



HAL
open science

Annotation et visualisation interactives de documents hypermédias

Laurent Robert

► **To cite this version:**

Laurent Robert. Annotation et visualisation interactives de documents hypermédias. Interface homme-machine [cs.HC]. Télécom ParisTech, 2001. Français. NNT: . tel-00005631

HAL Id: tel-00005631

<https://pastel.hal.science/tel-00005631>

Submitted on 5 Apr 2004

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

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

THÈSE

présentée pour obtenir le grade de

Docteur

de l'École Nationale Supérieure des Télécommunications

Spécialité : Informatique

par

Laurent Robert

Annotation et visualisation interactives de documents hypermédias

Directeur de thèse : Eric Lecolinet

Soutenue le 29 Juin 2001 devant la commission d'examen composée de

Michel Beaudouin-Lafon	Professeur, LRI Orsay	Président
Jocelyne Nanard	Professeur, LIRM Montpellier	Rapporteurs
Alain Derycke	Professeur, TRIGONE Lille	
Claudie Faure	Chargée de Recherche, CNRS Paris	Examineurs
Jean-Daniel Fekete	Maître Assistant, EMN Nantes	
Jean-Louis Lebrave	Directeur de Recherche, CNRS Paris	
Laurence Nigay	Maître de Conférence, CLIPS-IMAG Grenoble	
Eric Lecolinet	Maître de Conférence, ENST Paris	Directeur de Thèse

Résumé

Le développement du multimédia, des capacités de stockage et du matériel de numérisation permet de mettre à disposition sous une forme électronique un nombre toujours croissant de documents « matériels » (journaux, cartes, photographies, peintures, cahiers d'expérimentations, originaux d'œuvres littéraires, rapports, etc.). La numérisation, le stockage et la diffusion de ces matériaux sources ne posent aujourd'hui plus réellement problème. Néanmoins, il s'avère que les utilisateurs ont souvent des difficultés à exploiter ces documents en milieu informatique. Cet état de fait est la conséquence de plusieurs problèmes. Tout d'abord, peu d'outils logiciels permettent aux utilisateurs de réaliser les tâches qu'ils ont l'habitude d'effectuer avec des documents papiers (annoter, comparer, associer des idées, etc.). Ensuite, l'accès à l'information dans un espace informationnel de vaste envergure s'avère être un problème non trivial. Les utilisateurs éprouvent des difficultés à trouver les données recherchées, à comprendre leur organisation et leurs relations, ainsi qu'à collecter et organiser celles jugées intéressantes afin de pouvoir les retrouver plus rapidement par la suite.

A cette problématique de l'utilisation des documents en milieu informatique, nous proposons deux approches complémentaires issues des domaines de l'interaction homme-machine et de la visualisation d'information.

La première approche repose sur le concept de lecture active qui vise à transformer le « lecteur » en « acteur » en lui permettant d'enrichir interactivement les documents qu'il est en train de lire par un ensemble de signes graphiques (surlignages, cerclages, etc.) et d'informations textuelles (notes, commentaires, etc.). L'idée consiste à permettre aux utilisateurs de construire leur propre système de repérage afin de faciliter la compréhension et la réutilisation des documents. Une attention particulière est portée à la manière d'associer ces méta-données « ajoutées » (que nous nommons, d'une manière générique, « annotations ») aux documents sources au moyen de liens hypertextes, ainsi qu'à leur codage informatique dans un format normalisé (dérivé de XML).

La seconde approche consiste à faciliter l'exploration d'espaces documentaires en exploitant les capacités perceptives et mémorielles humaines au moyen de techniques interactives de visualisation. Nous proposons tout d'abord un système de navigation qui repose sur un fort couplage entre vues globales et vues locales pour aider à trouver les documents. Les vues globales montrent l'organisation de l'espace d'information et en permettent un survol rapide. Elles sont utilisées pour atteindre des zones d'intérêts. Les vues locales présentent le contenu de plusieurs documents en contexte afin d'aider l'utilisateur à les comparer et à choisir celui qu'il va

consulter. Nous proposons ensuite un environnement visuel de gestion de bookmarks dans le but de pouvoir retrouver le plus simplement possible les documents préalablement consultés. Les utilisateurs organisent leurs bookmarks de la même manière qu'ils le feraient avec un ensemble de documents papiers sur un bureau. Ils définissent interactivement leur propre organisation conceptuelle de l'espace documentaire en se basant sur des propriétés visuelles et spatiales. Ces propriétés aideront les utilisateurs à retrouver plus efficacement les documents sauvegardés par une stimulation de leurs capacités mémorielles.

Remerciements

Je souhaite tout d'abord remercier Monsieur Eric Lecolinet, qui a dirigé mes travaux au sein du département informatique et réseaux de l'École Nationale Supérieure des Télécommunications, pour la confiance qu'il m'a accordée, pour la compréhension dont il a fait preuve (souvent !), pour ses conseils avisés, ainsi pour son expérience partagée au quotidien. Je tiens également à lui exprimer ma reconnaissance pour m'avoir permis de prendre une part active à plusieurs projets de recherche, ainsi qu'aux activités d'enseignements de l'école. Enfin, je tiens à signaler que son contact a contribué à mon évolution, tant du point de vue humain que du point de vue professionnel. Qu'il soit ici assuré de ma sincère amitié.

Pour l'importance qu'ils ont accordée à mon travail, vis-à-vis des projets de recherche dont ils avaient la direction et auxquels je pris activement part, je tiens à remercier Jean-Louis Lebrave, Bernard Cerquiglini et Bernadette Dorizzi (notamment pour sa patience et sa gentillesse). Dans le cadre de ces projets, que soient également remerciés pour leur collaboration Jean Daniel Fekete, François Role, Laurence Likforman-Sulem, Jacques André, Hélène Richy, ainsi que Pierre-Marc de Biasi.

Michel Beaudouin-Lafon m'a fait l'honneur d'être le président de mon jury de thèse. Je lui exprime ici ma sincère reconnaissance. Merci à Jocelyne Nanard et Alain Derycke qui se sont intéressés à mon travail et ont bien voulu en être les rapporteurs. Merci également aux autres membres du jury, à savoir Jean-Daniel Fekete, Jean-Louis Lebrave, Laurence Nigay et tout particulièrement Claudie Faure qui fut à mon égard attentionnée et d'une égale gentillesse durant tout mon séjour à l'ENST.

Je souhaite également louer la qualité du cadre de travail offert par l'ENST, ainsi que la chaleureuse ambiance du département informatique et réseaux, deux facteurs qui ont permis la réalisation de mon travail dans de très bonnes conditions. Que soit remercié l'ensemble des membres du département pour leur camaraderie et leur gentillesse. Une mention particulière à Stuart, avec qui j'ai eu la joie de partager mon bureau, pour son amitié, sa bonne humeur et la qualité de ses compétences techniques, toujours appréciées à leur juste valeur.

Enfin, merci aux membres de ma famille et à tous mes amis (en particulier, Fabien, pour son travail de relecture), chacun d'eux ayant contribué à la pérennité de l'équilibre entre ma vie privée et ma vie professionnelle, garant de mon bien être durant la formidable et très enrichissante période des années de thèse !

Table des matières

<i>Liste des figures</i>	<i>11</i>
<i>Liste des tableaux</i>	<i>15</i>
I. Introduction	17
II. Manipulation et représentation des documents	23
1. Systèmes hypermédias	25
1.1 Concepts et définitions	25
1.1.1 Les nœuds	26
1.1.2 Les liens	27
1.2 Historique	28
1.2.1 Evolution du domaine (échelle de temps)	28
1.2.2 Pionniers et applications	30
1.2.2.1 Vannevar Bush: Memex (1945)	30
1.2.2.2 Theodor Nelson: Xanadu (1965)	31
1.2.2.3 Andy Van Dam: Hypertext Editing System (1967)	32
1.2.2.4 Douglas Engelbart: Augment: NLS (1968)	33
1.2.2.5 Robert Akscyn: ZOG & KMS (1975)	34
1.2.2.6 Alan Kay: Dynabook (1977)	35
1.2.2.7 Andrew Lippman: Aspen Movie Map (1978)	36
1.2.2.8 Ben Shneiderman: HyperTies (1983)	36
1.2.2.9 Janet Walker: Document Examiner (1985)	37
1.2.2.10 Norman Meyrowitz: Intermedia / IRIS (1985)	39
1.2.2.11 Peter Brown: Guide (1986)	41
1.2.2.12 Frank Halasz: NoteCards (1987)	42
1.2.2.13 Bill Atkinson: HyperCard (1987)	45
1.2.2.14 Le World Wide Web	46
1.3 Modèles	48
1.3.1 Architecture générale des systèmes hypertextes	48
1.3.2 Discussion	50
2. Augmentation interactive de documents électroniques	53
2.1 Introduction	53
2.1.1 Lecture Active	55
2.1.2 Objectifs	55
2.1.3 Illustration : le projet Philectre	57
2.2 Le poste de lecture active : principes de conception	59
2.2.1 Couplage fac-similé / annotation	59
2.2.2 Format de codage	60
2.2.3 Aide à la production de documents annotés	63
2.3 Couplage texte-image interactif	65

2.3.1	Transcription	67
2.3.2	Annotation.....	70
2.4	Codage XML/TEI	72
2.4.1	Codage de la transcription.....	72
2.4.2	Codage des annotations.....	73
2.4.3	Codage des associations.....	73
2.5	Modélisation.....	75
2.6	Rendu visuel des données XML	76
2.6.1	Conversion XSLT	77
2.6.2	Unification des représentations XML et graphique	80
2.7	Conclusion.....	84
III.	Exploration interactive	85
3.	Désorientation	87
3.1	Manque de structuration.....	87
3.2	Manque de mécanismes appropriés	88
4.	Techniques d'aide à l'exploration	91
4.1	Présentation du contenu	91
4.1.1	Outils dérivés du livre traditionnel.....	91
4.1.2	Points de repères	92
4.1.3	Visite guidée.....	93
4.2	Suivi du parcours de navigation.....	95
4.2.1	Retour arrière	95
4.2.2	Historique de navigation	95
4.3	Personnalisation	97
4.3.1	Marquage et annotation.....	97
4.3.2	Marque-page (Bookmark).....	98
4.4	Augmentation de la structuration	99
4.4.1	Typage des nœuds et des liens	100
4.4.2	Transformations structurelles.....	101
4.5	Métaphore de la navigation spatiale.....	103
4.6	Conclusion.....	104
5.	Visualisation	109
5.1	Taxonomies	109
5.1.1	Le type des données	110
5.1.2	Les traitements	112
5.1.3	Les tâches	117
5.1.4	Synthèse	118
5.2	Transformations de l'affichage	121
5.2.1	Distorsions.....	121
5.2.1.1	L'affichage bifocal	122
5.2.1.2	Les projections poly-focales	124
5.2.1.3	Les vues Fisheyes.....	126

5.2.1.4	Le tableau optique	128
5.2.2	Zoom	130
5.2.2.1	Space-scale Diagrams : représentation des ZUIs	131
5.2.2.2	Les interfaces zoomables d'exploration.....	133
5.3	Autres paradigmes d'interaction	136
5.3.1	Filtrage dynamique.....	136
5.3.2	Lentilles magiques	138
5.3.3	Excentric labeling.....	141
5.4	Structures linéaires et matricielles	142
5.4.1	Le mur perspectif	144
5.4.2	Le document optique.....	146
5.5	Structures hiérarchiques	147
5.5.1	Approche Pavage : le système Treemap	149
5.5.2	Approche 3D : les arbres coniques.....	150
5.5.3	Approche géométrique : les arbres hyperboliques	152
5.6	Structures de graphes	155
5.7	Espaces informationnels de travail.....	157
5.7.1	Les WebBooks et le Web Forager	158
5.7.2	Le Data Mountain	162
6.	Trouver l'information	165
6.1	Association vues globales / vues locales.....	165
6.2	Exemples de vues globales.....	166
6.2.1	Représentation perspective.....	166
6.2.2	Représentation concentrique hiérarchique	170
6.2.2.1	Visualisation 2D.....	170
6.2.2.2	Visualisation 3D.....	173
6.3	ZoomTree : un système de vues locales.....	175
6.3.1	Description	175
6.3.2	Interaction	179
6.4	Conclusion.....	185
7.	Retrouver l'information	187
7.1	Mémoire spatiale et visuelle.....	188
7.1.1	Localisation spatiale.....	188
7.1.2	Organisation	188
7.1.3	Représentation des documents	189
7.2	Tables de travail 2D	190
7.2.1	Description	190
7.2.2	Structuration hiérarchique.....	191
7.2.3	Navigation	194
7.3	Conclusion.....	198
IV.	Conclusion et perspectives.....	199
<i>Index</i>	<i>201</i>

Publications 205
Bibliographie 207

Liste des figures

Figure I.1 : Exemples de documents numérisés.....	17
Figure I.2 : Exemple d'intégration informatique.	18
Figure 1.1 : Hypermédia, organisation non séquentielle de l'information.	25
Figure 1.2 : Système Memex.	30
Figure 1.3 : Système Xanadu, visualisation des connexions (1972).....	31
Figure 1.4 : Système Xanadu (réalisations de spécifications publiées en 1972).....	32
Figure 1.5 : Une « Frame » dans ZOG/KMS.	34
Figure 1.6 : Modèle en carton du Dynabook.....	35
Figure 1.7 : Système Hyperties.	37
Figure 1.8 : Système Document Examiner.	38
Figure 1.9 : Hyperliens du système Document Examiner.....	38
Figure 1.10 : Système Intermédia.	39
Figure 1.11 : Une carte dynamique de localisation dans Intermédia.	40
Figure 1.12 : Système Guide.....	42
Figure 1.13 : Système NoteCards.....	43
Figure 1.14 : Système NoteCards, exemple de cartes « browser » et « filebox ».....	44
Figure 1.15 : Schéma conceptuel de l'environnement HyperCard.	45
Figure 1.16 : Le modèle Dexter.	49
Figure 2.1 : Exemples de livres électroniques (e-books).	53
Figure 2.2 : Tablette graphique à affichage haute résolution (système XLibris).....	54
Figure 2.3 : Exemples de manuscrits d'œuvres littéraires.	57
Figure 2.4 : Indirection induite par les annotations des fac-similés numériques.	59
Figure 2.5 : Image de manuscrit particulièrement difficile à analyser.....	64
Figure 2.6 : Le système d'annotation et de transcription.	65
Figure 2.7 : Sélectionner des zones spécifiques dans un fac-similé numérique.	66
Figure 2.8 : Précision de la sélection avec un marqueur de type lasso.	66
Figure 2.9 : L'interface d'annotation / transcription.	67
Figure 2.10 : Consultation contextuelle des éléments de la transcription.....	69
Figure 2.11 : Segmentation automatique de lignes de textes manuscrits, composantes connexes et alignements.....	69
Figure 2.12 : Une annotation de type lien, avec effet de transparence.	70

Figure 2.13 : Une annotation de type commentaire, sans effet de transparence.	71
Figure 2.14 : Sémantique de la balise <code><xptr></code>	75
Figure 2.15 : Modélisation hypertexte du poste de lecture active.....	76
Figure 2.16 : Deux méthodes de représentation des données XML/TEI.	77
Figure 2.17 : Rendu XML/TEI dans un navigateur Web après conversion XSLT....	78
Figure 2.18 : Combinaison de briques graphiques avec le toolkit Ubit.....	80
Figure 2.19 : Correspondance entre l'arbre XML et l'arbre Ubit.	82
Figure 2.20 : Projection graphique de l'arbre XML/Ubit.	83
Figure 4.1 : Points de repères sur la page d'accueil d'un site Web.	93
Figure 4.2 : Visites guidées dans NoteCards.	94
Figure 4.3 : Exemples d'historiques de navigation présentés sous forme de listes. ...	96
Figure 4.4 : Système PadPrints, présentation graphique d'un historique de navigation Web.	97
Figure 4.5 : Système XLibris, exemple d'annotation d'un document électronique...	98
Figure 4.6 : Bookmarks temporaires et persistants dans Deckview.....	99
Figure 4.7 : Système MacWeb, une vue du <i>web</i> des types (à gauche) et du web principal.....	101
Figure 4.8 : Système WebTOC, une vue d'ensemble hiérarchique d'un site Web..	102
Figure 5.1 : Taxonomie proposée par Leung.	113
Figure 5.2 : Taxonomie BVI proposée par Chuah.	114
Figure 5.3 : Modélisation du processus de visualisation proposée par Chi.	116
Figure 5.4 : Cadre d'étude du processus de visualisation proposé par Vernier.	119
Figure 5.5 : Principe de l'affichage bifocal.....	122
Figure 5.6 : L'affichage bifocal.	123
Figure 5.7 : Implémentation d'un affichage bifocal 2D.....	123
Figure 5.8 : Projection poly-focale.....	124
Figure 5.9 : Distorsions 3D de surfaces planes.	125
Figure 5.10 : Distorsions de formes 3D.	126
Figure 5.11 : Vue initiale (issue de Sarkar et Brown (1992)).	127
Figure 5.12 : Vue fisheye (issue de Sarkar et Brown (1992)).....	128
Figure 5.13 : Le tableau optique.	129
Figure 5.14 : Exemple de zoom géométrique appliqué à l'exploration d'une base d'images hiérarchique (Holmquist 1998).....	130
Figure 5.15 : Exemple schématique de zoom sémantique.	131
Figure 5.16 : Construction d'un « space-scale diagram » à partir d'une image 2D.	132

Figure 5.17 : Représentation de la fenêtre de visualisation utilisateur dans un « space-scale diagram ».....	132
Figure 5.18 : Trajectoires d'exploration dans une interface zoomable.	133
Figure 5.19 : Système Pad++, exploration d'une base documentaire.	134
Figure 5.20 : Système Zomit, exploration d'une base de données génétiques.	135
Figure 5.21 : Système HomeFinder.	137
Figure 5.22 : Système FilmFinder.....	138
Figure 5.23 : Exemples de lentilles magiques.....	139
Figure 5.24 : Composition de deux lentilles magiques.	140
Figure 5.25 : Lentilles magiques « click-through » de changement de couleur.	140
Figure 5.26 : Lentille volumétrique dans un environnement 3D.	141
Figure 5.27 : Excentric labeling.	142
Figure 5.28 : L'alphaslider.	143
Figure 5.29 : Système Infogrid, affichage d'une matrice de documents.	144
Figure 5.30 : Le mur perspectif.....	145
Figure 5.31 : Le document optique (vue initiale).....	146
Figure 5.32 : Le document optique (vue focus+contexte pyramidale).	147
Figure 5.33 : Une arborescence de fichiers (Explorateur Windows).	148
Figure 5.34 : Le système Treemap.	150
Figure 5.35 : Les arbres coniques.	151
Figure 5.36 : Les arbres coniques (visualisation horizontale).....	152
Figure 5.37: Affichage hyperbolique 2D d'une hiérarchie.	153
Figure 5.38 : Affichage hyperbolique 3D d'une hiérarchie.	154
Figure 5.39 : Système Narcissus.	156
Figure 5.40 : L'Information Visualizer.	158
Figure 5.41 : Un WebBook.	159
Figure 5.42 : Feuilletage d'un WebBook.	160
Figure 5.43 : Vue d'un WebBook avec le système Document Lens.	161
Figure 5.44 : Le Web Forager.	162
Figure 5.45 : Le Data Mountain.	163
Figure 6.1 : Vue structurelle perspective (version 2D et 3D),	167
Figure 6.2 : Modularité du niveau de détail.	168
Figure 6.3 : Les hyperliens sont affichés sur la face arrière.....	169
Figure 6.4 : Représentation concentrique (vue structurelle hiérarchique 2D).	171
Figure 6.5 : Exploration de la hiérarchie.....	172

Figure 6.6 (a & b) : Affichage des hyperliens.....	173
Figure 6.7 : Représentation sphérique perspective.	174
Figure 6.8 : ZoomTree, un système de vues locales	176
Figure 6.9 : Vue locale focus+contexte de plusieurs pages Web.....	177
Figure 6.10 : Une représentation multi-focus.	178
Figure 6.11 : Changement de taille proportionnel d'une page Web.	180
Figure 6.12 : Vue initiale des Figures 6.13 et 6.14.	181
Figure 6.13 : Zoom Hiérarchique sur la partie A de la Figure 6.12.....	182
Figure 6.14 : Zoom contextuel à partir de B sur la Figure 6.12.....	183
Figure 6.15 : Vue initiale de la Figure 6.16 (avant fermeture).	184
Figure 6.16 : Résultat d'une opération de fermeture appliquée à la fenêtre-document A de la Figure 6.15.....	185
Figure 7.1 : Une table de travail 2D.....	191
Figure 7.2 : Organisation hiérarchique des tables de travail.....	192
Figure 7.3 : Compatibilité des bookmarks entre les tables de travail et les navigateurs Web classiques.....	193
Figure 7.4 : Une vue globale concentrique de la hiérarchie des tables de travail. ...	194
Figure 7.5 : Un aperçu sur une table de travail à partir de la vue globale.....	195
Figure 7.6 : Un aperçu affiché dans une table de travail.....	196
Figure 7.7 : Une vue mutli-focus d'aperçus.....	197
Figure 7.8 : Résultat d'une transformation Fisheye sur la Figure 7.5.....	198

Liste des tableaux

Tableau I.1 : Tâches de haut niveau associées à l'exploration hyper-documentaire.	20
Tableau 1.1 : Développement des hypermédias.....	29
Tableau 4.1 : Aides aux différentes activités liées à l'exploration hyper-documentaire.	106
Tableau 5.1 : Types de données dans la taxonomie de Shneiderman.	111
Tableau 5.2 : Tâches dans la taxonomie de Shneiderman.....	112
Tableau 5.3 : Etapes des données dans DSM (Chi).	117
Tableau 5.4 : Opérateurs de transformation dans DSM (Chi).	117

I. Introduction

Le développement du multimédia, des capacités de stockage et du matériel de numérisation permet de mettre à disposition sous une forme électronique un nombre toujours croissant de documents « matériels » (journaux, plans, cartes, peintures, partitions musicales, photographies, cahiers d'expérimentations, originaux d'œuvres littéraires, rapports, notes et autres documents manuscrits, etc.).

La numérisation, le stockage et la diffusion de ces matériaux sources (Figure I.1) ont fait l'objet de nombreux travaux et ne posent aujourd'hui plus réellement problème. Toutefois, il semble que moins d'attention ait été portée à la conception et la réalisation d'environnements interactifs destinés à faciliter une intégration « forte » de ces matériaux en milieu informatique.



Figure I.1 : Exemples de documents numérisés.

On observe en effet que la mise à disposition de documents sous forme électronique entraîne un accroissement considérable du nombre de documents papiers. Les utilisateurs impriment le plus souvent les documents qu'ils jugent intéressants avant de travailler avec. Cette pratique s'explique en grande partie par un manque d'outils logiciels qui permettraient aux utilisateurs de mettre à profit leurs habitudes de

travail, acquises dans le « monde réel », afin de supporter la réalisation de leur *activité documentaire*.

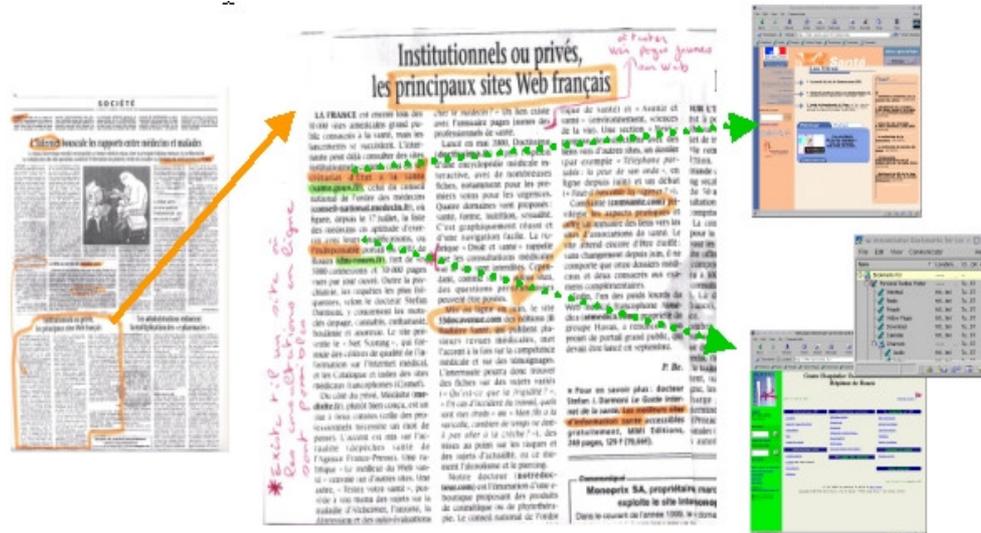


Figure I.2 : Exemple d'intégration informatique.

annoter, collecter, visualiser... en contexte dans un même environnement.

Activité documentaire

Le concept d'*activité documentaire* [Schilit99] concerne la manière dont les gens consultent, manipulent et travaillent avec un ensemble varié de documents papiers en vue de l'accomplissement d'un ou plusieurs objectifs. Ce concept, qui sous sa forme la plus simple correspond à la lecture linéaire d'un document, regroupe un ensemble de tâches plus ou moins complexes qui sont souvent en étroites relations.

Afin d'illustrer ceci, prenons l'exemple d'un enseignant en histoire qui souhaite préparer un cours sur un événement particulier.

Cet enseignant devra d'abord *trouver* les documents relatifs à l'événement étudié (cartes, rapports, notes et observations d'époque, documentation connexe, etc.), les *lire* ou les *survoler*.

Il sera ensuite amené à *identifier* les éléments pertinents, à les *collecter* et à les *organiser* afin de pouvoir les réutiliser, ce qui nécessitera de pouvoir les *retrouver* aisément par la suite.

Il devra également *analyser* l'information trouvée afin de produire son cours. Cette phase s'accompagne souvent d'une *prise de notes* (soulignements, liens, schémas, commentaires, etc.), ce qui a pour effet de transformer le lecteur en « acteur ». Ce dernier s'approprie le document en y ajoutant des marques pour focaliser son

attention sur certaines parties, pour faciliter l'assimilation des notions rencontrées, etc. La consultation d'un document se transforme alors en *lecture active*.

C'est la réalisation de ces activités, qui nous paraît naturelle et assez simple sur documents papiers, qui s'avère non triviale dans un environnement informatique par manque de mécanismes appropriés.

L'objectif de cette thèse consiste à faciliter l'intégration informatique de documents électroniques en facilitant l'activité documentaire des utilisateurs au moyen des capacités offertes par les systèmes hypermédias, l'interaction homme-machine et la visualisation d'information. Cette intégration peut être vue comme une *dématérialisation des documents* qui vise à réduire la rupture entre la mise à disposition et l'utilisation des documents (Figure I.2).

Hypermédia

Le développement des systèmes hypermédias a permis de mettre à la disposition d'un large public de grandes quantités d'informations de nature variée (texte, image, son, etc.). Ces systèmes organisent les documents en termes de nœuds et de liens, et proposent un paradigme intuitif d'accès à l'information [Nielsen90a], à savoir la navigation interactive qui s'effectue simplement par associations d'idées en suivant les liens rencontrés.

Si les systèmes hypermédias fournissent une base solide à l'intégration informatique des documents grâce à leurs possibilités d'accès (proche de la nature associative de l'esprit humain) et de structuration, les mécanismes qu'ils proposent s'avèrent souvent insuffisants pour la gestion de l'activité documentaire.

Au sein de tels systèmes, l'activité documentaire peut être rebaptisée activité *hyper-documentaire*. Dans ce mémoire, nous montrerons comment faciliter la partie de cette activité hyper-documentaire liée à la manipulation et à la consultation des documents, que nous appelons *exploration hyper-documentaire*.

Nous identifions cette exploration hyper-documentaire par quatre tâches de haut niveau (Tableau I.1).

Trouver : cette activité consiste à accéder aux documents recherchés en fonction d'un objectif, d'un centre d'intérêt, etc. Cet accès peut s'effectuer par navigation en suivant les liens rencontrés, mais également au moyen de moteurs de recherche (comme s'est souvent le cas sur le Web par exemple), de tables d'index ou encore de vues qui montrent la structure des documents dans le système hypermédia.

Comprendre : cette activité consiste à analyser, comparer et interpréter le contenu des documents ainsi que les relations sémantiques et/ou structurelles qui associent ces documents. Par exemple, l'affichage d'une vue d'ensemble peut donner des

informations sur l'organisation ou le nombre des documents, l'affichage simultané de deux documents dans des fenêtres différentes placées côte à côte peut aider à les comparer, etc.

Organiser : cette activité consiste à classer les documents jugés intéressants en vue d'une utilisation future. Elle nécessite de collecter les documents et de pouvoir en garder une trace. Les gestionnaires de fichiers traditionnels sont des exemples de systèmes qui permettent aux utilisateurs de gérer interactivement l'organisation de leurs données.

Retrouver : cette activité consiste à accéder une nouvelle fois à des documents préalablement consultés, sans repasser par les étapes de recherche (interrogation, navigation). Les bookmarks Web sont des exemples de mécanismes qui permettent ce type d'activité. Les icônes que l'on dispose sur le bureau de travail (MacOS, Windows) fournissent également un moyen de retrouver plus rapidement les documents qu'elles représentent.

Tâches	Description
Trouver	Accéder aux documents recherchés (en fonction d'un objectif, d'un centre d'intérêt, etc.).
Comprendre	Identifier les relations sémantiques et/ou structurelles entre les documents, faire des recoupements, associer des idées, analyser le contenu, etc.
Organiser	Etablir un schéma de classification personnalisé des documents jugés intéressants, après les avoir collecter.
Retrouver	Accéder une nouvelle fois, plus simplement et plus rapidement, à des documents préalablement consultés.

Tableau I.1 : Tâches de haut niveau associées à l'exploration hyper-documentaire.

Cette thèse propose deux types d'approches pour faciliter la réalisation de ces activités en milieu informatique.

La première consiste à permettre aux utilisateurs d'enrichir les documents sources au moyen de techniques hypertextes d'annotation. Les utilisateurs créent ainsi leur propre système de repérage et d'information qui est utilisé pour faciliter la compréhension et la réutilisation de ces documents. Une attention particulière est portée à la manière de représenter les données ajoutées et de structurer les différentes informations manipulées.

La seconde approche consiste à faciliter la manipulation et l'accès aux documents au sein de leur base informationnelle au moyen de techniques interactives de visualisation.

Organisation du mémoire de thèse

Ce mémoire de thèse s'organise en deux parties. La première partie inclue une présentation du domaine des hypermédias (Chapitre 1) et la description des techniques hypertextes d'annotation proposées (Chapitre 2). La deuxième partie décrit tout d'abord les problèmes liés à l'exploration interactive de base de données hypermédias (Chapitre 3). Elle présente ensuite les méthodes couramment employées pour réduire ces problèmes (Chapitre 4) en se focalisant sur des techniques de visualisation d'information (Chapitre 5). Cette partie se termine par la description des techniques interactives de visualisation que nous avons développées pour aider à trouver (Chapitre 6) et à retrouver (Chapitre 7) les documents au sein d'un espace informationnel de vaste envergure.

II. Manipulation et représentation des documents

1. Systèmes hypermédias

1.1 Concepts et définitions

Le terme hypertexte est apparu en 1965 dans un article de Theodor Nelson [Nelson65] pour désigner un ensemble d'écritures interconnectées de manière complexe ne pouvant pas être présentées efficacement sur un support papier. L'hypertexte s'est ensuite généralisé afin de représenter des morceaux de texte associés entre eux par des liens qui constituent un réseau à l'intérieur duquel il est possible d'effectuer des parcours non séquentiels.

