un bilan de la conception et les perspectives au système.

1 Introduction

MulTab (MULTimodal TABLE) est un système de visualisation et d'édition de très grandes tables de données. Nous constatons que les tableurs actuellement commercialisés comme Excel [Excel] n'ont pas été conçus pour permettre la manipulation de très grandes tables. La solution partiellement satisfaisante offerte par Excel consiste à découper l'espace créé pour visualiser une table en plusieurs tableaux. Chaque tableau est alors réparti sur une page et l'utilisateur navigue d'une page à l'autre en utilisant les onglets, mis en évidence à la Figure 86. Le découpage du tableau induit donc de nombreuses tâches articulatoires de navigation (changer de page ou faire défiler la page en cours) et d'édition (référencer la bonne page dans les formules). Ce découpage peut aussi induire des erreurs puisqu'il existe plusieurs cellules avec la même référence (par exemple A1).

Notre approche consiste à appliquer la vue en colline, présentée au chapitre précédent, au type de données d'un tableur, pour que ce dernier puisse présenter toutes les données dans un seul tableau.

	B	C	D	E	F	G	H	J	K	L	M	N	O	P	Q	R	S
26						0		Mohammed	x	15	x	14	x				14,5
27	Hicham	x	14	x	15	14,5											#DIV/0!
28						0		Sebastien	x	12	x	13	x		x		12,5
29	Salah	x	13	x	14	13,5		Ibrahim	x	12	x	13	x		x		12,5
30	Paul	x	13	x	14	13,5											#DIV/0!
31						0		Benjamin	x	15	x	14	x				14,5
32	Sylvain	x	13	x	13	13											#DIV/0!
33						0		Mathieu	x	14	x	14					14
34	Emmanuel	x	15	x-	9	12		Pierre	x	14	x	14					14
35	Pierre	x	15	x-	9	12											#DIV/0!
36						0		Robert	x	15	x	15	x		x		15
37	Guillaume	x	14	x	14	14		Eric	x	15	x	15	x		x		15
38	Olivier	x	14	x	14	14											#DIV/0!
39						0		Sebastien	x	14	x	15	x		x		14,5
40	Vincent	x	13	x	14	13,5		thomas	x	14	x	15	x		x		14,5
41	Christophe-Pi	x	13	x	14	13,5											#DIV/0!
42						0		Benoit	x	14	x	14	x				14
43	B.	x	14	x	15	14,5											#DIV/0!
44	D.	x	14	x	15	14,5		Jerome	x	14	x	15	x				14,5
45						0		Dimitri	x	14	x	15	x				14,5
46	Gregoire	x	14	x	14	14											#DIV/0!
47						0		Jean	x	15	x	15	x		x		15
48	Mathieu	x	12	x	13	12,5											#DIV/0!
49	Francois	x	12	x	13	12,5		Omar	x	14	x	14	x				14
50						0		Cherif	x	14	x	14	x				14
51	Fabrice	x	11	x	12	11,5											
52	Jean-Marie	x	11	x	12	11,5											
53						0											
54	Hugo	x	15	x	15	15											
55						0											
56	Nicolas	x	14	x	14	14											
57	Salama	x	14	x	14	14											
58																	
59	21 groupes					Moy TP1	Moy TP2										
60						13,9	13,79		15 groupes			14,2		14,06			

Onglets pour changer de page

Figure 86 : Interface du logiciel Excel [Excel] avec trois pages contenant des tableaux de données.

2 Conception de l'interface

Comme VITESSE, MulTab est principalement une technique de visualisation. Nous présentons donc la conception de l'interface selon les trois dernières étapes du processus de visualisation, que nous avons présentées au chapitre IV. Le point de vue sur les données est présenté au paragraphe 2.1, l'espace construit est présenté au paragraphe 2.2 et enfin le point de vue sur l'espace est présenté au paragraphe 2.3.

2.1 Point de vue sur les données

MulTab permet la manipulation de larges tables de données numériques ou alphanumériques, avec un maximum de 20 000 cases. Les données sont de type <2D non structurée, Ordonnée>, car nous supposons que l'ordre des lignes et des colonnes est important. Excel, avec son système à plusieurs pages, présente un point de vue sur les données qui est <3D non structurée, Ordonnée>. Cette structure "3D non structurée" est présentée dans l'espace par la métaphore de plusieurs feuilles de papier quadrillées, posées les unes sur les autres. Dans l'exemple de la Figure 86, la feuille 1 est posée sur la feuille 2, elle-même posée sur la feuille 3.

2.2 Espace

Comme le montre la Figure 87, l'espace de MulTab est composé d'un tableau qui est une modalité de visualisation globale de la table. Les cellules sont référencées comme les cases de bataille navale (A1, C3, etc.) et contiennent un texte (ou label) d'une seule ligne. La Figure 87 illustre un tableau de 200 lignes par 60 colonnes (soit 12 000 cellules). Les textes sont alignés à droite lorsqu'il s'agit de nombres et à gauche lorsqu'il s'agit de mots. Ce choix de modalité n'est pas arbitraire au sens de Bernsen puisqu'il correspond à la présentation d'Excel [Excel], un tableur communément utilisé.

Chaque cellule utilise également un trait graphique, la couleur, pour traduire la valeur de la cellule. La Figure 87 est composée de cellules dont les couleurs varient du rouge au bleu pour des valeurs qui varient de 0 à 20. Cette modalité, M1 est non-linguistique et arbitraire au sens de Bernsen, alors que le texte de la valeur dans la cellule est issu d'une modalité linguistique, M2. Nous avons choisi de colorier en gris clair les cellules contenant des mots et en blanc les cellules dont la valeur sort de l'échelle de valeurs et les cellules vides. La modalité M1, qui utilise les couleurs comme langage d'interaction, est arbitraire, et n'est donc pas compréhensible par l'utilisateur sans une légende. La légende présente l'échelle de valeurs selon une modalité active M3, en bas de l'espace. Certaines parties graphiques de cette légende, visibles en totalité à la Figure 87 et en détail à la Figure 88, s'expliquent car le système permet à l'utilisateur de changer la légende. Nous reviendrons sur cette fonctionnalité au paragraphe 2.3.2, consacré à la navigation.

Nous caractérisons la composition des modalités M1 et M3 comme suit :

- ▶ simultanée pour l'aspect temporel,
- ▶ adjacente pour l'aspect spatial,
- ▶ d'extension pour l'aspect syntaxique (le tableau présentant d'autres informations que les valeurs),

- complémentaire pour la sémantique.
- L'aspect articulatoire n'est pas ici considéré dans la composition puisque M1 et M3 reposent sur le même dispositif, l'écran.

Le Patron N°4 sur les modalités complémentaires et simultanées est vérifié à condition que le point de vue sur l'espace permette de rendre observable en même temps les informations véhiculées par M1 et M3. Nous verrons au paragraphe suivant que c'est toujours le cas.

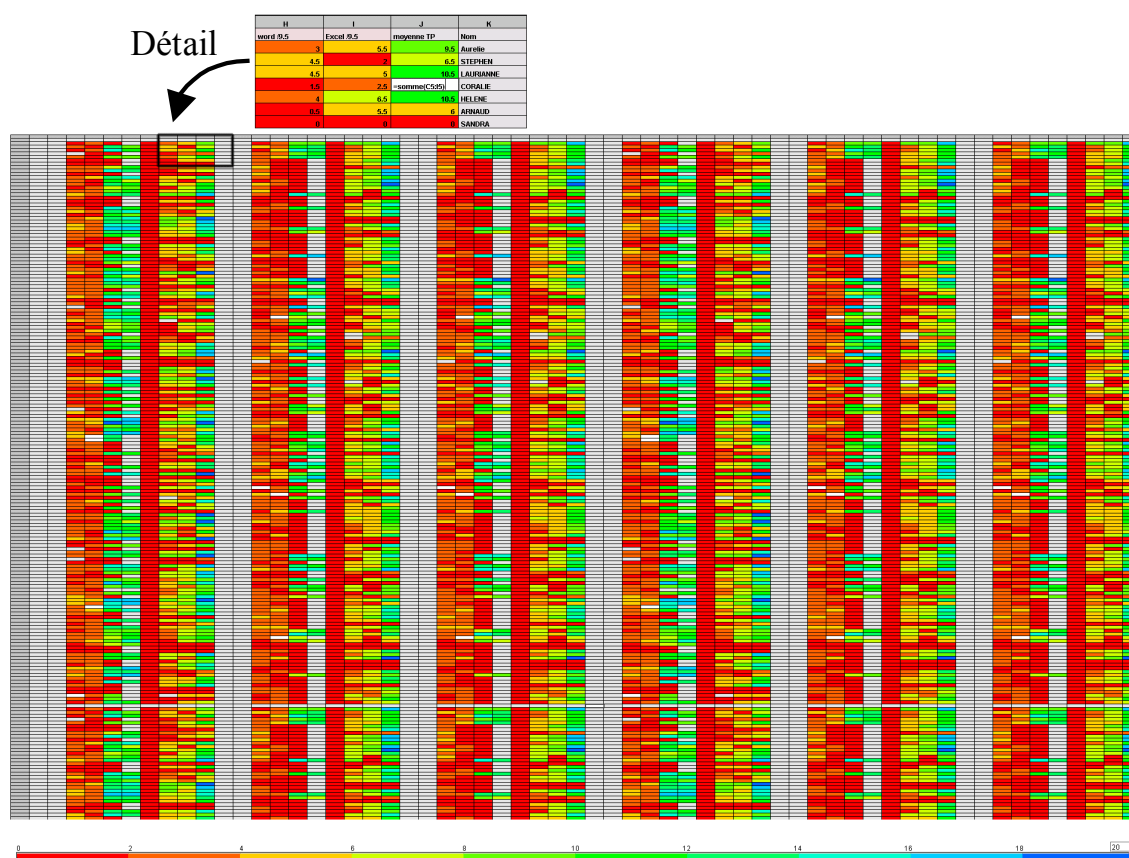


Figure 87 : Espace de MulTab composé d'un tableau de données. Les cellules sont coloriées et une légende accompagne l'espace. Dans cet exemple, 12000 cellules sont présentées.

2.3 Point de vue sur l'espace

Comme pour la présentation du système VITESSE, nous présentons d'abord les aspects de visualisation (modalités en sortie) au paragraphe 2.3.1, puis de navigation (couplage des modalités en entrée et en sortie) au paragraphe 2.3.2.

Certaines modalités de visualisation sont composées avec des modalités sonores. Aussi la première opération du système consiste à tester le dispositif sonore. Si le dispositif fonctionne, un

message sonore indique que les sons sont activés afin que l'utilisateur ne soit pas étonné ensuite des informations sonores du système. Si au contraire les haut-parleurs ne fonctionnent pas, le système démarre sans message sonore d'accueil. Comme l'utilisateur n'aura pas entendu le message d'accueil, il sera moins surpris de ne pas entendre de sons dans l'interaction. Nous vérifions ainsi le Patron N°1, sur le test du dispositif sonore.

2.3.1 Visualisation

Comme pour VITESSE, nous constatons qu'un tableau de grande taille ne peut pas afficher en détail le texte de chaque cellule car aucun dispositif d'affichage n'est assez grand. Excel utilise par exemple une modalité locale et précise que l'utilisateur déplace au moyen d'outils de navigation. Il s'agit d'un point de vue partiel de l'espace mais qui permet toutes les tâches d'édition.

Dans MulTab, nous adoptons une approche différente et nous proposons les deux modalités de sortie de la Figure 88.

La première modalité définit une version sans texte de l'espace, présentée à la Figure 87 et à la Figure 88 dans la partie gauche. Cette modalité est globale mais vague pour toutes les tâches d'édition et de consultation des données. Au regard de l'espace de conception présenté au paragraphe 3.3.1 du chapitre IV (page 114), il s'agit du contexte. Afin d'utiliser au maximum les capacités de l'écran, la résolution de l'écran est testée. Si le nombre de cellules implique que la taille des cellules est inférieure à une taille minimum fixée à 4 pixels de haut et 12 de large, le système refuse d'afficher une interface illisible et demande une nouvelle taille de tableau à afficher. Nous vérifions ainsi le patron N°1 sur le test du dispositif graphique.

La seconde modalité définit une vue locale et déformée de la table. Comme nous le voyons à la Figure 88, cette modalité déformée reprend la vue hybride, entre la sphère et la pyramide. La partie avant est aplanie afin de décomposer cette modalité en deux (une modalité encore déformée sur les bords et une modalité non déformée au centre). Le facteur de grossissement de la déformation permet d'afficher le texte des cellules dans la partie centrale de la modalité. Selon notre espace de conception, il s'agit du **focus**.

Le patron N°5 sur la continuité spatiale est vérifié. En effet la composition de ces deux modalités décrite comme suit :

- ▶ simultanée pour l'aspect temporel,
- ▶ adjacente pour l'aspect spatial,
- ▶ d'extension pour l'aspect syntaxique,
- ▶ partiellement redondante pour l'aspect sémantique.

Nous remarquons également que le patron N°8 sur la compréhension de la fonction de déformation est vérifié car la déformation de la grille offre les perspectives d'une forme analogique. Nous comparons cette forme à un cube placé entre une table et la nappe qui la couvre.

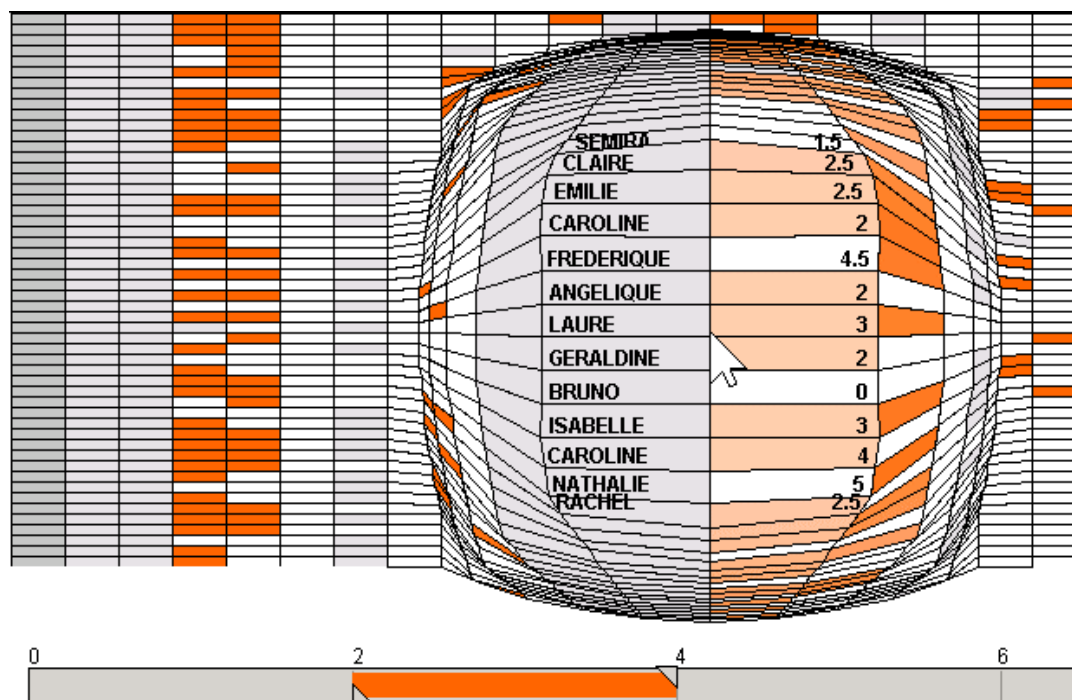


Figure 88 : Affichage d'une modalité détaillée (un focus) par dessus le tableau. La forme hybride, entre la sphère et la pyramide, présente des couleurs éclaircies pour une meilleure lisibilité.

La Figure 90 illustre différents cas de formes hybrides présentes dans MulTab. Le patron N°8 sur la compréhension de la déformation est respecté dans chacun d'eux. Nous avons respecté le patron N°3 sur l'adaptabilité des modalités de sortie en déléguant à l'utilisateur le choix de la forme hybride.

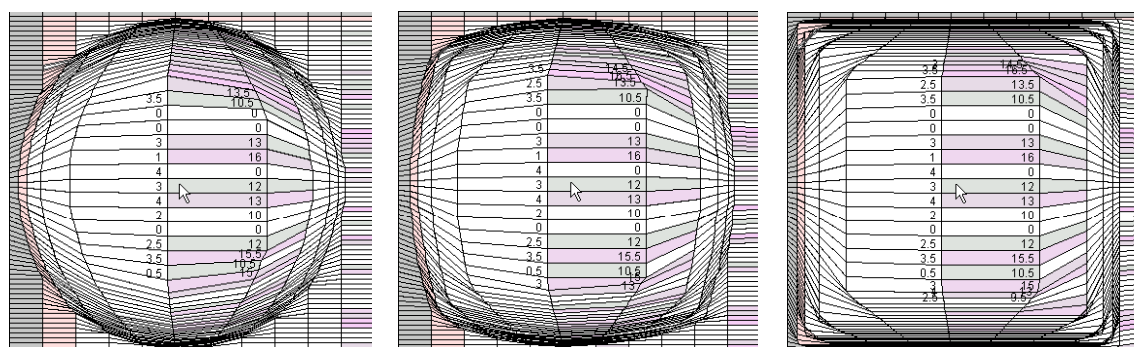


Figure 90 : Différentes formes de la modalité "focus" dans MulTab : circulaire à gauche, pyramidale à droite, hybride (50%-50%) entre les deux.

Comme le montre la Figure 91, l'utilisateur peut également faire apparaître plusieurs foci définis par une modalité déformante. Le point de vue sur l'espace proposé par MulTab autorise jusqu'à quatre zones déformées.



Figure 91 : Vue générale de l'application MulTab avec quatre foci.

Plusieurs foci mettent en œuvre une composition de modalités aux profils et syntaxes identiques. Pour la conception, nous avons besoin de décrire une composition :

- ▶ coïncidente pour l'aspect temporel (un des focus étant toujours présent, les autres apparaissant et disparaissant dynamiquement),
- ▶ complémentaire pour l'aspect sémantique (puisque des parties différentes du tableau sont véhiculées par les modalités).

Le patron N°4 sur les modalités complémentaires et simultanées est respecté, car c'est l'utilisateur qui décide de faire apparaître les foci, et sait quand les deux modalités véhiculent des informations complémentaires pour la réalisation de sa tâche.

L'aspect spatial de la composition peut correspondre à plusieurs schémas, car l'utilisateur déplace les foci dynamiquement. Nous n'autorisons pas la composition recouvrante ou imbriquée car un seul focus suffit dans ces cas là. La composition spatiale peut être :

- disjointe ou adjacente si l'utilisateur maintient éloignés les foci,
- intersectée si l'utilisateur rapproche deux foci.

Cette analyse nous a amenés à implémenter un algorithme qui gère cette collision en respectant le patron N°5 sur la continuité spatiale entre les modalités. Nous présenterons cet algorithme au paragraphe 3.2.1. Nous reproduisons à la Figure 92 trois exemples de collision, pour montrer que les patrons N°8 (compréhension de la déformation) et N°5 (continuité spatiale) sont vérifiés. Le cas extrême est la collision de trois pyramides, qui est reproduit à la Figure 92c. Dans ce cas, deux cotés de la pyramide restent intacts. Nous estimons que dans le cas limite présenté à la Figure 92c, l'utilisateur sera capable de comprendre, grâce à l'assemblage géométrique, que la cellule sous le pointeur de la souris est la cellule F30.

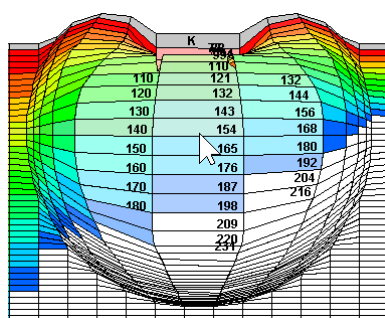


Figure 92a : Composition spatiale de deux foci en forme de sphère.

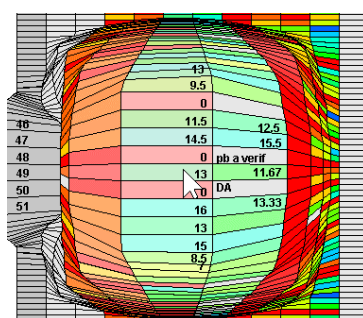


Figure 92b : Composition spatiale de deux foci de formes hybrides.

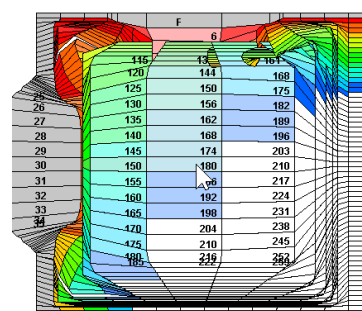


Figure 92c : Composition spatiale de trois foci en forme de pyramide.