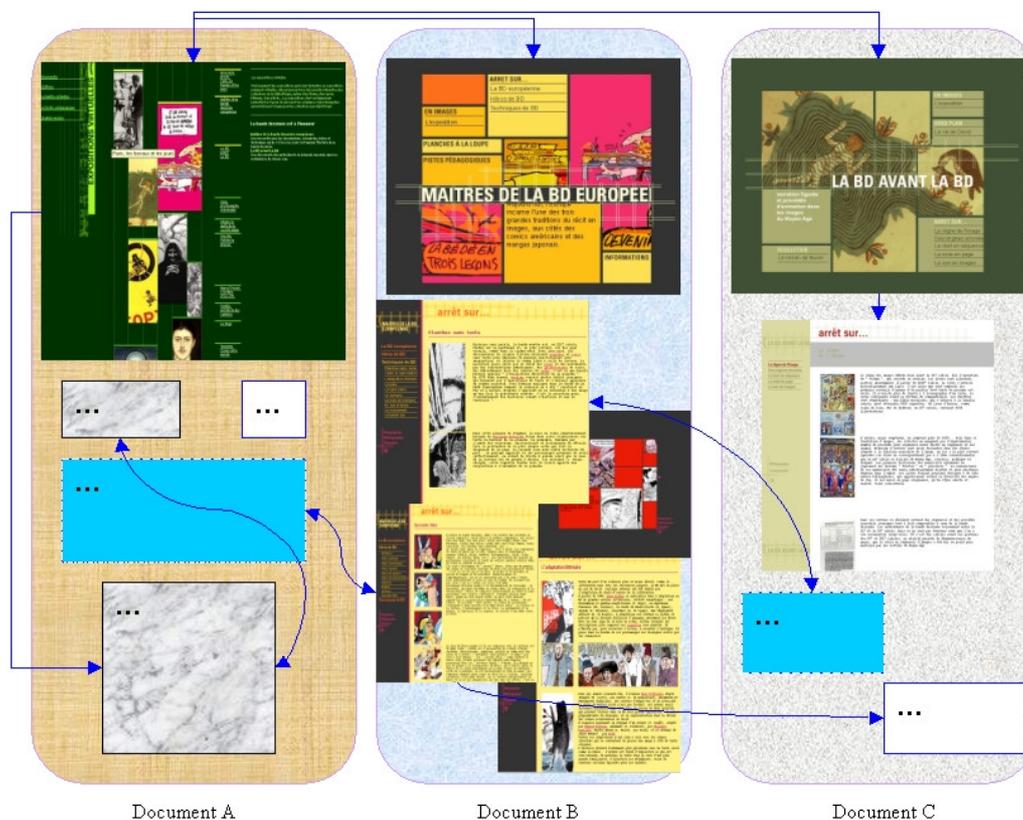


Figure 1.1 : Hypermédia, organisation non séquentielle de l'information.

Le concept d'hypermédia est une généralisation du concept d'hypertexte lorsque celui-ci est pris dans le sens restrictif d'écrit non linéaire. Il y a hypermédia dès lors que les unités d'information (les nœuds) du réseau hypertexte ne se limitent plus à contenir du texte mais des graphiques, des schémas, des séquences audio ou vidéo, etc.

« un hypermédia est un ensemble d'informations appartenant à différents types de médias pouvant être lus, écoutés, suivant de multiples parcours de lectures, en utilisant également la possibilité du multi-fenêtrage. Ce qui différencie essentiellement l'hypermédia de l'hypertexte n'est ainsi que la nature symbolique des codages d'informations utilisés. Un hypermédia n'est rien d'autre qu'un hypertexte gérant des textes supportés par des médias divers » [Balpe90].

Le concept d'hypermédia est donc né de la rencontre de l'hypertexte et du multimédia. La réalisation informatique de systèmes hypermédiés tire avantage du fait que, par nature, la numérisation permet de casser la séquentialité imposée par les supports traditionnels de communication (livre, film, disque, cassette audio).

Actuellement, les termes hypertexte et hypermédia tendent à désigner les mêmes choses. Dans la suite de ce mémoire nous emploierons donc indifféremment l'un et l'autre de ces deux termes (ainsi que le terme plus général d'hyperdocument) pour désigner tout document électronique accessible de manière non linéaire grâce à un cheminement sur un réseau de liens [Sandvad89]. Enfin, le terme hyperbase est utilisé pour désigner un espace informationnel structuré par un ensemble de liens.

En plus des entités structurelles sur lesquelles repose l'organisation des données (*les nœuds* d'information et les *liens*), les systèmes hypermédiés contiennent également l'outillage nécessaire à la consultation de ces données et au déplacement de l'utilisateur dans le réseau de liens. Ils proposent un nouveau paradigme d'accès à l'information, appelé la navigation, plus intuitif que les techniques d'interrogation utilisées dans les bases de données documentaires (requêtes, filtrage, éléments pertinents) ou que les techniques de déduction et d'induction sur lesquelles s'appuient les systèmes experts. La navigation libre est un moyen original d'appréhender l'information en permettant des associations non nécessairement guidées par des nécessités déductives, de cause à effet ou de tout autre nature logique, mais par des nécessités purement intuitives ou d'associations d'idées. Comme nous le verrons dans la suite de ce mémoire, la navigation nécessite la mise en place de mécanismes adéquats qui peuvent tirer avantage de techniques d'interaction et de visualisation.

1.1.1 Les nœuds

Les nœuds sont les unités d'information de l'hyperdocument. L'information contenue dans un nœud peut être d'un type unique (texte, graphique, son, vidéo, etc.), ou faire appel à plusieurs médias, comme c'est par exemple le cas d'un nombre important de pages Web. La notion de nœud peut également être généralisée à chaque objet d'information sur lequel le système peut raisonner [Bieber89]. Cette

définition permet de voir un nœud comme du code exécutable qui est appelé au moment où le lien menant à ce nœud est traversé. Concernant leur construction, les nœuds sont soit édités, soit calculés. Les nœuds calculés sont construits par traitement automatique ou assisté sur le contenu des nœuds édités, et servent généralement à structurer l'hyperbase. Différents types de nœuds peuvent être rencontrés dans un même hyperdocument (section 4.4.1).

1.1.2 Les liens

Les liens hypermédias (ou hyperliens) permettent d'associer les nœuds d'information entre eux afin de former un réseau à l'intérieur duquel il est possible de se déplacer. Ils constituent le principal moyen pour organiser les documents de manière non séquentielle et fournissent la base du système d'exploration: « *lorsque l'on traverse un lien, on quitte le nœud d'information source pour se rendre au nœud d'information destination* » [Halasz90]. Un lien peut être uni-directionnel ou bi-directionnel. Dans le cas d'un lien unidirectionnel on parle de référence pour la source, et de référent pour la destination.

Les ancrs correspondent aux extrémités d'un lien, c'est-à-dire à son origine et à sa (ou ses) destination(s). La destination d'un lien peut être un nœud ou une entité contenue à l'intérieur d'un nœud (ex: une zone dans une image). Dans le cas d'un nœud structuré, cette entité sera un élément de la structure (ex : une balise dans un fichier HTML, XML, etc.) alors que dans le cas d'un nœud non structuré l'entité sera définie par une quelconque mesure de localisation dépendant du contenu du nœud (des caractéristiques physiques des données). Cette localisation peut par exemple correspondre à une suite de caractères dans un texte, à une zone repérée par ses coordonnées en deux dimensions dans une image bitmap, ou encore à un laps de temps dans une bande sonore. L'ancre associée à l'origine d'un lien doit repérer une unité sémantique de niveau inférieure à celui du nœud afin que l'utilisateur puisse repérer et sélectionner le lien associé lors de la consultation du nœud. Elle correspond donc forcément soit à un élément d'un nœud structuré soit à une localisation précise d'un nœud non structuré.

La forme la plus simple d'un lien correspond à une connexion fixée entre deux ancrs (des nœuds ou des entités à l'intérieur des nœuds). Le traitement de base consiste à réaliser la séquence « sélectionner - traverser - afficher ». Cependant la forme et le traitement d'un lien sont des notions qui peuvent toutes deux être étendues [Bieber89, DeRose89, DeRose98a]. Un lien peut être dynamique et se créer en cours d'exécution en réponse à un événement particulier. La traversée d'un lien peut entraîner une modification des données contenues dans les nœuds sources et destination. Le nœud destination peut être inséré dans le nœud source à la position de l'ancre du lien qui les associe [DeRose98a], etc.

La création des liens s'effectue soit manuellement (au moyen d'un système interactif ou d'un langage de description) soit automatiquement. Les liens calculés, résultats d'une procédure automatique, sont dits fixes s'ils ont été construits pendant la phase d'édition de l'hyperdocument. Ils sont dynamiques s'ils découlent d'un calcul en temps réel (par exemple, les moteurs de recherche du Web permettent au lecteur de créer des liens qui sont calculés au moment de la consultation et qui sont présentés sous forme de résultats de requêtes).

Différents types de liens peuvent être rencontrés dans un même hyperdocument (voir section 4.4.1).

1.2 Historique

Le développement des hypermédias résulte de nombreux travaux et d'interconnexions entre sciences technologiques et sciences humaines. Les limitations imposées par le matériel, en termes de stockage, de puissance de calcul, de graphisme, etc., ont longtemps freiné le développement des idées fondatrices du domaine. Brosser l'historique des hypermédias, en guise d'introduction pour donner une rapide vue d'ensemble, peut se résumer comme suit. En 1945, Vannevar Bush [Bush45] donne une première idée d'un système de type hypertexte. C'est seulement vingt ans plus tard que Douglas Engelbart propose une première réalisation de ces idées [Engelbart63, Engelbart68]. Les termes hypertexte et hypermédia sont inventés par Theodor Nelson à la même époque [Nelson65]. Le monde informatique découvre réellement le concept d'hypertexte en 1987 grâce à deux événements majeurs. Apple introduit le logiciel *HyperCard* dans sa gamme de produits et l'ACM (*Association of Computer Machinery*) organise la première conférence internationale du domaine. Il se développe alors un réel engouement pour le domaine des hypertextes. Ce phénomène s'est ensuite accru avec les avancés technologiques (interfaces graphiques, capacités de stockage, etc.), le développement de « l'audiovisuel interactif », du multimédia et des jeux vidéo.

1.2.1 Evolution du domaine (échelle de temps)

Les principales évolutions du domaine des hypermédias résultent d'idées et d'applications novatrices, référencées dans le tableau suivant. Loin d'être exhaustive, la liste proposée ci-dessous définit les grandes lignes de l'historique en citant les travaux les plus mentionnés dans les études de références [Conklin87, Nielsen90a, Nyce91, Tomek91]. Les dates indiquées permettent de situer les évolutions répertoriées. Elles sont toutefois approximatives en raison de la durée des travaux de recherches pouvant être étalés sur plusieurs années.

Nous avons pris le parti d'arrêter l'historique proposé ci-dessous en 1993 pour deux raisons. Tout d'abord, à cause de la profusion des applications qui ont été proposées

depuis l'avènement des ordinateurs multimédia. Ensuite, car depuis le début des années 90, le développement de ces systèmes est lié de manière inextricable à celui de l'Internet (un sujet à part entière, traité dans cette thèse uniquement du point de vue de l'exploration interactive de l'espace documentaire).

Année	Concepteur	Système
1945	Vannevar Bush	Memex
1965	Theodor Nelson	Xanadu
1967	Andy Van Dam	HES, FRESS
1968	Douglas Engelbart	Augment / NLS
1975	Robert Akscyn	ZOG / KMS
1977	Alan Kay	Dynabook
1978	Andrew Lippman	Aspen Movie Map
1983	Ben Shneiderman	Hyperties
1985	Janet Walker	Document Examiner
1985	Norman Meyrowitz	Intermedia
1986	Peter Brown	Guide
1987	Frank Halasz	NoteCards
1987	Bill Atkinson	Hypercard
1989	Tim Berners-Lee	World Wide Web
1993	NCSA	Mosaic

Tableau 1.1 : Développement des hypermédias.

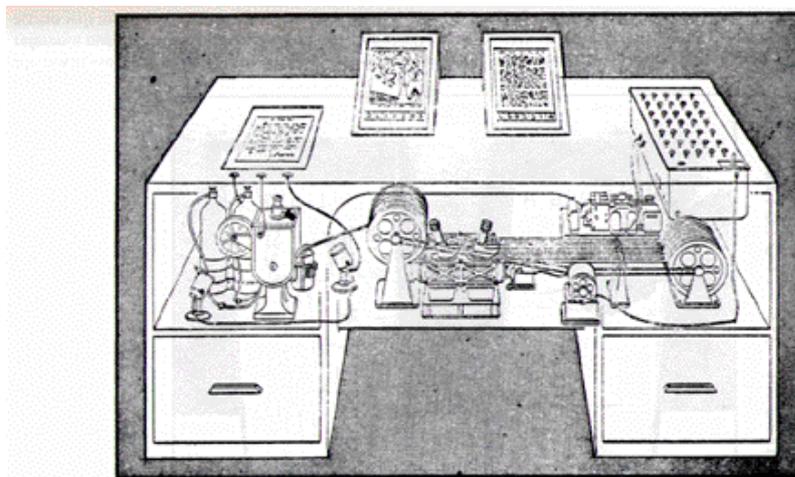
1.2.2 Pionniers et applications

1.2.2.1 Vannevar Bush: Memex (1945)

En 1945, Vannevar Bush propose l'idée d'un système de type hypertexte dans un article intitulé « As we may think » [Bush45]. Il y décrit le système *Memex*, pour *memory expander*, un dispositif mécanique destiné à « étendre la mémoire humaine ».

« A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory ».

Bush considère l'idée de mémoire associative comme primordiale. Il introduit le concept de « rails » pour lier des documents au moyen d'embranchements. Les documents sont organisés de manière non linéaire afin de simuler la nature associative de l'esprit humain. Le système Memex propose également des mécanismes qui permettent d'explorer et d'annoter de l'information textuelle ou graphique. Bien que ces idées aient été proposées pour des médiums de stockage tels que des cartes perforées ou des microfilms, l'analogie avec les hypertextes modernes est flagrante.



Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference (*LIFE* 19(11), p. 123).

Figure 1.2 : Système Memex.

Au vu des possibilités techniques de l'époque, aucun prototype fonctionnel de ce système n'a cependant pu être développé.

Nouveauté : stockage et recherche d'information par association d'idées.

1.2.2.2 Theodor Nelson: Xanadu (1965)

Theodor Nelson invente les termes hypertexte et hypermédia en 1965 alors qu'il travaille sur le projet Xanadu [Nelson65]. Ce projet, conçu au début des années 60 et inspiré des idées de Vannevar Bush [Nelson72], a pour objectif principal la construction d'un système universel de publication en hypertexte où les auteurs seront rémunérés automatiquement grâce à un « mécanisme de micro-paiement de royalties ». Pour ce faire, le projet Xanadu se base sur les paradigmes d'édition électronique et de bibliothèques distribuées. L'idée consiste à disposer d'un réseau, accessible en ligne, en temps réel, qui permettrait à chacun de lire, écrire, commenter, comparer, relier des documents de tous types (« the magic place of literary memory where nothing is forgotten » [Nelson87]). Le World Wide Web peut être considéré comme une forme de concrétisation de cette idée de Nelson, même si « Xanadu » propose des fonctionnalités plus riches comme le *backtracking* ou le *versioning* pour la comparaison des documents.

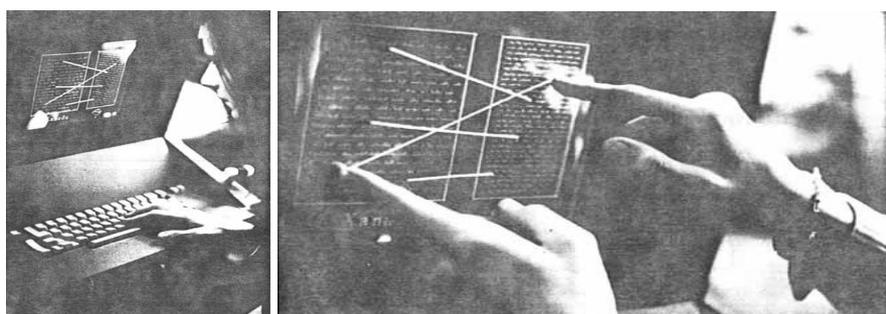


Figure 1.3 : Système Xanadu, visualisation des connections (1972).

Depuis 1994, Theodor Nelson travaille sur une nouvelle version de Xanadu (Figure 1.4) qui se construit autour de Netscape et du World Wide Web. Cette version possède un mécanisme baptisé la *transclusion*, qui permet à un document d'être simultanément à plusieurs endroits. Un même document ou un même extrait (texte, illustration, etc.) peut alors être intégré dans plusieurs autres contextes. Il ne s'agit pas de copies, mais d'un original unique : lorsqu'on crée un lien avec ce document dans un environnement donné, on retrouve ce lien dans tous les autres contextes. Le document n'est donc pas dupliqué, mais inclus simultanément dans divers environnements. Le concept de transclusion vise à faire découvrir aux lecteurs les différents sens possibles d'un document selon le contexte. Le concept de transclusion permet également une gestion du droit d'auteur évoluée : comme le texte n'est pas

dupliqué, il est possible de rémunérer l'auteur en fonction du nombre de caractères de son texte lus sur chaque écran à travers le monde.

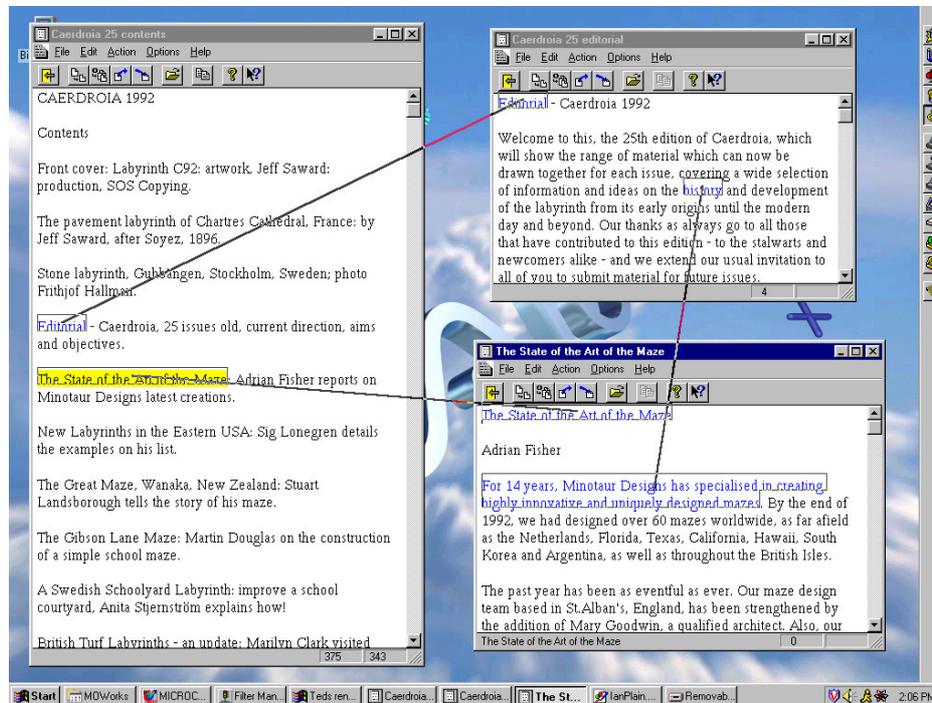


Figure 1.4 : Système Xanadu (réalisations de spécifications publiées en 1972).
Les liens adhèrent au contenu même lors du déplacement des fenêtres à l'écran ou du défilement de leur contenu.

Nouveautés : termes hypertexte et hypermédia,
transclusion,
backtracking, versioning,
copyright, royalties.

1.2.2.3 Andy Van Dam: Hypertext Editing System (1967)

Le premier système hypertexte entièrement fonctionnel (The Hypertext Editing System, HES) a été conçu et implémenté à la fin des années 60 à l'université de Brown par une équipe incluant Andy van Dame et Theodor Nelson [Carmody69]. Le système organise les données en segments de textes qui peuvent être connectés librement au moyen de deux types de liens. Les liens « In-text », repérées par un astérisque, permettent des déplacements intra ou inter segments. Les embranchements permettent de sélectionner un nouveau segment à partir d'un menu situé à la fin du segment courant. HES offre également des possibilités d'annotation.

La navigation et l'édition s'effectue uniquement au moyen de fonctions claviers. L'affichage se décompose en trois parties distinctes : une zone de texte munie d'ascenseurs qui permettent de faire défiler le texte, un espace réservé aux annotations et une zone de contrôle qui donne accès aux options du systèmes.

Le système HES a tout d'abord été acheté par la NASA afin de produire la documentation du programme Apollo. Les évolutions successives du système ont donné lieu à une version commerciale qui a été distribuée sous le nom de FRESS (a File Retrieval and Editing System). FRESS, comme son prédécesseur, est un système purement textuel. Les auteurs créent leurs documents au moyen d'un éditeur de texte interactif qui dispose de fonctionnalités de formatage. La création d'un lien nécessite la pose d'une marque qui peut être insérée à n'importe quel point du document. Ces marques sont les sources des liens qui pointent vers un autre élément textuel appartenant soit au même document soit à un document extérieur. Deux types de liens, les balises et les sauts, sont disponibles. Les balises sont des liens unidirectionnels qui indiquent une connexion à un élément unique (une annotation, une définition ou une note de bas de page). Les « sauts » sont des liens bidirectionnels qui indiquent un passage vers un autre document. L'affichage du document destination est pris en charge par une nouvelle fenêtre, le lecteur pouvant ainsi facilement comparer différentes parties de l'hyperdocument. Les auteurs peuvent attacher des mots-clés aux liens qui servent à filtrer l'information lors de la consultation (recherche et navigation). FRESS a donné lieu à une dernière version (The Electronic Document System) en tirant avantage des avancées en informatique graphique (gestion des couleurs, aides à la navigation, etc.).

Nouveauté : premier système hypertexte complètement fonctionnel.

1.2.2.4 Douglas Engelbart: Augment: NLS (1968)

Le système NLS (oN-Line System) a été développé comme une partie du projet Augment. Le but de ces travaux consistait à montrer « qu'une assistance de l'être humain par l'ordinateur en matière de manipulation de l'information pouvait améliorer les capacités intellectuelles de l'utilisateur ».

« a conceptual framework that could augment and structure cognitive processes, concepts, symbols, processes, and physical artifacts »
[Engelbart63].

NLS est sans doute la première implémentation d'un système hypertexte « simple » d'utilisation. Ce système permet aux utilisateurs de stocker et de gérer leurs données (plans, programmes, rapports, bibliographies, etc.) dans un environnement de travail personnalisé afin de faciliter leurs tâches quotidiennes. Le principal apport de NLS

concerne l'utilisation de la souris comme un moyen de navigation par activation des liens (pointer et cliquer).

Nouveauté : interface souris.

1.2.2.5 Robert Akscyn: ZOG & KMS (1975)

KMS (Knowledge Management System) [Akscyn88] est un descendant du système ZOG dont le développement a débuté en 1972 à l'université Carnegie-Mellon. ZOG fut tout d'abord testé à bord d'un avion de transport de l'armée américaine comme un système de gestion de tâches et de documentation des procédures de vols. C'est en 1981 que Robert Akscyn et Donald McCracken ont commencé le développement de KMS, une version commerciale du système ZOG.

KMS (Figure 1.5) est un système hypermédia distribué dédié à la gestion de connaissances organisées en structures hiérarchiques. KMS repose sur paradigme extrêmement simple. Chaque écran contient soit un cadre (un nœud) soit deux cadres affichées côte à côte. Les nœuds peuvent contenir du texte et du graphique. Les références sont repérées par des cercles vides et les commandes par des cercles noirs. Deux types de liens sont disponibles. Les liens structurels désignent des relations hiérarchiques. Les liens annotation, repérés par le caractère « @ », définissent quant à eux des relations de nature associative.

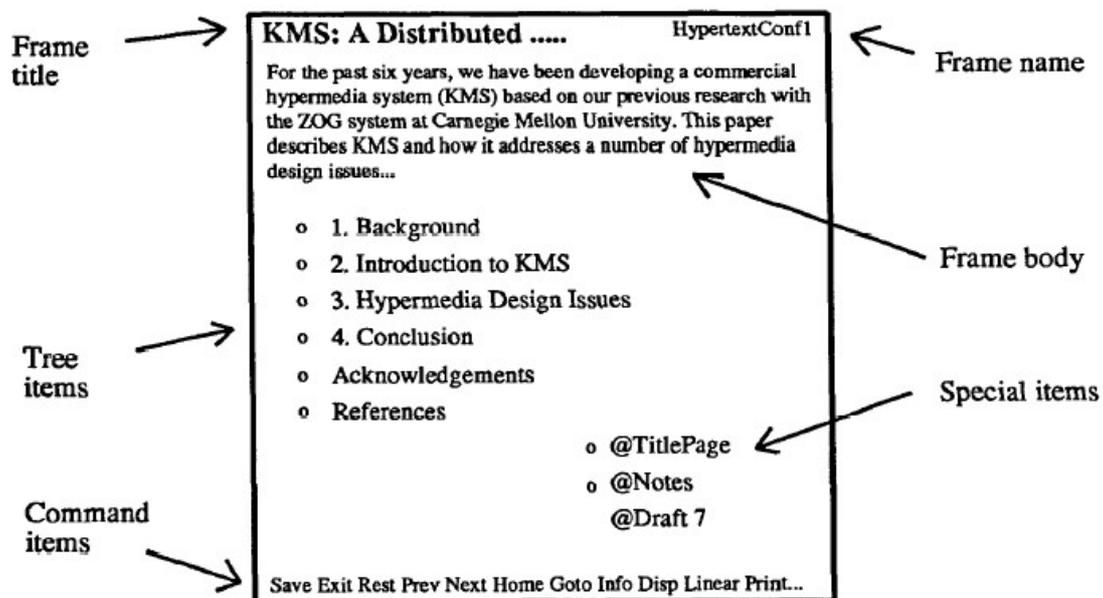


Figure 1.5 : Une « Frame » dans ZOG/KMS.

L'interface utilisateur se caractérise par l'absence d'interacteurs standard comme les menus déroulants, les boutons, les fenêtres, etc. L'idée consiste à faciliter l'utilisation du système en simplifiant et en unifiant les procédures d'interaction afin de diminuer le nombre de concepts à apprendre. L'utilisateur peut alors se concentrer sur l'opération en cours et non sur l'analyse des moyens mis à sa disposition pour la réaliser. Ainsi, aucune distinction n'est faite entre les opérations d'édition, de visualisation et de navigation. De même, le passage d'un nœud à un autre est préféré à l'enchaînement des données à l'écran via l'utilisation d'une barre de défilement. Les actions de l'utilisateur sont simplement déclenchées au moyen d'une souris trois boutons, le curseur à l'écran changeant de forme en fonction de sa position pour indiquer les opérations possibles. De cette manière, les fonctions d'édition et de navigation sont toutes exécutées sans avoir à passer par un quelconque système de menus. Des commandes globales additionnelles sont toutefois disponibles à partir d'une ligne en bas de l'écran, comme illustré à la Figure 1.5. Le système fournit également des fonctionnalités d'aide en ligne, d'importation de documents et de gestion de versions. Des recherches simples peuvent être effectuées sur un nœud et sa descendance (de manière hiérarchique), les résultats étant eux-mêmes affichés sous la forme d'un cadre.

KMS est un système fonctionnel des plus achevés qui fournit des fonctionnalités hypertextes avec un minimum de surcharge, accomplissant en cela les objectifs des concepteurs et développeurs.

Nouveautés : multi-utilisateurs, versioning, annotation

1.2.2.6 Alan Kay: Dynabook (1977)

Alan Kay propose en 1977, alors qu'il travaille au Xerox Palo Alto Research Center (PARC), l'idée d'un système hypermédia portable nommé Dynabook [Kay77].

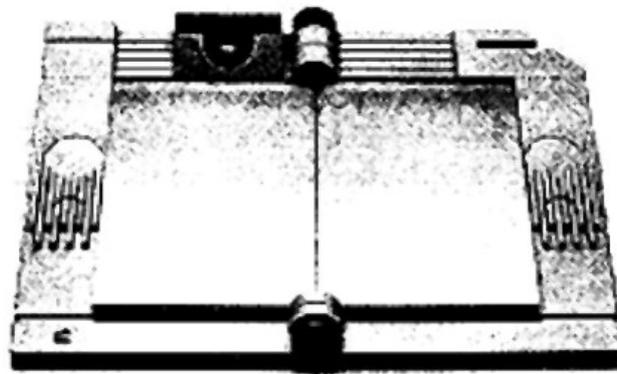


Figure 1.6 : Modèle en carton du Dynabook.

L'intention originale du Dynabook visait à concevoir un ordinateur de la taille d'un livre disposant d'un affichage sur un écran plat couleur haute résolution, d'une interface graphique, d'un environnement lecture/écriture complet et de liens de communication (via lignes téléphoniques et ondes radio) vers d'autres Dynabook et ressources réseaux. Seule une version non portable du système fut implémentée. Il servit toutefois de base vers la fin des années 70 à la création d'un ordinateur qui tenait sur un bureau et apportait plusieurs innovations dont un langage du nom de SmallTalk facilitant la manipulation directe et l'utilisation d'icônes dans une métaphore du bureau.

Nouveauté : idée d'un assistant numérique hypermédia portable.

1.2.2.7 Andrew Lippman: Aspen Movie Map (1978)

L'arrivée de nouveaux médias (graphique, vidéo, animation, etc.) dans le courant des années 70s-80s offre de nouvelles possibilités pour le développement de systèmes de type hypertexte. En 1978, Andrew Lippman et son équipe développe ainsi au MediaLab du MIT (Massachusetts Institute of Technology) le premier véritable système hypermédia (un système multimédia pourvu d'un réseau d'hyperliens). Ce système, baptisé Aspen Movie Map, permet l'exploration virtuelle et interactive d'une ville du Colorado (Aspen), au moyen de cartes géographiques, de vidéos de la ville (rues, bâtiments, etc.) et de liens hypertextes.

Le système utilise deux écrans, le premier sert à afficher la carte des rues de la ville, le second est quant à lui utilisé pour visualiser les vidéos. L'utilisateur peut naviguer à travers la carte en se déplaçant vers l'avant, l'arrière, la gauche ou la droite, et découvre la ville par l'intermédiaire des vidéos associées. Il est également possible de pointer n'importe quel lieu de la carte et sauter directement aux vidéos correspondantes sans avoir à définir explicitement son chemin à travers la ville.

Nouveauté : premier système hypermédia fonctionnel.

1.2.2.8 Ben Shneiderman: HyperTies (1983)

En 1983, Ben Shneiderman développe à l'université du Maryland le système TIES (The Interactive Encyclopedia System). Hyperties [Shneiderman87] est une évolution de ce premier système, dédiée à l'exploration d'information textuelle et graphique (Figure 1.7). Hyperties se caractérise par une interface « easy-to-use » (nombre de fonctionnalités raisonnable, présentation sous forme de page, interaction « touch-screen »), particulièrement bien adaptée aux systèmes d'information

publique (musées, aéroports, etc.). Le système de navigation contient un mécanisme d'historique, un module de recherche booléenne et un index. La principale nouveauté consiste à fournir aux utilisateurs des informations contextuelles sur la destination d'un lien afin de limiter les mauvais choix de navigation. Lorsque le pointeur de la souris est placé sur une ancre, une brève description du document destination est affichée en bas de l'écran.

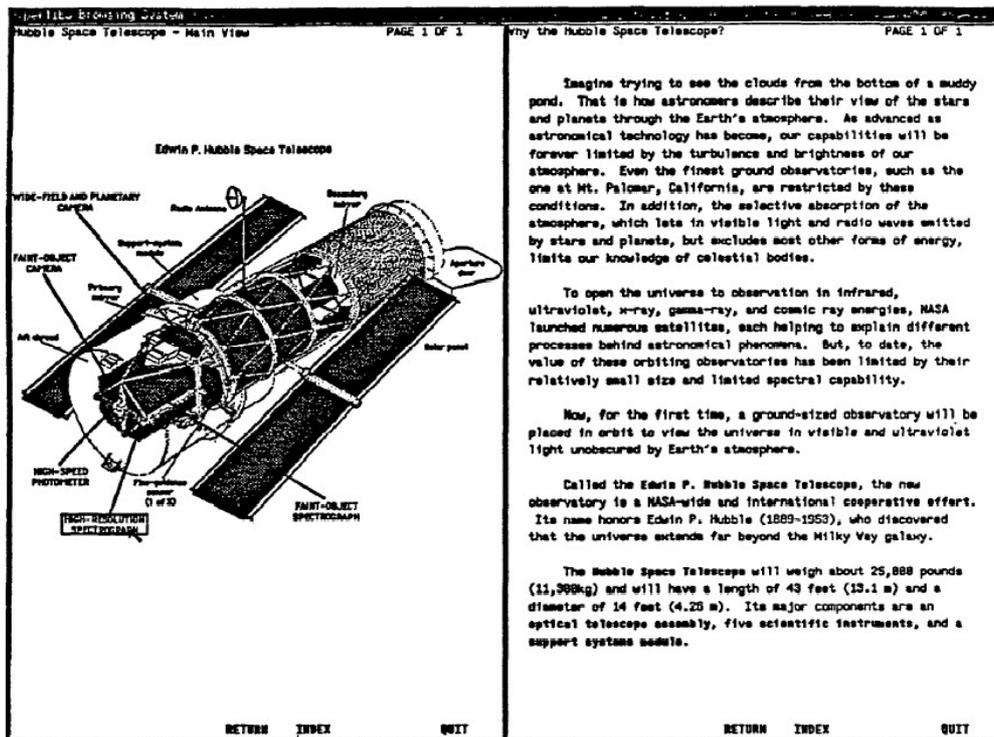


Figure 1.7 : Système Hypertex.

Nouveauté : interface « easy-to-use ».

1.2.2.9 Janet Walker: Document Examiner (1985)

Document Examiner [Walker87] est le premier système hypertexte professionnel à avoir été commercialisé. Il a été conçu comme un système d'aide à la consultation de documentations techniques (Figure 1.8) stockées dans des bases de données symboliques de taille moyenne.

Document Examiner se base sur la métaphore du livre, mais contrairement à des systèmes comme Xanadu ou NoteCards, sépare ses fonctionnalités auteurs de celles réservées aux lecteurs.

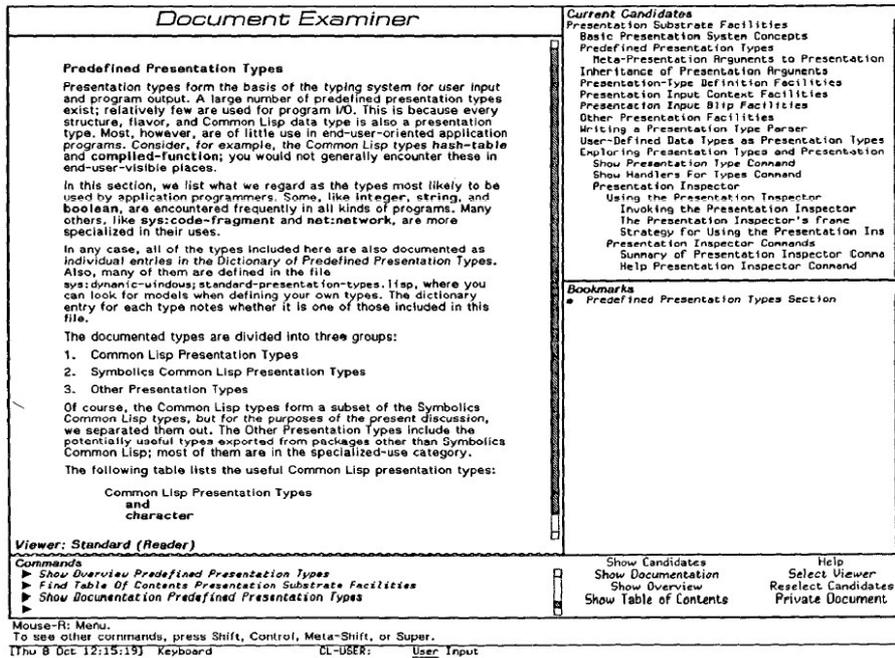


Figure 1.8 : Système Document Examiner.

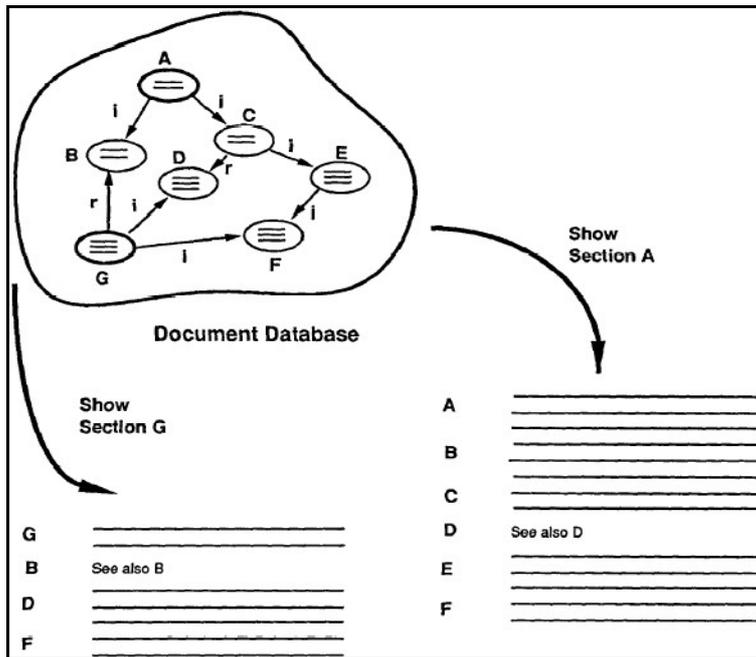


Figure 1.9 : Hyperliens du système Document Examiner.

Les flèches qui relient les nœuds correspondent aux hyperliens ; elles sont marquées d'un *i* pour indiquer l'inclusion ou d'un *r* pour la référence.

Les nœuds sont appelés des « enregistrements ». Chacun d'eux contient un document (ou un morceau de document) ainsi qu'un ensemble de méta-données comme le

numéro de version, un ensemble de mots-clés, le type de l'enregistrement ou encore une description de son contenu informatif. Les enregistrements sont connectés entre eux au moyen de deux types d'hyperliens (Figure 1.9). Les liens d'inclusion permettent d'insérer le contenu du document destination dans le document courant. Les liens de référence permettent de remplacer le document source par le document destination. Document Examiner dispose d'une fonctionnalité de recherche par mots-clés et implémente un mécanisme de bookmarks.

Nouveauté : application réelle, métaphore du livre, bookmarks.

1.2.2.10 Norman Meyrowitz: Intermedia / IRIS (1985)

Intermédia [Yankelovich88] a été conçu à l'IRIS (Brown University's Institute for Research in Information Scholarship), sur la base de l'expérience acquise lors du développement des systèmes HES, FRESS et EDS (section 1.2.2.3). Il est l'un des premiers systèmes à prendre en compte les problèmes inhérents aux hypermédias en réseau. Intermedia dispose d'un système de multi-fenêtrage avec interface graphique (Figure 1.10).

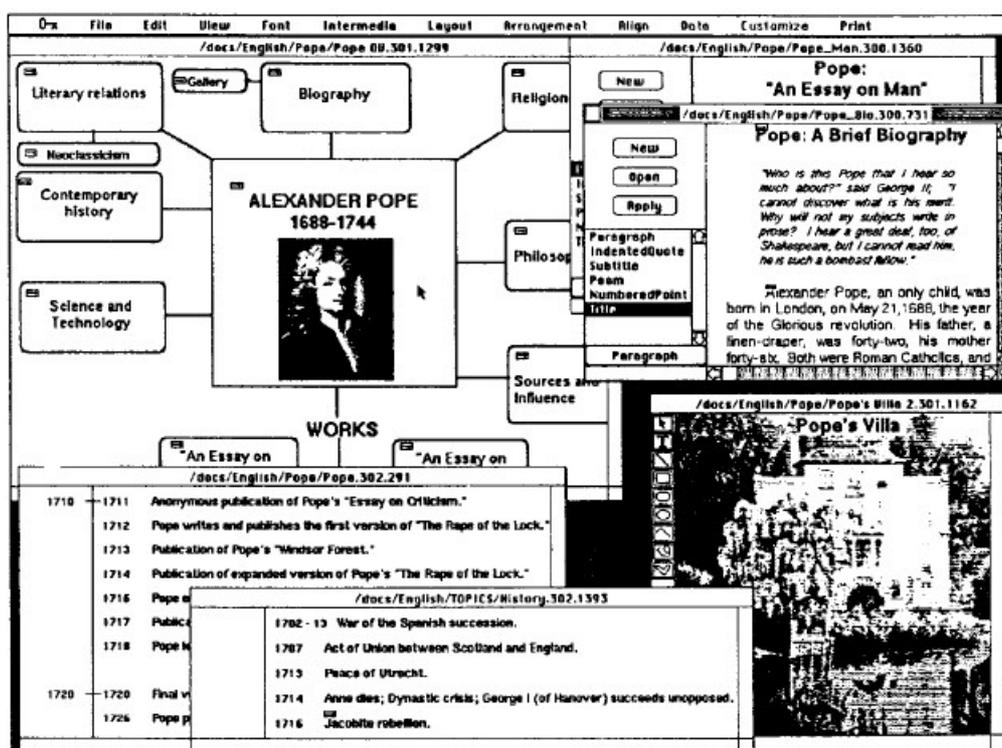


Figure 1.10 : Système Intermedia.

Les documents sont affichés dans des fenêtres qui peuvent se chevaucher et auxquelles sont associées des barres de défilements. Intermédia introduit l'idée d'utiliser une base de donnée séparée pour le stockage et la description des hyperliens (au lieu de les stocker à l'intérieur des documents), ceci afin de faciliter la gestion et la maintenance du réseau hypertexte. Les liens sont bidirectionnels. Les ancres sources peuvent être des lettres, des mots, des phrases, des paragraphes ou des icônes qui représentent des liens particuliers. Les ancres destinations ne se limitent pas à un document dans sa totalité, mais peuvent aussi correspondre à des chaînes de caractères à l'intérieur d'un document. Intermédia permet également la gestion et l'affichage de liens multiples.

Intermédia est un des premiers systèmes à proposer un mécanisme d'aide à la navigation. L'utilisateur peut visualiser sa position dans le réseau hypertexte au moyen d'une carte créée de manière dynamique qui affiche le nœud courant ainsi que tous les autres nœuds qui lui sont associés (les sources des liens qui pointent vers le nœud courant et les destinations des liens inclus dans ce nœud), comme illustré à la Figure 1.11.

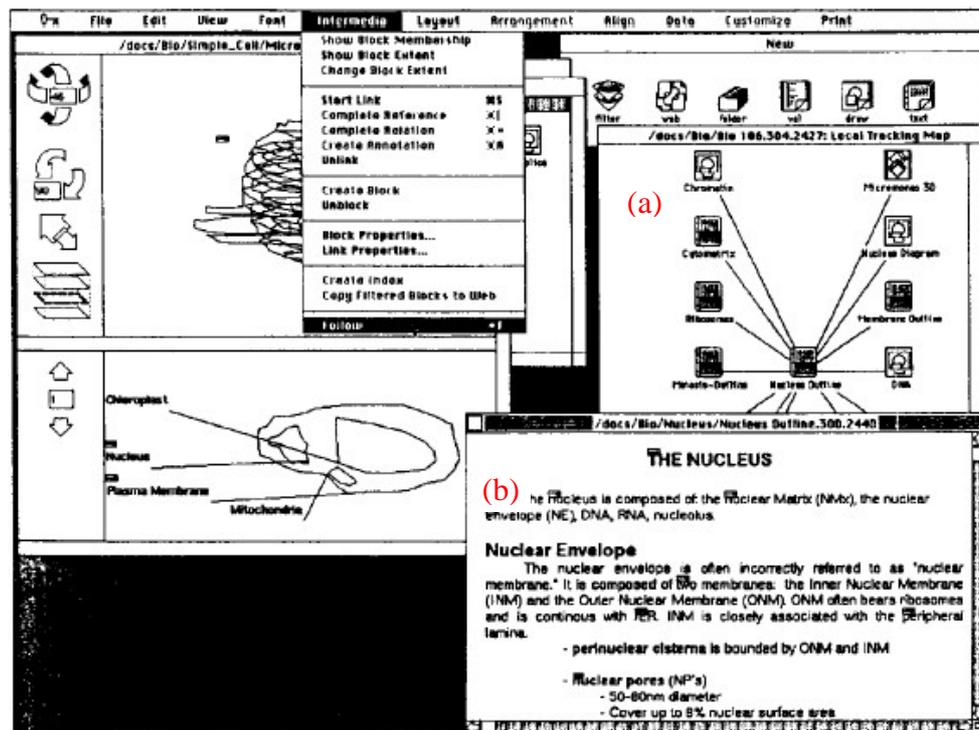


Figure 1.11 : Une carte dynamique de localisation dans Intermédia.

Le nœud affiché au centre de la carte, dans la fenêtre (a) correspond au nœud courant dont le contenu est affiché dans la fenêtre (b).

Intermédia est un outil d'édition et de consultation. Aucune distinction n'est faite entre les différents types d'utilisateurs. Le système, qui tourne sur des stations Unix

en réseaux, propose une large gamme d'applications pour éditer, lier, explorer ou encore annoter l'information disponible. Parmi ces applications, on trouve :

- ? InterText, un éditeur de texte qui permet également de définir des feuilles de style.
- ? InterDraw, un application de dessin qui contient une large palette outils graphiques
- ? InterPix, un application qui permet d'afficher, de découper, de copier et de coller des images scannées.
- ? InterSpect, une application qui permet de manipuler des modèles 3D.
- ? InterVal, une application qui permet de créer et de visualiser des données le long d'échelles de temps.

Intermédia favorise les transitions entre ces applications par une homogénéisation des traitements disponibles. Les utilisateurs sont ainsi capables d'apprendre à utiliser de nouvelles applications assez rapidement et de prédire le comportement de fonctionnalités qu'ils n'ont jamais utilisées auparavant. L'idée consiste à capitaliser l'expérience acquise pour améliorer l'utilisabilité de l'ensemble du système.

Nouveautés : système graphique « easy-to-use », carte, uniformisation des modes opératoires

1.2.2.11 Peter Brown: Guide (1986)

La première version du système Guide a été développée par Peter Brown à l'université de Kent [Brown87]. C'est le premier système hypertexte à avoir été commercialement distribué pour une gamme d'ordinateurs personnels. Guide assure la compatibilité des fichiers de données entre les ordinateurs Macintosh et les PC IBM. Guide est à la fois un outil d'édition et de consultation qui dispose d'une interface « easy-to-use ». Une des caractéristiques intéressantes du système est que différentes formes de curseurs sont utilisées pour indiquer les différents types de liens : une flèche pour les hyperliens traditionnels, une étoile pour les liens d'annotation et un cercle barré pour les liens d'inclusion. Un autre point intéressant concerne la possibilité de créer les liens par manipulation directe au moyen de la souris.



(a) Vue initiale d'un document.

(b) Vue résultante de la sélection de la troisième occurrence du mot *More* dans la vue (a).

Figure 1.12 : Système Guide.

Nouveauté : premier système hypertexte pour ordinateurs personnels.

1.2.2.12 Frank Halasz: NoteCards (1987)

NoteCards [Halasz87, Halasz88] est un environnement hypermédia qui a été conçu au Xerox PARC pour aider les auteurs, chercheurs, concepteurs, etc. à mieux travailler avec leurs idées et les informations associées (Figure 1.13). NoteCards permet de créer un réseau sémantique de cartes à notes électroniques au moyen de liens typés. Ce réseau représente un médium par lequel les utilisateurs peuvent représenter un ensemble d'idées interconnectées. Il correspond à une structure qui est

utilisée pour organiser, stocker et accéder à l'information. Le système propose aux utilisateurs un ensemble d'outils pour afficher, modifier, manipuler et explorer ce réseau. Il contient également un ensemble de méthodes et de protocoles pour créer des programmes qui permettent de manipuler les informations contenues dans le réseau.

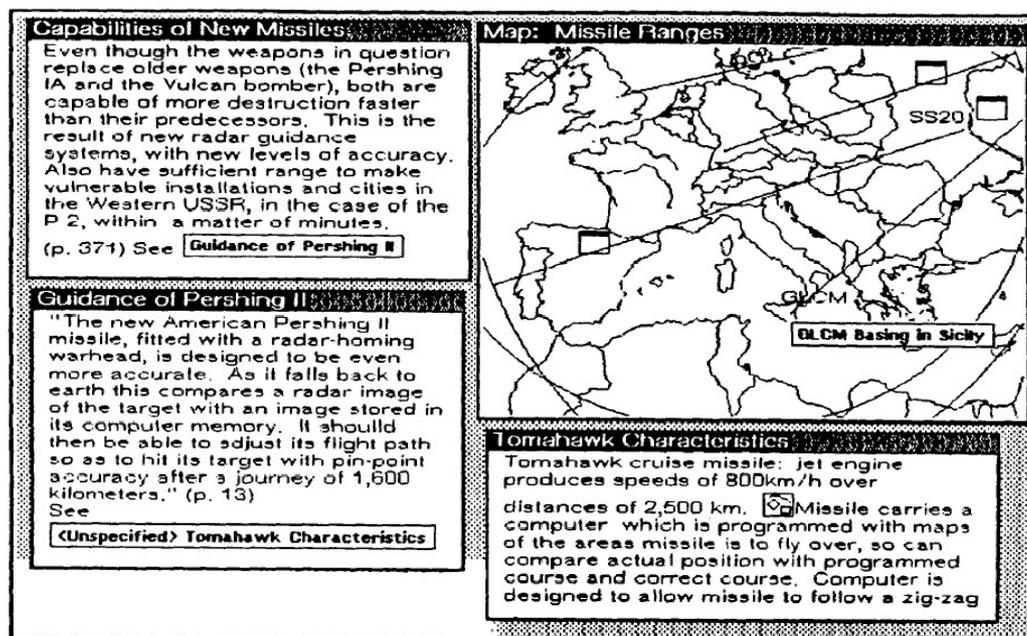


Figure 1.13 : Système NoteCards.

NoteCards gère deux types d'éléments structurels: les cartes à notes et les liens. Chaque carte possède un titre et contient un ensemble de données (une portion de texte, un dessin structuré ou une image bitmap) qui sont sensées représenter une idée. A l'écran, les cartes sont affichées dans des fenêtres distinctes, comme montré à la Figure 1.14. Les utilisateurs ont la possibilité d'éditer le contenu des cartes au moyen d'éditeurs spécifiques. Il existe plusieurs types de cartes qui sont différenciées en partie par la nature de leur contenu (texte ou graphique). NoteCards dispose également de fonctionnalités pour créer de nouveaux types de cartes (qui peuvent être définis sur la base des types existants au moyen de quelques transformations). Les liens servent à créer des réseaux sémantiques (ou structures) de cartes. Les liens sont directionnels et typés, chacun d'eux établit une connexion entre une carte source et une carte destination. Le type d'un lien est un label choisi par l'utilisateur qui spécifie la nature de la relation à représenter. Les liens sont ancrés à l'intérieur des cartes sources au moyen d'icônes qui peuvent être placées à n'importe quelle position. La destination d'un lien ne peut être qu'une carte (vue comme une entité prise dans son ensemble) et non un élément particulier de son contenu.

NoteCards dispose de deux types de cartes (*browser* et *filebox*) particulières dont le rôle consiste à aider les utilisateurs à gérer les réseaux de cartes de grandes tailles (Figure 1.14). Une carte de type « browser » affiche la structure d'un réseau de cartes. Chaque carte est représentée par une icône sensitive qui contient son titre. Lorsque l'utilisateur clique sur une telle icône, la carte correspondante s'affiche dans une nouvelle fenêtre. Les liens sont représentés par des flèches. Les attributs graphiques de ces flèches permettent d'identifier les différents types de liens. Ce type de diagramme est automatiquement créé par le système à la demande de l'utilisateur. Une fois créé, un « browser » a le même comportement qu'une carte standard. En addition de l'information visuelle fournie aux utilisateurs, un browser peut également être utilisé pour éditer la structure sous-jacente d'un réseau de cartes (i.e. créer, supprimer ou modifier les cartes et les liens du réseau). Une carte de type « filebox » permet d'organiser et de classifier un ensemble d'autres cartes de n'importe quel type (y compris des « fileboxes »). Dans NoteCards, chaque carte doit être rangée dans une « filebox ». Toutes les cartes sont donc intégrées à une structure hiérarchique de « fileboxes » dont le rôle consiste à faciliter l'organisation et la récupération des données.

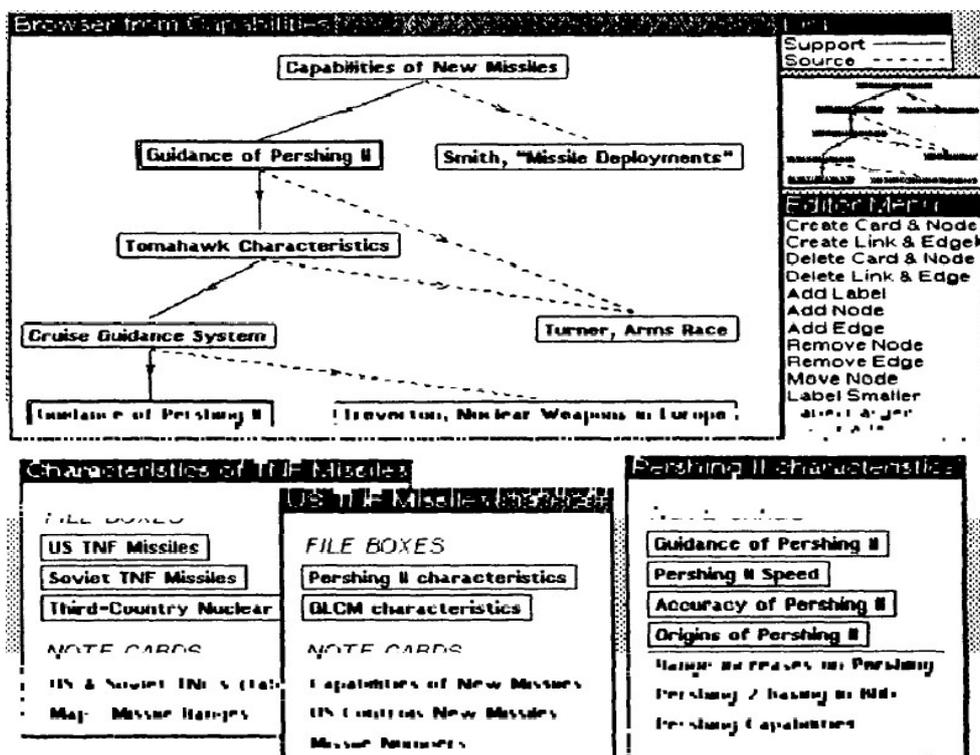


Figure 1.14 : Système NoteCards, exemple de cartes « browser » et « filebox ».

Les travaux de Frank Halasz sur NoteCards ont donné lieu à un papier historique « *Reflections on NoteCards : Seven Issues for the Next Generation of Hypermedia* »

Systems » [Halasz87] qui identifie et décrit les problèmes à prendre en compte lors du développement de systèmes hypermédias. Cette étude a fournie un premier cadre à la conception de systèmes hypermédias.

Nouveauté : cadre de conception.

1.2.2.13 Bill Atkinson: HyperCard (1987)

En 1987, Apple décide de distribuer le logiciel HyperCard, développé par Bill Atkinson [Goodman87, HyperCard89], avec chaque Macintosh. L'intégration en standard de ce logiciel à une gamme de produit grand public a complètement changé l'échelle de développement des systèmes hypertextes.

HyperCard peut être décrit comme un environnement d'édition multimédia interactif. Le modèle de conception d'HyperCard repose sur les concepts de la programmation orientée objet : les classes d'objets, les messages entre objets, les méthodes et l'héritage. Les objets communiquent par l'intermédiaire de messages qui déclenchent des méthodes et qui peuvent se propager de manière hiérarchique.

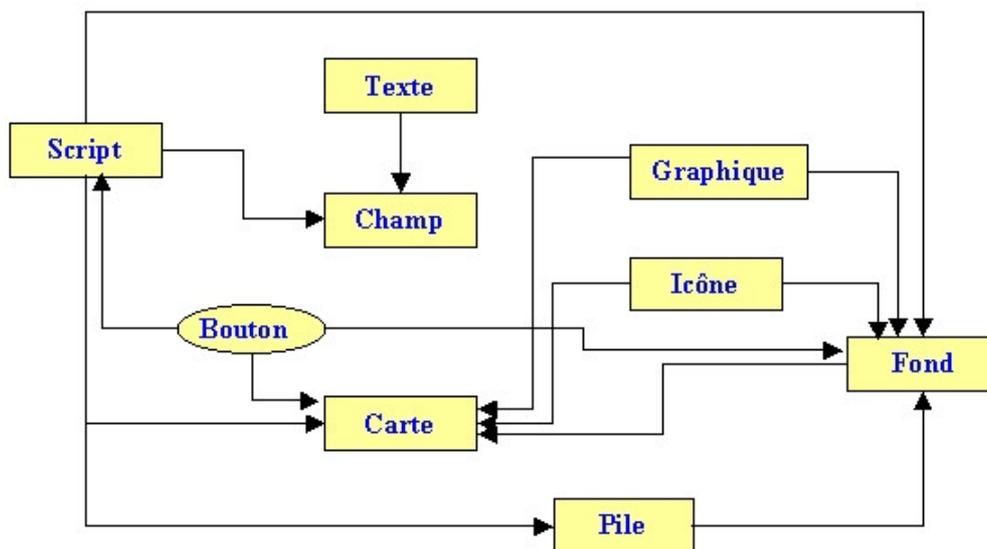


Figure 1.15 : Schéma conceptuel de l'environnement HyperCard.

Les utilisateurs disposent de cinq types d'objets : les cartes, les fonds, les piles, les boutons et les champs. Les cartes sont les unités élémentaires d'information (les nœuds). Elles peuvent contenir des données d'un ou plusieurs types (texte, graphique, image numérique, icône et script). Chaque carte doit être associée à un fond qui est le plus souvent commun à plusieurs cartes. Une pile est une liste

toujours cyclique (anneau) de cartes de même type. Les boutons et les champs appartiennent soit à une carte, soit à un fond. S'ils appartiennent à un fond, ils seront communs à toutes les cartes qui partagent ce fond. Les champs ne contiennent que du texte. Les boutons servent à déclencher des actions qui sont définies au moyen d'un langage de script (HyperTalk). La Figure 1.15 montre le schéma conceptuel d'HyperCard.

Les objets HyperCard communiquent au moyen de messages qui décrivent des événements, comme par exemple un clic souris ou l'ouverture d'une pile. Plusieurs objets peuvent recevoir un même message à tour de rôle, en fonction de leur position dans la hiérarchie d'objets et de la nature du message. Le langage de script HyperTalk permet de décrire les actions à effectuer lorsque un objet reçoit un certain type de message. Les scripts peuvent être associés à tous les types d'objets. Ils sont activés au moyen d'un bouton, d'une icône ou de n'importe quel événement (clic souris, entrée d'un caractère au clavier, etc.). L'exemple ci-dessous est un script qui indique au système de passer à la carte suivante (et de l'afficher) si l'utilisateur clique sur le bouton.

On mouseUp

Go to next card

End mouseUp

Lorsqu'un objet reçoit un message, son script est directement exécuté. Si les procédures décrites par le script traitent ce type de message, il n'est pas propagé, sinon il est transmis à l'objet suivant de la hiérarchie.

HyperCard dispose d'une interface graphique facilement utilisable par des novices permettant la manipulation de graphiques, de vidéos, de sons, etc., et de liens interactifs (boutons) vers d'autres cartes. Il dispose également de fonctionnalités de retour-arrière et d'historique de navigation.

Nouveauté : interface graphique multimédia simple, langage de script.

1.2.2.14 Le World Wide Web

En 1989, Tim Berners-Lee commence le développement au CERN (Centre Européen pour la Recherche Nucléaire) d'un système hypertexte distribué accessible via l'Internet [Berners-Lee92, Berners-Lee94], qui sera par la suite nommé le World Wide Web (W3). L'objectif de ce projet était de mettre en commun les résultats scientifiques, les publications et les documents dont les chercheurs avaient besoin. Le système a été mis à disposition en décembre 1990 et amélioré jusqu'en 1993.

Le WWW repose sur une architecture client/serveur. Chaque serveur de documents ne gère et ne connaît que l'ensemble des documents dont il est responsable. Il reçoit

des requêtes de clients pour l'accès aux documents qu'il délivre alors selon un codage normalisé [CNRS95]. Cet échange des données via le WWW se base sur deux standards. HTTP (*Hypertext Transfer Protocol*) qui prend en charge l'aspect réseau, utilisé pour le dialogue entre clients et serveurs. HTML (*Hypertext Markup Language*), une DTD (*Document Type Definition*) de SGML [ISO86], qui permet de décrire les documents dans un format textuel structuré au moyen d'un système de balisage. Les documents décrits au format HTML contiennent du texte et peuvent faire référence à des images, des vidéos, des sons codés dans la plupart des formats usuels.

Dans cette approche, les liens hypertextes sont unidirectionnels, statiques et conservés avec le document source. Ils sont codés en HTML à l'intérieur de la description du document. L'utilisateur reçoit un document et les liens qui en partent. Un interpréteur sur la station client assure la mise en page décrite en HTML et gère l'ancrage des liens dans le document. Chaque lien repère une ancre, un document destination et identifie le serveur de ce document sur le réseau sous forme d'URL (*Unified Resource Locator*). Le choix effectué par le lecteur en sélectionnant une ancre est matérialisé par l'envoi d'une requête au serveur qui gère le document cible. Dans ce modèle, chaque auteur ne gère qu'un ensemble limité de documents dans lesquels il est le seul autorisé à définir des liens vers d'autres documents. En revanche, un auteur peut référencer des documents qui ne lui appartiennent pas dans les documents dont il est responsable. La structure de liens est asymétrique. Elle permet d'interconnecter librement des cellules de connaissances. Il est ainsi possible de naviguer à partir d'un document, en profitant à tour de rôle de la connaissance des divers auteurs, jusqu'à atteindre la cible recherchée (ce qui peut engendrer des problèmes de digression).

En 1993, le Web prend un nouvel essor avec l'apparition de Mosaic [Andreessen93], le premier navigateur « easy-to-use ». Le système de navigation proposé repose sur une interface « point-and-clic » qui permet aux utilisateurs de passer d'un document à un autre en activant les liens rencontrés au moyen de clics souris. La dernière avancée en matière de technologies Web est un nouveau standard de description de documents nommé XML [W3C-XML01] disposant de fonctionnalités hypertextes étendues (section 2.2.2).

Le Web a largement favorisé l'acceptation et l'utilisation des systèmes hypermédias par un large public. Il est à la base de nombreux travaux qui ont mis en évidence le rôle primordial des domaines de l'interaction homme-machine et des modèles de documents structurés, deux domaines autour desquels s'articule une importante partie des travaux présentés dans ce mémoire de thèse.

Nouveauté : hypertexte en réseau accessible via l'Internet.

1.3 Modèles

1.3.1 Architecture générale des systèmes hypertextes

Face à l'évolution des systèmes hypertextes, un groupe de concepteurs s'est réuni au Dexter Inn dans le New Hampshire (Etats unis) en 1988 pour trouver un consensus dans la terminologie et la sémantique des concepts. Cette réunion a donné lieu au modèle Dexter (étendu par la suite pour donner le modèle Amsterdam) qui spécifie formellement les parties essentielles de l'architecture d'un système hypertexte et fournit une terminologie générale pour comparer les différents systèmes développés.

Le modèle Dexter

Le modèle de référence Dexter [Halasz90, Halasz94] identifie trois couches pour décrire un système hypertexte (Figure 1.16).