L'espace présenté à la Figure 91 présente un outil pour paramétrer la colorisation des cellules, que nous avons appelé BiSlider. Nous avons vu que la présentation de cet outil servait de légende pour les couleurs utilisées dans le tableau. Le BiSlider présente des segments de couleurs dégradés séparés par des valeurs intermédiaires entre la borne inférieure et la borne supérieure de la colorisation.

Le BiSlider, illustré à la Figure 93, offre de nombreuses possibilités à l'utilisateur pour régler les couleurs associées aux valeurs du tableau. L'utilisation de cet outil de navigation nécessite de redessiner entièrement le tableau pour prendre en compte la nouvelle façon de colorier.

Cette fonction n'est pour l'instant pas assez rapide pour être traitée par un outil de navigation fortement couplé. Le BiSlider est donc un outil de navigation relâché.

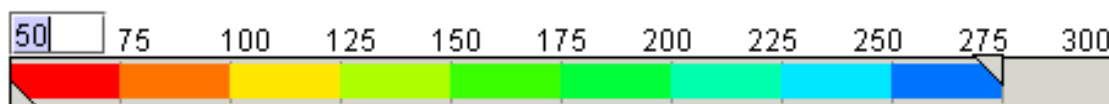


Figure 93 : Légende des couleurs affichée par l'outil BiSlider.

Les fonctionnalités offertes par cet outil sont les suivantes :

- ▶ Edition de la borne inférieure et de la borne supérieure de l'outil : l'édition de la borne inférieure "50" est illustrée à la Figure 93.
- ▶ Changement de la première et de la dernière valeur coloriées en faisant glisser les triangles de gauche à droite : en manipulant le triangle de droite, l'utilisateur a fixé la dernière valeur coloriée à 275 dans l'exemple de Figure 93.
- ▶ Sélection d'un seul segment à colorier en cliquant deux fois dessus : par exemple cliquer deux fois sur le segment jaune de la Figure 93 implique que seules les valeurs entre 100 et 125 seront coloriées.
- ▶ Sélection de plusieurs segments à colorier en cliquant deux fois sur chacun et en maintenant la touche "Shift" du clavier enfoncée.
- ▶ Changement de la granularité de l'outil, c'est-à-dire du nombre de segments de couleur, en déplaçant de gauche à droite le premier nombre ("50" à la Figure 93).

2.3.2 *Déplacement des foci dans l'espace : navigation dans l'espace*

Pour la conception, nous avons considéré, pour un tableau, les tâches de sélection, d'édition et de copier-coller. Notre approche est de toujours présenter un focus sous le pointeur de la souris (focus principal) car les tâches que nous venons d'énoncer comportent toutes des sous-tâches de désignation d'une cellule. En proposant une modalité précise sous le pointeur de la souris, nous permettons à l'utilisateur de toujours voir la valeur de la cellule qu'il peut désigner avec le bouton gauche de sa souris. Comme nous reconnaissons ici un outil de navigation fortement couplé, nous savons que notre approche impose des contraintes de performance. Les algorithmes de déformation et d'affichage doivent permettre de réactualiser plusieurs fois par seconde l'interface.

L'apparition des autres foci est contrôlée par l'utilisateur. Les deux boutons en haut à gauche de l'interface servent à faire apparaître et disparaître les foci, qui restent attachés respectivement à la première ligne et à la première colonne du tableau. Comme nous le voyons à la Figure 91, le focus dédié à la première ligne reste toujours sur la même colonne que le focus

principal. De même, le focus attaché à la première colonne reste toujours sur la même ligne que le focus principal. Ces deux foci servent à connaître la référence de la cellule qui se trouve sous le pointeur de la souris. Nous voyons ainsi que la cellule sous le pointeur de la souris à la Figure 91 est la case "AJ85" qui a pour valeur 12,67. Pour aider l'utilisateur à identifier le numéro de la ligne ou la lettre de la colonne d'une cellule, nous avons rendu un peu plus rouge toute la colonne au dessus de la cellule et toute la ligne à gauche de la cellule. Le numéro de la ligne et la lettre de la colonne dont le fond est entre le gris et le rouge sont donc la référence exacte de la cellule sous le pointeur de la souris.

L'apparition et la disparition des foci respectent le patron N°6 sur la continuité temporelle. En effet, lorsque le point de vue sur l'espace perd ou gagne une modalité précise, la modalité composée résultante n'est plus la même. Pour rendre continu le passage d'une modalité à n foci vers une modalité à $n-1$ ou $n+1$ foci, nous avons implémenté une animation faisant disparaître ou apparaître petit à petit un focus. Comme le préconise le patron N°6 (continuité temporelle), cette animation est renforcée par un son. Le son annonce qu'un focus est en train d'apparaître ou de disparaître. L'utilisation d'un son respecte le patron N°2 sur l'utilisation du canal le moins occupé. En effet, l'écran est très chargé et l'attention visuelle de l'utilisateur est dirigée vers le focus qui apparaît ou disparaît. Le son et l'animation sont liés par une composition :

- ▶ simultanée pour l'aspect temporel,
- ▶ indépendante pour l'aspect articulatoire,
- ▶ différente pour l'aspect syntaxique,
- ▶ redondante pour l'aspect sémantique.

L'aspect spatial n'est pas pris en compte dans une composition qui fait intervenir un son non spatialisé, comme c'est le cas dans MulTab.

Le quatrième focus apparaît lorsque l'utilisateur maintient le bouton droit de la souris enfoncé tout en déplaçant la souris. Le focus sous le pointeur se sépare alors en deux. Le premier focus continue de suivre le pointeur de la souris alors que le second se comporte comme le focus de première ligne ou de première colonne. Si le mouvement après le clic droit est horizontal, le deuxième focus reste attaché à la colonne où le clic a eu lieu. Ce cas est illustré à la Figure 91 où un clic droit a été réalisé sur la colonne des prénoms. Si le mouvement après le clic droit est vertical, le deuxième focus reste dédié à la ligne où le clic droit a eu lieu. Le nouveau focus disparaît lorsque le bouton droit de la souris est relâché. La disparition du focus se fait progressivement, par une animation, comme pour les foci de première ligne et de première colonne. Cet outil permet de faire apparaître deux foci pour comparer deux lignes ou deux colonnes. Nous

identifions trois avantages par rapport au système Excel [Excel] qui propose de scinder l'espace en deux espaces alignés. Deux modalités précises mais locales sont donc mises en œuvre côte à côte. Le premier avantage de notre système est de pouvoir évaluer la distance entre les parties comparées. Le second avantage est la facilité avec laquelle l'utilisateur crée un focus pour réaliser une tâche de comparaison. L'utilisateur indique en un geste de glisser-déposer :

- ▶ qu'elle/il veut comparer deux valeurs,
- ▶ que les valeurs à comparer sont sur la même ligne ou la même colonne,
- ▶ quels sont les lignes ou colonnes, objets de comparaison.

Dans Excel chacune de ces actions nécessite la manipulation d'une partie différente de l'interface. Enfin, le système Excel ne garantit pas que les espaces soient organisées comme les données initiales, la partie droite des données peut apparaître dans la partie gauche de l'interface par exemple.

Un autre outil de navigation que nous proposons à l'utilisateur est la réglette intitulée "Focus Size" qui permet de régler la taille du focus qui suit le pointeur. Cet outil permet d'agrandir le focus, le temps de rechercher une information par exemple. Comme dans VITESSE, nous modifions le paramètre R de la fonction de déformation pour changer la taille d'un focus. Cette possibilité offerte à l'utilisateur respecte le patron N°3 sur l'adaptabilité des modalités de sortie. Deux autres réglettes permettent de modifier le paramètre Z de la fonction de déformation et le coefficient CoeffMax. La première réglette permet à l'utilisateur de choisir la taille de la zone déformée entre le centre du focus et les cellules du contexte. La seconde réglette permet de choisir la taille des cellules dans le centre des foci. La conception de ces deux outils correspond également au patron N°3.

3 *Réalisation logicielle*

Dans ce paragraphe, nous décrivons les éléments de réalisation logicielle qui permettent de reproduire l'interface décrite au paragraphe précédent. Nous décrivons au paragraphe 3.1 notre solution d'architecture logicielle pour réaliser l'interface de MulTab selon le modèle PAC-Amodeus que nous avons présenté au chapitre V. Le paragraphe 3.2 est consacré à des points notables d'implémentation.

3.1 *Architecture logicielle*

Il convient de noter que certains composants de l'architecture logicielle présentée dans ce paragraphe ne sont que partiellement mis en place dans notre implémentation. MulTab est

opérationnel, mais sa structure logicielle selon notre solution générique de réalisation du Chapitre V nécessite des remaniements du code. En particulier les composants traditionnels (boutons, glissières, cases à cocher) ont été rajoutés dans l'interface pour rendre possible la manipulation de la technique de visualisation, sans appliquer nos principes de conception logicielle. Enfin l'architecture réalisant les foci multiples n'est pas achevée.

La Figure 94 présente la hiérarchie d'agents du composant Contrôleur de Dialogue lorsque trois foci sont présents. L'ajout d'un quatrième focus s'opère au même niveau que les trois autres. Nous appliquons ici notre solution générique de réalisation logicielle de la composition du focus et du contexte exposée au Chapitre V.

Les événements de la souris arrivent dans la facette Présentation du focus principal qui est toujours présent. Ce focus en informe son père, l'agent ciment appelé "FisheyeTable", qui répercute cet événement aux autres foci éventuellement présents. La facette Présentation de l'agent "Contexte" peut afficher tout le contexte ou une partie seulement. Quand un focus se déplace, l'agent ciment "FisheyeTable", lui demande de s'afficher à sa nouvelle position, mais demande auparavant à l'agent "Contexte" de rafraîchir l'ancien emplacement du focus. Afin de ne pas rafraîchir de zones inutilement, nous avons mis au point un algorithme de rafraîchissement minimal qui sera décrit dans le paragraphe suivant "Implémentation". Enfin, lorsqu'un focus rentre en collision avec un autre focus, l'agent ciment demande à l'un des agents de calculer sa déformation sans tenir compte de l'autre et sans afficher le résultat de sa déformation. Ce résultat est récupéré par le ciment qui le passe au second agent. Cet agent a la charge de calculer et d'afficher la forme de collision en tenant compte des données dans son abstraction et des données fournies par son père. La fonction de cet agent "Focus" change donc au cours du temps. L'algorithme de calcul de la collision sera également présenté au paragraphe suivant.

L'agent BiSlider transmet des messages de haut niveau qui décrivent quels intervalles de valeurs doivent être coloriés et par quelles couleurs.

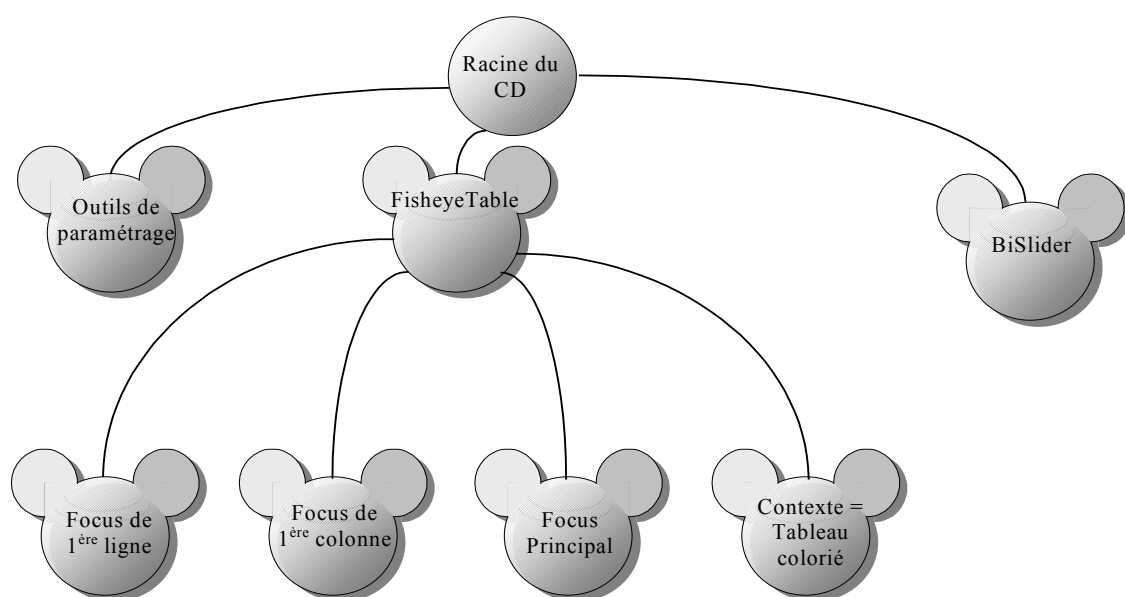


Figure 94 : Hiérarchie d'agents PAC du Contrôleur de Dialogue de MulTab.

Nous venons de décrire le Contrôleur de Dialogue (CD). Le NF et l'ANF, deux autres composants du modèle PAC-AMODEUS ne sont pas réalisés : l'utilisateur ne peut donc pas sauvegarder une table. Le CTP de MulTab repose sur la boîte à outils SWING.

3.2 Implémentation

Nous détaillons deux points remarquables de l'implémentation de MulTab : l'algorithme de calcul de collision et celui de rafraîchissement minimal.

3.2.1 Algorithme de calcul de la collision entre deux foci

Pour obtenir l'effet de collision entre deux foci, nous avons dû utiliser une astuce d'implémentation. Nous avons vu dans le paragraphe 3.1 que l'architecture impliquait que les collisions se calculaient en effectuant les déformations de focus les uns après les autres. Cette approche est adaptée à notre système car elle permet d'avoir plusieurs collisions, comme la Figure 92c l'illustre. Dans la collision de deux foci, nous considérons donc un premier focus, normalement calculé, et un second focus, différent de ceux déjà présentés. Alors que la continuité entre un focus normal et le contexte est garantie par les propriétés géométriques du point de vue sur l'espace, le second focus utilise une autre technique pour assurer la continuité spatiale et vérifier le patron N°5.

Considérons un point (x, y) que nous voulons déformer dans le second focus et pour lequel notre fonction de déformation a calculé la transformée (x', y') . Sans savoir si cette déformation fait intervenir la distance d du point par rapport au centre du focus, nous savons néanmoins que lorsque cette distance est supérieure à une distance D , le point n'est pas déformé puisqu'il appartient au

contexte. Notre technique de remplacement pour assurer la continuité consiste à déformer le point (x, y) en un point médian entre (x, y) et (x', y') . Quand le point est proche du centre du focus, un point médian proche de (x', y') est choisi afin d'assurer le grossissement des informations. Au contraire, quand le point s'éloigne du centre et se rapproche de la frontière avec le contexte, un point médian proche de (x, y) est choisi afin d'assurer la continuité avec le contexte. Nous implémentons ce compromis entre (x, y) et (x', y') par l'interpolation linéaire suivante :

- $CoeffdeContexte = d / D$ ($CoeffdeContexte$ vaut 1 près du contexte et 0 au centre)
- $x'' = x \times CoeffdeContexte + x' \times (1 - CoeffdeContexte)$
- $y'' = y \times CoeffdeContexte + y' \times (1 - CoeffdeContexte)$

Cette technique permet de rapprocher un focus d'une zone déjà déformée, mais le résultat n'est pas satisfaisant pour l'utilisateur : en effet des informations petites et déformées peuvent être affichées sous le curseur de la souris. Pour résoudre ce problème, nous avons adopté une approche consistant à diminuer la taille du focus le moins important au fur et à mesure que deux foci se rapprochent. Enfin, nous effaçons complètement le focus le moins important quand les deux foci sont très proches. Dans MulTab, le focus sous le pointeur est toujours le plus important.

3.2.2 Algorithme du rafraîchissement minimal

Le second point d'implémentation développé est l'algorithme du rafraîchissement minimal. Cet algorithme évite de rafraîchir l'intégralité de l'interface à chaque déplacement d'un focus. En effet, les hypothèses de performances que nous avons posées pour que la navigation soit fluide impliquent qu'un rafraîchissement partiel soit effectué à chaque déplacement d'un focus.

Lorsqu'un focus change de position, uniquement les zones correspondant à l'ancienne et à la nouvelle position du focus sont rafraîchies. Pour cela, nous rappelons deux aspects de l'architecture retenue :

- La zone couverte par la nouvelle position du focus est traitée par la facette Presentation de l'agent PAC du focus en question
- La zone couverte par l'ancienne position du focus est traitée par la facette Presentation de l'agent Contexte

Nous remarquons toutefois que ces deux zones se recoupent très souvent en grande partie car, dans l'exemple d'un rafraîchissement tous les 20ème de seconde, l'utilisateur n'a pas le temps d'effectuer un déplacement important du focus. La Figure 95 illustre les zones de l'interface de MulTab qui sont redessinées lors d'un mouvement de la souris vers le haut et vers la droite. Les

zones noires sont celles où il faut effectivement redessiner le contexte. Les zones barrées d'un X sont celles où il n'est pas nécessaire de redessiner le contexte puisque un focus (zone en gris) sera redessiné par dessus à cet endroit.

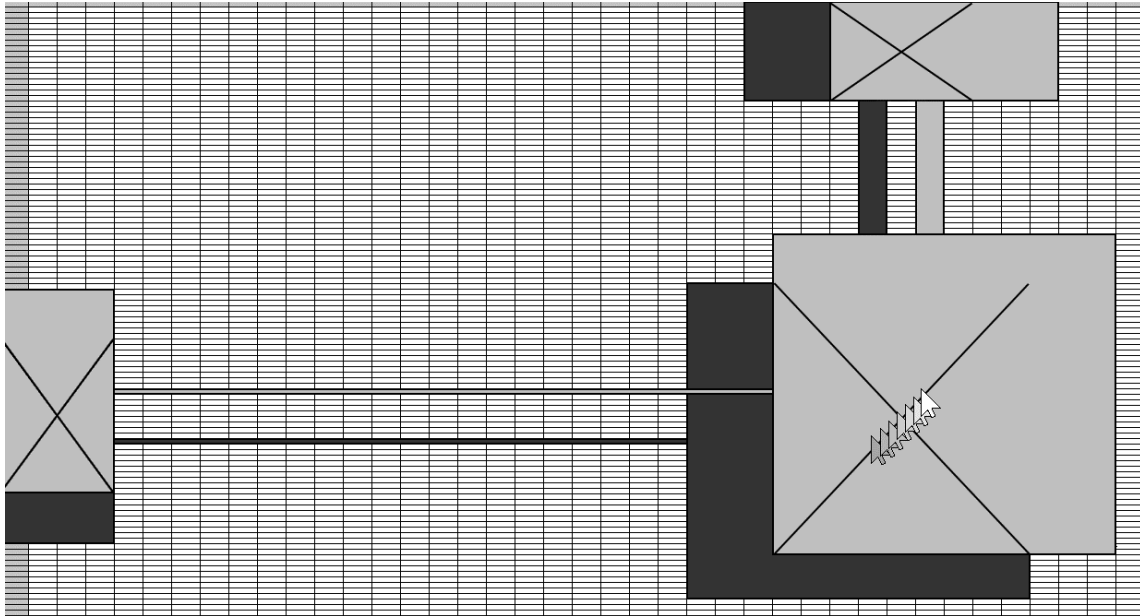


Figure 95 : Zones de l'interface à rafraîchir lors d'un déplacement de la souris.

Sans rentrer dans les détails de l'algorithme, nous décrivons son principe assez simplement en cinq étapes :

- ▶ Une liste vide est créée.
- ▶ Chaque zone à rafraîchir est ajoutée dans la liste par ordre d'affichage.
- ▶ Si la zone ajoutée n'intersecte aucune zone déjà présente dans la liste, elle est ajoutée sans modification.
- ▶ Si la zone ajoutée Z1 intersecte une zone déjà présente Z2 dans la liste, l'algorithme découpe la zone Z1 selon les bords de Z2 et demande récursivement l'ajout des parties de Z1 qui n'intersectent pas Z2 dans la liste.
- ▶ La liste est parcourue afin de demander à chaque agent de rafraîchir les zones qui restent dans la liste.

4 Conclusion

4.1 Bilan

Les patrons de conception nous ont guidés dans la conception de MulTab. De plus MulTab constitue une deuxième interface basée sur un point de vue déformé. MulTab a été conçu et réalisé

en deux étapes. Une évaluation ergonomique de la première version de MulTab a impliqué d'une part la modification de la vue en sphère remplacée par la vue hybride, d'autre part l'implémentation des foci multiples et enfin l'amélioration de l'outil de coloriage de l'espace d'information. Nous avons aussi beaucoup amélioré les performances qui, à notre avis, rendaient les débuts de l'expérimentation peu concluants.