La couche de stockage (*storage layer*) représente le noyau du système. Elle modélise le réseau hypertexte et fournit les fonctionnalités nécessaires à sa gestion (i.e. les mécanismes par lesquels les composants sont assemblés au moyen de liens relationnels). Dans cette couche, les composants sont traités comme des conteneurs génériques de données, sans prendre en compte le type ou la structure de ces données. La couche de contenu (*within-component layer*), au contraire, concerne spécifiquement la gestion du contenu (des morceaux de textes, de graphiques, d'images, etc.) et de la structure des composants. Ces deux premières couches considèrent l'hypertexte essentiellement comme une structure passive de données. Les systèmes hypertextes contiennent cependant un ensemble d'outils qui permettent aux utilisateurs d'accéder, de visualiser et de manipuler le réseau hypertexte (nœuds et liens). C'est la couche d'exécution (*run-time layer*) qui est en charge de cette partie.

L'indépendance entre ces couches est un point essentiel du modèle Dexter. Pour préserver cette indépendance, la communication inter-couches s'effectue au moyen de deux interfaces. La première, l'interface stockage-contenu est un mécanisme qui permet l'adressage des ancres des liens (boutons, régions, etc.) dans le contenu informatif à partir de la couche de stockage. La deuxième, l'interface stockage-exécution est un mécanisme qui permet de spécifier dans la couche de stockage les éléments de présentation inhérents aux nœuds et aux liens (i.e. indépendants de l'outil de présentation). Ainsi, la présentation d'un composant peut être déterminée par une fonctionnalité spécifique à un outil de présentation (définie dans la couche d'exécution), mais également par une propriété du composant lui-même et/ou du lien qui permet d'accéder à ce composant.

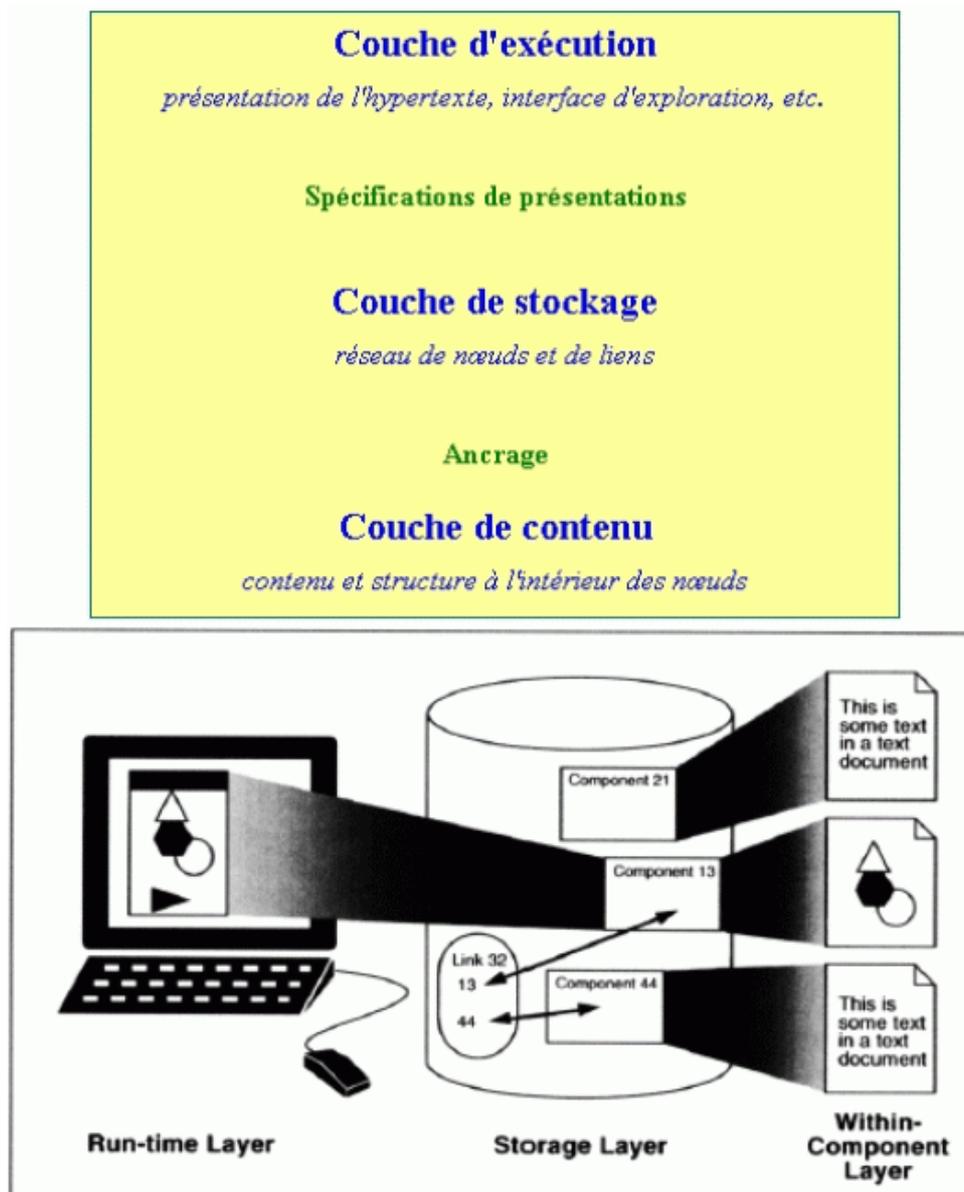


Figure 1.16 : Le modèle Dexter.

Le Modèle Amsterdam

Le AHM (*Amsterdam Hypermedia Model*) est une extension du modèle Dexter [Hardman94]. Il enrichit le modèle Dexter des notions de temps, de lien à l'intérieur d'un nœud pour simplifier l'accès à l'information, d'attributs de présentation de haut niveau et de contexte de lien. Ces caractéristiques permettent de décrire les éléments de base et les actions communes à une vaste gamme de systèmes hypermédias.

Ce modèle différencie deux types de nœuds. Les nœuds simples qui référencent l'information (les blocs de données). Les nœuds composés qui contiennent

uniquement des informations sur des ensembles de nœuds simples ou composés. Ces nœuds servent à mieux structurer la présentation de l'hyperdocument.

Architecture ouverte et fermée

Actuellement, deux types d'architectures sont communément proposées pour un système hypertexte.

Une architecture dite fermée qui implémente l'ensemble des couches décrites (stockage, contenu, présentation). Dans un système fermé, l'auteur ne peut pas accéder à la couche de stockage et ne dispose que d'un ensemble fixe d'éditeurs faisant partie du système.

Une architecture dite ouverte qui préserve l'indépendance entre la couche de stockage et la couche de contenu informatif au moyen d'une interface fonctionnelle. Cette interface permet entre autres l'intégration d'applications externes pour la gestion du contenu des nœuds. Les avantages d'une telle séparation sont nombreux [Kacmar91, Davis92, Davis94, Hill94]. Les utilisateurs peuvent gérer le contenu des documents au moyen de leurs logiciels préférés (éditeurs de texte, tableurs, etc.) ne sont pas restreints aux seuls outils proposés par le système hypertexte. L'implémentation et la maintenance ne nécessite plus d'éditeurs adaptés aux différents types d'applications, et s'en trouvent simplifiées. Enfin, ce type d'approche offre la possibilité d'intégrer des données et des applications de façon uniforme et transparente pour les utilisateurs qui peuvent ensuite échanger, modifier, retrouver et annoter des informations partagées. Des systèmes comme Intermedia [Yankelovich88], Link Service [Pearl89] ou MultiCard [Rizk92] reposent sur ce type d'architecture ouverte.

1.3.2 Discussion

Si l'on cherche à caractériser ce qu'ont en commun des systèmes hypertextes aussi différents que Xanadu [Nelson65], KMS [Akscyn88], WWW [Berners-Lee94], etc., on constate qu'il est pratiquement impossible de les comparer, tant sur le plan fonctionnel, que sur celui de leur architecture interne effective. Seule l'intention et l'esprit avec lesquels ils approchent l'interaction homme-machine leur sont communs.

Bien que les systèmes hypertextes soient employés dans un nombre toujours croissant d'activités, ces derniers restent difficilement classables. Des taxonomies ont toutefois été proposées afin de mettre en évidence les différents types d'applications rencontrés.

Par exemple, Legget, Shnase et Kacmar proposent une classification des systèmes hypertextes en cinq catégories [Leggett90].

- ? les *hypertextes littéraires* se caractérisent par des possibilités d'annotation et la prédominance des liens sur la structuration interne des nœuds. Ils sont surtout utilisés dans le domaine de l'édition et de l'éducation. Xanadu [Nelson65] et Intermédia [Yankelovich88] en sont des exemples.
- ? les *hypertextes structurels* attribuent aux nœuds d'information plus d'importance qu'aux liens et offrent des possibilités d'annotation plutôt réduites. Ils servent généralement de systèmes de gestion d'information et d'aide à l'argumentation. KMS [Akscyn88] et NoteCards [Halasz88] peuvent être classés dans cette catégorie.
- ? les *hypertextes de présentation* ont les mêmes caractéristiques que les hypertextes structurels. Ils se caractérisent par la séparation des outils d'édition et des outils de consultation. Ils sont principalement utilisés pour les manuels de référence et la documentation technique. C'est le cas du système Hyperties [Shneiderman87].
- ? les *hypertextes de travail coopératif* dans lesquels les liens et les nœuds ont la même importance. Ils sont par exemple utilisés en ingénierie du logiciel, et permettent la gestion d'informations à l'intérieur d'une organisation.
- ? les *hypertextes d'exploration* présentent une interface utilisateur centrée sur des métaphores spatiales. A l'écran, les unités d'informations sont représentées par des entités concrètes (icônes, labels, images réduites, etc.) que les utilisateurs peuvent manipuler et soumettre à divers traitements. KMS [Akscyn88] et Intermédia [Yankelovich88] proposent ce type de service.

Ce type de classification reste toutefois arbitraire. Peu de systèmes hypertextes sont limités à une catégorie particulière d'applications. Conklin propose une alternative à cette classification par type d'application en réduisant à quatre le nombre de catégories [Conklin87]. Il identifie les hypertextes macro-littéraires, d'exploration de problèmes, de consultation libre et d'usage général. Halasz [Halasz88] quant-à-lui propose une classification basée sur trois critères, qui recourent en partie les précédents : le nombre d'utilisateurs et la quantité d'information gérée, l'orientation vers la consultation ou la création, le domaine d'applications (spécifique ou général).

Tendance et besoin

Le développement du Web et l'introduction du langage HTML sont des facteurs qui ont joué un rôle considérable dans l'évolution des systèmes hypermédias. Ils ont amorcé une tendance qui consiste à représenter les hyperdocuments au moyen de formats de codage textuels, structurés et standardisés. Cette tendance favorise

grandement la mise en place d'architectures hypertextes ouvertes en libérant la représentation des données de contraintes logicielles.

Toutefois, les capacités de description et de liaison hypertexte proposées par le langage HTML s'avèrent très réduites. Pour faire face à cet état de fait, le W3C (*World Wide Web Consortium*) a développé un nouvel ensemble de spécifications regroupées autour d'un méta-langage de balisage nommé XML (*Extensible Markup Language*) [W3C-XML01] (section 2.2.2).

En observant les divers systèmes hypermédias existants, il peut également être constaté qu'ils proposent généralement peu de mécanismes intuitifs pour faciliter la manipulation et l'exploration des données présentées (l'annotation, la structuration, le rangement, la mise en évidence de documents intéressants, etc.). Un axe de recherche possible face à ce manque « d'outillage » consiste à mettre en œuvre des techniques d'interaction et de visualisation pour simuler les habitudes documentaires des utilisateurs. Cet axe de recherche repose sur la métaphore de la bibliothèque (liée aux aspects de visualisation et d'organisation) et sur la métaphore du document papier (liée aux aspects de lecture, d'analyse, etc.).

2. Augmentation interactive de documents électroniques

2.1 Introduction

Malgré les nombreux avantages offerts par les machines (facilités de stockage, d'accès et de traitements automatiques), les écrans d'ordinateurs ne sont pas de très bons supports de lecture comparativement au papier [O'Hara97]. Il a été observé que des activités comme la lecture, l'analyse ou la synthèse de documents électroniques s'effectuent la plupart du temps après impression de ces documents [Price98]. Ceci s'explique par la difficulté de lecture sur écran et surtout par le fait que l'outillage logiciel proposé ne permet généralement pas aux utilisateurs de mettre à profit leurs procédés et habitudes de travail, acquis de longue date sur documents papiers (la prise de notes, le regroupement sémantique d'informations, la réalisation d'associations d'idées, etc.). Au quotidien, les gens prennent souvent des notes pour mettre en valeur ou en relation des idées, pour donner une interprétation de concepts rencontrés, pour se remémorer des informations qu'ils jugent importantes, etc.



Figure 2.1 : Exemples de livres électroniques (e-books).
SoftBook par SoftBook Press et Rocket eBook par Nuvo Media.

Cette pratique « de recherche et d'impression » pose plusieurs problèmes. Par exemple, elle engendre, de manière paradoxale, un accroissement considérable du nombre de documents papiers [Johnson97].

Elle entraîne également une rupture entre la recherche de l'information et la lecture [Price98]. Cette rupture s'explique en partie par le temps d'impression, ce qui a pour effet de détourner l'attention du lecteur. Elle s'explique aussi par le fait que les

informations que l'utilisateur peut ajouter au document imprimé (notes, commentaires, schémas, etc.) ne seront pas mises à profit dans l'environnement informatique. De retour face à l'écran, l'utilisateur perd le système de repérage qu'il s'est construit en enrichissant le document papier de notes et de marques diverses. Ceci lui demande en conséquence un effort cognitif supplémentaire pour se refocaliser sur le sujet.

De plus, il est assez fréquent que le lecteur s'aperçoive que le document ne correspond pas à son attente lors de sa consultation. Dans ce cas, les phases de recherche et d'impression sont à recommencer, ce qui amplifie le phénomène de rupture.

L'arrivée de nouveaux supports matériels comme les livres électroniques (e-books) [Schilit99, Harrison00] fournit un élément de réponse à la lecture sur dispositif électronique (Figure 2.1). Le système XLibris, développé au Xerox Parc [Schilit98a], simule le support papier au moyen d'une tablette graphique portable à affichage haute résolution (Figure 2.2). La mobilité physique de ce type de support, la disponibilité d'un stylo électronique pour la prise de notes et la possibilité de feuilleter les documents complètent la métaphore du document papier. Ce rapprochement entre le monde réel et le monde virtuel (d'un point de vue documentaire) ne doit pas s'arrêter au niveau matériel mais doit être supporté par un outillage logiciel adapté. C'est le propos de notre sujet de recherche qui repose sur le concept de *Lecture Active*.

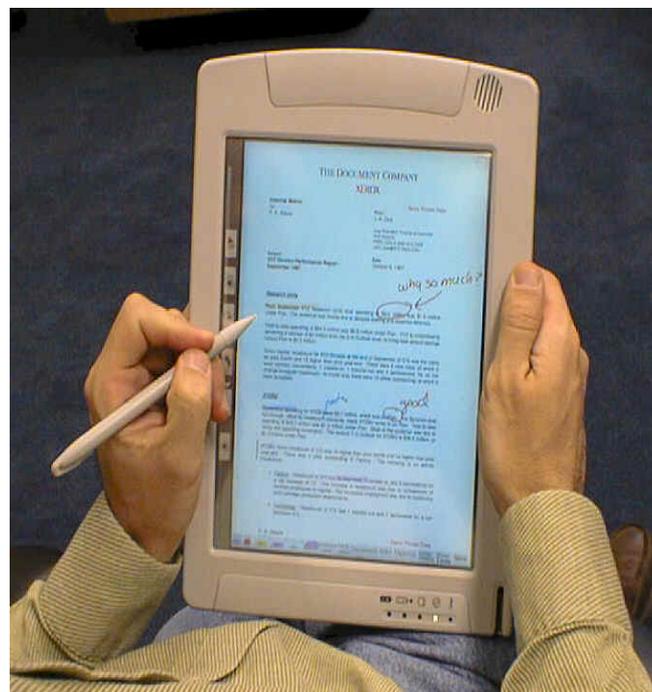


Figure 2.2 : Tablette graphique à affichage haute résolution (système XLibris).

2.1.1 Lecture Active

Le concept de Lecture Active a été introduit en 1972 par Adler pour distinguer l'ensemble des activités associées à la lecture de la simple consultation de mots sur une page [Adler72]. Ce concept a donné jour à un nouveau domaine de recherche en informatique, toutefois encore peu développé malgré l'ampleur des applications possibles.

Ce domaine de recherche tend à transformer le « lecteur » en « acteur » en lui permettant d'enrichir interactivement les documents qu'il est en train de lire au moyen de signes graphiques (ratures, sur- et sous-lignages, cerclages, flèches et liens entre zones d'intérêt, etc.) et d'annotations textuelles. Ces marques graphiques peuvent être simplement *passives* (elles agissent alors comme un « calque logique » qui se superpose au document d'origine), ou être *actives* et permettre de produire une version modifiée ou « augmentée » du document source.

La lecture active offre plusieurs avantages. Elle permet tout d'abord au lecteur de focaliser son attention, comme l'a remarqué Marshall au cours d'une étude comportementale basée sur l'observation de prises de notes par des étudiants sur des livres de cours [Marshall98]. Certains des étudiants interrogés ont déclaré prendre des notes en marge et/ou utiliser une couleur de stylo différente pour mettre en évidence un mot, un paragraphe, etc., et pour aider à assimiler les concepts rencontrés. La lecture active fournit également le moyen de rendre les documents réutilisables. Les marques et les annotations ajoutées aux documents sont des méta-données d'ordre sémantique qui facilitent l'analyse et l'interprétation lors de consultations ultérieures. Elles permettent également « de créer de l'ordre dans les données » [Marshall98]. Les flèches, renvois et autres marques de liaisons peuvent être vues comme des liens hypertextes utilisés pour structurer les documents. Cette organisation sémantique offre un moyen de regrouper des ensembles de documents en fonction de critères propres à chaque lecteur. Dès lors qu'une collection est ordonnée, il devient possible d'en tirer des informations cohérentes et complémentaires, et de les réutiliser, en faisant par exemple des recherches thématiques sur les textes des annotations s'il est possible de les reconnaître informatiquement.

2.1.2 Objectifs

Nos travaux, présentés dans ce chapitre, décrivent comment mettre à profit ces principes de lecture active au moyen de techniques hypertextes d'annotation. Les documents consultés sont enrichis de manière interactive par des marques graphiques, des notes textuelles et des liens hypertextes.

Dans le cadre de cette étude, nous nous intéressons plus particulièrement au cas de documents matériels (des papiers pour la plupart et parfois anciens) qui ont été numérisés car l'intégration informatique de ce type de documents pose un certain nombre de problèmes spécifiques. En effet, d'un point de vue informatique, ces documents sont des images « bitmaps » sur lesquelles il est difficile d'effectuer des traitements automatiques (recherche d'information, indexation, etc.) en raison de leur nature graphique (un ensemble de pixels représentant du texte, des schémas, des dessins, des idéogrammes, etc.). Des techniques d'annotation peuvent être avantageusement utilisées pour associer à ces documents des données encodées sous une forme textuelle. Cette association facilite l'intégration des documents sources, l'analyse ultérieure de leur contenu pouvant s'effectuer par l'intermédiaire des annotations.

Par la suite, nous emploierons le terme *fac-similé numérique* pour désigner la représentation informatique (codée dans un format image tel que GIF, PNG, JPEG, etc.) d'un document numérisé, ceci afin d'éviter toute confusion quant à la nature des données traitées.

Outre les techniques proposées pour marquer et annoter des documents électroniques de manière interactive, nos travaux prennent également en compte :

- ? la structuration des documents au moyen de liens hypertextes via des techniques d'annotation,
- ? la visualisation des données ajoutées,
- ? la sauvegarde de ces données (qui ne doivent pas nécessairement détruire ou modifier les documents originaux) et leur encodage dans un format normalisé facilitant la réalisation de traitements automatiques,
- ? la mise en correspondance logique entre les représentations externes (visibles à l'écran) et internes (le format d'encodage).

Ces différents points ont été développés dans le cadre de deux projets de recherche. L'un d'entre eux, le projet LADIA (Lecture Active et structuration de Documents multimédia¹) concerne explicitement l'utilisation des principes de la lecture active pour faciliter l'activité documentaire en milieu informatique. L'autre, le projet Philectre, nous a permis d'appliquer ces principes au cas d'hypermédias construits à partir de fac-similés numériques représentant des manuscrits originaux d'œuvres littéraires.

¹ Ce projet a été réalisé sous l'égide du GET (Groupe des Ecoles des Télécommunications).

2.1.3 Illustration : le projet Philectre

Le projet Philectre (PHILologie ELECTRONique²) [Cerquiglini94, Lecolinet98, Gusnard99, Lecolinet99a] avait pour objectif d'explorer les techniques informatiques nécessaires aux chercheurs en sciences littéraires pour réaliser l'édition critique d'une œuvre à partir de ses manuscrits originaux (Figure 2.3).

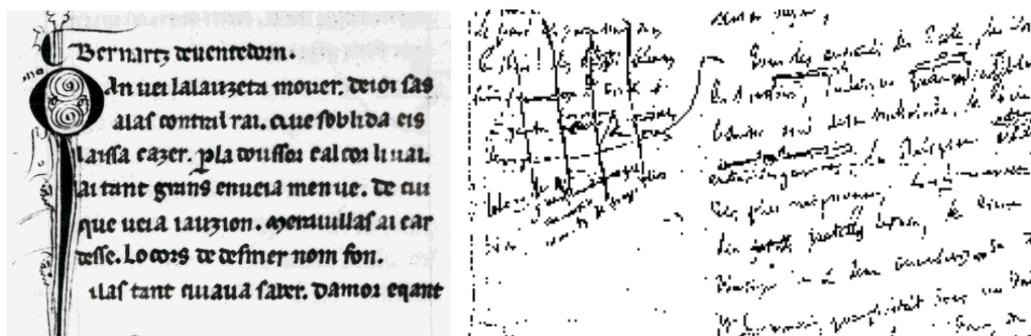


Figure 2.3 : Exemples de manuscrits d'œuvres littéraires.

Du point de vue de ces spécialistes, une œuvre littéraire ne se réduit pas à un texte « canonique » mais consiste en une collection de données hétérogènes telles que les copies manuscrites originales (comprenant généralement nombre de versions ou variantes), des transcriptions, d'éventuelles traductions, des éditions savantes, des commentaires, des notes ou des informations complémentaires de diverses natures (gravures, photos, cartes, bandes sonores, films, autres textes, etc.). C'est cette masse de documents qui permet de reconstituer la genèse des œuvres.

Parmi les activités des littéraires, il a été observé que les plus importantes consistent :

- ? à déchiffrer et transcrire les fac-similés numériques (les manuscrits originaux),
- ? à associer aux manuscrits les informations qui s'y rapportent,
- ? à comprendre et reconstituer le genèse de ces manuscrits, c'est-à-dire à les classer et les analyser,
- ? à communiquer les résultats de ce travail en les visualisant (par exemple sous la forme d'une édition).

Dans le cadre de cette étude, nos travaux ont donné lieu à la conception d'un prototype de poste de lecture active hypermédia dédié à la transcription, l'annotation et la consultation de fac-similés numériques (principalement des images de manuscrits littéraires). L'opération de transcription consiste à produire une version « texte » qui doit être la plus fidèle possible aux sources manuscrites (en respectant la

² Ce projet a été réalisé sous l'égide du GIS « Sciences de la Cognition ».

disposition spatiale des lignes, en indiquant les termes raturés et les principales marques d'édition, etc.). Comme nous le verrons dans la suite du chapitre, la transcription d'images de manuscrits peut être vue comme un cas particulier d'annotation.

Nous avons principalement focalisé notre attention sur le développement de techniques permettant de faciliter l'interprétation, l'accès et la réutilisation des fac-similés sources. L'ensemble de notre travail a été réalisé en étroite collaboration avec les spécialistes en sciences littéraires, en particulier des chercheurs de l'ITEM³ et de l'IRHT⁴.

Ces travaux entrent dans un cadre de recherche plus général lié à l'intégration de documents anciens aux bibliothèques numériques et au développement de postes hypermédias d'édition et de consultation de ce type de matériaux.

Contexte du projet Philectre

De nombreux travaux ont été réalisés ces dernières années dans le domaine des hypermédias littéraires, mais ceux-ci portent généralement soit sur la conception et l'édition d'œuvres hypermédias (par exemple le Canterbury Tales Project [Tales01] ou le projet HyperNietzche [D'Ioro98]), soit sur l'évolution des langages d'édition structurée en ce qui concerne les œuvres littéraires (par exemple les travaux relatifs à la Text Encoding Initiative [Sperberg94, Role96]). Les travaux portant sur la réalisation de postes de travail sont cependant nettement moins courants.

Un des projets pionniers en ce domaine fut celui du PLAO (Poste de Lecture Assisté par Ordinateur), développé entre 1990 et 1993 par la Bibliothèque Nationale de France [Virbel93]. Cet environnement, dédié à la recherche littéraire sur les documents, ne se limitait pas à la lecture, mais fournissait aussi des outils de travail pour traiter les annotations, la mise en relation de parties de documents, voire la reconnaissance de caractères. Malheureusement, ce projet ne fut jamais mené à son terme.

Le projet Bambi (Better Access to Manuscripts and Browsing of Images) [Bozzi97, Bozzi99] a été développé dans le cadre du programme « Télématique » de la Communauté Européenne de recherche sur les bibliothèques électroniques. Son objectif était de développer une « station de travail philologique » devant faciliter l'étude et l'édition des textes médiévaux. Le système réalisé permet de transcrire les images de manuscrits. Il dispose d'un module d'analyse dédié à la segmentation automatique de lignes et de mots dans le cas de manuscrits dont la mise en page (la disposition spatiale des éléments textuels) est nette et régulière. Une correspondance

³ Institut des Textes et Manuscrits Modernes

⁴ Institut de Recherche en Histoire des Textes

de mots entre l'image et la transcription peut être produite de manière semi-automatique en utilisant les résultats du module d'analyse. Le système permet également d'associer des annotations (commentaires, appareillage critique) aux transcriptions.

D'autres systèmes du même type ont été développés, comme par exemple MVED (Multi-Variant Contents Editor) [Hu99] qui est dédié à l'édition de documents pour lesquels existent plusieurs variantes.

2.2 Le poste de lecture active : principes de conception

2.2.1 Couplage fac-similé / annotation

L'exploitation informatique du contenu des fac-similés numériques par traitement automatique nécessite l'utilisation de modules d'analyse d'image souvent coûteux ou difficiles à mettre en œuvre. Des opérations telles que l'indexation automatique ou la recherche par le contenu ne sont pas toujours disponibles ni assez fiables dans le cas général. De plus, les modules d'analyse automatique sont fortement dépendants de la nature des fac-similés numériques à traiter (manuscrits littéraires, plans, journaux, etc.). Ils s'appliquent donc difficilement à la gestion d'une base documentaire de vaste envergure du fait de la diversité des documents qu'elle contient.

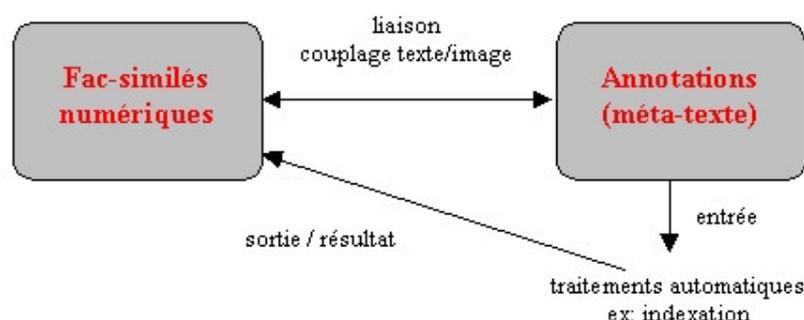


Figure 2.4 : Indirection induite par les annotations des fac-similés numériques.

La solution proposée consiste à associer des annotations aux fac-similés numériques ou à certaines de leurs parties. Les traitements automatiques (comme l'indexation) peuvent alors s'effectuer *indirectement* par le contenu via ces annotations (Figure 2.4). Par annotation, on désigne ici un ensemble de données textuelles (du *méta-texte*) regroupant diverses informations (notes, commentaires, mots clés, hyperliens, etc.) ainsi que leur codage informatique. La liaison entre les annotations et les fac-similés numériques auxquelles elles se rapportent sera quant à elle représentée par l'expression *couplage texte/image*.

Le système développé permet également de « transcrire » les fac-similés numériques. La transcription est vue comme un cas particulier d'annotation qui consiste à éditer une version du contenu textuel d'un fac-similé numérique en préservant l'association « texte/image » correspondante, ce qui peut s'avérer particulièrement utile pour l'exploitation de documents manuscrits difficilement lisibles.

2.2.2 Format de codage

Dans notre application, les annotations et transcriptions associées à un fac-similé numérique sont codées au format XML (*eXtensible Markup Language*) [W3C-XML01].

XML

XML est un descendant de SGML (*Standard Generalized Markup Language*) [Iso86, Role91, Sandoval94]. Ce sont tous les deux des méta-langages de balisage qui permettent de définir des modèles de documents structurés spécifiques à des domaines d'applications. Chaque modèle correspond à une classe de documents et tous les documents appartenant à une classe sont de même type (la classe des documents de type « livre » par exemple). A partir de la définition d'un modèle (une structure générique), XML et SGML permettent de dériver un langage de balisage qui est utilisé pour décrire un type particulier de documents. Le marquage est de type déclaratif. Les marques (ou balises) délimitent et identifient les éléments de la structure et leurs associent des attributs.

Un modèle de document est décrit au moyen d'une DTD (*Document Type Definition*). Une DTD identifie les types d'éléments que l'on peut trouver dans un modèle de document et donne un nom à chacun d'eux. Elle leur associe éventuellement des attributs qui les caractérisent. Elle décrit les relations structurelles qui lient ces éléments. Par exemple, un chapitre contient un titre obligatoire suivi d'un nombre variable de sections, etc. SGML et XML proposent une sémantique assez riche pour décrire ces relations entre éléments : séquence, choix, optionalité, répétitivité, etc. Les relations entre éléments sont hiérarchiques. Les types d'éléments rencontrés dans une DTD jouent individuellement le rôle d'une classe au sens orienté objet du terme. Les instances de ces classes sont les balises dans chaque document spécifique. Pour chaque type d'élément sont déclarés :

- ? les sous éléments qu'il peut contenir et leur organisation,
- ? les attributs qui peuvent lui être associés.

C'est le langage de balisage qui résulte d'une DTD qui est utilisé pour décrire et structurer les documents. HTML est un exemple de DTD SGML développée pour le Web. Un document est une instance de sa classe et donc de sa DTD. Un document

doit respecter la structure du type de document auquel il appartient, c'est-à-dire les règles définies dans sa DTD. Ceci est vérifié de façon automatique à l'aide d'outils que l'on nomme *parsers* (analyseurs).

XML n'impose aux documents que le respect d'une syntaxe basée sur des règles simples. Des balises de début et de fin encadrent le contenu d'un élément : `<nom_élément> contenu </nom_élément>`. XML n'autorise pas de minimisations : des omissions comme celles d'une balise dans une séquence prédéfinie, d'un attribut obligatoire, d'une balise de fin, etc., ne sont pas permises. La seule exception concerne les éléments qui ne possèdent pas de contenu. Ces derniers peuvent être représentés au moyen d'une balise unique dont la syntaxe est la suivante `<nom_élément/>`. Enfin, l'imbrication de balises XML doit correspondre à une arborescence bien formée. Lorsque cela est nécessaire, XML permet de contrôler la conformité d'un document bien formé avec une DTD, on parle alors de documents *valides*.