La nouvelle interface de MulTab, telle que nous l'avons décrite dans ce chapitre, permet maintenant non seulement de comparer des informations sur la même ligne ou sur la même colonne, mais également de voir à tout moment la référence de la cellule qui se trouve sous le pointeur de la souris.

4.2 *Perspectives*

Nous voulons d'abord atteindre un niveau de performance suffisant pour pouvoir mener une évaluation comparative dans laquelle notre système ne souffrira pas de ses performances d'affichage. Pour cela, nous envisageons de changer le composant SWING du CTP responsable de l'affichage des polygones par un composant OpenGL. Cette technologie bénéficie en effet d'un large support de la part des constructeurs de cartes graphiques. Ces cartes permettent l'optimisation et donc l'accélération de l'affichage des systèmes qui utilisent cette interface de programmation.

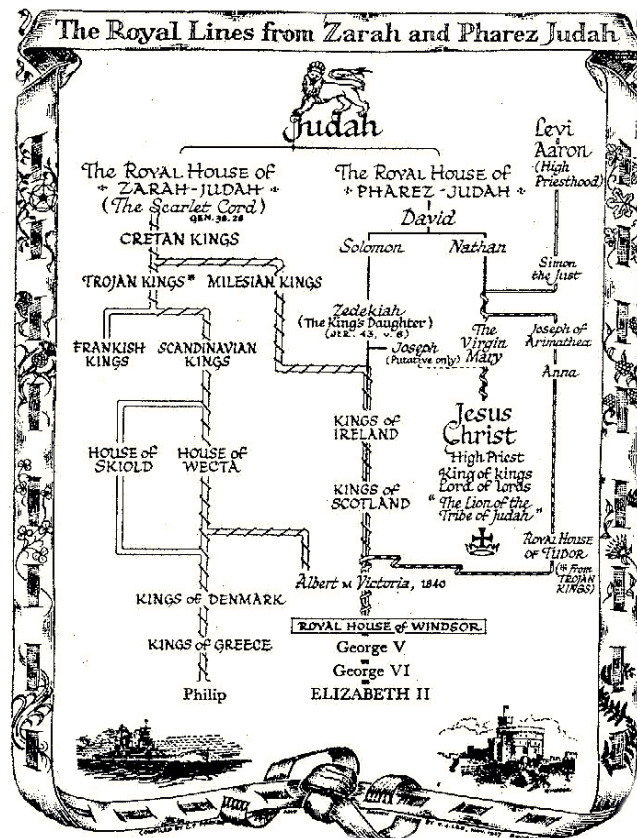
Notre système affiche actuellement entre cinq et quinze images par seconde (selon le nombre de foci, le nombre de couleurs et la résolution de l'écran) avec un Pentium II cadencé à 400MHz. Nous pensons dépasser la performance de quinze images par seconde dans le pire cas en utilisant une machine deux fois plus rapide et l'API OpenGL. Ces performances ouvriront la voie vers une nouvelle évaluation comparative de MulTab. Nous voulons notamment comparer l'interface de MulTab à celle d'Excel, qui est aujourd'hui la référence en matière de tableur de données.

Si cette évaluation révèle, comme nous nous y attendons, un gain d'utilisabilité pour certains types de tâches, nous envisageons, sur une proposition de Jock Mackinlay, d'étudier les liens possibles avec les travaux de Takeo Igarashi [Igarashi 98] sur le "Fluid Spreadsheet". Ces travaux proposent en effet de nouvelles solutions pour visualiser la structure des liens entre les cellules. Le système "Fluid Spreadsheet" ne présente toutefois qu'une modalité locale, comparable à Excel, pour rendre observable cette structure. Notre système fournit un cadre adéquat pour étendre les travaux de Takeo Igarashi à la visualisation de la structure globale du tableau. Leurs travaux (au niveau espace) sont orthogonaux aux nôtres (au niveau point de vue de l'espace).

Une autre perspective au système MulTab consiste à étudier la possibilité d'insérer dans l'espace d'information des graphes et des histogrammes comme ceux générés dans Excel. La taille d'une figure est beaucoup plus grande que la taille d'une cellule. Nous anticipons donc qu'il sera nécessaire d'adapter notre technique de déformation, insuffisante pour visualiser la totalité d'un graphe ou d'un histogramme sans déformation.

CHAPITRE VIII

PARENT : VISUALISATION D'UNE HIERARCHIE DE FICHIERS



répertoire à l'autre. Nous remarquons aussi que l'algorithme des Treemaps produit des rectangles très allongés verticalement ou horizontalement, qui ont deux défauts :

- ▶ Il est difficile d'y inscrire un texte.
- ▶ Il est difficile de sélectionner à la souris de tels objets.

Nous avons conçu Parent pour résoudre ces deux problèmes et pour offrir des actions de glisser-déposer dans cet espace d'information de grande taille

2 *Conception de l'interface*

Contrairement aux systèmes VITESSE et MulTab, le système Parent ne propose pas de point de vue sur l'espace différent de l'espace lui-même. Aussi dans cette section, nous ne présentons que le point de vue sur les données (paragraphe 2.1) et l'espace construit (paragraphe 2.2). Le point de vue sur l'espace correspond à une vue pleine écran de l'espace après que la résolution de l'écran ait été testée. Nous avons appliqué le patron N°1 pour le test du dispositif graphique.

2.1 *Point de vue sur les données*

La structure des données de Parent est évidemment une hiérarchie, comme pour le système TreeViz. La granularité des données est par contre différente. Nous avons en effet choisi de présenter des données nominales plutôt qu'ordonnées. Dans TreeViz, une liste de fichiers est présentée de manière à respecter un ordre. L'utilisateur averti peut constater l'ordre alphabétique des fichiers dans la Figure 95.

Pour produire l'espace que nous allons présenter au paragraphe suivant, nous nous sommes affranchi de cette contrainte. Cette différence de point de vue sur les données entre TreeViz et notre système implique des différences de fonctionnalités entre les systèmes (trier les fichiers dans un répertoire par exemple).

2.2 *Espace*

L'espace de Parent repose sur une extension de l'algorithme des **Treemaps** utilisé dans le système TreeViz. Nous expliquons donc les principes de l'algorithme des Treemaps au paragraphe 2.2.1 puis ceux de notre algorithme baptisé algorithme des **Fieldmaps** (paragraphe 2.2.2). Nous présentons ensuite, au paragraphe 2.2.3, un repère visuel en forme d'escalier, que nous avons inclus dans l'espace, afin d'aider l'utilisateur à situer l'information. Nous détaillons dans le paragraphe 2.2.4, une vue alternative contrôlée par l'utilisateur. Nous terminerons la

présentation de l'espace par l'exposé des outils de manipulation de l'espace et des données (paragraphe 2.2.5).

2.2.1 Algorithme des Treemaps

Comme ses auteurs, nous désignons par algorithme des Treemaps [Johnson 91][Shneiderman 91] l'algorithme étendu des Treemaps qui s'appelle également "Nested Treemaps". L'algorithme des Treemaps permet d'afficher une hiérarchie d'éléments en représentant chaque feuille de l'arbre par un rectangle et chaque nœud par un cadre. Cet algorithme permet enfin de choisir une taille pour chaque élément. Par exemple dans TreeViz, chaque rectangle a une taille proportionnelle à la taille du fichier qu'il représente.

Pour expliquer le placement des éléments dans cet algorithme, nous utiliserons les exemples de la Figure 97 qui présentent le même arbre sous trois formes différentes. La Figure 92a est une vue en graphe. La Figure 97b est une vue en diagramme de Venn du même arbre et la Figure 97c une vue en Treemaps. Le graphe d'arbre utilise le lien comme trait graphique pour représenter la structure arborescente. Dans le diagramme de Venn [Venn 71], c'est l'inclusion spatiale qui est le trait graphique utilisé pour rendre observable cette structure. L'algorithme des Treemaps est une version rectangulaire du diagramme de Venn, qui tend à produire un espace entièrement occupé par ses éléments (Space Filling approach).

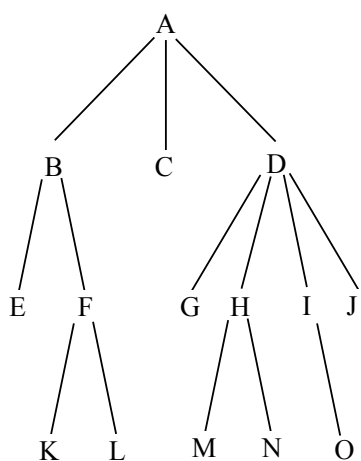


Figure 97a : Graphe d'arbre traditionnel

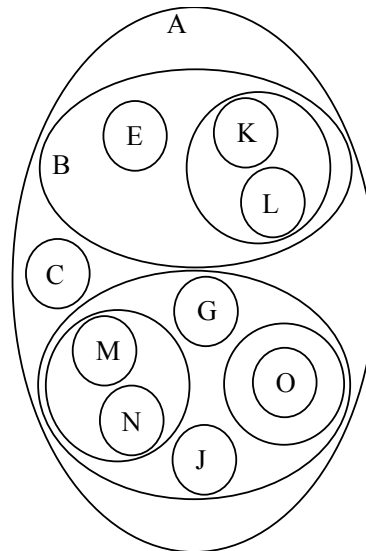


Figure 97b : Diagramme de Venn [Venn 71].

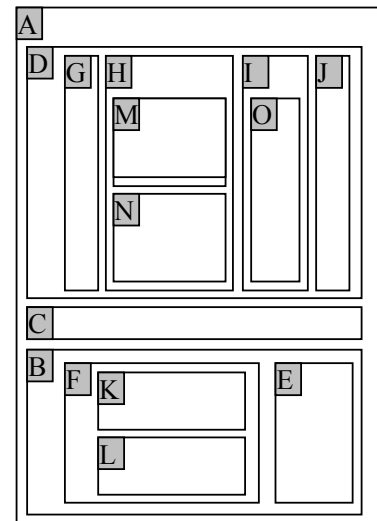


Figure 97c : Treemaps

L'algorithme des treemaps considère récursivement le placement des éléments en partant du haut de l'arbre. Le nœud de plus haut niveau (A) occupe la totalité de la zone

rectangulaire qui représente tout l'espace. Si, comme à la Figure 97c, l'espace est plus haut que large, l'espace est divisé horizontalement. Dans le cas contraire, l'espace est divisé verticalement. Nous appelons ce choix l'heuristique de départ.

L'espace occupé par ce nœud A est ensuite divisé en sous-zones, rectangulaires également. Le découpage ne se fait que selon un seul sens (horizontal ou vertical) et tient compte du cadre à garder pour le nœud A. La surface de chaque zone rectangulaire peut être choisie relativement aux autres : à la Figure 97c, nous observons une petite surface pour C, une surface moyenne pour B et une grande surface pour D.

Chaque zone créée à l'étape précédente est traitée de la même façon en remplaçant la division verticale par une division horizontale. Si une zone représente une feuille de l'arbre plutôt qu'un nœud, la division s'arrête et la zone est utilisée pour afficher la feuille de l'arbre. A la Figure 97c nous pouvons ainsi remarquer que les zones G, H, I et J sont réparties verticalement. Les zones E et F également. La zone C correspond à une feuille de l'arbre et n'est pas divisée.

Chaque niveau de l'arbre fait ensuite l'objet d'un changement entre division verticale et division horizontale. Les auteurs des Treemaps appellent cet aspect de l'algorithme le "Slice-Dice".

Pour allouer une surface précise à chaque feuille proportionnelle à un poids calculé, il convient de donner comme poids à chaque nœud, un poids équivalent à la somme des poids des fils du nœud. Dans l'exemple de la Figure 97c, pour donner les poids relatifs à B, C et D, les sommes de poids $(K+L+E)$, (C) et $(G+M+N+O+J)$ sont respectivement calculées. Ce calcul s'effectue en remontant en partant du bas de l'arbre, alors que l'algorithme des Treemaps s'opère en descendant à partir de la racine. Comme ce calcul doit être fait avant le traitement associé aux Treemaps, ces deux étapes ne peuvent donc pas être incluses dans le même parcours récursif de l'arbre.

2.2.2 Algorithme des Fieldmaps

Par rapport à l'algorithme des Treemaps, l'algorithme des Fieldmaps [Vernier 00b] recompose une suite de rectangles alignés verticalement ou horizontalement en utilisant la même zone. La Figure 98 illustre le cas de la recomposition d'une suite de rectangles horizontaux. L'algorithme des Fieldmaps recompose une liste de rectangles afin que chaque élément ressemble le plus possible à une proportion jugée idéale. L'exemple de la Figure 98 présente, à gauche, une suite de rectangles beaucoup plus larges que hauts. Notre algorithme

recompose cette suite de rectangles pour que chacun ressemble le plus possible à la proportion du carré que nous notons (1x1). Notre algorithme est récursif. Une étape de ce processus est présentée à la Figure 98 où une suite de huit rectangles est recomposée en deux suites de 5 et 3 rectangles. Chacune de ces deux suites (5 et 3) est ensuite recomposée. Notre algorithme s'arrête quand les listes à recomposer ne contiennent plus qu'un seul élément.

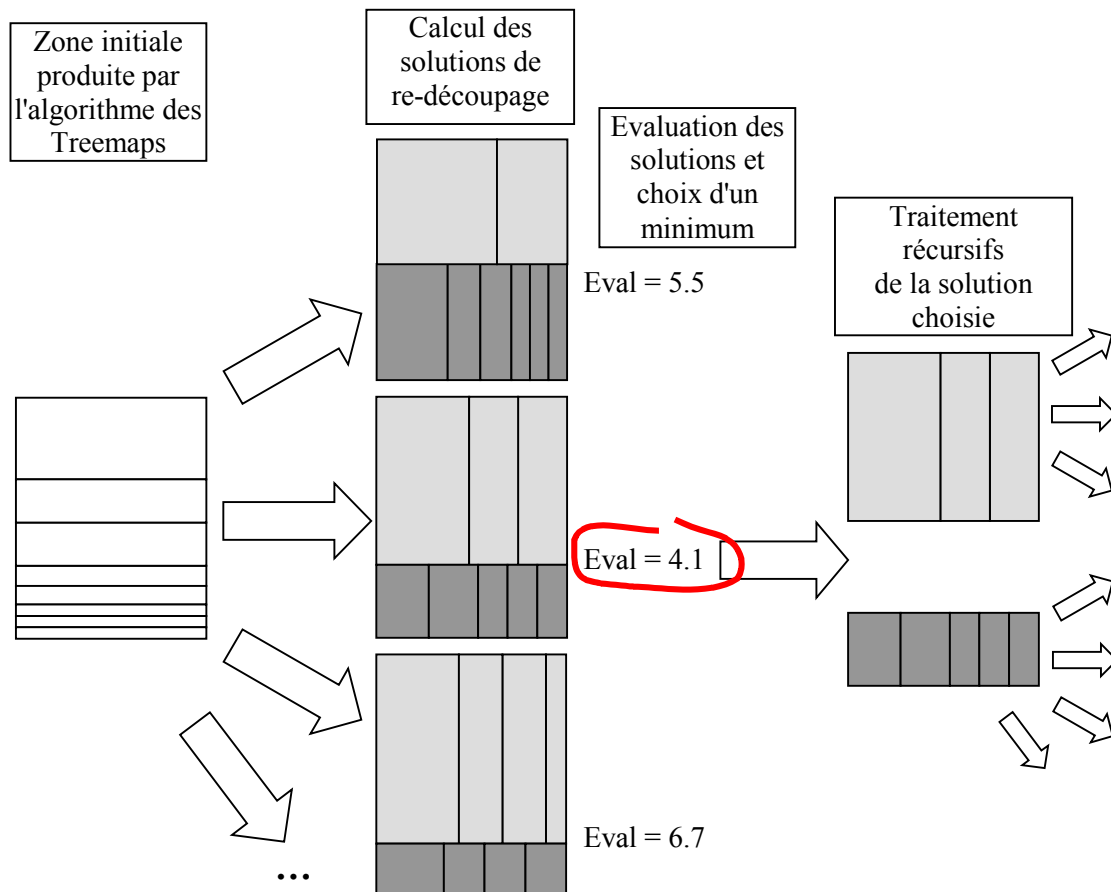


Figure 98 : Recomposition d'une suite de rectangles par l'algorithme des Fieldmaps.

Pour calculer la meilleure recomposition d'une suite de rectangles, notre algorithme teste chaque solution consistant à re-découper la suite et à la recomposer en deux suites. Chacune des deux suites est arrangée selon l'heuristique de départ de l'algorithme des Treemaps. Cette étape revient à ajouter un nœud invisible (sans cadre) dans l'algorithme des Treemaps ôté du mécanisme "Slice-Dice". Nous rappelons que l'heuristique de départ des Treemaps divise systématiquement horizontalement lorsque la zone est plus haute que large et verticalement lorsque la zone est plus large que haute.

L'évaluation de chaque solution envisagée repose sur une fonction d'évaluation. Pour une suite de rectangles, la fonction d'évaluation retourne la somme de l'évaluation de chacun des

rectangles. Pour un rectangle $R1$ (proportion $\text{largeur1} \times \text{hauteur1}$) que nous évaluons par rapport à un rectangle $R2$ (proportion $\text{largeur2} \times \text{hauteur2}$), la fonction d'évaluation est la suivante :

$$Eval(R1, R2) = \left| \frac{\text{largeur1}}{\text{hauteur1}} - \frac{\text{largeur2}}{\text{hauteur2}} \right|$$

La Figure 99 illustre l'espace de Parent avec 325 fichiers et répertoires. L'aspect des fichiers se rapproche d'un rectangle idéal horizontal ($1 \times 0,15$). La surface de chaque rectangle est proportionnelle à la taille du fichier indiquée entre parenthèses dans le label. Chaque nœud est affiché avec un cadre gris foncé fin à gauche, en bas et à droite mais épais en haut pour permettre d'afficher le nom du répertoire en gris clair sur le fond gris foncé. Le cadre d'un nœud a une apparence proche d'une fenêtre pour rappeler l'espace défini par une fenêtre d'exploration de fichiers de Windows [Windows]. Nous constatons à la Figure 99 que les répertoires contenant de nombreux fichiers (comme celui intitulé "images") présentent leur contenu par des rectangles très proches du rectangle idéal définis ($1 \times 0,15$), tandis que les répertoires qui ne contiennent que deux ou trois fichiers comme ceux en haut à droite de la Figure 99 s'éloignent de la proportion idéale.

La proportion idéale des rectangles peut être choisie par l'utilisateur grâce à une glissière qui se trouve dans une fenêtre regroupant les options et les paramètres de la modalité. D'autres paramètres que la taille idéale sont disponibles dans cette fenêtre, comme la taille minimale du texte. Le patron N°3 sur l'adaptabilité des modalités de sortie est vérifié pour ces variables. Nous avons choisi comme valeur par défaut du rectangle idéal la proportion ($1 \times 0,15$) car elle correspond aux proportions moyennes d'un label dans le cas particulier d'un nom de fichier.

Cet aspect des Fieldmaps est à mettre en relation avec les travaux de Mark Bruls sur les "Squarified Treemaps" [Bruls 00]. Notre algorithme est toutefois différent puisque la proportion du rectangle idéal est paramétrable alors que les "Squarified Treemaps" utilisent un algorithme totalement différent qui tend uniquement à produire des proportions carrées. Cette extension des Treemaps, les "Squarified Treemaps", ne corrige donc que le deuxième défaut des Treemaps énoncé à la section 1.

► /Proto/Vitesse2/DC/DPA/ISCartesian/C_ISCartesian.java.

Nous avons utilisé le trait graphique du lien pour relier les répertoires d'un chemin. Ces liens, illustrés à la Figure 100, sont des barres verticales noires : le répertoire "Vitesse2" est lié au répertoire "Proto" car "Vitesse2" est inclus dans "Proto". Ce lien n'est pas visible entre "Vitesse2" et "DC" car les deux barres de titre sont attachées mais il est encore visible entre "DC" et "DPA" et entre "DPA" et "ISCartesian". Cette mise en évidence du chemin, de la racine à l'objet pointé, ressemble à un escalier qui part en haut, à gauche de l'interface et descend vers la droite en direction de l'objet pointé par la souris.

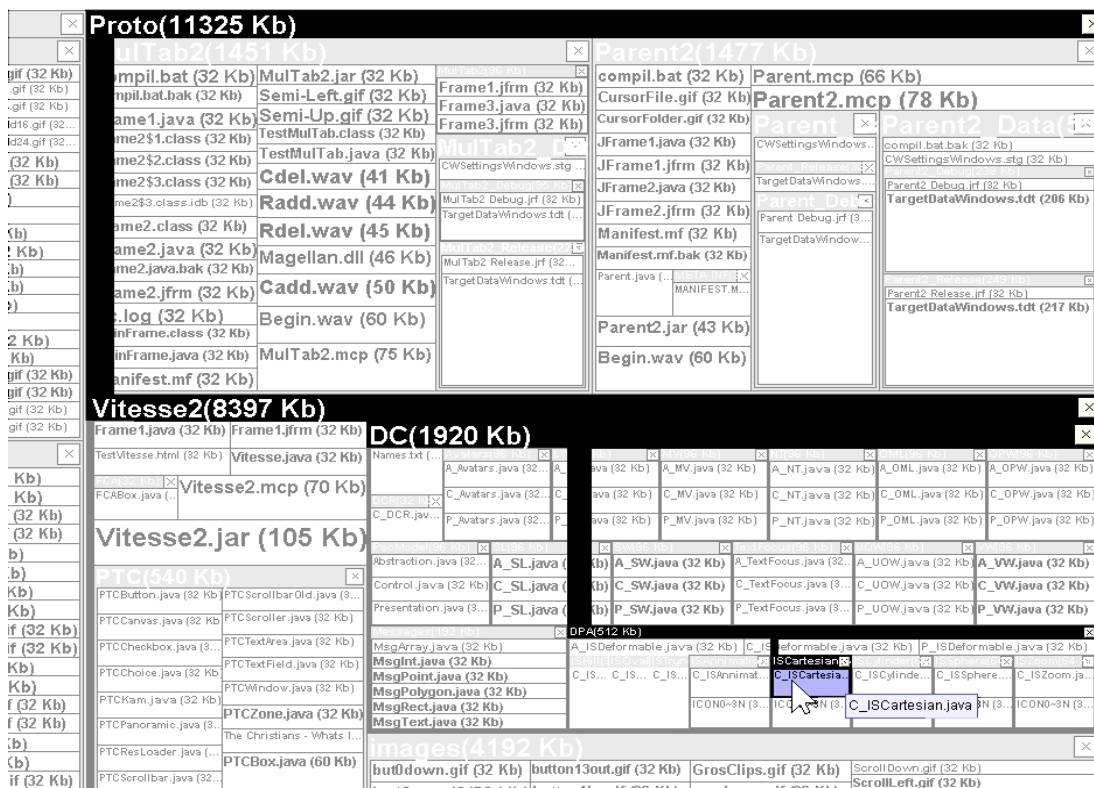


Figure 100 : Repère visuel en forme d'escalier noir mis en évidence dans l'espace de Parent pour montrer le chemin de la racine à l'objet pointé.

2.2.4 Vue alternative

La Figure 102 illustre la vue alternative qui peut remplacer une sous partie de l'espace d'information à la demande de l'utilisateur. Cette modification de l'espace correspond à une vue en liste indentée équivalente à la vue en liste de MacOS [MacOS] et à la vue du volet d'exploration des dossiers de Windows [Windows].

Nous avons choisi de placer cette vue exactement dans la même zone que la vue en Fieldmaps pour respecter le patron N°5 sur la continuité spatiale des modalités. Cette contrainte

de place entraîne que tous les éléments inclus dans ce répertoire ne peuvent pas être affichés en même temps. Comme dans Windows, et MacOS, nous plaçons une barre de défilement à droite pour atteindre tous les éléments.

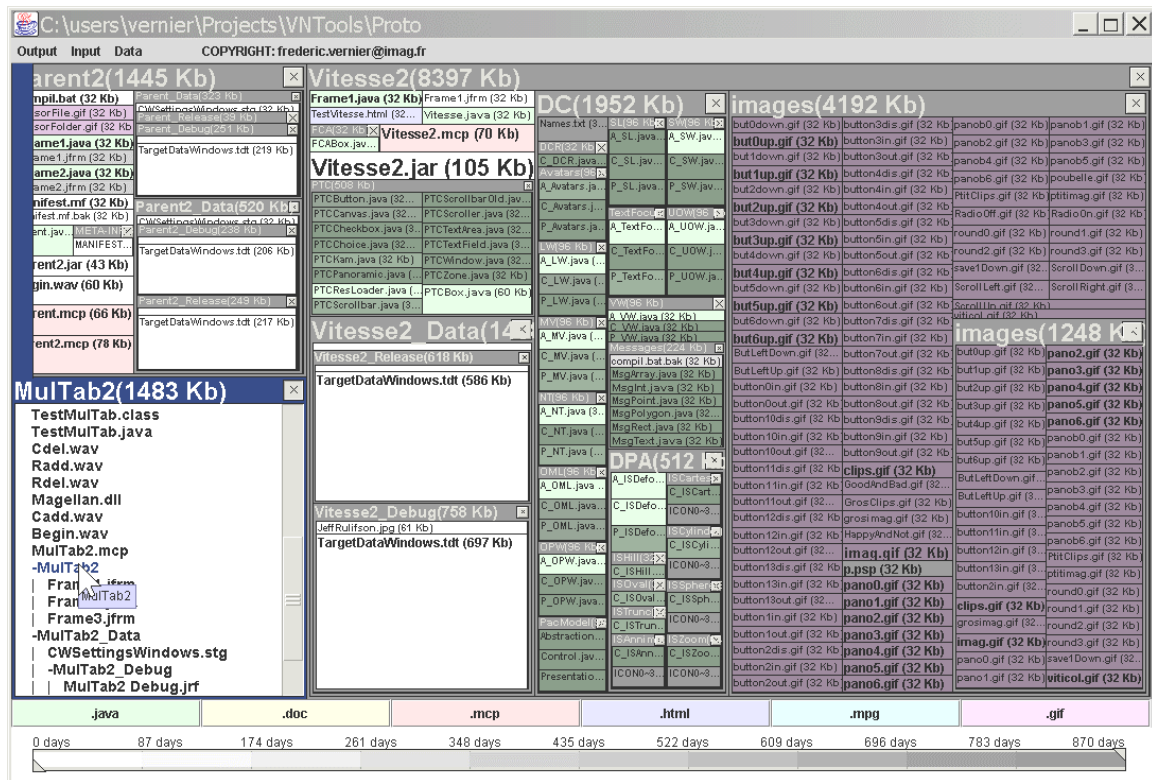


Figure 102 : Vue alternative, sous forme d'une liste indentée, du répertoire intitulé MulTab2.

2.2.5 Outils de manipulation de l'espace

Parent reprend également l'outil de TreeViz qui permet de colorier les fichiers. La Figure 95 (page 215) présente l'interface de TreeViz qui colorise sept types de fichiers (fichiers systèmes en rouge, fichiers graphique en vert, fichiers de type inconnu en jaune, etc.). La légende en bas de l'interface établit la correspondance entre les couleurs et les types de fichiers.

Parent présente deux informations dans le trait graphique de la couleur car c'est un trait graphique décomposable (teinte, luminosité et saturation). Le patron N°7 sur le plaquage des données en traits visuels nous incite à utiliser la couleur puisque les labels sont déjà très utilisés. La teinte de la couleur est utilisée pour coder le type du fichier et la luminosité pour coder le nombre de jours depuis lequel le fichier n'a pas été modifié. Parent présente donc deux légendes de couleur pour expliquer ces deux dimensions de la colorisation à l'utilisateur. Nous remarquons que ces légendes sont toujours présentes en parallèle de l'espace d'information pour vérifier le patron N°3 sur les modalités complémentaires et simultanées.

Le choix des deux légendes, présentes en bas de la Figure 103, repose sur les constats suivants :

- ▶ Le type de fichier est une information nominale.
- ▶ Le nombre de jours sans modification est une information quantitative.

Les deux légendes correspondent à des modalités différentes puisque les informations sont de natures différentes. Pour la première légende, les teintes sont présentées sur une palette de six boutons de couleurs. L'utilisateur peut glisser-déposer un fichier sur un des boutons pour changer le type de fichier associé à la couleur du bouton. L'utilisateur peut également enfoncer un bouton pour activer ou désactiver la colorisation d'un type de fichier. Le type est représenté dans le label du bouton par l'extension des fichiers de ce type (".gif" pour une image, ".mpg" pour une séquence vidéo, etc.) La seconde légende repose sur le BiSlider de MulTab (paragraphe 2.3.1, chapitre VII, page 199). Les couleurs à chaque extrême du dégradé sont le blanc et le gris qui correspondent aux luminosités des couleurs des fichiers. Nous avons suivi le patron N°9 sur la réutilisation des outils de navigation dans Parent mais pas dans MulTab. Toutefois les bénéfices attendus pour le respect de ce patron sont à posteriori les mêmes pour MulTab.

message à traduire en sons, nous avons choisi un langage d'interaction composé de notes de piano. Chaque traversée de répertoire est codée par une suite de trois notes rapides. Les notes montent dans la gamme si le déplacement se fait vers le haut (par exemple do, ré, mi). Les notes descendent dans la gamme si le déplacement se fait vers le bas (par exemple mi, ré, do). Un temps d'attente est perceptible entre chaque groupe de notes et le temps d'attente entre les notes montantes et les notes descendantes est encore plus grand. La mélodie jouée pour un déplacement qui monte deux fois dans la hiérarchie et qui descend trois fois est celle de la Figure 105.

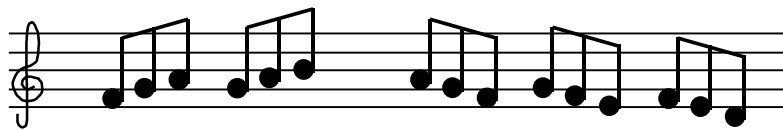


Figure 105 : Mélodie jouée pour accompagner le déplacement de fichiers.

3 *Réalisation logicielle*

L'interface étant présentée, nous présentons maintenant ses aspects de réalisation logicielle. Nous décrivons dans un premier temps, l'architecture logicielle (paragraphe 3.1) avant de présenter dans un second temps, les détails d'implémentation spécifiques à Parent (paragraphe 3.2).

3.1 *Architecture logicielle*

Dans cette section, nous ne détaillons que le Contrôleur de Dialogue (CD) en termes d'agents. Comme pour le système MulTab, le NF et l'ANF n'ont pas été développés et le CTP repose sur Swing.

La première hiérarchie d'agents du CD, que nous avons conçue, contenait un agent PAC par fichier et par répertoire. La lenteur du système obtenu a impliqué un remaniement du code. Toutes les facettes Présentation ont été fusionnées ensemble. Cette solution transgresse les principes du modèle en rendant tous les agents dépendants des uns des autres. Dans cette section, nous expliquons la solution que nous allons implémenter dans la prochaine version de Parent.

Cette solution consiste à créer un agent par répertoire et non pas un agent par fichier. En effet, le nombre de fichiers visualisés dans Parent peut être de plusieurs milliers. Le nombre de répertoires ne dépasse jamais quelques centaines. De plus la présentation des fichiers est très simple et ne varie presque jamais, ce qui nous incite à ne pas en faire un agent. De plus un agent

par répertoire facilite la mise en œuvre de sa vue alternative. La hiérarchie obtenue est présentée à la Figure 106.

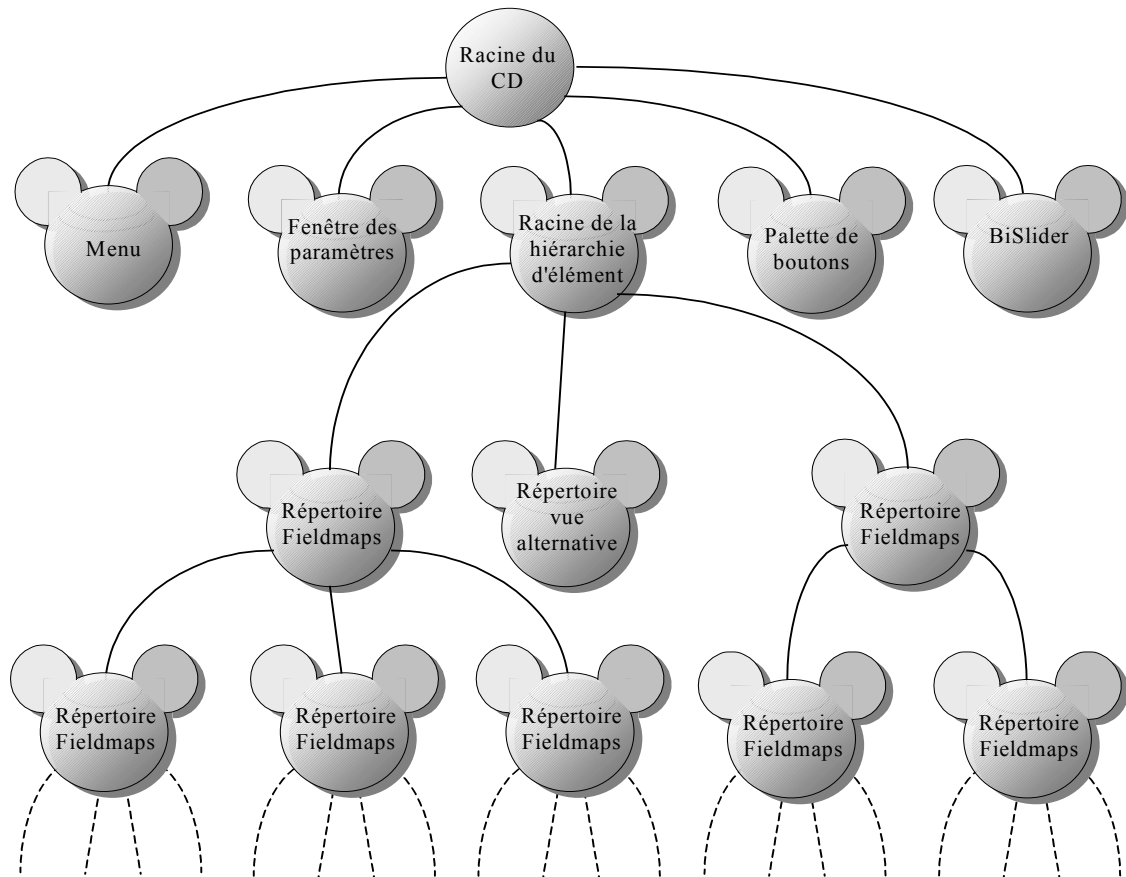


Figure 106 : Hiérarchie d'agents PAC proposée pour le système Parent.

Les agents "Menu", "Fenêtre des paramètres", "Palette de boutons" et "BiSlider" sont classiques et nous les avons déjà présentés aux chapitres précédents. Nous notons néanmoins l'utilisation intensive d'agents, instances de la même classe (agents "Répertoire Fieldmaps") auxquels peuvent se substituer un autre agent pendant l'interaction (agent "Répertoire vue Alternative").

3.2 Implémentation

Dans les paragraphes précédents, nous avons mis l'accent sur l'intégration de Parent avec des outils déjà existants dans les systèmes de gestion de fichiers de Windows [Windows] et de MacOS [MacOS]. Nous avons intégré le glisser-déposer qui était l'objectif avoué de Parent. Nous avons également mis en œuvre un système de vues multiples similaire aux vues multiples proposées par les systèmes de gestion de fichiers (voir Figure 6, page 41). Pour parfaire cette intégration, nous avons mis en œuvre d'autres mécanismes dont les détails d'implémentation

sont importants. Nous avons choisi l'environnement Windows [Windows] pour améliorer cette intégration.

Nous avons tout d'abord ajouté une option supplémentaire dans le menu contextuel associé aux répertoires. Cette possibilité permet de démarrer Parent depuis l'interface de Windows (voir Figure 107). Réciproquement, Parent permet de visualiser n'importe quel répertoire, présenté dans l'espace, avec l'interface de Windows en double cliquant sur la barre de fenêtre associée à ce répertoire.

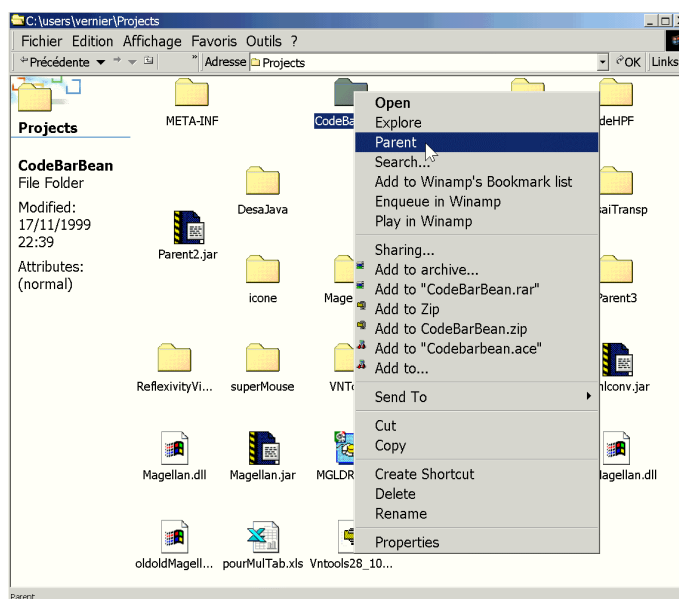


Figure 107 : Nouvelle entrée dans le menu contextuel associé à un répertoire pour lancer le système Parent.

De plus, en double cliquant sur un fichier, l'utilisateur lance l'application correspondante au fichier. C'est l'action la plus utilisée dans les systèmes graphiques de gestion de fichiers.

Enfin Parent offre la possibilité de glisser-déposer des fichiers depuis l'interface de Parent vers celle de Windows et vice-versa. L'intégration d'un outil de navigation similaire à Windows dans Parent, a rendu possible cette action inter-système. La tâche de glisser-déposer des fichiers dans la vue alternative se fait en deux actions :

- changer de vue pour la vue alternative,
- glisser-déposer les fichiers dans cette vue.

Cette tâche peut également être réalisée par les trois actions suivantes :

- ouvrir le répertoire avec Windows (une nouvelle fenêtre est ouverte)

- ▶ afficher le volet d'exploration des répertoires,
- ▶ glisser-déposer les fichiers dans la fenêtre de Windows.

Cette action à toutefois une trajectoire d'interaction plus longue et le patron N°5 sur la continuité spatiale des modalités n'est pas vérifié. La vue alternative reste donc justifiée.

4 Conclusion

4.1 Bilan

Parent est un système différent de MulTab et VITESSE car il traite des problèmes de construction de l'espace et non pas de point de vue sur l'espace. Reposant sur l'algorithme Fieldmaps, une extension des Treemaps, Parent offre une vue globale des fichiers sans trop sacrifier la lisibilité des labels. De plus il autorise des actions similaires à ceux de Windows [Windows] et MacOS [MacOS] pour la manipulation de fichiers. Nous avons vu de nouveaux types d'interaction émerger en utilisant notre modalité en sortie proche de celle de TreeViz [TreeViz] avec une modalité en entrée issue des systèmes traditionnels de gestion de fichiers (le glisser-déposer des fichiers). Nous avons également apporté plusieurs contributions originales :

- ▶ Notre algorithme des Fieldmaps permet d'améliorer la lisibilité des labels et la saisie des fichiers.
- ▶ Le repère visuel en forme d'escalier et le retour sonore après un glisser-déposer permettent d'éviter certaines erreurs de manipulation de fichier.
- ▶ Le système de colorisation des fichiers est hautement paramétrable (notamment grâce à l'outil BiSlider) et permet de faire apparaître des tendances et des motifs dans l'espace d'information.

4.2 Perspectives

Parent est un système pour lequel nous avons de nombreuses perspectives. Toutefois nous estimons avoir atteint les limites en terme d'évolutivité avec l'architecture logicielle actuelle. Aussi une réorganisation du code selon l'architecture présentée dans ce chapitre est nécessaire avant toute extension.

Nous souhaitons aussi consolider notre algorithme afin de le rendre plus rapide et mieux adapté a certains cas limites comme des répertoires avec très peu de fichiers. Lorsque nous aurons atteint cet objectif, nous pourrons comparer expérimentalement Parent avec les systèmes de référence.

Les nouvelles formes d'interaction identifiées nous incitent à approfondir l'intégration entre l'interface de TreeViz et celles des systèmes traditionnels de gestion de fichiers. Parmi les fonctionnalités qu'il nous semble possible d'adapter, nous en relevons deux qui soulèvent des problèmes intéressants de conception d'interface :

- ▶ Une fonction de recherche et filtrage équivalente à Sherlock [MacOS] ou au volet d'exploration "Rechercher" de Windows [Windows].
- ▶ Une fonction de sélection au lasso qui permet de sélectionner un groupe de fichiers.

CHAPITRE IX

CONCLUSION



Ce travail pluridisciplinaire jette un regard nouveau sur la conception de techniques de visualisation pour de grandes quantités d'information : celui de la multimodalité en sortie. A la section 1, nous dressons un bilan de nos travaux selon ces deux facettes : interface multimodale et interface de visualisation. Nous envisageons pour nos travaux de multiples perspectives. Ces dernières s'organisent en deux parties, dans la section 2 : les extensions et les prolongements à plus long terme.