Les motivations qui ont donné lieu à XML sont de deux types : la communauté du Web voulait un format extensible et propre ; la communauté informatique voulait un standard pour la représentation de données. XML est né de deux observations : l'inflexibilité de HTML et la complexité de SGML. HTML implémente un seul modèle de document (c'est une DTD SGML), bien adapté au Web mais dont les applications sont limitées par une bibliothèque figée et réduite de balises. Il ne permet donc pas de prendre en compte les besoins de domaines d'applications spécifiques. SGML est un méta-langage de balisage normalisé qui permet de définir des DTDs riches en sémantique, mais dont le nombre de caractéristiques le rend relativement lourd à mettre en œuvre et inadapté au Web.

XML offre la facilité de mise en œuvre de HTML associée à des fonctionnalités hypertextes étendues. XML est extensible dans le sens où c'est un méta-langage utilisé pour créer d'autres langages. Cette capacité, issue de SGML, permet de mettre en œuvre une richesse sémantique importante au niveau des DTDs. HTML, au contraire, est un simple langage qui définit de façon rigide comment décrire un certain type de document.

Une autre différence avec HTML concerne la présentation des documents. HTML contient également des balises procédurales (ou de formatages), c'est-à-dire des balises qui n'apportent aucune information à propos de la structure mais qui contrôlent l'apparence du texte balisé (`<i>` pour italique, `` pour gras, etc.). En XML, la structure et le contenu sont séparés de la présentation. Le rendu graphique d'un document se définit au moyen de feuilles de style. Cette approche, également adoptée dans SGML, offre plusieurs avantages. Elle permet de libérer les documents d'un dispositif d'affichage particulier (navigateurs Web, imprimantes, téléphones, télévisions, ordinateurs de poches, etc.), de simplifier les modifications de styles et

de partager les styles. Elle facilite aussi l'opération de balisage, les éditeurs se focalisant soit sur la description du contenu soit sur le rendu visuel.

La DTD TEI

Pour le codage des annotations et des transcriptions, nous avons choisi d'utiliser un sous-ensemble de la DTD TEI (*Text Encoding Initiative*) [Sperberg94, Role96, Fekete00] mise au format XML. Le développement de la TEI a commencé en 1987. L'objectif était de définir un format électronique pour faciliter la représentation et l'échange de documents dans les domaines des sciences humaines et de la linguistique. Les recommandations de la TEI s'adaptent particulièrement bien à la description de sources littéraires. Elles offrent entre autres une puissance d'expression de haut niveau pour décrire des phénomènes para-textuels (insertions, corrections, textes illisibles, etc.), pour associer aux éléments du texte diverses analyses et interprétations, ou encore pour effectuer l'édition critique des documents étudiés. Le fait de disposer d'une telle puissance d'expression tout en restant compatible avec un existant conséquent constitue la principale raison qui a orienté notre choix vers la TEI.

Avantages

Le choix d'un format de codage textuel standardisé comme XML offre plusieurs avantages. Il facilite le partage et la réutilisation des données, et assure la compatibilité avec des logiciels préexistants ou à venir. Il permet également le développement de systèmes basés sur une architecture ouverte. Un ensemble de modules séparés, partageant les données XML, peuvent être développés, chacun prenant en charge une tâche spécifique. L'ajout ou l'évolution des modules ne remet ainsi pas en cause l'architecture du système. Ceci permet aux utilisateurs de choisir l'outil le plus approprié à leur besoin. Ils ne sont plus contraints d'utiliser un logiciel unique pouvant devenir obsolète ou n'étant pas assez flexible pour réaliser un ensemble varié de tâches données.

L'utilisation de balises facilite le repérage et la manipulation des éléments d'information contenus dans le document. Elle permet de simplifier et d'améliorer les traitements informatiques ultérieurs tels que la recherche d'occurrences, la création de tables d'index, le calcul de statistiques ou l'utilisation de moteurs de recherche. En effet, en plus du contenu du document, les balises permettent de stocker un ensemble varié d'informations (des méta-données) sur lesquelles il est possible d'effectuer diverses recherches et traitements. La structure hiérarchique induite par le positionnement des balises fournit également des informations pouvant aider à l'analyse du document (il est par exemple possible d'appliquer un traitement informatique en limitant son exécution aux descendants d'une balise spécifique).

Cette notion de hiérarchie des balises permet dans un logiciel d'édition XML d'indiquer à tout moment aux auteurs les balises admises dans une section du document, assurant ainsi une validation interactive du document dès l'étape de création. Un autre avantage concerne le fait que les données soient représentées sous un format textuel et associées à des informations sémantiques. Elles peuvent donc être lues et comprises par un utilisateur, contrairement aux codes « machines » de la majorité des logiciels de traitement de texte.

Enfin, L'utilisation des mécanismes de pointage offert par ce langage permet d'inclure aux documents XML des liens qui interconnectent les divers éléments en présence (fac-similés ou parties de fac-similés, annotations, transcriptions et documentation annexe). Le codage XML des annotations contient donc, en plus du contenu même des annotations et autres méta-données sémantiques, les hyperliens nécessaires au couplage texte/image.

2.2.3 Aide à la production de documents annotés

Annoter un fac-similé numérique revient à le personnaliser en y ajoutant des informations textuelles (commentaires, mots-clés, etc.), des marques graphiques (schémas, soulignages, etc.) ou des hyperliens. Cette activité dépend fortement du type du document (photographies, plans, notes manuscrites, etc.), de la sémantique de son contenu et des choix de l'utilisateur. La création des annotations ne peut alors s'effectuer que « manuellement ».

En ce qui concerne le cas particulier d'une transcription, il est toutefois possible d'avoir recours à des systèmes d'analyse automatique pour aider à la production du document correspondant.

Si le fac-similé numérique est une image de texte imprimé, l'utilisation d'un OCR permet de récupérer, dans certains cas, le texte contenu avec un degré de fiabilité satisfaisant pour en générer une version ASCII. La principale difficulté consiste ici à savoir dans « quel ordre » récupérer le texte, ou en d'autres termes à déterminer la structure du document source (colonnes, lignes, paragraphes, etc.). Ce problème, qui motive de nombreux travaux [Azokly95, Faure99], s'avère particulièrement complexe de par la grande variabilité des mises en page, et rend, en conséquence, difficile la production automatique de transcriptions. On peut toutefois imaginer des systèmes semi-automatiques laissant à l'utilisateur le soin de décrire la structure par sélection interactive de parties des fac-similés, la reconnaissance pouvant alors s'effectuer sur les blocs sélectionnés.

Par contre, la complexité des images de textes manuscrits rend difficile, voire impossible, la récupération automatique du texte et de sa structure. En effet, ces images contiennent souvent de nombreux blocs de textes qui ne sont pas disposés linéairement. Des notes, des surcharges, des schémas, des dessins, des corrections ou

des références peuvent souvent intervenir dans les marges ou être imbriqués entre les lignes (Figure 2.5). Plusieurs blocs, non successifs, peuvent parfois être reliés par des flèches ou diverses marques. Ceci constitue autant d'éléments qui compliquent les traitements d'analyse et de reconnaissance automatique.

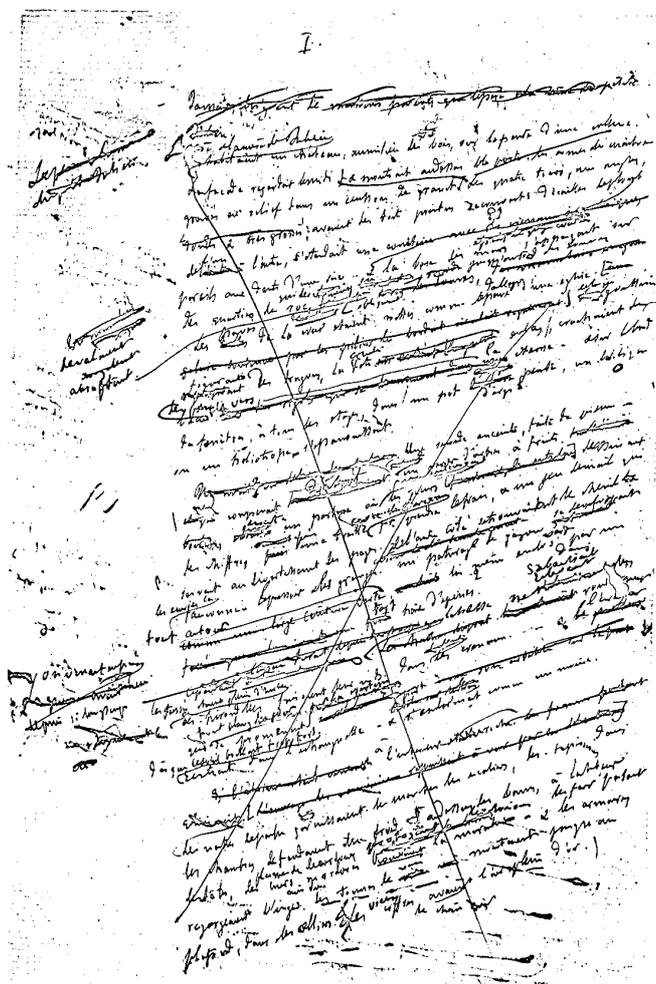


Figure 2.5 : Image de manuscrit particulièrement difficile à analyser. un brouillon d'une œuvre de Flaubert.

Actuellement, le codage de fac-similés numériques (annotation, transcription) s'effectue souvent « à la main » via des traitements de textes classiques ou des éditeurs de documents structurés, tels que Thot [Bonhomme96], FrameMaker/SGML [Adobe97] ou même Emacs [Stallman84]. De tels logiciels fournissent souvent des aides à l'édition et disposent généralement d'un module de vérification de la structure du document produit. Le principal désavantage de cette approche réside dans le fait que ces logiciels ne permettent généralement pas de mettre en relation les éléments XML produits et les parties des fac-similés numériques auxquelles ils se réfèrent. Les documents sources sont simplement utilisés comme un support visuel à

la création des annotations et des transcriptions. Il s'avère ainsi difficile de réaliser le couplage texte/image, pourtant indispensable à l'exploitation indirecte du contenu des fac-similés numériques, comme par exemple pour la mise en place de procédures d'indexation automatique.

Notre système repose sur la définition de techniques interactives permettant d'associer, de manière intuitive, du texte structuré à des parties d'un fac-similé numérique. Le codage s'effectue par interaction directe avec l'image. Une fois les annotations et les éléments de transcription créés, un document structuré, codé au format XML, est généré de manière automatique. Une attention particulière est consacrée au codage des informations nécessaires au couplage texte/image.

2.3 Couplage texte-image interactif

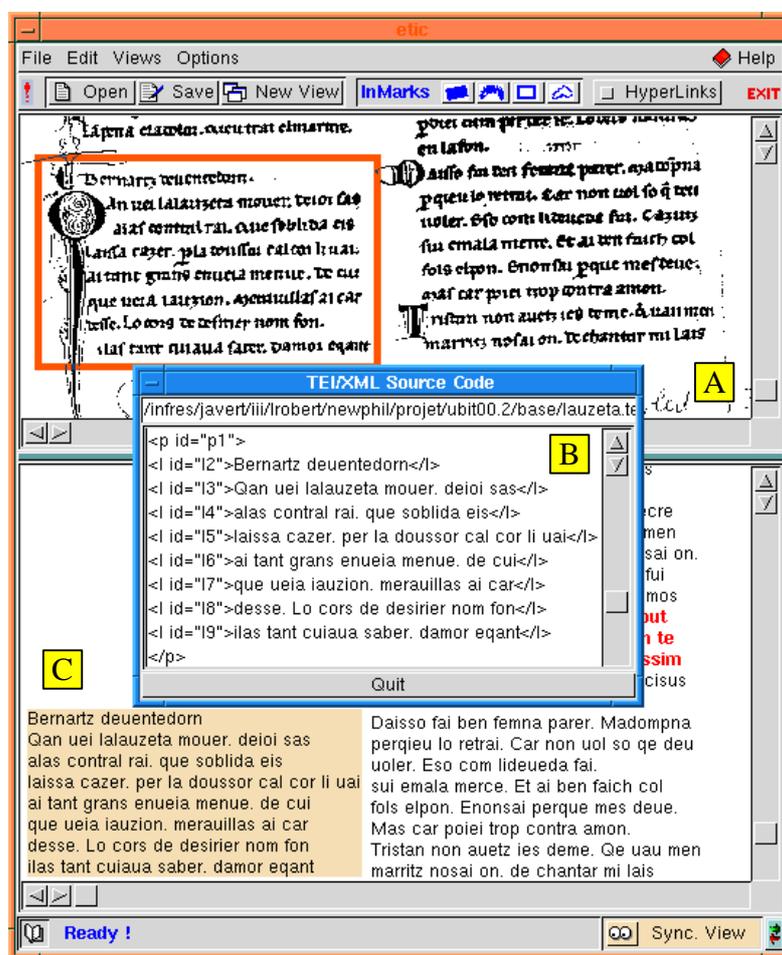


Figure 2.6 : Le système d'annotation et de transcription.

A: image d'un manuscrit. B: code XML. C: représentation graphique du code XML.

Dans notre système, illustré à la Figure 2.6, les opérations interactives de transcription et d'annotation d'un fac-similé numérique s'effectuent en deux étapes successives. La première étape consiste à sélectionner une zone de l'image. La deuxième étape consiste soit à transcrire le contenu textuel de cette zone, soit à l'annoter.

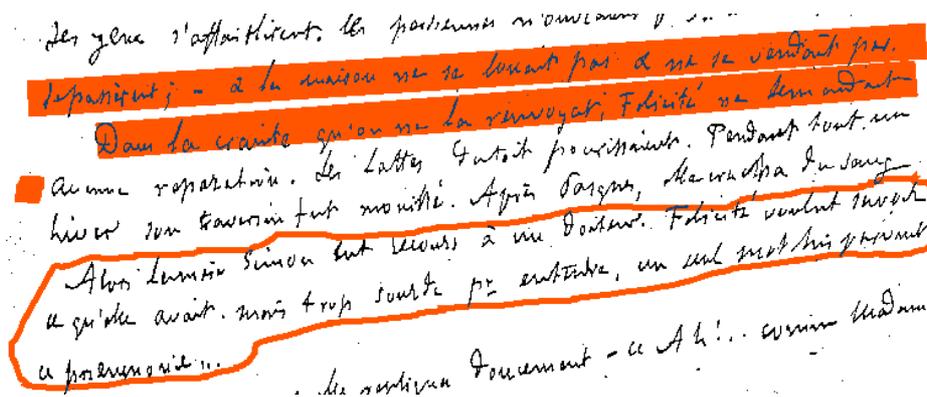


Figure 2.7 : Sélectionner des zones spécifiques dans un fac-similé numérique.

Pour sélectionner une région particulière de l'image, l'utilisateur doit déposer une trace graphique à l'aide de la souris sur ou autour de la zone souhaitée, comme illustré à la Figure 2.7. Quatre types de marqueurs interactifs ont été implémentés (ligne, stabilo, rectangle, lasso) pour définir ces traces graphiques. Les deux premiers permettent de désigner des zones linéaires, soit en définissant un rectangle englobant, soit en surlignant une zone curviligne de forme plus ou moins fluctuante. Utilisés sur une image de texte manuscrit, ils s'adaptent par exemple à la sélection de lignes, de mots ou de groupes de mots. Les deux autres marqueurs servent à désigner des blocs (un paragraphe, une note ou un dessin en marge du texte principal, la zone de titre, etc.) soit au moyen d'un rectangle, soit en dessinant un « lasso » encerclant la zone d'intérêt.

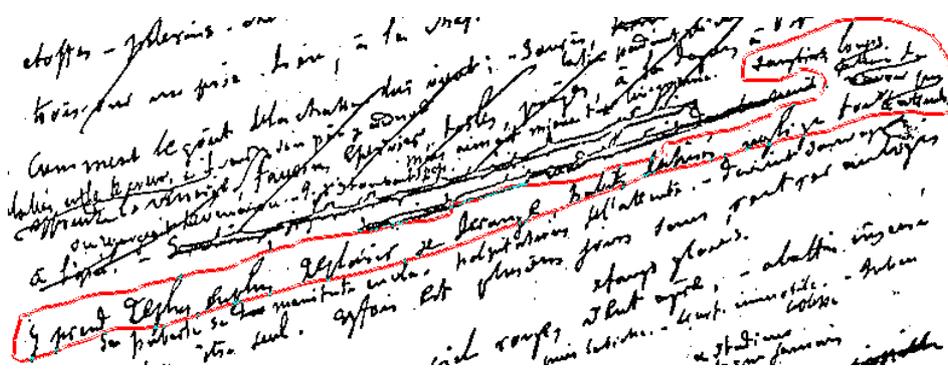


Figure 2.8 : Précision de la sélection avec un marqueur de type lasso.

De l'avis des utilisateurs, le lasso est le marqueur le plus difficile à utiliser mais s'avère toutefois indispensable de par le niveau de précision qu'il permet d'atteindre. Ce marqueur répond aux besoins des littéraires qui ont souvent à manipuler des fac-similés surchargés de marques d'insertions, de schémas, ou encore de ratures qui rendent difficile la séparation des différents éléments affichés (Figure 2.8).

Après avoir défini une zone dans l'image, il est alors possible soit de transcrire le texte manuscrit qu'elle contient, soit de l'annoter.

2.3.1 Transcription

Une fois la zone de texte sélectionnée, une interface de transcription apparaît sur l'image à côté de cette zone (Figure 2.9). L'utilisateur y entre alors la transcription du texte manuscrit et choisit parmi un ensemble de balises, affichées dans un menu déroulant, celle qui sera associée à cette partie du texte afin d'en spécifier la « nature » (ligne, paragraphe, colonne).

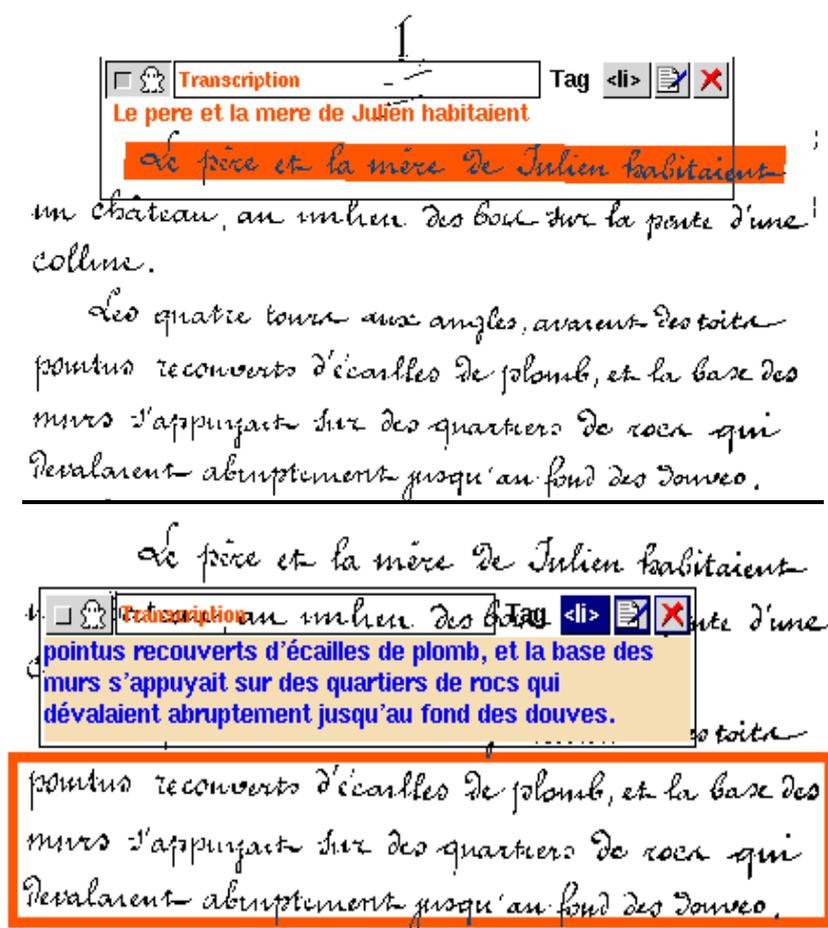


Figure 2.9 : L'interface d'annotation / transcription.
La transparence peut être désactivée au besoin.

La proximité spatiale entre le texte manuscrit et la zone d'édition permet à l'utilisateur de se concentrer sur la tâche de transcription (interprétation et édition du texte), en lui évitant d'avoir à observer deux parties différentes (voire éloignées) de l'écran. Comme le montre la Figure 2.9, cette interface est composée d'un objet graphique transparent qui augmente cet effet de proximité visuelle. L'utilisateur a l'impression d'écrire sur l'image et peut ainsi transcrire le texte en contexte. Cet effet de transparence peut être désactivé au besoin, si la superposition du texte manuscrit et du texte entré au clavier rend leur lisibilité difficile. Ce choix de conception repose sur les observations d'auteurs comme Kurtenbach [Kurtenbach97] ou Harrison [Harrison96], qui ont montré que les utilisateurs focalisent plus facilement leur attention sur des objets rassemblés dans une même zone de l'écran. Ces auteurs préconisent l'utilisation de la transparence pour favoriser le rapprochement visuel entre l'espace de la tâche et l'espace des outils, tout en réduisant les problèmes d'occlusion.

Après validation du texte transcrit, le système crée l'élément XML correspondant et l'ajoute à la représentation interne du document. La correspondance entre la zone image sélectionnée et cet élément XML s'effectue alors automatiquement, un lien bidirectionnel étant implicitement créé. Les informations nécessaires à ce couplage texte/image sont elles-mêmes codées en XML (section 2.4.3) et stockées dans le document structuré produit (la transcription).

Visualisation

A l'écran, l'association texte/image est matérialisée par un surlignage dans les deux zones en correspondance chaque fois que la souris passe sur l'une des zones concernées (Figure 2.6, partie A et C). Un second mode de visualisation permet de superposer les éléments de la transcription à l'image source. Comme illustré à la Figure 2.10, ces éléments sont affichés à proximité des fragments d'images qui leurs sont associés, ce qui semble favoriser la consultation en contexte. Ce procédé permet également à l'utilisateur d'allouer la totalité de l'espace à l'affichage du fac-similé tout en gardant un accès aux éléments de la transcription. L'utilisateur visualise une plus grande partie du fac-similé, ce qui réduit le nombre d'interactions (défilements) nécessaires à sa consultation et à la comparaison d'éléments de son contenu.

De plus, le système préserve la disposition spatiale des éléments transcrits afin d'aider l'utilisateur à effectuer la correspondance visuelle entre image et texte. Les coordonnées du polygone sélectionné dans l'image sont sauvegardées, tandis que le texte est automatiquement ajusté dans la vue graphique (Figure 2.6, partie C) de manière à être approximativement disposé comme dans l'image.

Le père et la mère de Julien habitaient
un château, au milieu des bois sur la pente d'une
colline.

Les quatre tours aux angles, avaient des toits
pointus recouverts d'écailles de plomb, et la base des
murs s'appuyait sur des quartiers de rocs qui
devaient abruptement jusqu'au fond des douves.

pointus recouverts d'écailles de plomb, et la base des
murs s'appuyait sur des quartiers de rocs qui
devaient abruptement jusqu'au fond des douves comme le tal-
les figurant

Figure 2.10 : Consultation contextuelle des éléments de la transcription.

Transcription semi-automatique

Du fait de la masse de documents qu'il faut généralement traiter, l'étape de sélection, simple à réaliser, peut toutefois s'avérer assez fastidieuse. Afin d'améliorer ce processus de sélection, nous proposons une approche semi-automatique. Celle-ci laisse à l'utilisateur la possibilité de valider ou de corriger interactivement des hypothèses de segmentation, qui lui sont fournies par un module d'analyse automatique de document [Likforman97, Lecolinet98]. La définition de tels traitements dépend cependant fortement de la nature graphique des images à traiter (voir par exemple la Figure 2.5), ce qui les rend difficiles à appliquer de manière systématique.

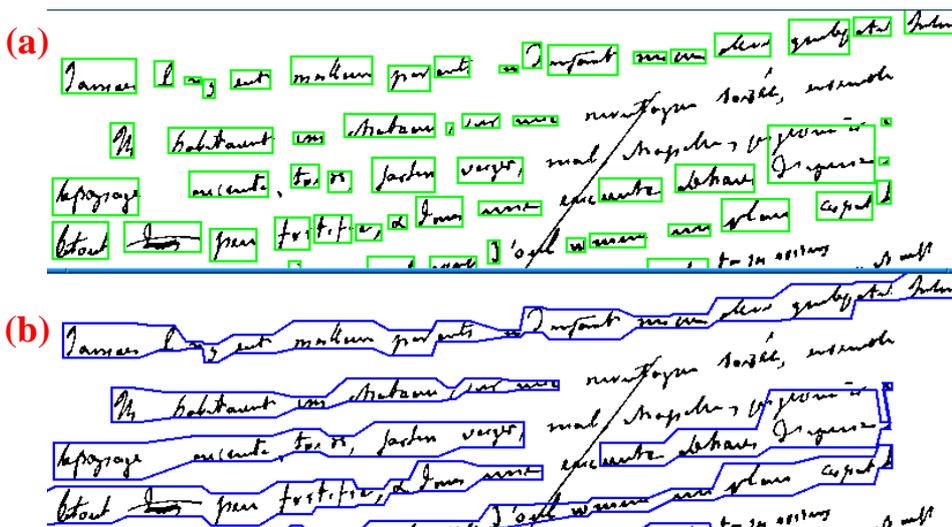


Figure 2.11 : Segmentation automatique de lignes de textes manuscrits, composantes connexes et alignements.

Le module d'analyse proposé a été spécialement adapté, suite à un développement existant [Likorman94], au cadre du projet Philectre [Lecolinet98, Gusnard99, Lecolinet01] pour la détection de lignes de textes dans des images de documents manuscrits. C'est un processus ascendant qui regroupe les composantes connexes de l'image en lignes de textes. Les composantes, qui présentent une direction fiable, sont tout d'abord sélectionnées comme points d'ancrage directionnels, c'est-à-dire comme points de départ des alignements. A partir de ces points d'ancrage, les alignements sont construits par regroupements itératifs de composantes en se basant sur des contraintes locales de perception combinées à des informations globales. Les contraintes locales consistent à vérifier que deux composantes voisines dans un alignement vérifient des critères de proximité, de similarité et de continuité de direction. Les informations globales concernent la qualité des alignements obtenus. Cette qualité est évaluée à chaque itération.

Les conflits d'alignement sont résolus en appliquant un ensemble de règles qui prennent en compte la configuration des alignements et leur qualité. A chaque itération, la contrainte de proximité est réduite afin de pouvoir regrouper les composants plus éloignés. La segmentation s'arrête quand les alignements ne sont plus modifiés entre deux itérations et fournit pour chaque ligne trouvée les coordonnées des composantes connexes qui la composent (Figure 2.11a). Chaque alignement est ensuite associé à une trace graphique afin d'être affiché à l'écran, comme illustré à la Figure 2.11b.

Ce processus d'analyse de documents peut être exécuté à tout moment par l'utilisateur et s'effectue en temps réel. Les alignements détectés sont soumis à validation, les utilisateurs pouvant supprimer ou modifier interactivement ces hypothèses de sélection.

2.3.2 Annotation

Quatre types d'annotations sont actuellement disponibles dans le système. Il est ainsi possible d'associer à une portion d'image des mots clés, des commentaires, des entrées d'index ou encore des liens hypertextes soit vers un document local à la base informationnelle soit vers un document extérieur au moyen d'un URL.

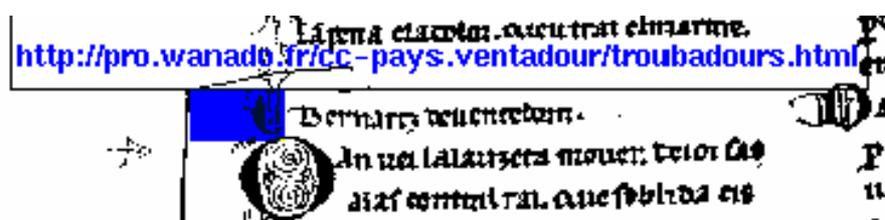


Figure 2.12 : Une annotation de type lien, avec effet de transparence.

Comme dans le cas d'une opération de transcription, une interface d'édition apparaît à proximité de la zone sélectionnée afin d'autoriser la saisie des informations à associer à cette zone (texte, adresse d'un lien, etc.). L'utilisateur sélectionne le type de l'annotation via un menu déroulant associé à l'interface d'édition. Après la validation de la saisie, le système crée automatiquement un élément XML qui décrit l'annotation (type, contenu, portion d'image associée) et l'ajoute à la représentation du document. L'utilisation de la transparence est encore une fois laissée au choix de l'utilisateur. La Figure 2.12 montre un exemple de définition d'un lien vers un site Web. A la Figure 2.13, c'est un commentaire qui est associé au texte manuscrit « Bernartz deutedorn » (le nom de l'auteur). Lors de la création d'un commentaire nécessitant un texte important, l'utilisateur a la possibilité d'effectuer la saisie dans une fenêtre d'édition séparée.

Le fait de catégoriser les annotations est un avantage pour la réalisation de traitements automatiques sur les documents XML produits. Il est alors possible, par exemple, de filtrer les balises pour affiner ou améliorer les performances de ces traitements (recherche par mots clés uniquement sur les commentaires, listing des liens hypertexte extérieurs, etc.). De même, lors d'une consultation, l'utilisateur peut afficher sélectivement les divers types d'annotations afin de ne pas surcharger inutilement l'image, lui évitant ainsi d'avoir à interpréter des informations non pertinentes par rapport à sa tâche courante.

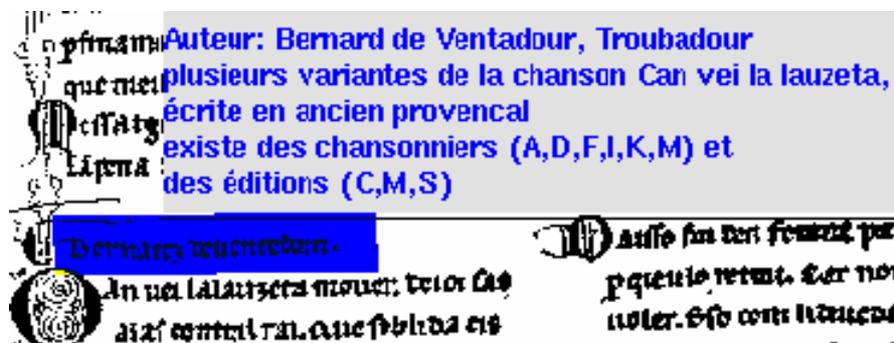


Figure 2.13 : Une annotation de type commentaire, sans effet de transparence.

Le système offre également la possibilité d'associer au document XML des informations plus générales (des annotations de haut niveau) telles que le nom de l'auteur, l'appellation de l'image, la description du sujet, etc. Ces « méta-données » pourront être exploitées avantageusement, par exemple, lors de l'organisation de la base documentaire (classification, recherche, etc.).

Enfin, il est intéressant de remarquer que la technique d'interaction proposée est applicable à tout type de fac-similés numériques (photographies, plans, cahiers d'expériences, etc.).

2.4 Codage XML/TEI

A chaque fac-similé numérique est associé un fichier texte qui contient toutes les informations codées en XML relatives à ce fac-similé. Dans un souci de clarté du fichier produit, le corps du document XML se divise en trois parties distinctes : la première décrit le codage de la transcription, la seconde le codage des annotations et la troisième le codage des associations (i.e. les liaisons transcription/fac-similé, annotation/fac-similé, annotation/transcription).