1 *Contribution de la thèse*

Nos travaux de recherche ont trait à la conception d'interfaces multimodales en sortie pour manipuler de grandes quantités d'information. Dans le domaine de l'Interaction Homme-Machine (IHM), la conception de techniques d'interaction qui permettent aux utilisateurs d'explorer, comprendre et appréhender une grande quantité d'information définit un sujet d'étude important. En effet, le problème actuel n'est plus tant la disponibilité de l'information mais la capacité de sélection d'une information qui réponde au besoin grâce à la mise en œuvre d'une ou plusieurs techniques de visualisation et plus généralement d'interaction. La multiplicité des techniques de visualisation témoigne du dynamisme de cet axe d'étude.

Dans ce contexte, aux avancées brutales se traduisant par une profusion de techniques de visualisation, notre contribution concerne l'établissement de référentiels applicables à la conception afin de mieux cerner et comprendre l'espace de conception des interfaces dédiées à de grandes quantités d'information. A ces fins, nous adoptons une démarche d'analyse nouvelle consistant à considérer une technique de visualisation comme une modalité d'interaction atomique ou composée, définie par le couple (langage d'interaction, dispositif physique). Nous jetons donc un nouveau regard sur les techniques de visualisation en adoptant le point de vue de la multimodalité en sortie. Ainsi notre travail contribue au domaine de l'IHM sous deux formes complémentaires : un cadre de conception des interfaces multimodales et un cadre de conception des interfaces de visualisation. Ces deux espaces identifient et organisent des éléments de conception nécessaires à l'établissement d'un langage de patrons de conception (Chapitre II).

Nous cernons la conception d'une interface multimodale en sortie (Chapitre III) au choix motivé par des critères d'ergonomie d'une modalité simple ou composée pour un contexte d'interaction donné. Les critères de choix identifiés de nature très différente ainsi que les caractéristiques des données à rendre perceptibles définissent le contexte d'interaction. Adoptant cet approche de conception d'interfaces multimodales, nous décrivons l'espace des possibilités en définissant deux espaces de caractérisation complémentaires : l'un dédié aux modalités,

l'autre aux compositions de modalités. Ce dernier définit 25 types de composition de modalités selon deux axes orthogonaux : le schéma et l'aspect de la composition. Ces deux espaces de caractérisation définissent des éléments conceptuels génériques utiles à la mise en place de patrons de conception. Nous validons notre approche par la définition de quatre patrons de conception d'interfaces multimodales :

- ▶ Patron du test du dispositif
- ▶ Patron du canal le moins occupé
- ▶ Patron d'adaptabilité des modalités de sortie
- ▶ Patron des modalités complémentaires et simultanées

De notre cadre de conception des interfaces multimodales, nous dérivons un cadre fédérateur dédié à la conception de techniques de visualisation (Chapitre IV). Pour cela il convient d'infléchir et d'affiner nos éléments conceptuels aux cas de la visualisation de grandes quantités d'information. Ce cadre, fédérateur des approches existantes, organise les étapes du processus de visualisation. Reposant sur ces étapes de conception, deux espaces de caractérisation, l'un dédié aux techniques de visualisation, l'autre à la composition de ces techniques, cernent là encore l'espace des possibilités. Comme pour le cas plus général de la multimodalité en sortie, nous validons notre approche en définissant cinq patrons de conception d'interfaces de visualisation de grandes quantités d'information.

- ▶ Patron de la continuité spatiale des modalités de visualisation
- ▶ Patron de la continuité temporelle des modalités de visualisation
- ▶ Patron du plaquage des données en traits graphiques
- ▶ Patron de la déformation de l'espace d'information
- ▶ Patron de réutilisation des outils de navigation

Nous établissons ensuite des liens entre nos éléments conceptuels et la réalisation logicielle d'interfaces de visualisation (Chapitre V). La correspondance entre les étapes du processus de visualisation et le modèle d'architecture PAC-AMODEUS définit une solution générique pour la réalisation logicielle.

Nous validons ces résultats de conception et de réalisation logicielle avec la conception et la réalisation de trois systèmes indépendants, illustrant l'application des neuf patrons : VITESSE (Chapitre VI), MulTab (Chapitre VII) et Parent (Chapitre VIII).

- ▶ **VITESSE** est un système de visualisation d'une grande quantité de réponses retournées par un moteur de recherche du Web. Ce système est basé sur une technique de vues multiples, les unes hyperboliques, les autres à différents facteurs de zoom.
- ▶ **MulTab** est un système de visualisation d'un très grand tableau de données. Contrairement à VITESSE, l'interface repose entièrement sur une seule vue hyperbolique. Néanmoins cette vue unique est composée de multiples foci.
- ▶ **Parent** est un système de visualisation d'une arborescence de fichiers. L'espace d'information construit repose sur un nouvel algorithme, étendu de celui des Treemaps, que nous avons baptisé Fieldmaps. La composition de modalités en sortie permet l'édition des fichiers par manipulation directe.

Nous résumons nos résultats selon les étapes de notre approche de travail à la Figure 108. Nous y retrouvons la dualité de nos travaux, multimodalité et visualisation.

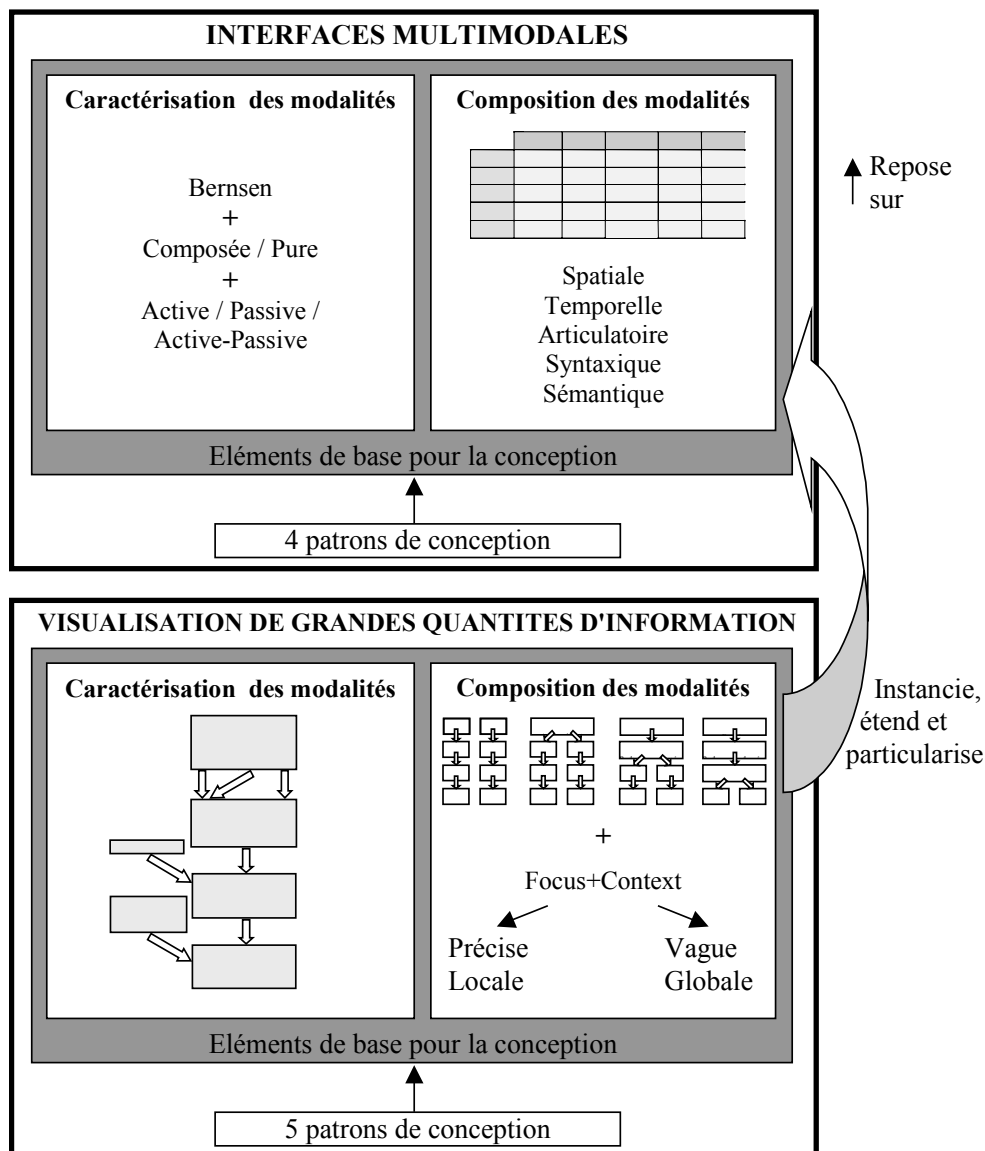


Figure 108 : Démarche d'analyse et contributions.

Pour conclure la synthèse de notre contribution, nous choisissons trois patrons de conception qui nous semblent particulièrement importants et représentatifs de nos travaux :

- Le **Patron N°2**, préconisant le choix de la modalité de sortie selon le canal sensoriel le moins occupé, constitue une motivation importante de la multimodalité en sortie.
- Le **Patron N°7**, plaquage des données en traits graphiques, explique la préférence d'une modalité graphique pour visualiser une grande quantité d'information.
- Le **Patron N°5**, continuité spatiale des modalités graphiques, correspond à de nombreux travaux en visualisation qui visent à intégrer de façon continue le focus dans son contexte.

2 *Perspectives*

Nos travaux concernent deux axes de recherche du domaine de l'IHM : la multimodalité et la visualisation. L'étude entreprise est donc ambitieuse par son ampleur. Aussi nous identifions pour nos travaux de nombreuses perspectives. Nous présentons d'abord les extensions à réaliser sans plus attendre, avant d'expliquer les prolongements à plus long terme.

2.1 *Extensions*

Les extensions à nos travaux concernent d'une part nos réalisations logicielles et d'autre part les patrons de conception.

VITESSE, MulTab et Parent

Programmation

Nous avons identifié de nombreuses extensions particulières à chaque système : elles sont donc exposées en conclusion des chapitres décrivant les systèmes. Un travail de réalisation logicielle nécessaire aux trois systèmes vise à garantir la fluidité de l'interaction (temps de réponse dans la boucle d'interaction). Pour VITESSE, la fluidité de l'interaction nous semble satisfaisante. Pour MulTab, atteindre le même niveau de fluidité que VITESSE implique l'utilisation d'un ordinateur plus puissant. Parent, enfin, nécessite un travail plus important afin d'atteindre un temps de réponse acceptable dans le cas de la manipulation d'une très grande hiérarchie. Sans un temps de réponse correct, le système ne sera pas utilisé dans des tâches quotidiennes de manipulation de fichiers.

Outre une acceptation du système par l'utilisateur, la fluidité dans l'interaction est incontournable pour notre évaluation expérimentale par comparaison avec des systèmes existants.

Evaluation

Dans la présentation de nos systèmes, nous avons insisté sur leurs comparaisons avec des systèmes existants (VITESSE avec AltaVista [AltaVista], MulTab avec Excel [Excel]. et Parent avec Windows [Windows]. et MacOS [MacOS]). Ces comparaisons à priori impliquent naturellement une évaluation expérimentale à posteriori. Nous allons donc entreprendre des tests expérimentaux qui pour chaque système considéreront deux environnements de réalisation du scénario : l'un de nos systèmes et un système existant offrant des possibilités équivalentes sur les mêmes données. Ainsi VITESSE peut être comparé expérimentalement avec AltaVista, MulTab avec Excel et Parent avec Windows ou MacOS. Afin de définir des scénarios réalistes aboutissant à des tâches naturelles de nos sujets, il conviendra d'utiliser des données leur

appartenant. Les sujets seront donc des experts dans la manipulation de leurs données avec le système existant. Aussi une phase d'apprentissage de nos systèmes sera donc nécessaire afin de gommer les différences d'expertise dans la manipulation des deux systèmes comparés.

Patrons de conception

Patrons illustrés par un système fonctionnel

Comme nous l'avons expliqué, il est parfois difficile d'illustrer un patron de conception d'interfaces avec une copie d'écran. En effet, une présentation graphique statique peut parfois être inadéquate car l'interaction est par définition dynamique. Ce problème est expliqué dans la section 3 du chapitre II (page 22). De plus la validité d'un patron passe surtout par son usage, qui implique sa diffusion, mais aussi par sa mise en relation avec d'autres patrons pour former un langage de patrons.

Ces deux constats nous incitent à mettre nos patrons à disposition sur Internet en utilisant nos systèmes comme exemples puisqu'ils sont inclus dans des documents Web grâce au mécanisme d'Applet. Nous envisageons alors d'affiner la rubrique "Exemples illustratifs" (contenant au moins trois exemples) en distinguant les exemples cités, illustrés et fonctionnels.

Elargissement du langage de patrons

Nous avons identifié neuf patrons de conception. Notre expérience pratique, avec la conception et la réalisation de trois systèmes, nous permet d'envisager d'autres patrons. Il convient donc que nous restions attentifs à nos pratiques de conception afin de découvrir et d'écrire d'autres patrons qui puissent enrichir notre langage. Ce travail commencera bien évidemment avec les extensions d'implémentation que nous avons présentées ci-dessus.

2.2 Prolongements

Nos prolongements s'organisent selon deux axes : le premier axe de recherche traite des patrons de conception logicielle, tandis que le deuxième concerne l'application notre démarche d'analyse à d'autres domaines que la visualisation comme l'informatique mobile et la réalité augmentée.

Patron de conception logicielle

Dans le Chapitre V, la mise en place d'une architecture logicielle commune à nos systèmes nous a conduits à établir une correspondance entre les composants logiciels du modèle d'architecture logicielle PAC-AMODEUS et les étapes du processus de visualisation de notre cadre fédérateur (Chapitre IV). Nous avons donc établi des liens explicites entre le modèle PAC-AMODEUS et nos éléments de conception. Comme nos patrons se fondent sur nos

éléments de conception, leur mise en relation avec le modèle PAC-AMODEUS dessine les contours d'une démarche de traduction de nos patrons de conception d'interfaces en patrons de conception logicielle. En adhérant à cette approche, les patrons de conception logicielle seront donc dépendants du modèle PAC-AMODEUS. Le modèle PAC-Amodeus n'est bien sûr pas le seul modèle d'architecture avec lequel nous pouvons entreprendre ces travaux. Néanmoins ce modèle hybride semble être particulièrement adapté car il localise les étapes du processus de visualisation dans des composants logiciels distincts et supporte un affinement en agents logiciels des techniques de visualisation et de leurs compositions. Nous nous rapprocherons alors des patrons proposés par Gamma [Gamma 95] qui adressent le problème de la conception logicielle orientée objet.

Traduire les patrons de conception d'interfaces en patrons de conception logicielle constitue une approche prometteuse pour établir des passerelles entre la conception ergonomique centrée sur l'utilisateur et la conception logicielle. L'étude d'un langage de patrons de conception d'interfaces et de conception logicielle définit, à notre connaissance, un axe de recherche nouveau.

Multimodalité en sortie : un regard nouveau sur l'informatique mobile

L'informatique mobile est un domaine en plein essor. La multiplication des ordinateurs de poche en témoigne. Par exemple, le nouvel ordinateur de poche de Compaq, le Ipack, a été épuisé en Europe comme aux Etats-Unis dès sa sortie sur le marché. L'explosion de la téléphonie mobile constitue un autre exemple significatif de la formidable extension de ce domaine.

Dans ce domaine en pleine effervescence, la mise en place de référentiels applicables à la conception des interfaces nous semble donc cruciale. Adopter notre démarche d'analyse, en affinant notre cadre de conception d'interfaces multimodales au cas de l'informatique mobile se justifie doublement :

1) Ces artéfacts reposent sur des dispositifs physiques de sortie aux capacités restreintes, sortant du cadre classique d'un grand écran, d'un clavier et d'une souris. Adopter notre démarche d'analyse, basée sur une modalité définie comme le couple (langage, dispositif) nous conduira à caractériser finement les dispositifs physiques, puis à caractériser les langages d'interaction selon les dispositifs. Ce travail permettra donc d'étendre nos espaces de caractérisation, en particulier celui des dispositifs, qui pour l'instant est simplement basé sur les capacités sensorielles mises en jeu.

2) Le contexte d'utilisation de ces artefacts est par définition très variable. Comme nous l'avons souligné dans notre typologie de la multimodalité, l'adaptabilité et l'adaptativité constituent une motivation importante de la multimodalité. Dans cet étude, il conviendra de considérer l'adaptabilité et l'adaptativité aux contextes physiques d'utilisation. L'informatique mobile est donc un domaine privilégié de la multimodalité. Pour s'en convaincre, considérons le mode vibration qui existe sur la plupart des téléphones mobiles actuels !

Multimodalité en sortie : un regard nouveau sur la réalité augmentée

La réalité augmentée (RA) est un paradigme d'interaction qui est né de la volonté de fusionner les capacités de traitements informatiques et l'environnement physique. En réponse à l'avancée des nouvelles technologies d'interaction, les applications faisant usage du paradigme de RA sont de plus en plus diversifiées. Chaque domaine d'application adapte la RA à ses besoins et développe des principes de conception ad hoc. Un espace fédérateur de conception de systèmes de RA fait actuellement défaut, ce constat motivant cet axe de recherche.

Afin d'étudier la conception de systèmes de RA qui visent la fusion harmonieuse et continue du monde informatique et du monde réel, nous considérons la composition des modalités de sortie du système avec le monde réel perçu par l'utilisateur. Aussi nous faisons l'analogie entre percevoir un énoncé multimodal, et percevoir le monde réel augmenté par l'ordinateur grâce à l'usage de modalités de sortie. Nous constatons que notre espace de composition définissant 25 types de composition de modalités est un outil de conception pour la Réalité Augmentée. La composition spatiale par exemple, a été étudiée dans le système de chirurgie augmentée CASPER [Dubois 99] : une composition spatiale imbriquée a été définie entre la modalité de sortie basée sur un casque semi-transparent (dispositif) et la perception du champ opératoire. Afin d'appliquer notre cadre de conception, la première étape de travail consistera à caractériser la perception du monde réel ayant un impact sur l'accomplissement de la tâche de l'utilisateur en deux niveaux d'abstraction, comme nous l'avons fait pour une modalité de sortie ; (langage, dispositif).

REFERENCES

Nous avons organisé les références utilisées dans ce mémoire en deux parties distinctes. Compte tenu du grand nombre d'applications graphiques utilisées comme exemples, nous les avons distingués des articles de recherche. Aussi la section 1 est dédiée aux articles de recherche tandis que la section 2 présente la liste des systèmes référencés. Dans le manuscrit, les 26 références aux systèmes ne sont pas datées alors que les 106 références aux articles se terminent par les deux derniers chiffres de l'année de publication.

La seconde spécificité de cette bibliographie est le grand nombre d'adresses Internet. Si ces adresses vous permettent d'obtenir très rapidement l'article référencé, elles nécessitent parfois d'être membre d'une société savante (ACM ou IEEE).

1 Communications de recherche référencées

[Abowd 92]

Abowd G., Coutaz J. et Nigay L., “***Structuring the Space of Interactive System Properties***”. Actes de la conférence TC2/WG2.7 Engineering for Human Computer Interaction, édités par J. Larson & C. Unger, 1992, pp. 113-128.

[Ahlberg 94]

Ahlberg C. et Shneiderman B., “***Visual Information Seeking: Tight coupling of dynamic query filters with starfield displays***”. Actes de la conférence ACM CHI'94 (Boston, USA), 1994, pp 313-317 + color plates. Reprinted in Baecker, R. M., Grudin, J., Buxton, W. A. S., and Greenberg, S. (Editors), *Readings in Human-Computer Interaction: Toward the Year 2000*, Second Edition, Morgan Kaufmann Publishers, Inc., (San Francisco, USA), 1995, pp 450-456. <http://www.cs.umd.edu/hcil/spotfire/>.
<ftp://ftp.cs.umd.edu/pub/papers/papers/ncstr1.umcp/CS-TR-3131/CS-TR-3131.ps.Z>.

[Allen 83]

Allen J., “***Maintaining Knowledge about Temporal Intervals***”. *Journal Communication of the ACM* 26(11), Novembre 1983, pp. 832-843.

[Alexander 77]

Alexander C., Ishikawa S. et Silverstein M., “***A Pattern Language: Towns, Building, Construction***”. Livre édité par Oxford University Press, 1977. ISBN 0-19-501-919-9.