2.4.1 Codage de la transcription

Pour la transcription, les lignes, les paragraphes et les colonnes sont décrites au moyen des balises XML/TEI `<lb>`, `<p>` et `<cb>`. Chaque instantiation d'un de ces éléments correspond à un bloc de texte manuscrit qui a été sélectionné dans l'image au moyen d'un marqueur interactif (section 2.3). L'exemple ci-dessous montre une partie du code XML qui décrit la transcription du manuscrit visualisé à la Figure 2.6.

```
<text><body>
  <cb id="cb1">
    <p id="p1">
      <lb id="l1">Bernartz deutedorn</lb>
      <lb id="l2">Qan uei lalauzeta mouer. deioi sas</lb>
      <lb id="l3">alas contral rai. que soblida eis</lb>
      <lb id="l4">laisa cazer. per la doussor cal cor li uai</lb>
      <lb id="l5">ai tant grans enueia menue. de cui</lb>
      <lb id="l6">que ueia iauzion. merauillas ai car</lb>
      <lb id="l7">desse. Lo cors de desirier nom fon</lb>
      <lb id="l8">ilas tant cuiaua saber. damor eqant</lb>
    </p>
  </cb>
  <cb id="cb2">
    <p id="p2">
      ...
      <lb id="l9">petit ensai. car ieu damar nom puosc</lb>
      <lb id="l10">tener. Cellui don ia pro no aurai. tolt</lb>
      ...
    </p>...
  <p id="p3">
    <lb id="l41">Daisso fai ben femna parer. Madompna</lb>
    <lb id="l42">perqieu lo retrai. Car non uol so qe deu</lb>
    <lb id="l43">uoler. Eso com lideueda fai.</lb>
    <lb id="l44">sui emala merce. Et ai ben faich col</lb>
    <lb id="l45">fols elpon. Enonsai perque mes deue.</lb>
    <lb id="l46">Mas car poiei trop contra amon.</lb>
  </p>
  <lb id="l47">Tristan non auetz ies deme. Qe uau men</lb>
  <lb id="l48">marritz nosai on. de chantar mi lais</lb>
</cb>
</body></text>
```

La représentation graphique de cette transcription et l'image source associée correspondent respectivement aux parties C et A de la Figure 2.6.

2.4.2 Codage des annotations

Les annotations sont décrites au moyen de la balise `<note>`. Leur nature (commentaires, mot-clés, entrées d'index, liens) est déterminée par la valeur de l'attribut `type`. Toutes les annotations sont regroupées à la fin de la transcription au moyen de la balise `<noteGrp>`.

```
<noteGrp>
  <note id="n1" type="index">
    chansonnier laudetta
  </note>
  <note id="n2" type="comment">
    Auteur: Bernard de Ventadour, Troubadour
    plusieurs variantes de la chanson Can vei la lauzeta, écrite en ancien provençal
    existe des chansonnier (A,D,F,I,K,M) et des éditions (C,M,S)
  </note>
  <note id="n3" type="keyword">
    Troubadour
  </note>
  <note id="n4" type="link">
    <url>http://pro.wanadoo.fr/cc-pays.ventadour/troubadours.html</url>
  </note>
</noteGrp>
```

Les Figures Figure 2.12 et Figure 2.13 illustrent l'édition interactive de deux des annotations codées ci-dessus (le lien et le commentaire) à des zones de l'image.

2.4.3 Codage des associations

A chaque élément XML/TEI, qui décrit une partie de la transcription ou une annotation, est associé un attribut `id` qui sert à les différencier. Ces identificateurs fournissent également un moyen de référencer les éléments XML afin de les interconnecter ou de les lier aux portions d'images correspondantes. Lier une annotation à un élément de la transcription s'effectue ainsi simplement par une mise en correspondance de leurs identificateurs respectifs au moyen de la balise `<corresp>`, comme illustré ci-dessous.

```
<corresp targets="l1 n2" />
<corresp targets="l2 n1" />
```

Coder les informations nécessaires au couplage texte/image s'avère un peu plus délicat. Tout d'abord, les formats d'images courants (GIF, JPG, PNG, etc.) ne sont pas structurés. Ils ne fournissent pas de description sémantique de leur contenu (qui est stocké en bloc) via un ensemble de balises de style SGML. Il n'est donc pas

possible de référencer simplement une zone quelconque dans une image via une chaîne de caractères, ou plus généralement au moyen d'un mécanisme d'identification compatible XML. Ensuite, en raison de la nouveauté de XML, les mécanismes de pointage actuellement disponibles dans les spécifications XML ne permettent pas encore d'indexer une partie d'une image de manière standard. XPointer et XPath sont des mécanismes puissants qui permettent de référencer des parties spécifiques (des éléments, des attributs, des blocs de texte, etc.) mais uniquement à l'intérieur d'un document structuré. Afin de résoudre ce problème et ainsi pouvoir associer un élément XML (une balise) à une portion d'image (un polygone quelconque contenant un ensemble de pixel), nous proposons d'étendre la sémantique de l'élément XML/TEI `<xptr>` sur lequel repose un mécanisme de pointage existant dans la version SGML de la DTD TEI.

Cette mise en correspondance texte/image se code en deux parties. On crée tout d'abord un pointeur (une balise XML/TEI de nom `<xptr>`) vers chaque zone de l'image que l'on veut indexer. C'est la sémantique de cette balise `<xptr>` qu'il nous a fallu préciser. L'attribut `doc` permet d'indiquer le nom du fichier image traité. L'attribut `type` permet de savoir lequel de nos marqueurs interactifs a été utilisé pour définir la zone image pointée. Les coordonnées de cette zone sont stockées dans deux listes de l'attribut `from`, l'une pour les abscisses, l'autre pour les ordonnées. La Figure 2.14 illustre en détail la sémantique de la balise `<xptr>`, dont des exemples sont donnés ci-dessous.

```
<ptrGrp>
  <xptr id="xp1" type="rect" doc="lauz.gif" from="space (3d) (0 98 330) (5 640 788)" />
  <xptr id="xp2" type="line" doc="lauz.gif" from="space (3d) (0 372 608) (30 82 86)" />
  <xptr id="xp3" type="rect" doc="lauz.gif" from="space (3d) (0 370 600) (5 638 731)" />
  <xptr id="xp4" type="line" doc="lauz.gif" from="space (3d) (0 368 616) (30 752 758)" />
  <xptr id="xp5" type="line" doc="lauz.gif" from="space (3d) (0 100 328) (30 658 660)" />
  <xptr id="xp6" type="line" doc="lauz.gif" from="space (3d) (0 253 452) (30 54 55)" />
</ptrGrp>
```

La représentation graphique du pointeur identifié par la chaîne `xp1` est visualisé dans la partie A de la Figure 2.6. C'est un rectangle qui se définit par deux points, son coin supérieur gauche (98, 640) et son coin inférieur droit (330, 788), et par l'épaisseur de son contour (5).

Chaque balise `<xptr>` est ensuite associée au moyen de la balise `<corresp>` à un élément XML/TEI (qui décrit soit une annotation, soit une partie de la transcription) par l'intermédiaire de leur identifiant respectif, comme illustré ci-après. Dans cet exemple, l'association du pointeur `xp1` (la portion d'image correspondante) au « bloc transcription » `p1` est illustrée à la Figure 2.6 (partie A et C). La balise `<correspGrp>` est utilisée pour regrouper les différents types de liaisons en fin de fichier.

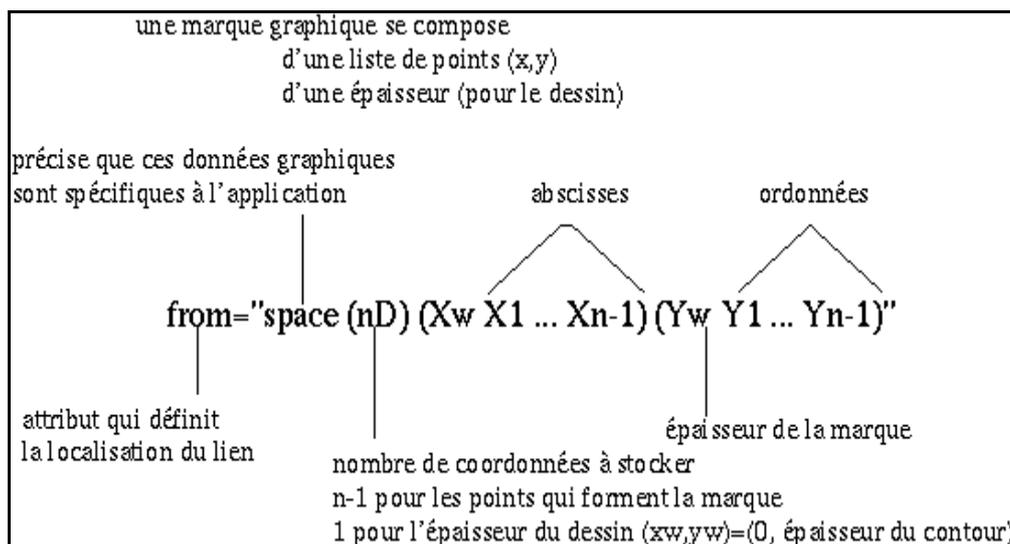


Figure 2.14 : Sémantique de la balise <xptr>.

```
<correspGrp>
  <corresp targets="p1 xp1" />           <!-- liaison transcription / image -->
  <corresp targets="l9 xp2" />
  <corresp targets="p2 xp3" />
  <corresp targets="n1 xp5" />         <!-- liaison transcription / image -->
  <corresp targets="n4 xp6" />
  <corresp targets="l1 n2" />         <!-- liaison transcription / annotation -->
  <corresp targets="l2 n1" />
</correspGrp>
```

Le procédé de codage utilisé permet également de représenter l'association de deux parties d'images via leurs pointeurs <xptr> respectifs, comme illustré ci-dessous.

```
<corresp targets="xpn xpm" />         <!-- liaison image / image -->
```

2.5 Modélisation

L'architecture de notre application repose sur le modèle Dexter [Halasz94] (section 1.3.1), qui décompose les éléments d'un système hypertexte en trois couches distinctes (contenu, stockage, exécution). La Figure 2.15 montre la schématisation du modèle sous-jacent à notre système.

Dans notre système, la couche de stockage correspond aux données XML qui décrivent les associations entre les différents composants (portions d'images, éléments de transcription et annotations), encapsulées par les balises <ptrGrp> et <correspGrp>. La couche de contenu correspond aux fac-similés numériques et aux données XML qui décrivent le contenu des transcriptions et des annotations. La couche d'exécution regroupe toutes les techniques de visualisation, d'interaction et d'analyse de l'application (comme par exemple la procédure interactive

d'annotation). L'interface entre les couches de stockage et de contenu permet de décrire l'ancrage d'un lien dans une image au moyen de la balise `<xptr>` (utilisée pour référencer une zone image quelconque). L'interface entre les couches de stockage et d'exécution ne contient aucune information. L'ensemble des effets de présentation est, dans l'état actuel, à la charge de l'application. Aucun attribut de formatage n'est associé aux divers composants de l'hypertexte (nœuds et liens) représentés par les balises `<cb>`, `<p>`, `<corresp>`, etc. Une des améliorations possibles consisterait à associer des feuilles de styles (de type CSS2 ou XSLT) aux données XML. Ces informations de formatage seraient alors prises en charge par l'interface stockage-exécution.

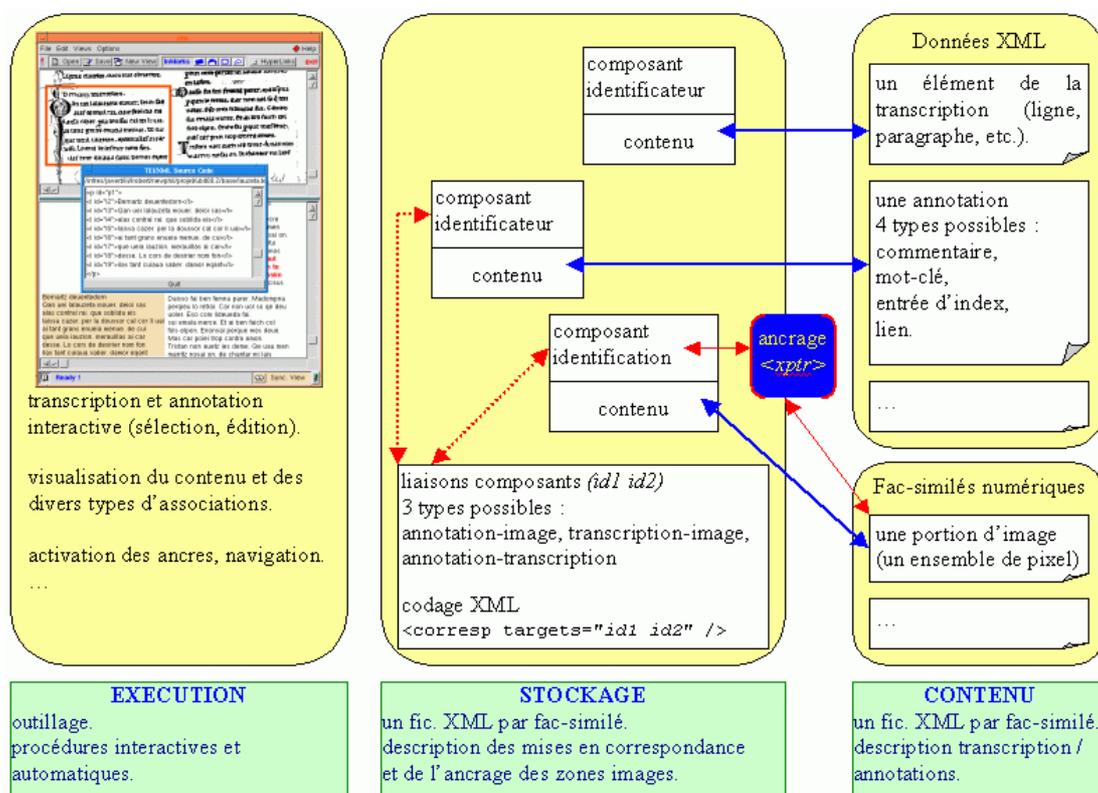


Figure 2.15 : Modélisation hypertexte du poste de lecture active.

2.6 Rendu visuel des données XML

Concernant la représentation visuelle (le rendu graphique) des données XML/TEI, il est possible d'utiliser deux méthodes distinctes (Figure 2.16). La première, liée au fait qu'il n'existe pas encore de logiciels d'affichage XML/TEI standard, consiste à transformer l'arbre XML/TEI en un arbre HTML facilement affichable par les navigateurs courants. Cette transformation s'effectue par l'intermédiaire d'un module de conversion XSLT et constitue un moyen efficace pour rester compatible avec

« l'industrie logicielle » du Web. La seconde possibilité consiste à utiliser les fonctions de rendu graphique de notre application. Ces fonctions ont été spécifiquement développées pour prendre en compte les spécificités des balises XML/TEI, ainsi que pour renforcer le couplage entre les données textuelles (annotations et transcriptions) contenues dans ces balises et les fac-similés numériques auxquels elles se rapportent. Ce fort couplage offre des possibilités intéressantes d'interaction, qui peuvent être mises à profit lors la consultation ou de l'édition des documents.

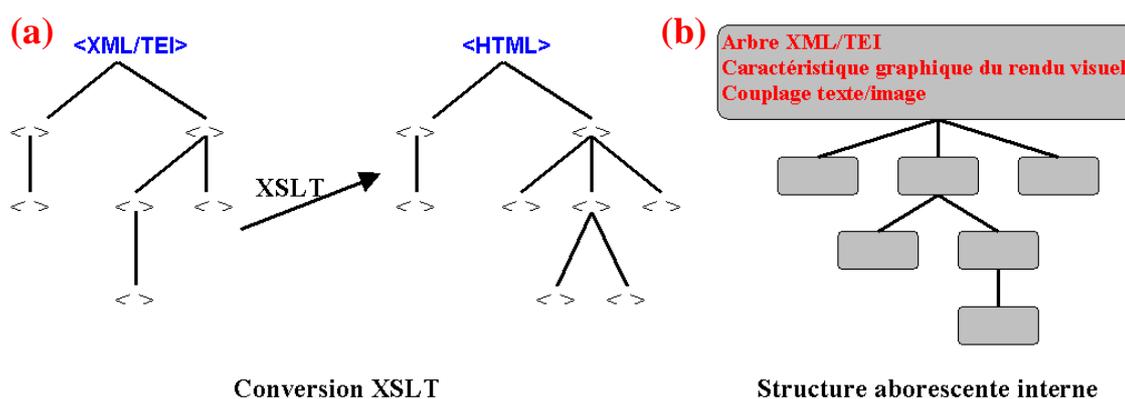


Figure 2.16 : Deux méthodes de représentation des données XML/TEI.

- a) transformer l'arbre XML en arbre HTML. b) intégrer dans une même représentation interne la description XML et les fonctions de rendu graphique associées.

2.6.1 Conversion XSLT

XSLT (Extensible Stylesheet Language Transformations) [W3C-XSLT01] est un langage de description, membre de la famille des spécifications XML, utilisé pour définir des feuilles de styles. Ce langage se décompose en deux parties. La première regroupe un ensemble d'éléments pour le formatage des documents XML (leur rendu graphique). La seconde propose un ensemble de règles qui permettent de convertir un document XML en n'importe quel autre type de document. Une des utilisations les plus courantes de XSLT consiste à transformer un document XML en un document HTML bien formé et facilement affichable. Lors de la transformation, ce sont les règles de conversion trouvées dans la feuille de style qui indiquent au module XSLT l'ordre de parcours du document XML source et les actions à effectuer lorsqu'un élément (ou une séquence) particulier est rencontré. Il est par exemple possible que la feuille de style contienne une règle qui spécifie au module XSLT de créer une balise HTML ** à chaque fois qu'une occurrence de l'élément XML/TEI *<emph>* est rencontrée.

Comme le décrit Role, un des acteurs du projet « Philectre », la conversion XSLT fournit un moyen assez simple pour afficher les fac-similés et les données XML/TEI

(annotations et transcriptions) qui leur sont associées dans les navigateurs Web courants [Lecolinet01]. Selon la méthode proposée par Role, les documents XML/TEI sont tout d'abord convertis en documents HTML. Chacun d'eux contient un lien vers le fac-similé source ainsi que le texte de la transcription et des annotations. Les informations de couplage texte/image sont stockées au moyen du système des « cartes » HTML qui permet de lier du texte à des zones d'une image par l'intermédiaire de scripts java. Lors du chargement du fichier HTML, le navigateur Web affiche le fac-similé numérique. Les données textuelles associées aux différentes parties de ce fac-similé apparaissent dans une fenêtre auxiliaire lorsque l'utilisateur clique sur une de ces parties (Figure 2.17).

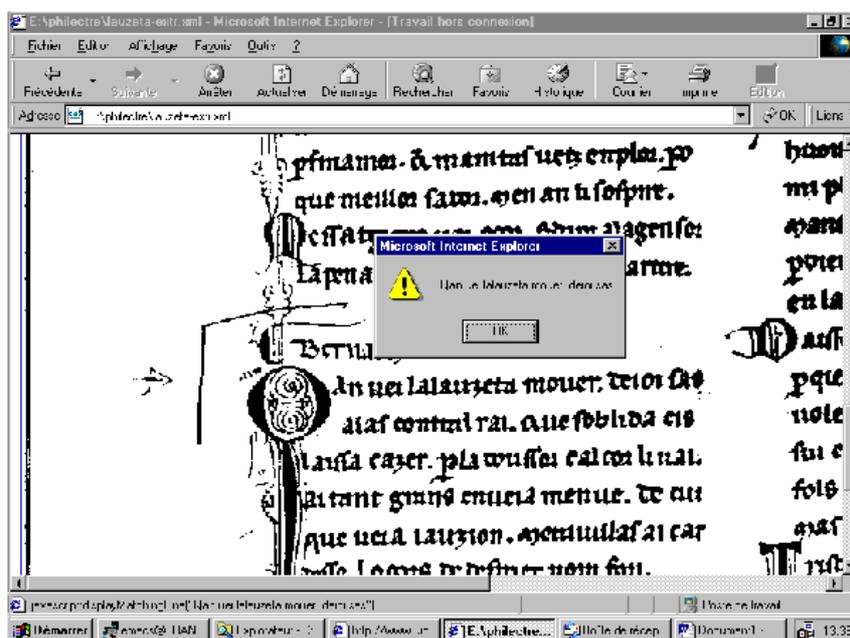


Figure 2.17 : Rendu XML/TEI dans un navigateur Web après conversion XSLT.

Les règles de conversion écrites permettent au module XSLT de créer un élément `<area>` pour chaque élément `<corresp>` rencontré dans le document XML/TEI. Un élément `<area>` associe à une région de l'image une fonction javascript de nom `displayMatchingLine(arg)` dont le rôle consiste à afficher le texte qui lui est passé en argument dans une boîte de dialogue. Rappelons ici qu'un élément `<corresp>` met en relation une région de l'image (codée au moyen de la balise `<xptr>`) et un élément de transcription (`<lb>`, `<p>`, etc.) ou d'annotation (`<note>`), comme décrit à la section 2.4. Cette mise en correspondance permet au module de conversion de récupérer de manière dynamique les coordonnées de la région et le texte associé, des informations qui sont ensuite encapsulées dans les attributs de l'élément `<area>` produit. Par exemple, pour les éléments XML/TEI suivants :

```
...
<lb id="l2">Qan uei lalauzeta mouer. deioi sas</l id="l2">
<lb id="l3">alas contral rai. que soblida eis</l>
...

<ptrGrp>
  <xptr id="img_lb2" doc="lauzeta.gif" type="rect"
    from="space (3d) (0 250 650) (0 1385 1405)"/>
  <xptr id="img_lb3" doc="lauzeta.gif" type="rect"
    from="space (3d) (0 250 650) (0 1425 1445)"/>
</ptrGrp>

...

<correspGrp>
  <corresp targets="l2 img_lb2" />
  <corresp targets="l3 img_lb3" />
</correspGrp>
```

le module XSLT produit la carte HTML décrite ci-dessous.

```

<map name="transcription">
  <area shape="rect"
    href='javascript:displayMatchingLine("Qan uei lalauzeta mouer. deioi sas")'
    coords=250,1385,650,1405'/>
  <area shape="rect"
    href='javascript:displayMatchingLine("alas contral rai. que soblida eis")'
    coords=250,1425,650,1445'/>
</map>
```

Ce type d'approche impose toutefois certaines limitations. Les possibilités de description offertes par la DTD HTML sont beaucoup moins riches que celles proposées par la DTD XML/TEI. En effet, les balises HTML ne permettent pas de prendre en compte les spécificités du domaine d'application des données qu'elles encodent. Au contraire, XML est un méta-langage qui permet d'adapter une DTD à un domaine particulier (section 2.2.2). Par exemple, la version XML de la DTD TEI utilisée dans le cadre de nos recherches facilite la description de textes littéraires. La conversion d'un document XML en document HTML s'accompagne donc généralement d'une perte d'information. En effet, les balises XML spécifiques n'ont souvent pas d'équivalents en HTML. Elles sont alors transformées en balises HTML plus générales. A titre d'illustration, prenons le cas d'une DTD XML qui permet de différencier trois types de paragraphes au moyen des balises *<p1>*, *<p2>* et *<p3>*. Après avoir été converti en HTML, ces différents paragraphes sont tous encodés au moyen de la balise *<p>*, la DTD HTML ne disposant que d'un seul élément paragraphe.

2.6.2 Unification des représentations XML et graphique

Dans notre poste de lecture active, décrit à la section 2.3, les aspects texte et image sont fortement couplés car ils sont intégrés dans une structure arborescente interne unique qui facilite l'implémentation d'interactions complexes entre ces deux types de ressources. Pour ce faire, l'architecture de notre système repose sur une nouvelle boîte à outils graphique, appelée *Ubit* (pour Ubiquitous Brick Interaction Toolkit) [Lecolinet99b, Lecolinet99c].

Le Toolkit Ubit

Le toolkit Ubit propose deux APIs C++ équivalentes et interchangeables. La première est une API objet classique qui ressemble un peu à celle du toolkit graphique Java AWT. La seconde API, également en C++, permet des descriptions déclaratives qui rappellent celles des langages de balisage tels que HTML et autres DTDs définies avec SGML ou XML.

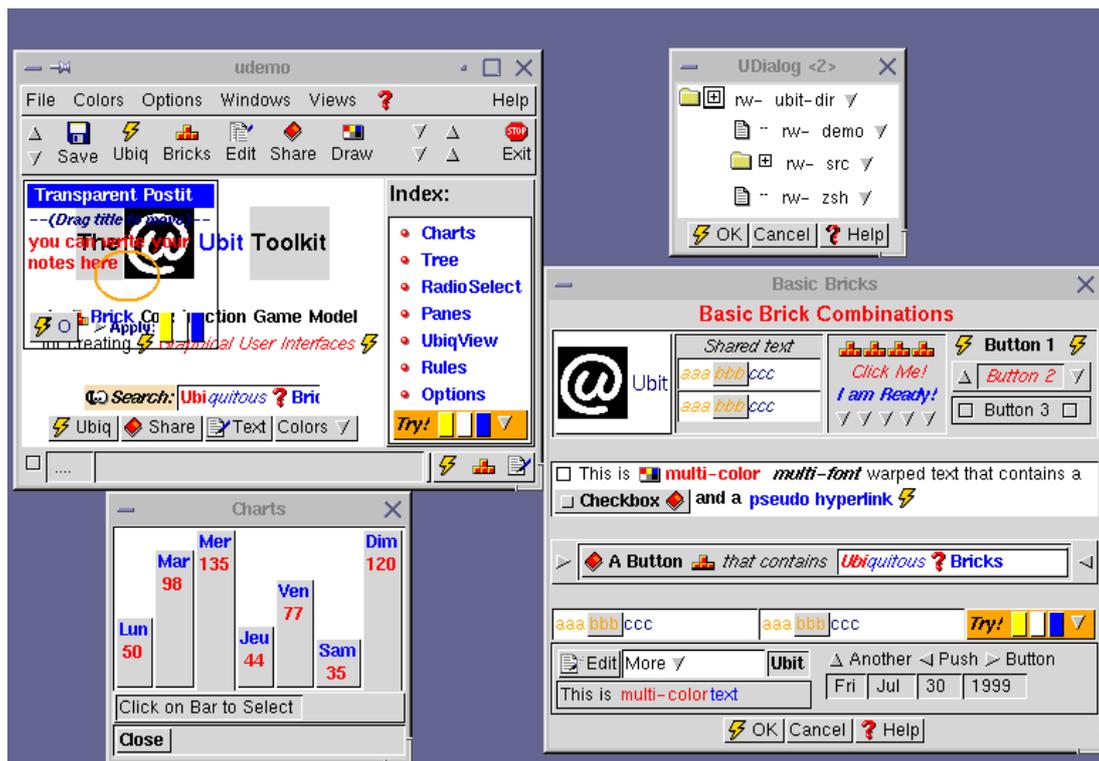


Figure 2.18 : Combinaison de briques graphiques avec le toolkit Ubit.

L'architecture du toolkit Ubit est fondée sur la notion de briques logicielles génériques et modulaires, appelées *briques de base*, qu'il est possible de combiner librement afin d'obtenir une grande variété de composants graphiques spécialisés

(Figure 2.18). Une brique de base est un petit objet qui implémente et qui contrôle une fonctionnalité bien précise. Les briques Ubit ne se limitent pas aux composants graphiques visibles (les traditionnels *widgets* ou *controls*), mais permettent aussi de représenter n'importe quelle propriété matérielle (une décoration, du texte, une image, un symbole, etc.) ou immatérielle (une couleur, un type de police, un comportement, une condition, etc.).

Ce schéma de conception offre plusieurs avantages. L'utilisation des briques de base permet de créer des objets graphiques qui correspondent aux besoins du domaine d'application. En effet, la généralité et la simplicité de ces briques les rendent facile à dériver en une grande variété d'objets pouvant accomplir des tâches très spécifiques. Une brique peut, par exemple, implémenter un objet particulier tel qu'un élément ou un attribut XML. La simplicité de ces briques de base permet également de disposer simultanément d'un grand nombre d'objets graphiques en utilisant une quantité de mémoire raisonnable, ce qui rend possible le développement d'applications pleinement orientées objets sans pour autant sacrifier les performances. Par opposition, la plupart des toolkits graphiques fournissent des composants qui implémentent de nombreuses caractéristiques standard (même lorsqu'ils sont utilisés pour des tâches spécifiques) et, en conséquence ont tendance à utiliser de grandes quantités de mémoire. Ce schéma de conception impose de sérieuses limitations sur le nombre d'objets graphiques qui peut raisonnablement être utilisé. En conséquence, les logiciels efficaces tentent généralement de mixer différents styles de programmation et d'introduire des astuces de codage qui accroissent leur complexité tout en limitant leurs capacités.

Outre ces divers aspects, le choix du toolkit Ubit s'explique surtout par le fait que le modèle structurel induit par la combinaison des briques de bases se rapproche fortement de celui des langages de balisages. Cette propriété offre la possibilité de représenter l'arbre qui décrit un document structuré au moyen d'un arbre de composants graphiques. Le toolkit Ubit semble donc fournir un cadre bien adapté au développement de logiciels de visualisation et d'édition de documents structurés. De plus, il dispose en standard de fonctionnalités comme le zoom et la transparence, et de composants comme les lentilles magiques, qui facilitent l'implémentation de techniques d'interactions avancées.

Mise en correspondance des éléments XML et des briques Ubit

Au niveau de notre application, nous tirons parti des spécificités du toolkit Ubit de la façon suivante. Au lieu de manipuler des représentations internes séparées de l'arbre XML et de l'arbre d'instanciation des objets graphiques (Figure 2.19), on les rassemble dans un graphe unique.

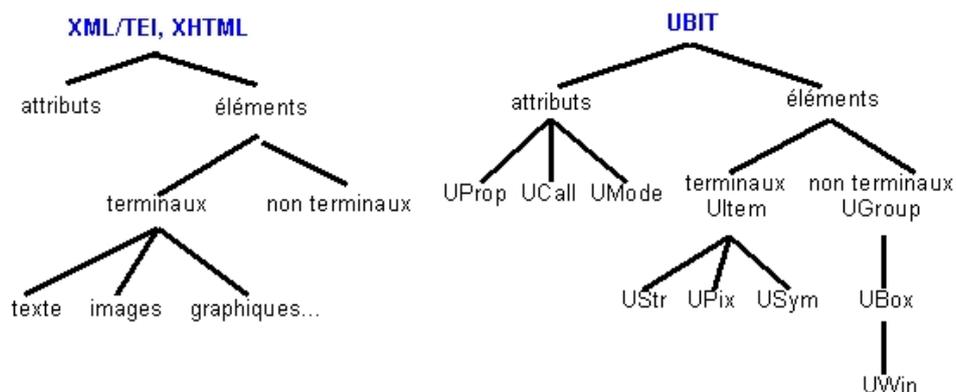


Figure 2.19 : Correspondance entre l'arbre XML et l'arbre Ubit.

Chaque nœud de l'arbre XML décrivant le document est représenté par une brique Ubit qui implémente les caractéristiques XML appropriées ainsi que leurs fonctions de rendu graphique. Cette représentation facilite la construction d'éditeurs interactifs puisque chaque objet (les balises) apparaissant effectivement à l'écran correspond à un objet réel et unique de la représentation interne. Ainsi, la représentation textuelle affichée à la Figure 2.20 (partie B) peut être vue comme une projection de l'arbre XML/Ubit qui est automatiquement gérée (et mise à jour à l'écran) par le toolkit. Ce schéma de conception possède également l'avantage de fournir directement la *correspondance inverse* de la représentation visuelle vers les données internes. La correspondance entre la position du curseur sur le texte et les nœuds XML est connue en permanence de telle sorte qu'un déplacement de la souris sur l'écran est conceptuellement équivalent à un déplacement de pointeur dans l'arbre XML/Ubit. Cette caractéristique offre des possibilités étendues d'édition et diverses autres fonctionnalités intéressantes, puisque l'utilisateur peut directement interagir avec une représentation visuelle qui est virtuellement équivalente aux données internes.

Enfin, une autre caractéristique de notre implémentation consiste à ajouter aux briques XML/Ubit les données spatiales qui permettent de lier le contenu des images avec leurs parties textuelles (annotations et transcriptions), c'est-à-dire les informations de couplage texte/image. Les nœuds XML/Ubit intègrent ainsi trois types de données dans une représentation unique :

- ? les caractéristiques des éléments XML, leurs attributs et leur contenu textuel,
- ? les caractéristiques graphiques du rendu approprié au texte structuré dans la transcription et la vue XML,
- ? les coordonnées spatiales de ce texte dans les fac-similés numériques.

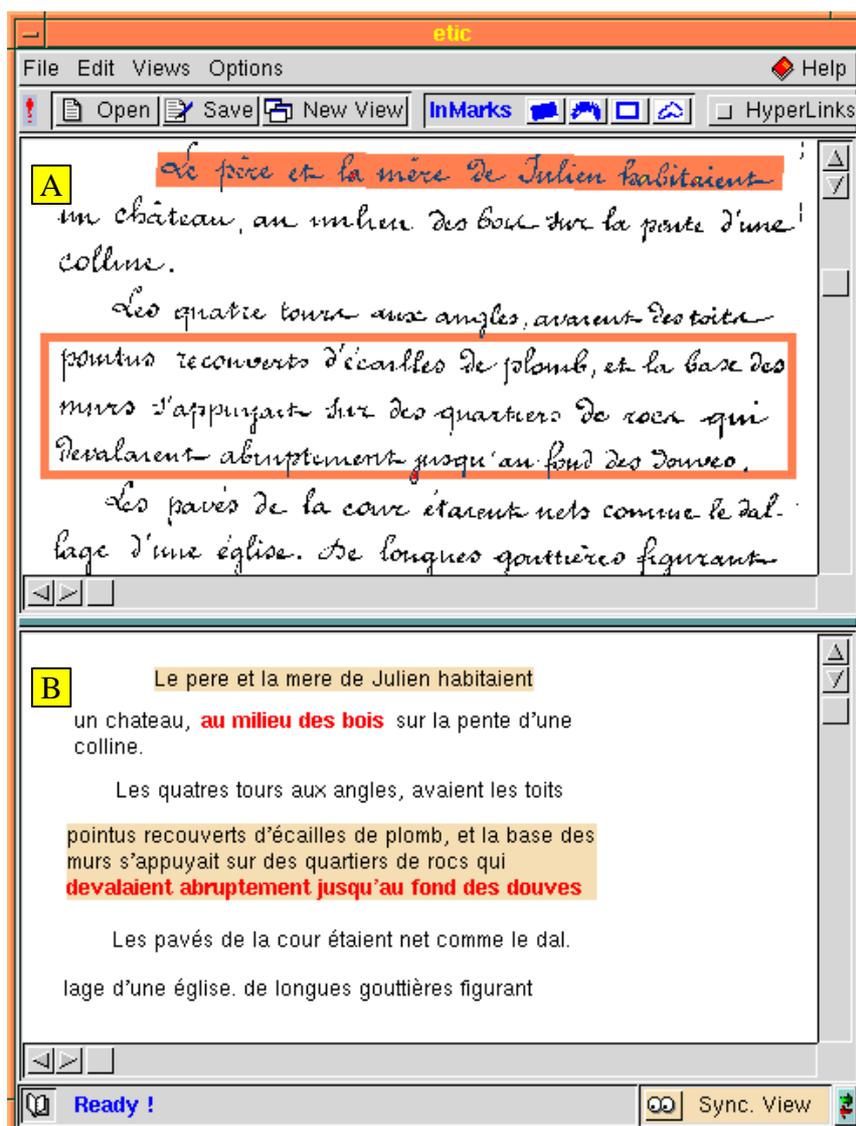


Figure 2.20 : Projection graphique de l'arbre XML/Ubit.

Fenêtre B : rendu visuel du code XML/TEI associé au fac-similé numérique affiché dans la fenêtre A.

Cette correspondance entre les nœuds XML et leurs représentations graphiques (à la fois dans les images et dans les vues textuelles) pourrait surprendre à première vue puisque le codage XML tend à séparer la structure et la sémantique du texte du rendu graphique. Tout d'abord, il s'avère toutefois important de rappeler, qu'au niveau de notre application, cette correspondance existe uniquement dans la représentation interne des données, afin d'améliorer la gestion de ces données et de faciliter l'interaction utilisateur, mais n'a aucun impacte sur les représentations externes (fichiers XML/TEI). Ensuite, il est intéressant de noter que les caractéristiques graphiques standards pour le rendu du texte ne sont pas stockées dans les nœuds XML/Ubit, mais dans des objets de *style* séparés. Ces styles, une autre caractéristique

standard du toolkit, fournissent un moyen simple et efficace de paramétrer les briques Ubit. Ce sont des collections de spécifications graphiques qui s'appliquent par défaut en l'absence de spécifications explicites. L'utilisation de styles permet de partager l'essentiel des ressources graphiques (fontes, couleurs...) entre les objets. Cette caractéristique facilite la translation entre les « mondes » XML/Ubit (document structuré / interface graphique) qui repose alors sur un double niveau de correspondance :

- ? la correspondance entre les balises (et attributs) XML et les briques Ubit pour le codage,
- ? la correspondance entre les feuilles de style XML et les objets de style Ubit pour le rendu.

Comme alternative à la méthode proposée, une autre solution consiste à avoir deux arbres en correspondance bi-univoque. Le premier décrit la structure du document XML et peut être construit au moyen d'une API standard (par exemple l'API XERCES du projet Apache [Apache01]). Le second est un arbre Ubit qui reproduit la structure XML au moyen des briques graphiques. Cette approche permet aux développeurs de mettre à profit toutes les opérations implémentées en standard dans l'API XML utilisée. Elle semble également faciliter la gestion de documents de grandes tailles.

2.7 Conclusion

Dans ce chapitre, nous avons montré comment mieux intégrer le document dans un milieu informatique en aidant les utilisateurs à réaliser leur activité documentaire. Des techniques d'annotation sont utilisées pour associer aux documents sources des marques graphiques, des informations textuelles ainsi que d'autres données (images, textes, etc.) au moyen de liens hypertextes. Ces méta-données sont « ajoutées » interactivement par les utilisateurs. Ils créent ainsi leur propre système de repérage qui les aide à *comprendre* et à *réutiliser* les documents. L'utilisation de liens hypertextes fournit un moyen pour *structurer l'organisation* des données manipulées. Les annotations (le texte et les marques) sont codées avec le méta-langage de balisage XML. La mise en correspondance de ces annotations XML avec les documents sources fournit une solution à l'accès, par indexation, à des documents difficilement exploitables par traitement automatique. Ce procédé d'indirection est particulièrement utile dans le cas de fac-similés numériques. Il permet de *trouver* et de *retrouver* les documents au moyen d'une recherche par le contenu via les annotations XML.

III. Exploration interactive

3. Désorientation

Le phénomène de désorientation est un des problèmes majeurs engendrés par l'exploration interactive d'espaces informationnels [Conklin87, Utting89, Nielsen90a, Nielsen90b, Berstein91b, Rivlin94]. Les utilisateurs ont souvent la sensation « d'être perdus » dans la masse de données disponibles. Ils éprouvent des difficultés à trouver, à collecter, à comparer ou à retrouver les informations qui les intéressent. Ces difficultés résultent principalement du manque de structuration des données présentées et du faible outillage proposé par les systèmes d'exploration.

3.1 Manque de structuration

L'exploration de documents hypermédias constitue le terrain privilégié du syndrome de désorientation [Brown87].

« The biggest problem in hypertext, which most of us admit in footnotes toward the end of papers extolling the virtues of our systems, is getting lost » [Brown87].

La faible structuration des réseaux hypertextes est une des principales causes de ce problème. L'agencement des liens forme souvent un graphe quelconque, un type de structure difficile à comprendre et qui ne facilite ni l'orientation ni la localisation (ce qui s'observe par exemple assez facilement dans le cas du Web). Nielsen montre que même dans un réseau de nœuds relativement restreint, les utilisateurs ont du mal à se repérer [Nielsen90b].

« 56% [of the users] agreed fully or partly with the statement: when reading the report I was often confused about where I was [...] 44% agreed with the statement: when reading the report I was often confused about how to get back where I came from ».

Foss [Foss89b] a identifié certains des symptômes de la désorientation au moyen d'une expérience. Celle-ci met en jeu deux hyperdocuments organisés différemment mais constitués des mêmes nœuds d'information. Les nœuds du premier sont structurés linéairement (sous la forme d'une liste chaînée) et ceux du second sous la forme d'un graphe qui empêche tout parcours linéaire de l'hyperdocument. Les participants disposent de 45 minutes pour parcourir l'un ou l'autre des hyperdocuments, puis sont soumis à un test de compréhension suivi d'un questionnaire rapide. Un dispositif enregistre les clics sur les boutons, détermine les nœuds qui ont été visités et le temps mis par l'utilisateur pour consulter ces nœuds. Les résultats montrent que les lecteurs de « l'hyperdocument linéaire » ont fait une « meilleure lecture » du texte que les autres. En effet, dans le premier cas les lecteurs

visitent en moyenne 80% des nœuds contre 73% pour le second cas; 64% des nœuds sont réellement lus contre seulement 40% dans le cas non-linéaire, et finalement les bouclages sont moins nombreux : seulement 18% des nœuds sont visités plus de trois fois dans le premier cas contre 36% dans le second. Pour expliquer ces différences, Foss fait état de trois difficultés typiquement liées au problème de désorientation et rencontrées par les lecteurs de l'hyperdocument non-linéaire:

- ? la difficulté de trouver l'ordre optimal de parcours des nœuds et même de choisir le nœud de départ.
- ? la difficulté d'évaluer par avance l'importance relative d'un nœud. Ceci peut se mesurer par le nombre de fois où un utilisateur revient au nœud précédent après un très court instant de lecture.
- ? la difficulté de procéder à une lecture systématique du contenu des nœuds. Ceci peut se mesurer par le calcul de la proportion des nœuds visités en un temps donné dans chacun des cas.

3.2 Manque de mécanismes appropriés

Le problème de désorientation est également dû à l'utilisation de systèmes d'exploration inadéquats ou faiblement outillés. Dans ce sens, il peut être abordé comme un problème d'interaction.

La plupart des interfaces d'exploration ne présentent que des vues parcellaires des nœuds d'information (comme c'est par exemple le cas sur le Web où les navigateurs courants n'affichent qu'une seule page à la fois et hors contexte). L'utilisateur éprouve des difficultés à comprendre la place du nœud consulté dans l'hyperdocument ainsi que ses relations aux autres nœuds. Il s'avère également compliqué de comparer le contenu de plusieurs nœuds et de réaliser des associations d'idées de cette manière (de par la difficulté d'afficher simultanément plusieurs composants ainsi que les chemins qui les relient). Ce manque d'information contextuelle peut être résolu par l'ajout de mécanismes appropriés au système d'exploration. Par exemple, les observations de Foss [Foss89a] sur l'utilisation du système NoteCards (section 1.2.2.12) [Halasz88] montrent que les lecteurs qui disposent de vues simultanées sur plusieurs nœuds ont plus de facilité à organiser leur navigation dans le réseau. Certains utilisateurs se servent même du positionnement des nœuds à l'écran comme procédé mnémorique pour garder la trace des nœuds visités et des chemins à poursuivre.

Selon notre point de vue, le phénomène de désorientation peut donc être atténué au moyen de mécanismes facilitant la recherche d'information, la localisation et la compréhension des données et de leur organisation. Comme nous le verrons dans le chapitre 4, ces mécanismes peuvent être de diverses natures, allant d'une aide au

suivi du parcours de navigation à la simplification de la structure initiale de l'hyperdocument. Au chapitre 5, nous ferons le point sur les techniques interactives de visualisation qui peuvent permettre de faciliter l'exploration (en particulier par l'utilisation de vues globales, locales ou encore contextuelles).

4. Techniques d'aide à l'exploration



A ce jour, différentes techniques ont été proposées pour aider les utilisateurs à trouver, comprendre, organiser et retrouver l'information au sein d'un espace de données. Nous avons classé ces techniques en cinq catégories.

4.1 Présentation du contenu

Ces techniques visent à faciliter la découverte et la compréhension des données présentes dans l'espace hypermédia.

4.1.1 Outils dérivés du livre traditionnel

Le livre traditionnel comporte de nombreux indices, qui une fois associés, forment un système de repérage complet. La table des matières donne l'ensemble des notions abordées dans l'ouvrage et l'articulation des différents thèmes les uns par rapport aux autres. L'index présente lui aussi les notions abordées dans l'ouvrage, mais cette fois par ordre alphabétique. Le glossaire donne la définition des thèmes et, éventuellement, les relations sémantiques qu'il peut y avoir entre ces thèmes. Le résumé et les mots-clés donnent au lecteur un aperçu synthétique du contenu des documents. Enfin, la bibliographie fournit des indices sur les chemins à suivre pour en savoir plus sur tel ou tel thème. Ces repères classiques, appliqués aux hypermédias, peuvent constituer des aides efficaces à la navigation. Ils permettent de donner de façon concise à l'utilisateur des informations qui l'aiguillent sur ce qu'il va pouvoir trouver dans les documents contenus dans l'hyperbase.

La création d'un tel système de repérage nécessite un processus d'indexation qui peut être réalisé manuellement, automatiquement ou de façon semi-automatique, et qui dépend fortement de la nature des données (texte, image, schéma, etc.).

L'indexation manuelle nécessite que chaque document soit analysé par un spécialiste du domaine correspondant avant d'être intégré à la base. En fonction de ses connaissances, le spécialiste détermine les unités syntaxiques qui lui semblent les plus significatives pour représenter le contenu du document. Ces unités sont regroupées dans une liste, plus ou moins structurée, qui est utilisée lors de toute opération d'indexation et d'interrogation. Les systèmes d'indexation manuelle sont généralement longs à mettre en œuvre en raison de la complexité de la tâche. Pour cette raison, ils sont relativement coûteux.

L'indexation automatique consiste à déterminer, par un processus informatisé, les unités syntaxiques représentatives des concepts et des thèmes exprimés dans les

documents. Le processus doit être à même de distinguer et de « comprendre » les notions qui y sont définies. Dans le cas de documents textuels, les diverses méthodes existantes vont, d'une simple extraction de mots, à des analyses linguistiques et/ou statistiques permettant de mettre en évidence les unités significatives et de les pondérer en fonction de leur importance. Dans le cas de documents hétérogènes contenant des images, des vidéos, etc., il s'avère difficile (voire impossible) de définir automatiquement les notions représentées par ces éléments. Des techniques d'analyse et de reconnaissance peuvent toutefois être utilisées pour essayer d'extraire les unités textuelles d'une image, d'un schéma, ou pour essayer de déterminer les mots-clés contenus dans une bande sonore.

L'indexation semi-automatique est une combinaison des deux méthodes précédentes. Le spécialiste du domaine concerné garde un certain contrôle sur le processus et reste maître du choix final. Contrairement à l'indexation manuelle, qui impose le choix des termes d'index document par document, l'indexation semi-automatique peut être réalisée de façon transversale sur tout un corpus, par modification, ajout ou suppression globale de termes. Ceci permet un meilleur contrôle de la cohérence du travail. Selon notre opinion, cette méthode est un bon compromis pour obtenir un système de repérage significatif et de qualité. Elle nécessite toutefois la mise en place de mécanismes d'interaction pour faciliter la validation des hypothèses fournies par la machine.

4.1.2 Points de repères

Dans un hypermédia, les points de repères correspondent à des nœuds qui doivent être accessibles rapidement et ce de n'importe quel point de l'hyperbase. La grande majorité des systèmes disposent d'un point de repère particulier, à savoir le nœud d'introduction. C'est notamment le cas de la carte « Home » d'Hypercard [Goddman87, HyperCard89]. De même, les navigateurs Web usuels proposent tous une interface contenant un bouton « Home » associé à une page Web particulière (à définir dans les préférences utilisateur) qui correspond au point de départ de chaque session de navigation.

L'utilisation de plusieurs points de repères semble permettre aux utilisateurs de mieux structurer leur navigation. Cette pratique est couramment employée. De nombreux hypermédiats sur cd-rom (encyclopédies, musées virtuels, etc.) et sites Web affichent sur la page d'accueil (i.e. l'entrée du système) des listes entières de points de repères. C'est par exemple le cas du site Web de Yahoo, comme le montre la Figure 4.1.

Nielsen [Nielsen90a] préconise l'élaboration de mécanismes automatiques pour la détection de points de repères potentiels, tout en laissant à l'auteur le soin de choisir parmi les résultats ceux qui seront effectivement utilisés.

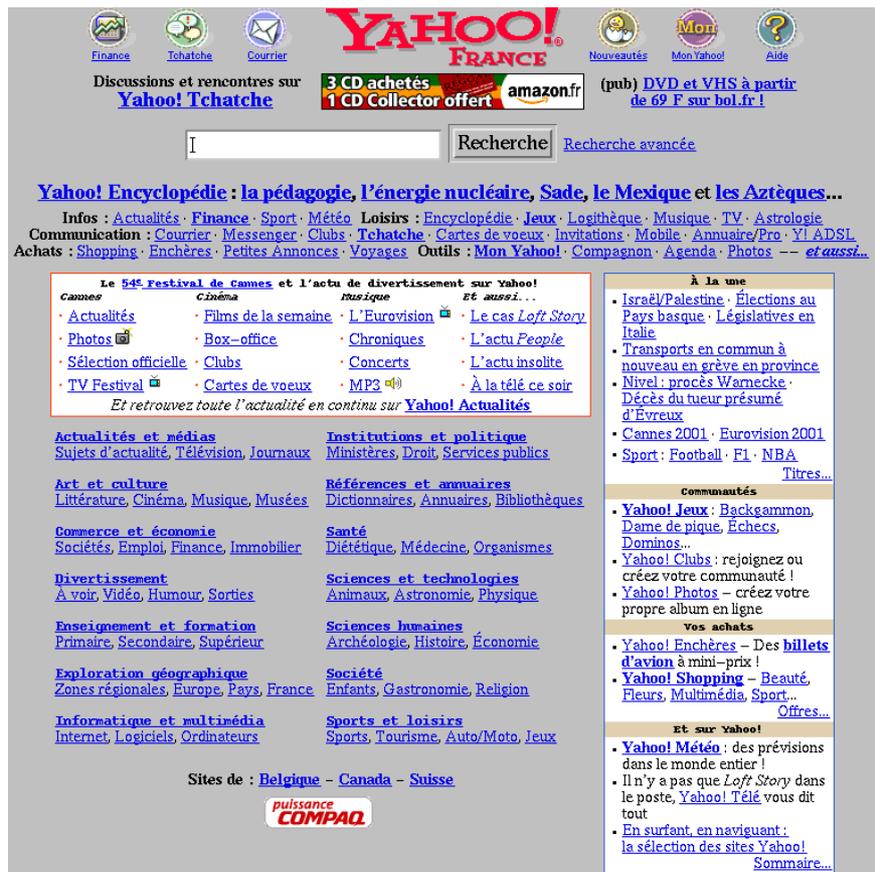


Figure 4.1 : Points de repères sur la page d'accueil d'un site Web.

4.1.3 Visite guidée

Cette technique consiste à définir des séquences ordonnées de liens (des chemins) à l'intérieur d'un réseau hypertexte. Le lecteur avance ou revient en arrière sans jamais s'écarter du chemin. Les visites guidées sont souvent utilisées comme un outil de découverte. Il est possible d'associer plusieurs visites guidées à un même hypertexte. Chacune d'elle peut être en charge de l'exploration d'une partie spécifique ou être destinée à un certain type d'utilisateurs (étudiants, professeurs, etc.). Une visite guidée peut être une liste quelconque de nœuds créée en fonction d'un certain contexte. La séquence de liens est alors virtuelle, le chemin qu'elle décrit ne se retrouvant pas dans le réseau hypertexte.

Selon Zellweger, un système de visite guidée efficace doit satisfaire trois critères majeurs [Zellweger89].

1. Il doit fournir aux auteurs d'importantes possibilités d'expression,
2. Il doit aider les auteurs à créer et à modifier les chemins,

3. Il doit aider les lecteurs à trouver les chemins et à les suivre de manière flexible.

Différentes implémentations de cette technique ont été proposées. Dans le système Textnet [Trigg88], les chemins sont de simples listes ordonnées de références à des nœuds. Ils sont stockés séparément du réseau hypertexte, ce qui permet aux utilisateurs de pouvoir facilement les modifier, en ajoutant ou en enlevant des références. Le contenu des nœuds qui font partie d'un chemin est affiché comme le résultat d'une concaténation du texte contenu dans chacun de ces nœuds. Ce procédé sert surtout à générer divers documents linéaires à partir d'un même réseau hypertexte. Le logiciel HyperCard dispose d'un objet spécial qui permet de stocker et de « rejouer » (faire défiler automatiquement) une séquence ordonnée de nœuds [Goodman87], ces derniers étant affichés à tour de rôle. Le système NoteCards [Halasz87] propose des mécanismes pour éditer des représentations graphiques de visites guidées (Figure 4.2).

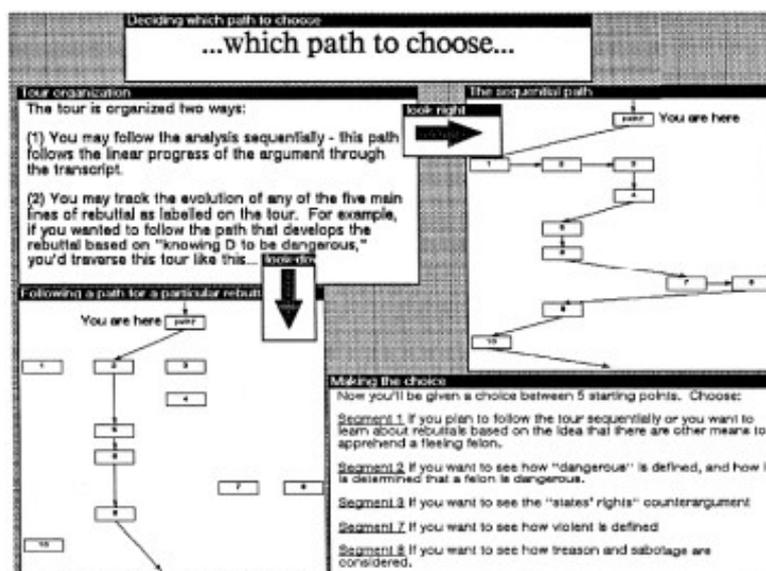


Figure 4.2 : Visites guidées dans NoteCards.

Les visites guidées sont soit visualisées au moyen d'interfaces spécifiques [Marshall89, Trigg88] soit au moyen de l'interface qui permet de consulter normalement les documents de l'hypertexte [Johnson93]. Les ancres des liens qui font partie de la visite sont alors mises en évidence par un artefact visuel (couleur, cadre, etc.), les autres liens étant temporairement inactifs.

Les visites guidées sont le plus souvent définies manuellement, soit par les concepteurs du système hypertexte, soit par les utilisateurs, ces derniers souhaitant pouvoir parcourir l'hyperdocument selon différents ordres. Certains mécanismes permettent toutefois de créer des visites guidées de manière automatique [Brown88]. Guinan [Guinan92] propose une technique pour présenter le résultat d'une recherche

par le contenu (initialisée via une requête) sous la forme d'une visite guidée. Les nœuds ne sont pas seulement triés par rapport à leur importance, mais également par rapport aux liens qui les associent. L'interrogation par le contenu pose cependant plusieurs problèmes liés à la localisation des documents retrouvés par rapport à la position de l'utilisateur et à son contexte de travail.

L'inconvénient des visites guidées est qu'elles orientent fortement l'exploration en fonction d'un contexte ou d'un point de vue.

4.2 Suivi du parcours de navigation

Ces techniques aident l'utilisateur à comprendre les chemins suivis et à retrouver les documents consultés lors de la session de navigation courante ou de sessions précédentes.

4.2.1 Retour arrière

Le retour-arrière permet de revenir au nœud précédent. L'utilisateur se sert de cette fonctionnalité pour retrouver les documents qu'il vient de consulter et pour corriger ses erreurs de parcours. Une utilisation intensive du retour arrière correspond souvent à une sorte d'insatisfaction. Le retour-arrière ne fournit pas en soi une solution au problème de la désorientation. Il est toutefois fort utile « faute de mieux ».

4.2.2 Historique de navigation

Le concept d'historique n'est pas propre aux systèmes hypermédias. De nombreux systèmes informatiques enregistrent les commandes des utilisateurs afin de pouvoir les réutiliser plus simplement et plus rapidement par la suite.

Créer un historique de navigation consiste à garder une trace de l'ensemble des nœuds déjà visités [Foss89a, Utting89]. De nombreux systèmes hypermédias présentent l'historique sous la forme d'une liste. C'est par exemple le cas du système HyperCard [Goodman87] et des navigateurs Web usuels (Microsoft Internet Explorer, Netscape Navigator). Comme illustré à la Figure 4.3, les listes de pages Web affichées par ces navigateurs peuvent être classées selon différents critères comme par exemple la date de dernière visite, la fréquence de visite, l'ordre de visite ou encore le site d'appartenance. Ce type de représentations ne permet toutefois pas de décrire les chemins suivis par l'utilisateur. Les relations de parcours entre les pages sont perdues. De plus, il peut être difficile d'identifier les pages au moyen des labels textuels (le plus souvent leur titre ou leur adresse) qui les représentent, ces derniers n'étant souvent pas assez informant.

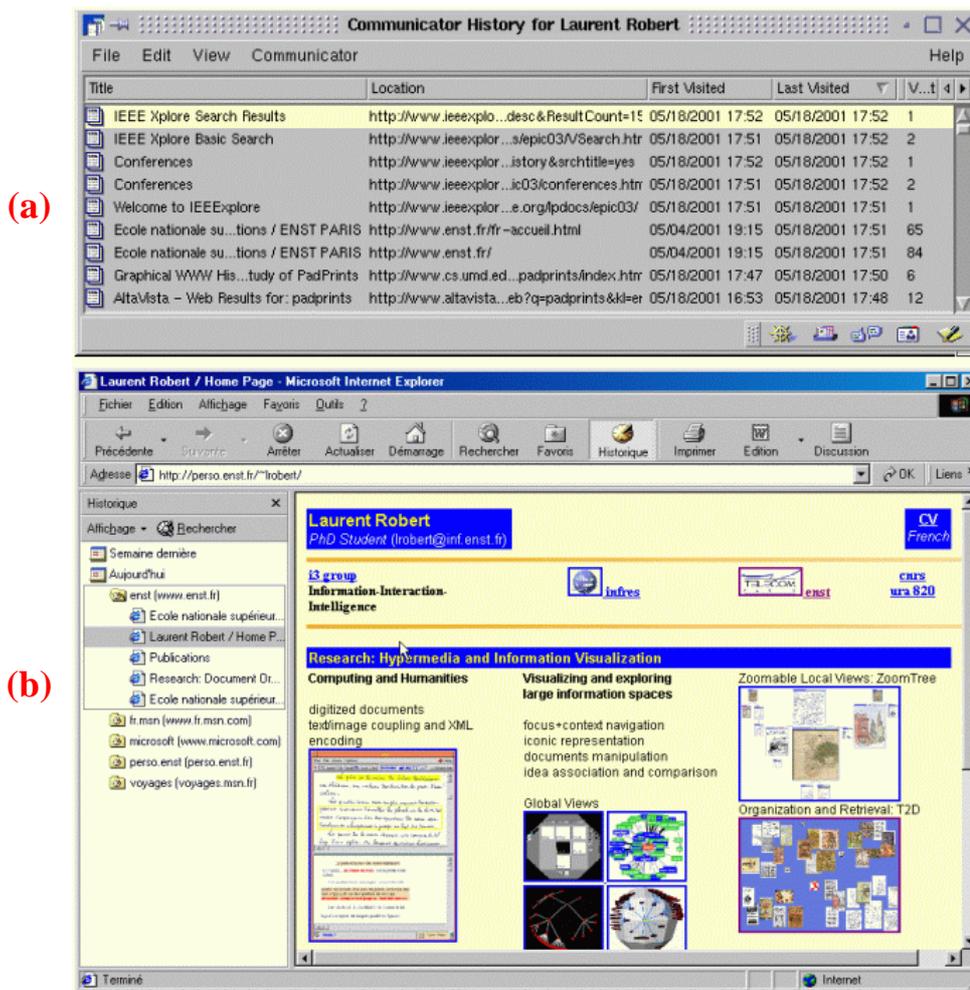


Figure 4.3 : Exemples d'historiques de navigation présentés sous forme de listes.

- (a) les pages sont classées par date de dernière visite (Netscape Navigator). (b) les pages sont classées par site d'appartenance (Internet Explorer).

Comme alternative, certains systèmes implémentent l'historique de navigation sous la forme d'une représentation graphique. C'est par exemple le cas des systèmes MosaicG [Ayers96] et PadPrints [Hightower98] qui affichent les pages Web consultées pendant une session de navigation sous la forme d'une arborescence. La hiérarchie se construit dynamiquement. Les pages sont représentées par leur titre et par une image de leur contenu. Cette image peut être affichée à différentes échelles, comme illustré à la Figure 4.4. Ce type de représentation permet aux utilisateurs de garder une trace des chemins empruntés, ce qui semble améliorer la localisation. L'utilisation d'images réduites facilite l'identification des pages, les utilisateurs faisant appel à leur mémoire visuelle. Le système PadPrints a été soumis à une étude d'utilisabilité afin de tester sa validité. Cette étude a montré que l'utilisation de PadPrints réduit le nombre de pages affichées et le temps d'accès aux pages préalablement consultées.

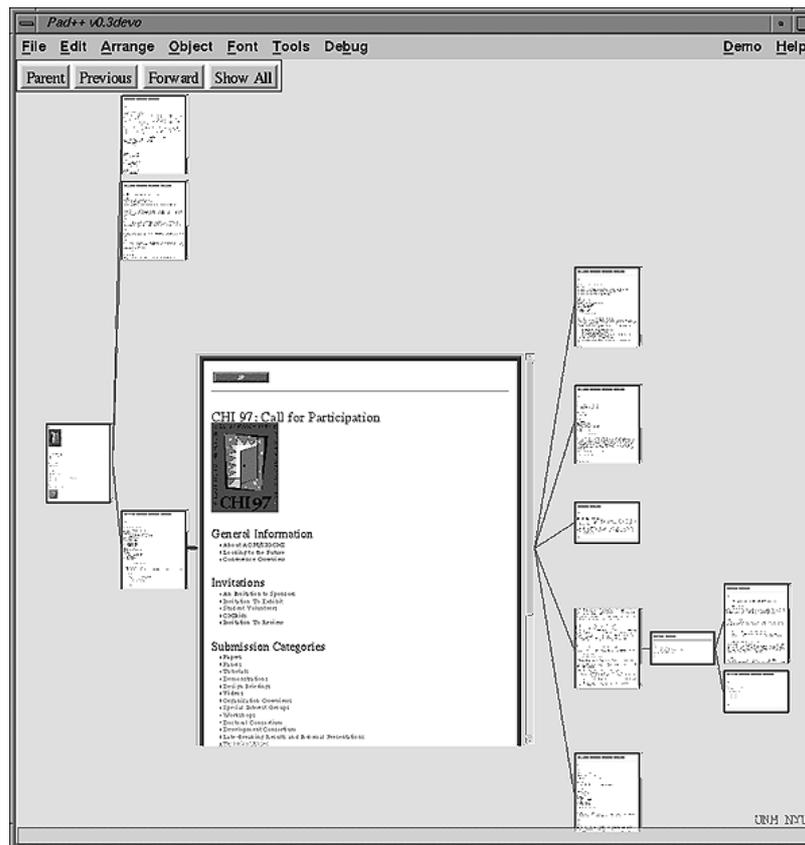


Figure 4.4 : Système PadPrints, présentation graphique d'un historique de navigation Web.

4.3 Personnalisation

Les techniques de cette catégorie permettent aux utilisateurs de personnaliser les documents ou leur environnement de travail afin de construire un système repérage facilitant la réutilisation et la recherche de données préalablement consultées.

4.3.1 Marquage et annotation

Les techniques de marquage permettent au lecteur d'enrichir les documents par un ensemble de signes graphiques et d'annotations textuelles. Les actions du lecteur sont intégrées à la fois comme données et comme instruments de navigation [Bernstein91a, Ginsburg96, Marshall98].