[André 98]

André E., Rist T. et Müller J., “***Guiding the User through Dynamically Generated Hypermedia Presentations with a Life-Like Presentation Agent***”. Actes de la conférence IUI'98 (San Francisco, USA), 1998, pp. 21-28. <http://www.dfki.de/imedia/papers/iui98.ps.gz>.

[Apperley 82]

Apperley M., Tzavaras I. et Spence R., “***A Bifocal Display Technique for Data Presentation***”. Actes de la conférence Eurographics'82 (Amsterdam, Hollande), pp. 27-43. Egalement présenté à : <http://www.ee.ic.ac.uk/research/information/www/Bobs.html>.

[Baldonado 00]

Baldonado M., Woodruff A. et Kuchinsky A., “***Guidelines for Using Multiple Views in Information Visualization***”. Actes de la conférence AVI 2000 (Palerme, Italie), 2000, pp 110-119.

[Barnard 85]

Barnard P.J., “ *Cognitive Ressources and the Learning of Computer Dialogs* ”. Livre Interfacing Thought, Cognitive aspects of Human Computer Interaction. J. M. Carroll, édité par MIT Press, 1985, pp. 112-158.

[Barnard 92]

Barnard P.J. et May J. “ *Real time blending of data streams: a key problem for the cognitive modelling of user behaviour with multimodal systems* ”. Article de travail du groupe Amodeus, UM/WP 10, 1992.

ftp://ftp.mrc-apu.cam.ac.uk/pub/amodeus/usemod/um_wp10.rtf.

[Barnard 93]

Barnard P. et May J., “ *Cognitive Modelling for User Requirements* ”. Livre édité par Byerley, Barnard, et May, Computers, Communication and Usability: Elsevier.

[Barnard 94]

Barnard P.J., May J. et Salber D. “ *Deixis and Points of View in Media Spaces, Amodeus Project* ”. Article de travail du groupe Amodeus, UM/WP 19, 1994.

ftp://ftp.mrc-apu.cam.ac.uk/pub/amodeus/usemod/um_wp19.rtf.

[Bayle 97]

Bayle E., Bellamy R., Casaday G., Erickson T., Fincher S., Grinter B., Gross B., Lehder D., Marmolin H., Potts C., Skousen G. et Thomas, J., “ *Putting It All Together: Towards a Pattern Language for Interaction Design* ”. Rapport résumé de l’atelier de la conférence CHI’97. (Atlanta, USA) 22-23 Mars 1997.

http://www.pliant.org/personal/Tom_Erickson/Patterns.WrkShpRep.html

Dans le bulletin SIGCHI de janvier 1998 :

<http://www.acm.org/sigchi/bulletin/1998.1/erickson.html>

[Beck 87]

Beck K. et Cunningham W., “ *Using pattern languages for object-oriented programs* ”. Rapport technique CR-87-43, Tektronic Inc. 17 Septembre 1987.

<http://c2.com/doc/oopsla87.html>.

[Beck 94]

Beck K. et Johnson R., “ *Patterns Generate Architectures* ”. Actes de la conférence ECOOP’94 (Bologna, Italie), 1994, pp. 139-149.

<ftp://st.cs.uiuc.edu/pub/patterns/papers/patterns-generate-archs.ps>.

[Bender 00]

Bender M., Klein R., Disch A. et Ebert A., “ *A Functional Framework for Web-Based Information Visualization Systems* ”. Journal IEEE Transactions On Visualization And Computer Graphics 6(1), Janvier-Mars 2000.

<http://dlib.computer.org/tg/books/tg2000/pdf/v0008.pdf>.

[Bérard 99]

Bérard F., “ *Vision par ordinateur pour l’interaction homme-machine fortement couplée* ”. Thèse de l’Université Joseph Fourier. Soutenue le 30 novembre 1999 à Grenoble.

http://iihm.imag.fr/publs/1999/THESE1999_Berard.pdf.

[Bernsen 93]

Bernsen N., “ *A revised generation of the taxonomy of output modalities* ”. Rapport du projet Esprit AMODEUS numéro RP5-TM-WP11, 1994, 13 pages.

ftp://ftp.mrc-apu.cam.ac.uk/pub/amodeus/taxmod/tm_wp11.rtf.

[Bertin 77]

Bertin J., “ *La graphique et le traitement graphique de l'information* ”. Livre édité par Flammarion, 1977.

Version anglaise : “ *Graphics and Graphic Information Processing* ”. Livre édité par Walter de Gruyter. Berlin. (New York, USA), 1981, 273 pages. ISBN 3110069016

[Bier 93]

Bier E., Stone M., Pier K., Buxton W. et DeRose T., “ *Toolglass and Magic Lenses: The See-Through Interface* ”. Actes de la conférence ACM Siggraph'93 (Anaheim, USA), 1993, pp 73-80. <http://www.parc.xerox.com/istl/projects/MagicLenses/93Siggraph.html>.

[Boehm 81]

Boehm B. W., “ *Software Engineering Economics* ”. Livre édité par Prentice-Hall, 1981. ISBN 0-13-822122-7

[Bourget 92]

Bourget M-L, “ *Conception et réalisation d'une interface de dialogue personne-machine multimodale* ”. thèse de l'Institut National Polytechnique de Grenoble, 1992, 206 pages.

[Bruley 99]

Bruley C., “ *Analyse des représentations graphiques de l'information – Extension aux représentations tridimensionnelles* ”. Thèse de l'Université Joseph Fourier. Soutenue le 3 juin 1999 à Grenoble. <ftp://ftp.inrialpes.fr/pub/romans/theses/bruley.pdf>.

[Bruls 00]

Bruls M., Huizing K. et van Wijk J., “ *Squarified Treemaps* ”. Actes de la conférence VisSym'00 (Amsterdam, Hollande), 2000, A paraître. <http://www.win.tue.nl/~vanwijk/stm.pdf>.

[Cadoz 94]

Cadoz C., “ *Le geste canal de communication homme/machine. La communication instrumentale* ”. Journal TSI 13(1), 1994, pp. 31-61.

[Card 97]

Card S. et Mackinlay J., “ *The Structure of the Information Visualization Design Space* ”. Actes de la conférence InfoVis'97 (Phoenix, Arizona), 1997, pp. 92-99. <http://dlib.computer.org/conferen/infovis/8189/pdf/81890092.pdf>.

[Carpendale 97]

Carpendale M., Cowperthwaite D. et Fracchia F. “ *Making Distortions Comprehensible Space* ”. Actes de la conférence IEEE Visual Languages'97 (Ile de Capri, Italie), 1997, pp. 36-45. <http://www.cpsc.ucalgary.ca/~sheelagh/personal/pubs/visl97.ps.gz>.

[Chatty 99]

Chatty S., Athènes S. et Bustico A., “ *Votre attention s'il vous plaît ! Eléments d'un espace de conception de signaux visuels* ”. Actes de la conférence IHM'99 (Montpellier, France), 1999, pp. 17-24. <http://www.tls.cena.fr/divisions/PII/Rapports/NR99-658.ps.gz>.

[Chi 98]

Chi E. et Riedl J., “ *An Operator Interaction Framework for Visualization Systems* ”. Actes de la conférences InfoVis '98 (Research Triangle Park, USA), 1998, pp. 63-70. <http://www-users.cs.umn.edu/~echi/papers/infovis98/operator.pdf>.

- [Coplien 96]
Coplien J., “ *Software Patterns* ”. Livre édité par SIGS Books & Multimedia,
<http://www.bell-labs.com/user/cope/Patterns/WhitePaper/SoftwarePatterns.pdf>.
- [Coutaz 87]
Coutaz J., “ *PAC: an Implementation Model for Dialog Design* ”. Actes de la conférence Interact'87, H-J. Bullinger, B. Shackel ed., North Holland (Stuttgart, Allemagne), 1987, pp. 431-436.
- [Coutaz 94]
Coutaz J. et Nigay L., “ *les proprietes “care” dans les interfaces multimodales* ”. Actes de la conférence IHM'94 (Lille, France), pp 7-14.
http://iihm.imag.fr/publs/1994/IHM94_CARE.Fr.pdf.
- [Coutaz 95]
Coutaz J., Nigay L., Salber D., Blandford A., May J. et Young R., “ *Four easy pieces for assessing the usability of multimodal interaction: The CARE properties* ”. Actes de la conférence Interact'95 (Lillehammer, Norvège), 1995, pp. 115-120.
http://iihm.imag.fr/publs/1995/Interact95_CARE.ps.
- [Coutaz 98]
Coutaz J., “ *Interfaces Homme-Machine : le Futur ne Manque pas d'Avenir* ”. Actes de la conférence ERGO-IA'98 (Biarritz, France), 1998, pp. 43-55.
http://iihm.imag.fr/publs/1998/ErgoIA_Coutaz.pdf.
- [Dieterich 93]
Dieterich H., Malinowski U., Kühme T. et Schneider-Hufschmidt M. “ *State of the Art in Adaptive User Interfaces* ”. Extrait du livre Adaptive User Interfaces: Principles and Practice édité par Schneider-Hufschmidt (North Holland), 1993, pp 13-48.
- [Dix 93]
Dix A., Finlay J., Abowd G. et Beale R., “ *Human-Computer Interaction* ”. Livre édité par Prentice Hall, New York, 1993. ISBN 0-13-458266-7.
- [Dourish 92]
Dourish P. et Bellotti V. “ *Awareness and Coordination in Shared Workspaces* ”. Actes de la conférence CSCW'92 (Toronto, Canada), 1992, pp. 107-114.
<ftp://ftp.parc.xerox.com/pub/dourish/cscw92-awareness.ps>.
- [Dubois 99]
Dubois E., Nigay L., Troccaz J., Chavanon O. et Carrat L., “ *Classification Space for Augmented Surgery, an Augmented Reality Case Study* ”. Actes de la conférence Interact'99 (Edimbourg, Angleterre), 1999, pp. 353-359.
http://iihm.imag.fr/publs/1999/INTERACT99_DUBOIS_Classif_RA.pdf.
- [Engelbart 62]
Engelbart D., “ *Augmenting Human Intellect: A Conceptual Framework* ”. Rapport du contrat AF 49(638)-1024, SRI Project 3578 (Menlo Park, USA), Octobre 1962.
<http://www.histech.rwth-aachen.de/www/quellen/engelbart/ahi62index.html>.
- [Feiner 91]
Feiner S. et McKeown K. “ *COMET: generating coordinated multimedia explanations* ”. Actes de la conférence CHI'91 (New Orleans, USA), 1991, pp. 449-450.

[Fincher 99]

Fincher S., “ *What is a Pattern Language* ”. Article soumis à CHI'99 et publié sur Internet.
<http://www.tk.uni-linz.ac.at/~jan/patterns/fincher.pdf>.

[Frohlich 91]

Frohlich D., “ *The Design Space of Interfaces, Multimedia Systems, Interaction and Applications* ”, Actes de la conférence Eurographics Workshop (Stockholm, Suède), 1991, pp. 53-69.

[Furnas 86]

Furnas G., “ *Generalized Fisheye Views* ”. Actes de la conférence CHI'86 (Boston USA), 1986, pp. 16-23. <http://www.acm.org/pubs/articles/proceedings/chi/22627/p16-furnas/p16-furnas.pdf>.

[Furnas 94]

Furnas G. et Zacks J., “ *Multitrees: Enriching and reusing hierarchical structure* ”. Actes de la conférence CHI '94 (Boston, USA), 1994, pp. 330-336.
<http://www.acm.org/pubs/articles/proceedings/chi/191666/p330-furnas/p330-furnas.pdf>.

[Gamma 95]

Gamma E., Helm R., Johnson R. et Vlissides J., “ *Design Patterns: Elements of Reusable Object-Oriented Software* ”. Livre édité par Addison-Wesley, 1995, 416 pages, ISBN 0-201-63361-2.

[Garido 97]

Garido L., “ *Evaluation ergonomique par prototypage d'une interface destinée à rendre perceptible un espace d'information de grande taille : utilisation de la multimodalité* ”. Rapport de DESS en ergonomie de l'université René Descartes, Août 1997, 53 pages.

[Gram 94]

Gram C. et Cockton G., “ *Design Principles for Interactive Software* ”. Livre édité par Chapman & Hall pour IFIP, 1994, pp. 133-187. ISBN : 0-412-72470-7.

[Grudin 89]

Grudin J., “ *The Case Against User Interface Consistency* ”. Journal Communication of the ACM 32(10), 1989, pp. 1164-1173. <http://www.acm.org/pubs/articles/journals/cacm/1989-32-10/p1164-grudin/p1164-grudin.pdf>.

[Herman 00]

Herman Y., Melançon G. et Marshall M.S., “ *Graph Visualization and Navigation in Information Visualization: A Survey* ”. Journal IEEE Transaction on Visualisation and Computer Graphism. 6(1), Janvier / Mars 2000, pp. 24-43.
<http://www.cwi.nl/InfoVisu/Survey/StarGraphVisuInInfoVis.html>.

[Hovestadt 95]

Hovestadt V., Gramberg O. et Deussen O., “ *Hyperbolic User Interfaces for Computer Aided Architectural Design* ”. Article court. Actes II companion de la conférence CHI'95 (Denver, USA), 1995, pp. 304-305. <http://www.ubka.uni-karlsruhe.de/vvv/1995/architektur/6/6.pdf>.

[Igarashi 98]

Igarashi T., Zellweger P., Chang B. et Mackinlay J., “ *Fluid Visualization of Spreadsheet Structures* ”. Actes de la conférence Visual Languages'98 (Halifax, Canada), 1998, pp. 118-125. <http://www.parc.xerox.com/istl/projects/uir/pubs/pdf/UIR-R-1998-07-Mackinlay-VL98-FluidSpreadsheet.pdf>.

[Jambon 96]

Jambon F., “ *Erreurs et Interruptions du Point de Vue de l'Ingénierie de l'Interaction Homme-Machine* ”. Thèse de l'Université Joseph Fourier, Grenoble 1, 1996.
http://iihm.imag.fr/pubs/1996/THESE1996_Jambon.pdf.

[Johnson 91]

Johnson B. et Shneiderman B., “ *Treemaps: a space-filling approach to the visualization of hierarchical information structures* ”. Actes de la conférence Visualization'91 (San Diego, USA), 1991, pp. 284-291. <ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/91-06html/91-06.html>.

[Keahey 96]

Keyhey A. et Robertson L., “ *Techniques for Non-Linear Magnification Transformations* ”. Actes de la conférence InfoVis'96 (San Francisco, USA), 1996, pp. 38-45.
<ftp://ftp.cs.indiana.edu/pub/tkeahey/papers/infovis.96.pdf>.

[Kerpedjiev 00]

Kerpedjiev S. et Roth S., “ *Mapping Communicative Goals into Conceptual Tasks to Generate Graphics in Discourse* ”. Actes de la conférence IUI '00 (New Orleans, USA), 2000, pp. 60-67.

[Koenig 95]

Koenig A., “ *Patterns and Antipatterns* ”. Journal Object-Oriented Programming 8(1), 1995, pp 46-48.

[Leung 94]

Leung Y. K. et Apperley M. D., “ *A Review and Taxonomy of Distortion-Oriented Presentation Techniques* ”. Journal ACM Transaction on Computer-Human Interaction. 1(2), Juin 1994, pp. 126-160. <http://www.acm.org/pubs/articles/journals/tochi/1994-1-2/p126-leung/p126-leung.pdf>.

[Papka 00]

Papka M. et Stevens R., “ *Next-Generation Visualization Displays: The Research Challenges of Building Tiled Displays* ”. Panel organisé lors de la conférence Visualization'2000 (Salt Lake City, USA) et introduit dans les actes de la conférences, pp. 527-528.

[Mahemoff 99]

Mahemoff M. et Johnston L., “ *Pattern Languages for Usability: An Investigation of Alternative Approaches* ”. Actes de la conférence Asia-Pacific Conference on Human Computer Interaction (Tanaka, Japon), 1998.
<http://www.cs.mu.oz.au/~moke/papers/candidate/>

[Mackinlay 86]

Mackinlay J., “ *Automating the Design of Graphical Presentations of Relational Information* ”. Journal ACM Transaction on Graphics 5(2), 1986, pp. 110-141.
<http://www.parc.xerox.com/istl/projects/uir/pubs/pdf/UIR-R-1986-02-Mackinlay-TOG-Automating.pdf>.

[Mackinlay 91]

Mackinlay J. et Robertson G. et Card S. “ *The Perspective Wall: Detail and context smoothly integrated* ”. Actes de la conférence CHI '91 (New York, USA), 1991, pp. 173–179. <http://www.parc.xerox.com/istl/projects/uir/pubs/pdf/UIR-R-1991-03-Mackinlay-CHI91-PerspectiveWall.pdf>.

[MacLean 90]

MacLean A., Young R., Bellotti V. et Moran T., “ *Questions, Options and Criteria: Elements of Design Space Analysis* ”. Journal Human Computer Interaction 6(3&4), 1990, pp. 201-250.

[Martin 94]

Martin J.C., “ *Etude de la multimodalité fondée sur des types et buts de coopération entre modalités* ”. Actes des troisièmes Journées Internationales sur l'Interface des Mondes Réels et Virtuels, 7-11 février, Montpellier, France. pp 97-106.

[Martin 99]

Martin J.C., “ *Six primitive types of cooperation for observing, evaluating and specifying cooperations* ”. Actes de la conférence AAAI Fall 1999 (Cape Cod, USA), 1999, pp. 61-68. <http://www.limsi.fr/Individu/martin/aaai99/html/martin-final-v7.html>.

[Maybury 93]

Maybury M., “ *Intelligent Multimedia Interfaces* ”. Livre édité par Menlo PARC, AAAI Press, 1993.

[Maybury 99]

Maybury M. et Stock O., “ *Multimedia Communication, including Text* ”. Chapitre 9 du rapport pour "US National Science Foundation", " European Commission's Language Engineering Office" et de "US Defense Advanced Research Projects Agency". Avril 1999. <http://www.cs.cmu.edu/~ref/mlim/chapter9.html>.

[Meszaros 98]

Mezaros G. et Doble J., “ *A Pattern Language for Pattern Writing* ”. Actes de la conférence PloP'98, Addison-Wesley, 1998. pp. 529-574. <http://www.joeyoder.com/papers/patterns/patwrit/>.

[Mynatt 95]

Mynatt E., “ *Transforming graphical interfaces into auditory interfaces* ”. Thèse de Georgia Institute of Technology, 1995.

[Nanard 98]

Nanard M., Nanard J. et Kahn P., “ *Pushing reuse in hypermedia design : golden rules, design patterns and constructive templates* ”. Actes de la conférence Hypertext'98 (Pittsburgh, USA), 1998, pp. 11-20. <http://webfuse.cqu.edu.au/Information/Resources/Readings/papers/p11-nanard.pdf>.

[Neal 88]

Neal J., Dobes Z., Bettinger K. et Byoun J., “ *Multi-Modal References in Human-Computer Dialogue* ”. Actes volume 2 de la conférence Artificial Intelligence'88 (Saint Paul, USA), Août 1988.

- [Nigay 94]
Nigay L., “ *Conception et modélisation logicielles des systèmes interactifs : Application aux interfaces multimodales* ”. Thèse de l’Université Joseph Fourier. Soutenue le 28 janvier 1994 à Grenoble. http://iihm.imag.fr/publs/1994/THESE1994_Nigay.pdf.
- [Nigay 95]
Nigay L. et Coutaz J. “ *A Generic Platform for Addressing the Multimodal Challenge* ”. Actes de la conférence CHI'95 (Denver, USA), 1995, pp. 98-105.
http://iihm.imag.fr/publs/1995/CHI95_Fusion.pdf
- [Nigay 96]
Nigay L. et Coutaz J., “ *Espaces conceptuels pour l'interaction multimédia et multimodale* ”. Journal TSI, spécial Multimédia et Collecticiel, AFCET & Hermès Publ. 15(9), 1996, pp. 1195-1225. <http://iihm.imag.fr/publs/1996/TSI96-Mu.pdf>.
- [Nigay 98]
Nigay L. et Vernier F., “ *Design method of interaction techniques for large information spaces* ”. Actes de la conférences AVI'98 (L'Aquila, Italie), 1998.
<http://iihm.imag.fr/publs/1998/AVI98.pdf>.
- [Patterson 94]
Patterson J.F., “ *A taxonomy of Architectures for Synchronous Groupware Applications* ”. Atelier Software Architectures de la conférence CSCW'94 (Chapel Hill, USA), 1994.
- [Rao 94]
Rao R. et Card S., “ *The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information* ”. Actes de la conférence ACM CHI '94 (Boston, USA), 1994, pp. 318-322 Rapport technique UIR-R-1994-07-Rao-CHI94-TableLens : <http://www.parc.xerox.com/istl/projects/uir/pubs/pdf/UIR-R-1994-07-Rao-CHI94-TableLens.pdf>.
- [Rasmussen 86]
Rasmussen, J. “ *Information processing and human-machine interaction, an approach to cognitive engineering* ”. Livre édité par Elsevier Science Publishing, 1986.
- [Riehle 96]
Riehle D. et Zllichoven H., “ *Understanding and Using Patterns in Software Development* ”. Journal Theory and Practice of Object Systems, 2 (1), 1996, pp. 3-13.
<http://www.riehle.org/papers/1996/tapos-1996-survey.pdf>.
- [Rist 97]
Rist T., André E. et Muller J. “ *Adding Animated Pre-sentation Agents to the Interface* ”. Actes de la conférence IUT'97, (Orlando, USA), pp 79-86.
- [Robertson 93]
Robertson G. et Mackinlay J., “ *The Document Lens* ”. Actes de la conférences UIST'93 (Atlanta, USA), pp. 101-108. <http://www.parc.xerox.com/istl/projects/uir/pubs/pdf/UIR-R-1993-08-Robertson-UIST93-DocumentLens.pdf>.
- [Robertson 96]
Robertson S., Wharton C., Ashworth C. et Franzke M., “ *Dual device user interface design: PDAs and interactive television* ”. Actes de la conférence CHI 96 (Vancouver, Canada), 1996, pp 79-86.

- [Roth 90a]
Roth, S., Mattis J, et Mesnard X. “ **Graphics and Natural Language as Components of Automatic Explanation** ”. Livre Architectures for Intelligent Interfaces: Elements and Prototypes édité par Sullivan J. and Tyler S., Addison-Wesley, Reading, Mass, 1990.
- [Roth 90b]
Roth, S. et Mattis J., “ **Data characterization for intelligent graphics presentation** ”. Journal Empowering people: Human factors in computing system: Numéro spécial du bulletin SIGCHI, 1990, pp. 193-200.
- [Salber 95a]
Salber D., “ **De l'interaction homme-machine individuelle aux systèmes multi-utilisateurs, L'exemple de la Communication Homme-Homme médiatisée** ”. Thèse de l'Université Joseph Fourier, Grenoble 1, 1995. http://iihm.imag.fr/publs/1995/THESE1995_Salber/.
- [Salber 95b]
Salber D., Coutaz J., Decouchant D. et Riveill M., “ **De l'observabilité et de l'honnêteté : le cas du contrôle d'accès dans la Communication Homme-Homme Médiatisée** ”. Conférence sur l'Ingénierie des Interfaces Homme-Machine, Toulouse, France, 1995. http://iihm.imag.fr/publs/1995/IHM95_Obs&Honnet.Fr.pdf.
- [Sarkar 92]
Sarkar M. et Brown M., “ **Graphical Fisheye Views of Graph** ”. Actes de la conférence CHI'92 (Monterey, USA), 1992, pp. 83-91. <ftp://ftp.digital.com/pub/DEC/SRC/research-reports/SRC-084a.pdf>.
- [Sarkar 93]
Sarkar M., Snibbe S., Tversky O. et Reiss S., “ **Stretching the Rubber Sheet: A Methaphor for Viewing Large Layouts on Small Screens** ”. Actes de la conférence UIST'93 (Atlanta USA), 1993, pp. 81-91. <ftp://ftp.cs.brown.edu/pub/techreports/93/cs93-39.ps.Z>.
- [Savidis 95]
Savidis A. et Stephanidis C., “ **Developing dual user interfaces for integrating blind and sighted users: the HOMER UIMS** ”. Actes de la conférence CHI'95 (Denver, USA), 1995, pp. 106-113.
- [Scapin 93]
Bastien J.M. et Scapin D., “ **Ergonomic criteria for the evaluation of human-computer interfaces** ”. Rapport technique de l'INRIA Rocquencourt N° RT-0156, Juin 1993, 79 pages.
- [Schmandt 95]
Schmandt C. et Mullins A., “ **AudioStreamer: exploiting simultaneity for listening** ”. Actes seconds de la conférence CHI'95 (Denver, USA), 1995, pp 218–219.
- [Sellen 92]
Sellen A., Kurtenbach G.P. et Buxton W., “ **The Prevention of Mode Errors through Sensory Feedback** ”. Journal Human Computer Interaction publié par Lawrence Erlbaum 7(2), 1992, pp. 141-164. http://reality.sgi.com/gordo_tor/papers/ModeErrors.JofHCI/ModeErrors.JofHCI.html.
- [Shneiderman 83]
Shneiderman B., “ **The future of interactive systems and the emergence of direct manipulation** ”. Journal Behaviour and Information Technology 1(3), 1982, pp. 237-256.

[Shneiderman 91]

Shneiderman B., “ *Tree visualization with treemaps: a 2-d space-filling approach* ”. Journal ACM Transactions on Graphics 11(1), Janvier 1992, pp. 92-99. Rapport technique CS-TR-2645. [ftp://ftp.cs.umd.edu/pub/papers/papers/ncstrl.umcp/CS-TR-2645/CS-TR-2645.ps.Z](http://ftp.cs.umd.edu/pub/papers/papers/ncstrl.umcp/CS-TR-2645/CS-TR-2645.ps.Z).

[Shneiderman 96]

Shneiderman B., “ *The eyes have it : A task by data taxonomy for information visualizations* ”. Actes de la conférence Visual Languages'96 (Boulder, USA), 1996, pp. 336-343. Rapport technique CS-TR-3665. [ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/96-13html/96-13.html](http://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/96-13html/96-13.html).

[Stasko 00]

Stasko J. et Zhang E., “ *Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations* ”. Actes de la conférence IEEE InfoVis'2000 (Salt Lake City, USA), 2000, pp. 57-65. Rapport technique : [ftp://ftp.cc.gatech.edu/pub/gvu/tech-reports/2000/00-12.pdf](http://ftp.cc.gatech.edu/pub/gvu/tech-reports/2000/00-12.pdf).

[Stevens 46]

Stevens S., “ *On the theory of scales of measurement* ”. Journal Sciences 103(2684), Juin 1946, pp. 677-680.

[Takeushi 93]

Takeuchi A. et Nagao K., “ *Communicative facial displays as a new conversational modality* ”. Actes de la conférence Human factors in computing systems (Amsterdam, hollande), 1993, pp. 187-193.

[Thevenin 99]

Thevenin D. et Coutaz J., “ *Plasticity of User Interfaces: Framework and Research Agenda* ”. Actes de la conférence Interact99, (Edinburgh, Ecosse), 1999, pp.110-117. http://iihm.imag.fr/publs/1999/INTERACT99_Plasticite.pdf.

[Tidwell 99]

Tidwell J., “ *Common grounds : A Pattern Language for Human-Computer Interface Design* ”. Non publié en dehors du World Wide Web. http://www.mit.edu/~jtidwell/common_ground.html.

[Tracz 95]

Tracz W., “ *Confessions of a Used Program Salesman : Institutionalizing Software Reuse* ”. Livre édité par Addison-Wesley, 1995, 336 pages. ISBN: 0201633698.

[Truillet 99]

Truillet P., “ *Modélisation de coopérations intermodales : application à l'interaction non visuelle* ”. Thèse de l'université Paul Sabatier, soutenue le 15 janvier 1999. <http://www.multimania.com/truillet/these.html>.

[Tsingos 98]

Tsingos N., “ *Simulating High Quality Dynamic Virtual Sound Fields For Interactive Graphics Applications* ”. Thèse de l'université J. Fourier, GRENOBLE I, Décembre 1998. <http://www.bell-labs.com/user/tsingos/publis/thesis.html>.

[Tufté 83]

Tufté E., “ *The Visual Display of Quantitative Information* ”. Livre édité par Graphics Press, Cheshire Connecticut, 1983, A97 pages, ISBN 096 1392 10X.

[Tweedie 97]

Tweedie L., “ *Characterizing Interactive Externalizations* ”. Actes de la conférence CHI'97 (Atlanta, USA), 1997, pp. 375-382.

<http://www.acm.org/pubs/articles/proceedings/chi/258549/p375-tweedie/p375-tweedie.pdf>.

[UIMS 92]

The UIMS Workshop Tool Developers : A Metamodel for the Runtime Architecture of an Interactive System, SIGCHI Bulletin, 24, 1, (Janvier 1992), pp. 32-37.

[Van Welie 00]

Van Welie M., Van der Veer G. et Eliëns A., “ *Patterns as Tools for User Interface Design* ”. Actes du groupes de travail "Tools for Working with Guidelines" (Biarritz, France), 2000, pp. 313-342. <http://www.cs.vu.nl/~martijn/gta/docs/TWG2000.pdf>.

[Venn 71]

Venn J., “ *Symbolic Logic* ”, Livre édité par Bronx: Chelsea Publishing Company, Seconde édition, 1971.

[Vernier 97]

Vernier F. et Nigay L., “ *Représentations Multiples d’une Grande Quantité d’Information* ”. Actes de la conférence IHM'97 (Poitiers, France), 1997, pp. 183-190.

http://iihm.imag.fr/pubs/1997/IHM97-Repr_Multiples-Vernier.pdf.

[Vernier 99]

Vernier F., Lachenal C., Nigay L. et Coutaz J., “ *Interface Augmentée par Effet Miroir* ”. Actes de la conférence IHM'99 (Montpellier, France), 1999, pp. 158-165.

http://iihm.imag.fr/pubs/1999/IHM99_Interfaces_Augmentees.pdf.

[Vernier 00a]

Vernier F. et Nigay L., “ *A Framework for the Combination and Characterization of Output Modalities* ”, Actes de la conférence DSV-IS'00 (Limerick, Ireland), 2000, pp. 32-48. http://iihm.imag.fr/pubs/2000/DSVIS2000_Vernier.pdf.

[Vernier 00b]

Vernier F. et Nigay L., “ *Modifiable Treemaps Containing Variable-Shaped Units* ”, Présenté comme travail en cours (Work in progress) et publié dans des actes à part de la conférence Visualisation'00 (Salt Lake City, USA), 2000, 2 pages.

http://iihm.imag.fr/pubs/2000/Visu2K_Vernier.pdf.

[Walster 93]

Wahlster W., André E., Finkler W., Profitlich H. et Rist T. “ *Plan-based integration of natural language and graphics generation* ”, Journal Artificial Intelligence, volume 63, pp. 387-427.

[Ware 95]

Ware C. et Lewis M., “ *The DragMag Image Magnifier* ”. Actes de la conférence CHI'95 (Denver, USA), 1995, pp. 407-408.

<http://www.acm.org/pubs/articles/proceedings/chi/223355/p407-ware/p407-ware.pdf>.

[Ware 96]

Ware, C., “ *Multimedia output devices and techniques* ”. Journal ACM Computing Surveys (28, 1), Mars 1996, pp. 181-183.

2 *Systèmes de référence*

[AltaVista]

AltaVista : Moteur de recherche sur le World Wide Web avec une interface en HTML.

<http://www.altavista.com>.

[Apache]

Apache : Serveur HTTP permettant la construction de serveur Web. Développé et distribué en logiciel libre. <http://www.apache.org>.

[Codewarrior]

Codewarrior : Environnement de programmation graphique pour les langages JAVA, C, C++ et Pascal. Développé et commercialisé par Metrowerks. <http://www.metrowerks.com>.

[CoVITESSE]

CoVITESSE : Version collaborative de notre système Vitesse [Vitesse] développé par Yann Laurillau. Version de démonstration par applet à l'adresse suivante :

<http://eprouvet.imag.fr/cgi-bin/CoVitesse/covitesse>.

[Eudora]

Eudora : Application graphique de réception et de traitement du courrier électronique.

Développé et commercialisé par Qualcomm. <http://www.qualcomm.com>.

[Excel]

Excel : Tableur. Développé et commercialisé par Microsoft dans la suite Microsoft Office.

<http://www.microsoft.com>.

[HiNote]

HiNote : Application de dessin utilisant la boîte à outil JAZZ pour réaliser un zoom par glisser-déposer. A tester par applet à l'adresse suivante :

<http://www.cs.umd.edu/hcil/jazz/play/hinote-0.5/hinoteapplet.html>.

[Illustrator]

Illustrator : Système d'édition pour les graphiques vectoriels pour Windows ou Macintosh.

Version de test téléchargeable à : <http://www.adobe.com/products/illustrator/demodnld.html>.

[JDK]

JDK, Java Development Kit, Ensemble d'outils et de bibliothèques autour du langage de programmation JAVA. Développé par Sun et téléchargeable à : <http://java.sun.com>.

[MacOS]

MacOs 9 et MacOS X : Famille de systèmes d'exploitation. Développé et commercialisé par Apple.

<http://www.apple.com>.

[MagniFind]

MagniFind : Système de visualisation de répertoire utilisant une vue hyperbolique.

Développé par Inxight et téléchargeable à : <http://www.inxight.com/>.

[MediaPlayer]

MediaPlayer : Système permettant de jouer de nombreux fichiers multimédia continus (son, vidéo). Développé par Microsoft. <http://www.microsoft.com>.

[MulTab]

MulTab : Système de visualisation d'un tableau de donnée. Décrit au chapitre VII de cette thèse. Version de démonstration par applet à l'adresse suivante :

<http://iihm.imag.fr/demos/MulTab/>.

[Netscape]

Netscape : Système d'affichage et de navigation dans les pages Web du Web. Développé par Netscape. <http://www.netscape.com>.

[Olfacom]

OLFACOM : Système de perception olfactif . Développé par la société A.C.2.I.

<http://www.olfacom.com/anglais/pages/presenta.html>.

[PalmPilot]

PalmPilot : Ordinateur de poche utilisant le système d'exploitation PalmOS. Développé et commercialisé par Palm. <http://www.palm.com> .

[Parent]

Parent : Système de visualisation d'une hiérarchie de fichier. Décrit au chapitre VIII de cette thèse. Version de démonstration par applet à l'adresse suivante :

<http://iihm.imag.fr/demos/Parent>.

[QuickTimePlayer]

QuickTimePlayer : Système permettant de jouer de nombreux fichiers multimédia continus (son, vidéo) Développé par Apple. Téléchargeable à : <http://www.apple.com/quicktime>.

[Sand]

Sand : Système d'information géographique développé par Eser Kandogan. Décrit à :

http://reality.sgi.com/kandogan_engr/.

[StarSchedule]

StarSchedule : Module Agenda de la suite StarOffice. Développé par SUN. Téléchargeable à : <http://www.sun.com/products/staroffice/>.

[TreeViz]

TreeViz : Système de visualisation d'une hiérarchie de fichier utilisant l'algorithme des Treemaps [Johnson 91][Shneiderman 91]. Présentation du système à :

<http://www.cs.umd.edu/hcil/treemaps/>.

[Vitesse]

VITESSE 2.5 : Système de visualisation des réponses d'un moteur de recherche du Web. .

Décrit au chapitre VI de cette thèse. Version de démonstration par applet à l'adresse suivante : <http://iihm.imag.fr/demos/VitesseDemos.html>.

[Warcraft]

Warcraft : Logiciel ludique de stratégie. Développé par Blizzard. Version de démonstration téléchargeable à <http://www.blizzard.com/war1/>.

[Winamp]

Winamp : Système de lecture audio de nombreux types de fichier sons. Téléchargeable à : <http://www.winamp.com>.

[Windows]

Windows 95, 98, NT et 2000 : Famille de systèmes d'exploitation. Développés et commercialisés par Microsoft. <http://www.microsoft.com>.

[Word]

Word : Logiciel de traitement de texte. Développé et commercialisé par Microsoft dans la suite Microsoft Office. <http://www.microsoft.com>.

ANNEXES

1 Annexe 1 : Liste des critères ergonomiques de l'équipe IIHM [Abowd 92][Gram 94]

Cette annexe consigne un ensemble de critères servant à caractériser la notion d'utilisabilité. Dans ce cadre, nous dirons que : Utilisabilité = Souplesse + Robustesse où :

- ▶ La Souplesse (de l'interaction) exprime l'éventail des choix (pour l'utilisateur et le système).
- ▶ La Robustesse (de l'interaction) vise la prévention des erreurs, l'augmentation des chances de succès de l'utilisateur.

Remarque : la facilité d'apprentissage n'est pas considérée.

Souplesse et robustesse sont déclinées en critères. Pour chacun d'eux, nous fournissons une définition complétée de remarques, d'exemples, de contre-exemples et, si possible, de métriques.

1.1 Souplesse

Atteignabilité

Définition : Capacité du système à permettre à l'utilisateur de naviguer dans l'ensemble des états observables du système. Un état q est atteignable à partir d'un état p s'il existe une suite de commandes $\{ c_i \}$ qui font passer de l'état p à l'état q .

Remarque : Propriété non vérifiée si analyse de tâche ou analyse fonctionnelle défectueuse.

Métrique : Longueur de la trajectoire d'interaction entre p et q .

Non-préemption

Définition : Le prochain but souhaité par l'utilisateur est directement atteignable. Pour l'utilisateur, il n'y a pas de contrainte dans la trajectoire interactionnelle.

Contre-exemple : Boîte de dialogue modale : à n'utiliser qu'à bon escient (cas des commandes irrévocables). Un système interruptible est, du point de vue utilisateur, non préemptif. Mais trop de liberté peut nuire : L'utilisateur risque d'être perdu. Penser au concept de tableau de bord qui indique à l'utilisateur sa localisation dans l'espace des tâches.

Métrique : Si la longueur de la trajectoire d'interaction entre l'état actuel et l'état souhaité vaut 1, alors la propriété de non préemption est vérifiée.

Préemption globale

Définition : Interdiction, pour l'utilisateur, d'effectuer toute autre action que celle requise par le système (ici la station de travail).

Préemption locale

Définition : Ne bloque qu'un fil de dialogue. Laisse l'utilisateur continuer les autres fils.

Préemption de ressources partagées par un utilisateur

Exemple : Collecticiels à "tour de parole" (turn-taking) : préemption du curseur partagé.

Interaction multifilaire

Définition : Capacité du système à permettre la réalisation "simumultanée" de plusieurs tâches.

Remarque : Analyse du caractère entrelacé ou parallèle à différents niveaux de granularité : niveau tâches au sein d'un logiciel donné, niveau actions (cf. les propriétés CARE).

Exemple : Edition de plusieurs documents à la fois.

Métrique : Nombre de tâches que l'on peut mener de manière entrelacée au regard de la charge cognitive.

Interaction multifilaire parallèle

Définition : Parallélisme vrai entre plusieurs tâches. Peu souhaitable si les compétences de l'utilisateur n'est pas du niveau expert ("skill level", au sens de Rasmussen [Rasmussen 86]).

Interaction multifilaire entrelacée

Définition : Les tâches peuvent être simultanées au sens de l'utilisateur, mais à un instant donné, l'interaction est restreinte à une seule tâche.

Multiplicité du rendu (ou représentation multiple d'un même concept)

Définition : Capacité du système à fournir plusieurs représentations pour un même concept.

Remarque : (1) En sortie : Formes différentes (par ex., pour le concept de température : un entier ou une représentation analogique comme un thermomètre. Contenus différents : le détail vs l'ensemble : Attention aux discontinuités visuelles. Penser aux techniques "fisheye" ou holophrastiques. (2) En entrée : Formes différentes : 6*4 et 24. (3) Principe d'égale opportunité : L'utilisateur choisit la nature de l'entrée et le système en déduit la sortie (principe des tableurs).

Réutilisabilité des données d'entrée et de sortie

Définition : Les sorties du système peuvent être utilisées comme des données d'entrée (couper-coller). Les entrées de l'utilisateur peuvent être réutilisées par le système en sortie (valeurs par défaut).

Remarque : Attention aux effets de bord dus aux conversions de types de données. Cas du couper-coller entre deux éditeurs graphiques, l'un vectoriel et l'autre du niveau pixel.

Adaptabilité

Définition : Personnalisation du système sur intervention explicite de l'utilisateur.

Remarque : Risque de perte de cohérence entre les systèmes d'une communauté d'utilisateurs sensés travailler ensemble. À considérer selon le degré de couplage des activités collectives. D'après [Grudin 89], les utilisateurs ne modifient pas les valeurs par défaut qui viennent à la livraison du collecticiel. Leçon à retenir : bien étudier les valeurs par défaut initiales en fonction des catégories/rôles des futurs utilisateurs.

Exemple : Menus et formulaires d'options et de préférences. Macro d'encapsulation de commandes à caractère répétitif dont le niveau d'abstraction est trop bas. Toute donnée lexicale (ex. nom des commandes) doit être dans un fichier de ressources, donc pas dans le code source du logiciel. Mesure de vérification : produire le logiciel dans une autre langue sans le recompiler.

Adaptativité

Définition : Capacité du système à s'adapter à l'utilisateur sans intervention explicite de l'utilisateur.

Remarque : L'adaptativité s'appuie sur un modèle embarqué de l'utilisateur. Veiller à ce que le système ait un comportement prévisible. Ne pas surprendre l'utilisateur (caractère disruptif).

Plasticité

Définition : Capacité du système à s'adapter aux variations des ressources interactionnelles, computationnelles, communicationnelles et environnementales tout en conservant la continuité ergonomique [Thevenin 99][Coutaz 98].

Exemple : IHM d'un agenda sur PalmPilot et sur PC.

Migrabilité de tâche

Définition : Capacité de délégation dynamique de tâches entre le système et l'utilisateur ou entre utilisateurs : changement dynamique de l'acteur(s) responsable(s) de l'accomplissement de la tâche.

Remarque : Manifestation à différents niveaux de granularité : (1) 6*4 et 24 (2) valeurs par défaut (3) détection de tâches répétitives puis prise en charge (Système Eager) (4) sauvegarde automatique de fichiers.

CARE (multimodalité)

Définition : Complémentarité, Assignation, Redondance, Equivalence [Nigay 94][Coutaz 95]. Caractérisation de la multimodalité offerte par un système. Modalité = <dispositif d'E/S, système représentationnel>. Complémentarité : plusieurs modalités distinctes sont nécessaires pour exprimer le but. Assignation : une seule modalité est disponible pour exprimer le but. Redondance : plusieurs modalités sont utilisables en "même temps" et expriment le même but. Equivalence : plusieurs modalités sont possibles pour exprimer le but. Une seule est utilisable à la fois. Notons que la redondance implique l'équivalence.

Exemple : Complémentarité : " mets ça là " vocal // geste. Redondance : " Montre-moi Grenoble " vocal // double-clic sur Grenoble affichée sur une carte.

CARE (collecticiel)

Définition : (1) Appliqué aux rôles des acteurs d'un collecticiel [Salber 95a] : Complémentarité de rôle (cas des jeux). Assignation de tâche à un rôle. (2) Appliqué aux moyens technologiques pour collaborer : CARE entre Fax, email, Vphone etc.

1.2 Robustesse

Observabilité

Définition : Capacité du système à rendre perceptible l'état pertinent du système. Capacité pour l'utilisateur à évaluer l'état actuel du système. L'utilisateur PEUT PERCEVOIR.

Remarque : Inspectabilité (browsability) : capacité pour l'utilisateur d'explorer l'état interne du système au moyen de commandes articulatoires (ou passives) (c'est-à-dire qui ne modifient pas l'état du noyau fonctionnel) telles que zoom, défilement, etc.

Contre-exemple : "Defense in depth design" (Rasmussen).

Observabilité publiée

Définition : Capacité pour un utilisateur de rendre observables des variables d'état personnelles [Salber 95a][Salber 95b].

Remarque : Exemples de variables d'état personnelles : présence, niveau de disponibilité. Une variable publiée peut être filtrée. Filtrage de variable : opération de transformation de la valeur de la variable visant à protéger l'espace privé. Un filtre ne doit pas être réversible.

Réciprocité

Définition : Dans un collectif, capacité d'observation/inspection mutuelle des variables d'état personnelles.

Exemple : " Si je vous vois, vous me voyez. "

Réflexivité

Définition : Capacité d'inspecter ou d'observer les variables d'état personnelles publiées à autrui.

Exemple : Vidéo miroir en vidéoconférence.

Insistance

Définition : Capacité du système à forcer la perception de l'état du système L'utilisateur DEVRA PERCEVOIR.

Remarque : Le retour d'information du système peut être pour un contexte donné : (1) Ephémère (ex. audio, vidéo) ou non (ex. écrit statique), (2) Evitable (retour visuel) ou inévitable (audio), (3) Entretenu par le système (ex. clignotement) ou par l'utilisateur (maintien d'un bouton enfoncé qui minimise les oublis ; cf. [Sellen 92]). Attributs perceptuels additifs ; l'intensité sonore et l'intensité des couleurs ont un effet additif. Le ton sonore et la teinte ne sont pas additifs. Rétro-action de groupe. Awareness : compréhension des activités d'autrui, fournissant à l'action individuelle un contexte situationnel collectif [Dourish 92].

Honnêteté

Définition : Capacité du système à rendre observable l'état du système sous une forme conforme à cet état ET qui engendre une interprétation correcte de la part de l'utilisateur. L'utilisateur AURA UN MODELE CORRECT de l'état du système.

Remarque : WYSIWYG (What You See is What You Get), WYSIWIS (What You See Is What I See) et les versions relâchées. Distorsion des informations (ex. Les données linéaires ne doivent pas être présentées en deux dimensions [Tufte 83]). Conformité de l'état interne et de la présentation pas toujours compatibles avec les temps de réponse attendus : utiliser un indicateur pour exprimer que l'information a changé et qu'elle n'est pas encore réactualisée dans le rendu. Intégrité des messages entre l'émetteur et le récepteur. Dans les formulaires, veiller à la formulation des unités de mesure et du format des données à saisir. Veiller à une terminologie précise en accord avec le métier, l'utilisateur, etc.

Honnêteté sociale

Définition : Le système peut être honnête mais peut être détourné socialement.

Exemple : Enclencher son répondeur téléphonique pour simuler l'absence.

Curabilité

Définition : Capacité pour l'utilisateur de corriger une situation non désirée.

Remarque : (1) Curabilité arrière : capacité de défaire. Le défaire de profondeur 1 est facile à réaliser : on ne modifie le noyau fonctionnel qu'à l'interaction suivante. (2) Curabilité avant : reconnaissance de l'état actuel et capacité de négociation pour atteindre le but désiré = atteignabilité indispensable. Messages d'erreur explicatifs et correctifs. Principe de l'effort commensurable [Dix 93] : ce qui est difficile à défaire doit être difficile à faire (exemple de la destruction d'un fichier). Voir [Jambon 96] pour une analyse détaillée.

Prévisibilité

Définition : Capacité pour l'utilisateur de prévoir, pour un état donné, l'effet d'une action.

Remarque : Cohérence : conformité aux règles/usages [Grudin 89]. Les règles/usages de l'utilisateur ne sont pas nécessairement celles du concepteur. Cohérence interne : cohérence lexicale, syntaxique, sémantique. Cohérence externe : conformité à des normes d'IHM, conformité à l'expérience dans le monde réel (analogie, métaphore). Retour d'information proactif : principe du "do-nothing" ou de résistance passive : éléments interdits en grisé. Prévisibilité et stabilité des temps de réponse : rassurer si temps de réponse long. Attention aux

solutions système de type ramasse-miette à la volée sur la stabilité dans le temps de réponse. Régularité des médias continus. Attention aux limites de tolérance (ICS [Barnard 85][Barnard 92][Barnard 94]).

Tolérance du rythme

Définition : L'utilisateur plutôt que le système décide quand il peut agir.

Remarque : Attention aux temporisations qui font sens (ex. durée de pression sur un dispositif, tels les téléphones portables et les distributeurs de boissons).

Exemple : Dans le cas de la saisie anticipée : nombre d'actions anticipables (noter que sous Word, ce nombre est variable.).

Métrique : Durée de tolérance.

Viscosité

Définition : L'action de l'utilisateur à un effet sur son plan de tâches ou, pour un collectif, sur celui des autres.

Exemple : Ajout d'une ligne dans un texte qui provoque des orphelins dans la pagination.

Métrique : Longueur de la trajectoire d'interaction nécessaire à la correction de l'effet de viscosité.

Rejouabilité

Définition : Capacité pour l'utilisateur de rejouer des séquences informationnelles audio/vidéo.

Remarque : S'appuie sur l'existence d'un historique. En communication médiatisée, la rejouabilité d'un message audio permet de ré-écouter [Salber 95a].

Révisabilité

Définition : Capacité pour l'utilisateur de réviser un message avant de l'émettre.

Remarque : La révisabilité implique la rejouabilité. La révisabilité est une forme de curabilité.

2 Annexe 2 : Principes ergonomiques de Dominique Scapin et Christian Bastien [Scapin 93]

Dans cet annexe, nous listons sans les détailler les principes ergonomiques définis et organisés par Dominique Scapin et Christian Bastien. De nombreux sites Internet contiennent également cette liste (par exemple : <http://clips.imag.fr/multicom/articles/critere.html>).

1. Guidage
 - 1.1. Incitation
 - 1.2. Groupement/ Distinction entre Items
 - 1.2.1. Groupement / Distinction par la Localisation*
 - 1.2.2. Groupement/ Distinction par le Format
 - 1.3. Feedback Immédiat
 - 1.4. Lisibilité
2. Charge de Travail
 - 2.1. Brièveté
 - 2.1.1. Concision
 - 2.1.2. Actions Minimales
 - 2.2. Densité Informationnelle
3. Contrôle Explicite
 - 3.1. Actions Explicites
 - 3.2. Contrôle Utilisateur
4. Adaptabilité
 - 4.1. Flexibilité
 - 4.2. Prise en Compte de l'Expérience de l'Utilisateur
5. Gestion des Erreurs
 - 5.1. Protection Contre les Erreurs
 - 5.2. Qualité des Messages d'Erreur
 - 5.3. Correction des Erreurs
6. Homogénéité/ Cohérence
7. Signification des Codes et Dénominations
8. Compatibilité

INDEX DES FIGURES

Figure 1 : Axes de classification des patrons de conception.....	27
Figure 2 : Exemple de schéma QOC pour la question "où dois-je faire construire ma maison ?".	31
Figure 3 : Niveaux physique et logique d'une modalité d'interaction.....	36
Figure 4 : Trois niveaux d'abstraction d'une modalité.....	37
Figure 5 : M ² LD, une classification des systèmes multimodaux en sortie selon la multiplicité des langages et des dispositifs de sortie. La valeur "non" sur un axe signifie que le système est "mono" pour cette dimension. La valeur "oui" correspond au cas "multi".	38
Figure 6a : Affichage d'une liste de grandes icônes dans Windows NT [Windows].	41
Figure 7 : Typologie de la multimodalité en sortie selon les acteurs du choix des modalités.....	43
Figure 8 : Cadre de conception de la multimodalité en sortie.....	47
Figure 9 : Six axes de l'espace MSM [Nigay 96].	53
Figure 10 : L'espace de conception des interfaces en sortie de Frohlich [Frohlich 91].....	56
Figure 11 : Exemples de deux icônes (a) l'une analogique, (b) l'autre non-analogique [Word]..	58
Figure 12a : Composition de modalités éloignées.....	61
Figure 13 : Lentilles magiques [Bier 93] : cas de deux lentilles ("Highlight Major Roads" et "Highlight Water" mettant en valeur les routes principales et les voies fluviales sur un fond de carte d'une ville.....	64
Figure 14 : Système des pixels miroir [Vernier 99] où la totalité de l'écran est utilisée pour afficher une carte météorologique et une vidéo en temps réel de l'utilisateur.....	64
Figure 15 : Système MagniFind [MagniFind]: Exemple de composition complémentaire et redondante de modalités de sortie.....	67
Figure 16 : Modalité passive-active composée (langage icônique, langage naturel).	71
Figure 17 : Message d'erreur dû à un dispositif déjà occupé sous Windows NT.....	74
Figure 18 : Windows NT [Windows] en mode d'affichage minimal de 640 par 480 pixels. La modalité associée à l'interacteur, qui permet de changer de mode fait plus de 480 pixels de haut. Dans le mode 640 par 480, l'utilisateur n'a alors pas accès au bouton "OK" pour changer de mode et se trouve piégé(e) dans ce mode.	75
Figure 19 : Options d'affichage du système Word [Word].....	79
Figure 20 : L'interface de Codewarrior : à droite un programme visualisé selon une modalité graphique paramétrable grâce au panneau de configuration présenté à gauche.....	80
Figure 21a : Visualisation de répertoires dans Windows NT [Windows].	82
Figure 22 : Système StarSchedule [StarSchedule].	83
Figure 23 : Processus de visualisation de Ed Chi [Chi 98].	91
Figure 24 : Taxonomie de Leung [Leung 94].	93
Figure 25b : Vue non déformée de l'abstraction graphique des villes de France	93
Figure 26 : Caractérisation du système FilmFinder [Ahlberg 94] par la taxonomie de Card [Card 97].	97
Figure 27 : Cadre d'étude fédérateur du processus de visualisation.....	98
Figure 28a : Vues multiples sur des données différentes.....	105
Figure 29 : Tableau récapitulatif des différents points de vue sur les données contenant un exemple dans chacune des quinze cases non grisées. Les données non-structurées à plus d'une dimension ne sont pas représentables dans cette taxonomie.	108
Figure 30a : Exemple de donnée 2D quantitative : le cycle thermodynamique Pression-Volume d'un moteur à explosion.	109
Figure 31 : Visualisation et navigation deux niveaux d'abstraction dans l'activité de l'utilisateur au niveau du point de vue sur l'espace.....	114
Figure 32 : Espace de conception des modalités représentant un point de vue sur l'espace d'information.....	115
Figure 33 : Système Warcraft [Warcraft] utilisant une vue radar.....	116
Figure 34 : Système DragMag [Ware 95] (Image aux contours retravaillés extraite de la vidéo de présentation du système DragMag).	117
Figure 35 : Taxonomie en trois axes des effets de déformation.....	118
Figure 36 : Système du Bifocal Display [Apperley 82].	119

Figure 37 : Système du mur en perspective ("Perspective Wall") [Mackinlay 91].	119
Figure 38 : Outils de navigation grossière et de navigation pas-à-pas dans le logiciel Word [Word].	120
Figure 39 : Le système FilmFinder [Ahlberg 94] avec un couplage fort entre les glissières et le champs de points.	121
Figure 40 : Plan du métro de Washington, déformé autour de la maison blanche [Keahey 96].	123
Figure 41 : Système Sand [Sand].	125
Figure 42 : Illustration de l'effet d'animation appelé "génie" dans MacOS [MacOS] lorsqu'une fenêtre est minimisée dans la barre des applications, en bas de l'écran.	127
Figure 43 : Système des TableLens [Rao 94].	130
Figure 44 : Déformation d'une carte géographique [Carpendale 97] avec affichage d'une grille et d'un effet d'ombre.	133
Figure 45 : Carte des USA déformée par une vue en œil de poisson (ou hyperbolique) [Sarkar 92][Sarkar 93].	134
Figure 46 : Trois outils de navigation très proches dans trois systèmes différents.	135
Figure 47 : Outil de recherche par mot clé dans le système Word [Word].	137
Figure 48 : Tracé de la fonction de déformation $FDeform(x)$ [Hovestadt 95] pour les valeurs : $r=300$, $\alpha=150$ et $z=112$.	143
Figure 49 : Modèle géométrique de projection en deux étapes correspondant à la fonction de déformation $FDeform(x)$ [Hovestadt 95].	144
Figure 50 : Déformation d'une rangée de carré.	144
Figure 51a : $(x,y) \rightarrow (x,y)$ Espace de départ.	146
Figure 52 : Déformation en forme de pyramide à base carrée, vue du sommet.	147
Figure 53 : Document Lens [Robertson 93].	148
Figure 54 : Déformation d'un polygone à l'intersection de l'arête de la pyramide.	149
Figure 55 : Aplanissement de la partie centrale de la déformation en limitant la valeur maximale de $CoeffDeform$.	149
Figure 56 : Aplanissement de la partie extérieure de la déformation en limitant la valeur minimale de $CoeffDeform$.	150
Figure 57 : Forme hybride entre la sphère (50%) et la pyramide (50%).	151
Figure 58 : Application de deux fonctions de déformation différentes sur chaque axe, dont résulte une forme de cylindre.	152
Figure 59 : Formes allongées obtenues en utilisant des valeurs du paramètre R différentes sur chaque axe.	153
Figure 60 : Le modèle PAC-Amodeus [Nigay 94].	154
Figure 61 : Proposition de correspondance entre les étapes du processus de visualisation (<i>Italique</i>) et les composants du modèle PAC-AMODEUS (Gras).	155
Figure 62a : Réalisation de la composition du focus et du contexte.	157
Figure 63 : Interface du moteur de recherche AltaVista [AltaVista].	168
Figure 64 : Espace des réponses dans VITESSE.	169
Figure 65 : Histogramme d'opinion en blanc, apparaissant sous l'effet d'une lentille magique.	170
Figure 66 : Interface de VITESSE affichant un des points de vue sur l'espace (Sphère). Les onglets permettent de changer ce point de vue. Les éléments verts sombres correspondent à des pages Web déjà visitées.	172
Figure 67 : Vue à différents facteurs de zoom, globale et vague (à gauche), puis précise et globale (à droite).	173
Figure 68 : Vue en sphère.	174
Figure 69 : Vue cartésienne.	176
Figure 70 : Vue en pyramide.	177
Figure 71 : Vue en colline.	178
Figure 72 : Vue en cylindre de l'interface de visualisation de VITESSE.	179
Figure 73 : Le système VITESSE avant et après un glisser-déposer (matérialisé par une flèche en pointillés).	181

Figure 74 : Partie supérieure de l'architecture PAC du Contrôleur de Dialogue de VITESSE.	184
Figure 75 : Partie inférieure de la hiérarchie d'agents PAC du Contrôleur de Dialogue de VITESSE. Seule est illustrée la présence des agents "vue en sphère" et "vue en pyramide".	184
Figure 76 : Réalisation graphique de l'agent "Fenêtre de l'opinion de l'utilisateur".	186
Figure 77 : Réalisation graphique de l'agent "Outils de navigation".	186
Figure 78 : Réalisation graphique de l'agent "Fenêtre de chargement d'un nouvel espace".	186
Figure 79 : Réalisation graphique de l'agent "Fenêtre de sauvegarde de l'espace affiché".	186
Figure 80 : Réalisation graphique de l'agent "Vues multiples".	186
Figure 81 : Réalisation graphique de l'agent "Lentille magique d'opinions".	186
Figure 82 : Réalisation graphique de l'agent "Texte dans le focus".	186
Figure 83 : Communication entre le coté serveur et le coté client de VITESSE.	187
Figure 84a : Le système VITRIHOL.	189
Figure 85 : Résultats du questionnaire pour identifier les informations à afficher à propos d'une page Web. Le questionnaire demandait de classer par ordre d'importance (de 1 à 13) treize informations différentes possibles à propos d'une page Web. Les colonnes sont ordonnées par ordre d'importance de l'information (de la plus importante = 1 à la moins importante = 13). Nous voyons par exemple dans la dernière colonne de droite, que presque 30% des utilisateurs questionnés considèrent que le temps de chargement est l'information la moins importante. Nous constatons à la colonne de gauche que les informations les plus importantes sont le titre et l'adresse URL de la page.	190
Figure 86 : Interface du logiciel Excel [Excel] avec trois pages contenant des tableaux de données.	196
Figure 87 : Espace de MulTab composé d'un tableau de données. Les cellules sont coloriées et une légende accompagne l'espace. Dans cet exemple, 12000 cellules sont présentées.	198
Figure 88 : Affichage d'une modalité détaillée (un focus) par dessus le tableau. La forme hybride, entre la sphère et la pyramide, présente des couleurs éclaircies pour une meilleure lisibilité.	200
Figure 89 : Différentes formes de la modalité "focus" dans MulTab : circulaire à gauche, pyramidale à droite, hybride (50%-50%) entre les deux.	200
Figure 90 : Vue générale de l'application MulTab avec quatre foci.	201
Figure 91a : Composition spatiale de deux foci en forme de sphère.	202
Figure 92 : Légende des couleurs affichée par l'outil BiSlider.	203
Figure 93 : Hiérarchie d'agents PAC du Contrôleur de Dialogue de MulTab.	207
Figure 94 : Zones de l'interface à rafraîchir lors d'un déplacement de la souris.	209
Figure 95 : Le système TreeViz [TreeViz] utilisant l'algorithme des Treemaps [Johnson 91][Shneiderman 91] pour afficher une hiérarchie de fichiers.	215
Figure 96a : Graphe d'arbre traditionnel.	217
Figure 97 : Recomposition d'une suite de rectangles par l'algorithme des Fieldmaps.	219
Figure 98 : L'espace de Parent, créé par l'algorithme des Fieldmaps, pour 325 fichiers et répertoires.	221
Figure 99 : Repère visuel en forme d'escalier noir mis en évidence dans l'espace de Parent pour montrer le chemin de la racine à l'objet pointé.	222
Figure 100 : Vue alternative, sous forme d'une liste indentée, du répertoire intitulé MulTab2.	223
Figure 101 : Interface globale de Parent avec deux légendes pour la couleur.	225
Figure 102 : Mélodie jouée pour accompagner le déplacement de fichiers.	226
Figure 103 : Hiérarchie d'agents PAC proposée pour le système Parent.	227
Figure 104 : Nouvelle entrée dans le menu contextuel associé à un répertoire pour lancer le système Parent.	228
Figure 105 : Démarche d'analyse et contributions.	236