Ce type de procédé rejoint le concept de lecture active (section 2.1.1). Le système XLibris, développé au Xerox Parc [Schilit98a], en propose une implémentation. Ce système présente les documents sur une tablette à affichage graphique et permet aux utilisateurs de griffonner des notes, de dessiner des figures, de souligner des mots, de mettre en relief ou de commenter du texte, au moyen d'un stylo électronique (Figure

4.5). Le système propose plusieurs stylos de surlignage, plusieurs couleurs d'encre et une gomme.

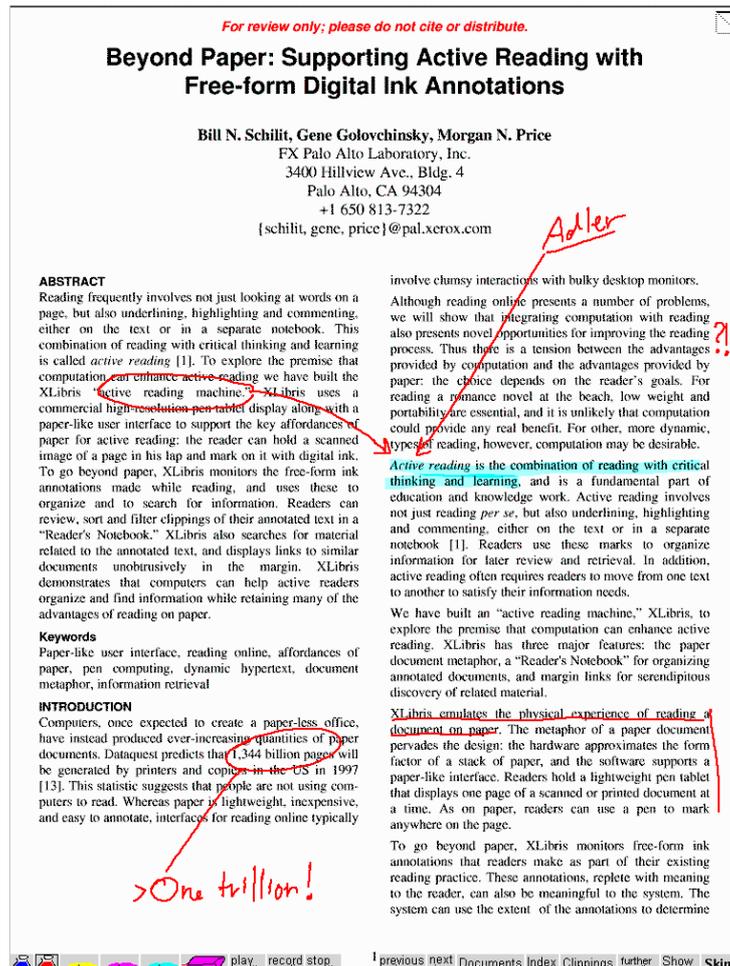


Figure 4.5 : Système XLibris, exemple d'annotation d'un document électronique.

Notre propre système, présenté au chapitre 2, utilise cette technique de marquage. Toutefois, peu de logiciels fournissent ce type de possibilités.

4.3.2 Marque-page (Bookmark)

Les bookmarks permettent aux utilisateurs d'identifier des nœuds comme points de repères particuliers. Ce procédé permet de retourner à certains nœuds jugés intéressants à n'importe quel moment et de n'importe quel point du système, sans avoir à recommencer une recherche par requête ou navigation. De nombreux systèmes hypermédiés exploitent ce procédé.

Par exemple, les navigateurs Web courants permettent de sauvegarder des listes d'URLs représentés par le titre des pages HTML qu'ils référencent. Les utilisateurs ont également la possibilité d'organiser ces références de manière hiérarchique. Les bookmarks peuvent être temporaires ou persistants. Certains systèmes comme Deckview [Ginsburg96] offrent ces deux possibilités (Figure 4.6).

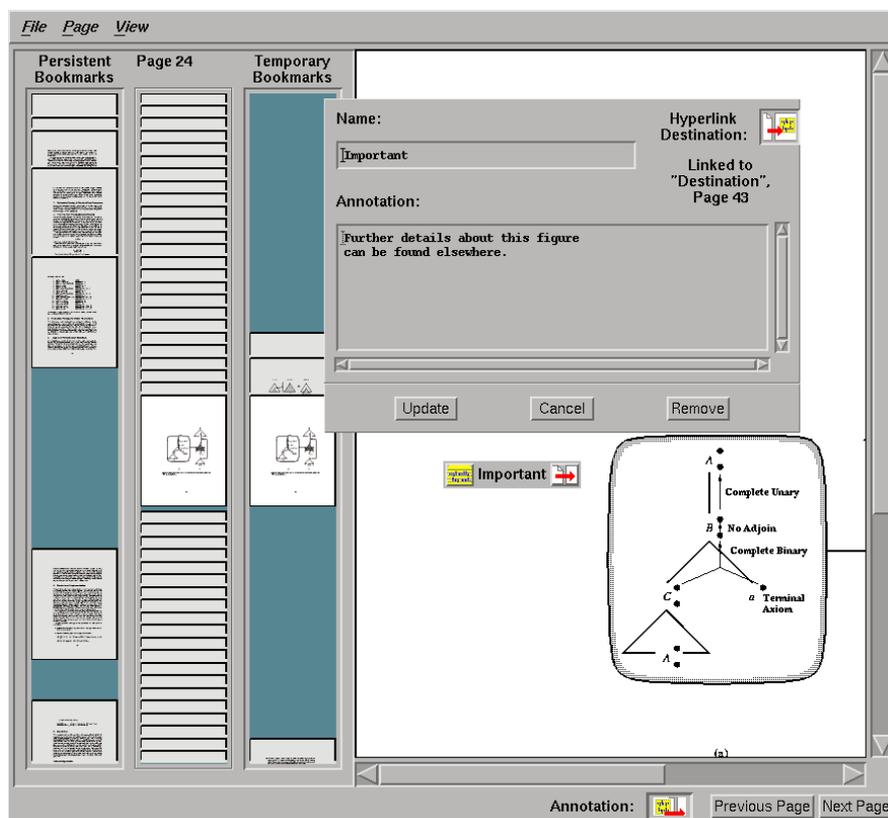


Figure 4.6 : Bookmarks temporaires et persistants dans Deckview.

L'utilisabilité des mécanismes de bookmarks dépend de la manière dont le système les représente et permet de gérer leur organisation. Comme dans le cas des historiques de navigation, la présentation sous forme de listes ou de hiérarchies de labels textuels ne s'avère pas optimale. Certaines études, notamment celles de Card [Card96] ou de Robertson [Robertson98] (section 5.7.1 et 5.7.2), proposent des techniques alternatives de représentation pour améliorer l'utilisation des bookmarks. Nous nous sommes également penché sur ce problème, comme nous le verrons au chapitre 7.

4.4 Augmentation de la structuration

Les techniques de cette catégorie ont pour but d'améliorer la structuration des informations présentées afin d'en faciliter la compréhension et l'interprétation.

4.4.1 Typage des nœuds et des liens

Le typage des nœuds et des liens est un moyen pour mieux structurer l'organisation des données et pour en fournir une description de haut niveau. Il permet de représenter les aspects généraux des domaines d'applications sous forme de schémas conceptuels. Ceux-ci permettent de distinguer les différents types d'informations et de relations disponibles [Garzotto91, Nanard91].

Cette approche offre plusieurs avantages. Divers traitements automatiques, comme la recherche par le contenu ou l'indexation, peuvent être limités à un type de nœuds particuliers, ou tirer profit des informations sémantiques introduites par le typage des nœuds et des liens. Elle permet également de filtrer les informations visualisées par les utilisateurs en fonction de leurs besoins ou de leurs objectifs. Par exemple, dans un hypertexte géographique, il peut y avoir un filtre pour les liens reliant les deux nœuds Océan Pacifique et Courants Maritimes, qui sera activé lors de la consultation de la base par un skipper et masqué pour un minéralogiste ou un volcanologue. Ce procédé permet de ne pas encombrer l'hypergraphe de liens inutiles pour la plupart des utilisateurs. Il peut également être employé pour des raisons de sécurité afin de restreindre l'accès de certains nœuds à des utilisateurs privilégiés.

On distingue deux catégories de systèmes qui utilisent le typage pour faciliter la description des informations traitées.

- ? Les systèmes restreints à certains domaines applications avec des types de nœuds et de liens prédéfinis, comme c'est par exemple le cas des systèmes gIBIS [Conklin88], SEPIA [Streitz92] ou PHIDIAS [McCall90]. Le système gIBIS [Conklin88] est basé sur une métaphore d'argumentation avec trois types de nœuds et neuf types de liens correspondant à leur rôle dans une discussion sur un problème. Plusieurs auteurs peuvent collaborer en créant des nœuds exprimant des arguments (documents de type *Argument*), des positions (documents de type *Position*) et des issues (documents de type *Issue*) reliés par des liens de différents types pour distinguer entre les arguments pour (*supports*) ou contre (*object-to*), les positions, etc.
- ? Les systèmes qui permettent de créer des applications nouvelles avec des types définis par les utilisateurs, comme c'est par exemple le cas des systèmes MacWeb [Nanard89], MORE [Lucarella93] et Gram [Amann92]. Dans MacWeb [Nanard89], un hypertexte est géré sous la forme d'un réseau (*web*) de nœuds (*chunk*) et de liens typés. Les types de nœuds et de liens peuvent être créés explicitement pour représenter des connaissances au sein de l'application hypertexte. Une approche orientée objet permet de spécifier des liens d'héritage entre les types et d'associer un comportement aux différents nœuds au moyen de scripts. Une originalité du système MacWeb est que les schémas conceptuels sont également gérés sous forme de *webs* (*webs* de types), c'est-à-dire de la même

façon que les réseaux hypertextes (Figure 4.7). Les types sont représentés par des nœuds hypertextes, et les relations entre les types (liens d'héritage, liens de composition, etc.) par des liens dans le *web* des types.

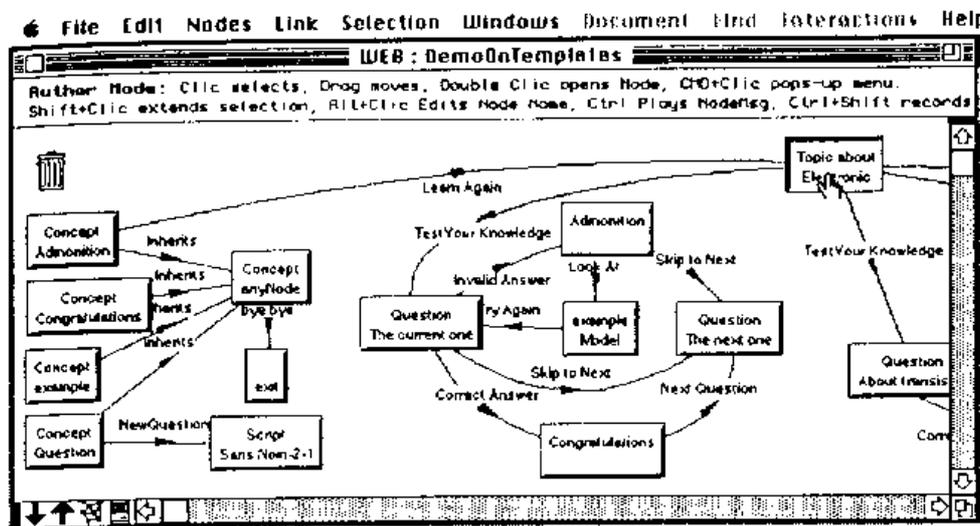


Figure 4.7 : Système MacWeb, une vue du *web* des types (à gauche) et du web principal.

4.4.2 Transformations structurelles

Ces techniques visent à simplifier la structure du réseau d'hyperliens afin de faciliter l'exploration. Il est couramment admis que plus la structure d'ensemble d'un hyperdocument est simple, plus les risques de désorientation sont faibles. Dans cette optique, on peut distinguer deux types d'approches : la simplification en structure hiérarchique et le regroupement de nœuds en nuées.

Simplification en structure hiérarchique

Une structure hiérarchique constitue un système de repérage couramment utilisé (tables de matières, questionnaires de fichiers, etc.) et peut être représentée plus facilement et plus efficacement qu'un graphe quelconque.

La réduction de la structure d'un hyperdocument à une structure hiérarchique est souvent une opération réalisée au moment de la conception. Par exemple, dans KMS (section 1.2.2.5) [Aksyn88], chaque nœud du réseau hypertexte est intégré au moment de sa création à une structure arborescente. Seul l'ajout de liens spéciaux permet de relier des nœuds quelconques.

Une autre possibilité, proposée par Shneiderman et al., consiste à créer de manière automatique des structures arborescentes à partir d'hyperdocuments ayant une

structure de graphe plus complexe [Shneiderman92]. La première étape consiste alors à détecter des racines potentielles d'après les critères suivants :

1. une racine est un nœud à partir duquel il existe un chemin vers tous les autres nœuds.
2. la distance moyenne entre la racine et tous les nœuds du graphe ne devrait pas être trop grande.
3. le nombre de descendants d'une racine candidate doit être « raisonnable ». En particulier, un nombre élevé de descendants caractérise souvent un nœud index (qui contient la liste des sujets et qui est donc directement lié à pratiquement tous les autres nœuds du réseau), il n'est pas souhaitable qu'un tel nœud soit racine.

Les auteurs définissent donc une mesure, qui permet de déterminer, à partir d'une analyse structurelle du réseau existant, les nœuds qui répondent le mieux à (1) et (2). Puis les nœuds index sont retirés de cet ensemble de résultats. Les nœuds restant constituent les candidats soumis aux auteurs pour validation. Une fois les racines déterminées, un algorithme construit un ou plusieurs hyperdocuments hiérarchiques. Cette opération n'est possible que si le graphe initial est entièrement connexe [Berge83].

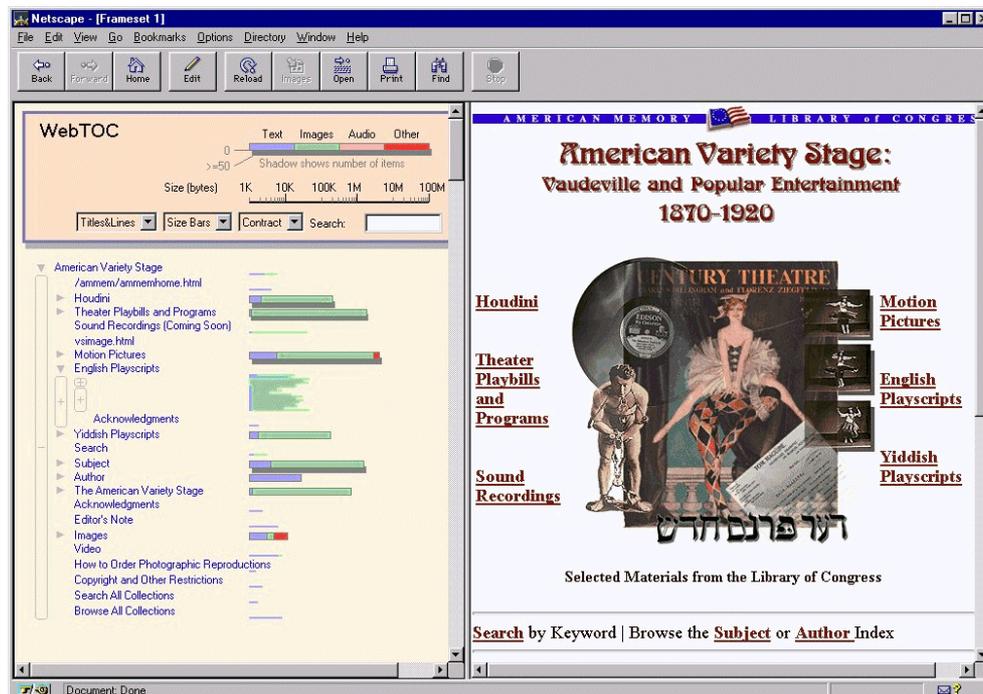


Figure 4.8 : Système WebTOC, une vue d'ensemble hiérarchique d'un site Web.

Dans le cas du Web, une solution consiste à présenter les pages d'un site sous forme hiérarchique en se basant sur la structuration physique des répertoires contenant ces

pages. Une autre possibilité est donnée par le système WebTOC [Nation97]. Ce système utilise un algorithme qui crée un arbre en suivant les liens rencontrés à partir de la racine du site (Figure 4.8). Les liens qui pointent vers des pages extérieures au site ne sont pas pris en compte. Seule la première occurrence d'un lien est ajoutée à l'arbre.

Regroupement de nœuds en nuées

L'idée est de réduire la complexité de la structure globale d'un hyperdocument en diminuant le nombre de nœuds qui la composent [Crouch89, Nielsen90a, Rivlin94]. La méthode proposée consiste à effectuer un regroupement de nœuds en nuées, les nuées prenant à leur tour le rôle de nœuds. Le graphe d'origine constitué d'un nombre important de nœuds se transforme en un graphe plus petit, composé de nuées et de liens entre ces nuées. Dans la plupart des cas, les liens entre les nœuds initiaux demeurent à l'intérieur des nuées et deviennent des liens spéciaux dans le nouvel hyperdocument. Il est d'usage de dissocier lors de la navigation les liens inter-classes des liens intra-classes par un artefact de présentation [Nielsen90a]. Par exemple, une animation graphique peut être associée à l'action de suivre un lien, la forme de celle-ci variant selon le type du lien traversé (inter ou intra-nuée).

4.5 Métaphore de la navigation spatiale

Une stratégie souvent utilisée dans les systèmes hypermédias consiste à proposer à l'utilisateur une carte (ou un diagramme) du réseau d'hyperliens pour faciliter le repérage et la localisation en visualisant la structure de l'hyperdocument. Des systèmes tels que Intermédia (section 1.2.2.10) [Yankelovich88] ou NoteCards (section 1.2.2.12) [Halasz88], proposent des navigateurs graphiques qui, tels des cartes routières, permettent de se situer dans un hyperespace documentaire. Ce type de navigateur peut être global ou local selon qu'il représente l'ensemble des nœuds et des liens du réseau hypermédia, ou uniquement les liens rattachés à un nœud particulier ou à un ensemble de tels nœuds [Conklin87, Halasz88].

De nombreux auteurs s'accordent à dire, que les diagrammes d'ensemble sont des instruments indispensables pour assurer une bonne navigation dans un hyperdocument. Cependant, l'implémentation et l'utilisation de ces diagrammes posent un certain nombre de problèmes [Fairchild88, Utting89]. L'affichage d'un nombre important de nœuds et de liens enchevêtrés est généralement difficile à comprendre et à interpréter. Une étude de Foss [Foss89] a montré que la conception de diagrammes demande plus que le simple affichage des nœuds et des liens à l'écran. Les utilisateurs ont besoin d'outils appropriés pour explorer ces diagrammes et pouvoir accéder à un élément particulier. Conklin [Conklin87] note la difficulté à utiliser des cartes de manière productive pour de grands systèmes, alors que Akscyn

[Aksyn88] considère que la conception d'un diagramme non adapté ne fournit pas une aide efficace à la navigation, en particulier pour des hypermédias complexes.

La plupart des systèmes hypertextes, qui permettent de réaliser cartes ou diagrammes, utilisent leurs propres moyens et peu de règles peuvent être tirées de ces réalisations. Les diagrammes sont souvent construits de manière manuelle [Nielsen90a] et « en dur » au moment de la conception. Cette solution n'est cependant pas adaptée, dès lors que la structure des hyperdocuments devient dynamique, ou que le nombre de nœuds devient élevé.

Dans les cas d'hyperdocuments hiérarchiques, les techniques bien connues sur la manipulation et la visualisation des structures arborescentes permettent la construction automatique des diagrammes. Mais la représentation de réseaux hypertextes ayant une structure plus complexe reste un problème ouvert. Certains systèmes, tels que Intermedia [Yankelovich88], proposent des mécanismes automatiques pour la construction de tels diagrammes. Cependant, les graphes hypertextuels sont généralement des graphes non coplanaires, complexes, pour lesquels il n'existe pas de technique de placement donnant un affichage à la fois esthétique, intelligible et utilisant efficacement l'espace de l'écran [Bernstein88]. C'est pourquoi les techniques de transformation structurelle présentées à la section 4.4.2 peuvent faciliter l'exploration en permettant la construction de représentations plus simples à afficher et à interpréter.

Plus généralement, comme il n'est pas possible de représenter tous les nœuds d'un hyperdocument dans un même diagramme, il s'avère nécessaire de faire des choix quant aux nœuds à représenter. Une possibilité consiste à prendre en compte le contenu des nœuds, afin d'extraire de la structure réelle du réseau, un diagramme suffisamment abstrait pour que le nombre de nœuds et de liens à afficher s'en trouve réduit. En fait, les diagrammes qui calquent directement la structure du réseau sont très vite limités par leur faible niveau d'abstraction. Une autre possibilité, dans le cas d'hypermédias typés (section 4.4.1), consiste à prendre en compte le type des nœuds ou des liens pour définir la représentation adéquate à la réalisation d'une tâche particulière.

4.6 Conclusion

La taxonomie proposée n'est pas la seule possible. Une alternative consiste à prendre en compte l'aspect statique ou dynamique des techniques présentées.

Par exemple, les outils dérivés du livre traditionnels (tables des matières, glossaires, etc.) et les points de repères sont statiques. Ils sont le plus souvent définis lors de la phase de conception. Au contraire, le retour arrière, les historiques de navigation, les bookmarks, le marquage des documents sont dynamiques. Ils évoluent au fur et à

mesure de l'exploration. Certaines de ces techniques peuvent être statiques ou dynamiques. C'est par exemple le cas pour la construction d'un diagramme.

Pour conclure, le Tableau 4.1 met en relation les techniques d'aide présentées avec les quatre activités (voir introduction) par lesquelles nous identifions l'exploration hyper-documentaire. Les cercles indiquent que la technique peut être utilisée pour faciliter la réalisation de la tâche au moyen d'un artefact particulier de conception ou d'implémentation.

Les *outils dérivés du livre traditionnel* sont utilisés pour *trouver* l'information et *comprendre* les différents concepts mis en jeu.

Les *points de repères* permettent de *trouver* et de *retrouver* (lors d'explorations ultérieures) des éléments jugés importants par les auteurs. Dans le cas où ces points de repères sont regroupés en catégories, ils fournissent également des informations facilitant la *compréhension* des données présentes dans l'hypermédia

Les *visites guidées* sont utilisées pour la découverte de l'hypermédia. Elles permettent principalement de *comprendre* la nature des données présentées. Elles peuvent également servir à *trouver* et à *retrouver* certains nœuds d'information. Si les utilisateurs ont la possibilité de définir leurs propres visites guidées, ils disposent alors d'un moyen de créer une organisation personnalisée des données, ce qui peut les aider à *retrouver* plus facilement certains éléments.

Le *retour arrière* ne constitue pas réellement une aide à l'exploration. Il est toutefois fort utile, dans les systèmes faiblement outillés, pour *retrouver* les documents consultés dans la session de navigation courante. Les *historiques de navigation* « efficaces » semblent fournir une meilleure solution pour *comprendre* les chemins suivis et *retrouver* les documents.

Le *marquage des documents* facilite la compréhension des données consultées. Les lecteurs ajoutent leurs propres marques (notes, dessins, etc.), créant ainsi un système de repérage qui sera mis à profit lors de consultations ultérieures. Comme montré au chapitre 2, le codage de ces marques dans un format normalisé facilite la réalisation de traitements automatiques comme l'indexation et la recherche par le contenu, aidant en cela à *trouver* et à *retrouver* les informations. L'utilisation de techniques hypertextes lors du marquage peut également aider à *organiser* l'information en la structurant au moyen d'hyperliens (qui se superposent aux documents d'origines).

La définition de *bookmarks* permet d'*organiser* un ensemble de documents afin d'aider à les *retrouver*. L'organisation sémantique définie par l'utilisateur, en regroupant par exemple les bookmarks par catégories, peut également l'aider à *comprendre* les notions présentes dans l'hypermédia.

	Trouver	Comprendre	Organiser	Retrouver
Outils du livre	X	X		
Points de repères	X	O		X
Visites guidées	X	X	O	X
Retour arrière				X
Historiques		X		X
Marquage	O	X	O	O
Bookmarks		X	X	X
Typages	X	X		
Transformations structurelles	X	X		
Diagrammes	X	X		X

Tableau 4.1 : Aides aux différentes activités liées à l'exploration hyper-documentaire.

Le *typage des nœuds et des liens* augmente la structuration des informations et introduit une sémantique qui facilite la *compréhension* de l'hypermédia. Cette sémantique améliore les possibilités d'interrogation et, en cela, aide à *trouver* l'information.

Les *transformations* structurent les données de l'hypermédia pour permettre aux utilisateurs de les *trouver* et de les *comprendre* plus simplement.

Les *diagrammes* permettent de *comprendre* l'organisation des données au moyen, par exemple, de vues d'ensembles ou détaillées. La visualisation d'un diagramme peut aider à *trouver* les documents ou à les *retrouver*, les utilisateurs pouvant se remémorer la localisation spatiale des divers éléments affichés.

5. Visualisation



Les stratégies de présentation conditionnent souvent les performances des interfaces destinées à l'exploration d'espaces informationnels mettant en œuvre de grandes quantités de données. Parmi ces stratégies, les techniques interactives de visualisation jouent un rôle clé en permettant de construire des vues synthétiques qui facilitent le survol de l'espace d'information et la compréhension de son organisation. Elles facilitent également la mise en correspondance, via des présentations graphiques adéquates, des données manipulées et des représentations mentales qu'en ont les utilisateurs (diminuant du même coup leur charge cognitive). Ces techniques peuvent être mises à profit pour améliorer l'utilisabilité des mécanismes d'aide à l'exploration présentés au chapitre 4 (visites guidées, historiques de navigation, bookmarks, etc.).

La conception de systèmes interactifs de visualisation s'avère toutefois non triviale. Les données à traiter sont généralement trop nombreuses pour être toutes affichées de manière efficace dans l'espace restreint que représente l'écran, ce qui entraîne de nombreuses questions parmi lesquelles :

- ? comment faire pour que l'organisation de la masse d'information présentée soit comprise par les utilisateurs ?
- ? comment éviter l'encombrement de l'affichage ?
- ? comment concentrer le détail sur des régions intéressantes ou importantes de l'affichage, sans en perdre le contexte ?
- ? comment localiser un élément particulier (navigation) ? comment l'interpréter ? et comment comprendre ses relations avec les autres éléments ?

Malgré le foisonnement des techniques développées pour résoudre ces difficultés, peu de travaux visent à faire le bilan du domaine afin de fournir un cadre réutilisable lors de la conception de systèmes de visualisation.

5.1 Taxonomies

Les principaux systèmes de classification en visualisation de l'information ont été construits autour de trois axes : le type des données, les traitements et les tâches.

5.1.1 Le type des données

La littérature sur la visualisation de l'information fait souvent référence aux travaux de Bertin [Bertin77] sur l'étude de la sémiologie graphique qui suggèrent que la forme de la représentation utilisée dépend du type des données à afficher. Selon Bertin, les données peuvent se représenter sous deux formes, soit en fonction de leur valeur, soit en fonction de leur structure. Les valeurs sont associées aux attributs (numériques, thématiques, etc.) inhérents à un problème. La structure comprend les relations (liens, contraintes) qui caractérisent les données comme un tout. A chacune de ces formes (et pour un type de données particulier) doit être associée une représentation graphique différente. Par exemple, un histogramme peut représenter des valeurs d'attributs, alors qu'un diagramme d'arbre représente les relations structurelles à l'intérieur d'un ensemble de données.

A partir des observations de Bertin, plusieurs taxonomies ont été construites en fonction du type des données à représenter. Parmi celles-ci, une des plus connues est celle de Shneiderman [Shneiderman96].

Taxonomie de Shneiderman

En addition au type des données, cette taxonomie prend en compte un ensemble de tâches que la représentation doit s'efforcer de faciliter [Shneiderman96]. Sept types de données (Tableau 5.1) et sept tâches (Tableau 5.2) ont été identifiés dans cette étude. La matrice résultante (type de données / tâche) fournit un cadre de conception qui permet de réfléchir aux tâches à offrir en fonction des types de données manipulés. Les tâches proposées définissent un cadre général, qui peut être étendu par le raffinement de ces tâches, ou par l'ajout de nouvelles tâches (en fonction des besoins des utilisateurs).

Types de données	Description
1D	Données organisées de manière linéaire ou séquentielle : documents textuels, lignes de code d'un programme, listes alphabétiques, etc.
2D	Données dont la localisation ou la géométrie dans le plan est primordiale : plans, cartes géographiques, mise en page de journaux, etc.

3D	Données qui accordent une prédominance à la localisation ou la géométrie dans l'espace : chimie, architecture, mécanique, etc.
Temporelles	Données qui ont une existence dans le temps : gestion de projet, suivi médical, etc.
Multi-dimensionnelles	Données dont le caractère spatial n'est pas dominant et qui possèdent un nombre d'attributs n : données statistiques, contenu de bases de données, etc.
Hiérarchiques	Données organisées de manière à ce que chaque entité possède un lien unique vers une entité parent (à l'exception de la racine) : système de fichiers, GUI, etc.
Relationnelles	Données qui forment un graphe dont la structure n'est pas arborescente : hyperdocuments, réseaux télécoms, etc.

Tableau 5.1 : Types de données dans la taxonomie de Shneiderman.

Tâches	Description
Donner une vue d'ensemble	Obtenir une vue globale sur la collection entière.
Zoomer	Zoomer sur une entité jugée intéressante.
Filtrer	Éliminer de la vue les données non pertinentes à un moment donné.
Détailler à la demande	Sélectionner une entité ou un groupe et en visualiser les détails si besoin.
Associer	Visualiser les relations entre les éléments.

Fournir l'historique	Tracer un historique afin de supporter l'annulation d'une ou plusieurs commandes, le retour arrière, le play-back, etc.
Extraire	Permettre la sauvegarde de sous-collections et de paramètres de requêtes.

Tableau 5.2 : Tâches dans la taxonomie de Shneiderman.

5.1.2 Les traitements

Un autre type d'approche pour classifier les systèmes de visualisation repose sur l'analyse des différents traitements proposés par ces systèmes. Un exemple possible concerne l'analyse des interactions en entrée, la classification s'effectuant en fonction de leur nature (sélection par manipulation directe ou indirecte), de leur niveau (singleton, groupe ou intégralité des objets ou des attributs) et de leur conséquence en terme de transformation graphique, de présentation et d'organisation [Tweedie97].

Plusieurs approches orientées traitements ont été proposées. Parmi celles-ci, la taxonomie de Leung classe les systèmes de visualisation en fonction des techniques utilisées pour faire évoluer la représentation au cours de l'exploration [Leung94]. La Taxonomie de Chuah et Roth repose sur un ensemble de primitives d'interaction qui permettent de décrire les opérations proposées par les systèmes [Chuah96]. La taxonomie de Chi repose quant à elle sur une modélisation du processus de visualisation [Chi98]. Elle permet de décrire les différentes étapes séparant les données de leurs représentations graphiques, ainsi que les opérateurs utilisés pour passer d'une étape à une autre.

Taxonomie de Leung

Leung [Leung94] classe les systèmes de visualisation en deux catégories en fonction des techniques utilisées pour accéder à l'information. La première catégorie regroupe l'ensemble des techniques qui permettent de déformer la représentation, la seconde regroupe les techniques non déformantes (Figure 5.1). La principale caractéristique des techniques de déformation consiste à permettre aux utilisateurs d'examiner en détail une région particulière de la représentation tout en gardant une vision globale de l'espace (voir section 5.2.1), ceci afin de faciliter la navigation. Leung affine cette classification en différenciant les données de nature graphique (i.e. qui possèdent des relations spatiales implicites), des données non graphiques, ces dernières pouvant toutefois être représentées sous une forme graphique dans de nombreux cas.

Les cartes et les plans sont de bons exemples de données graphiques, alors qu'un graphe hypertexte correspond à une donnée non graphique.

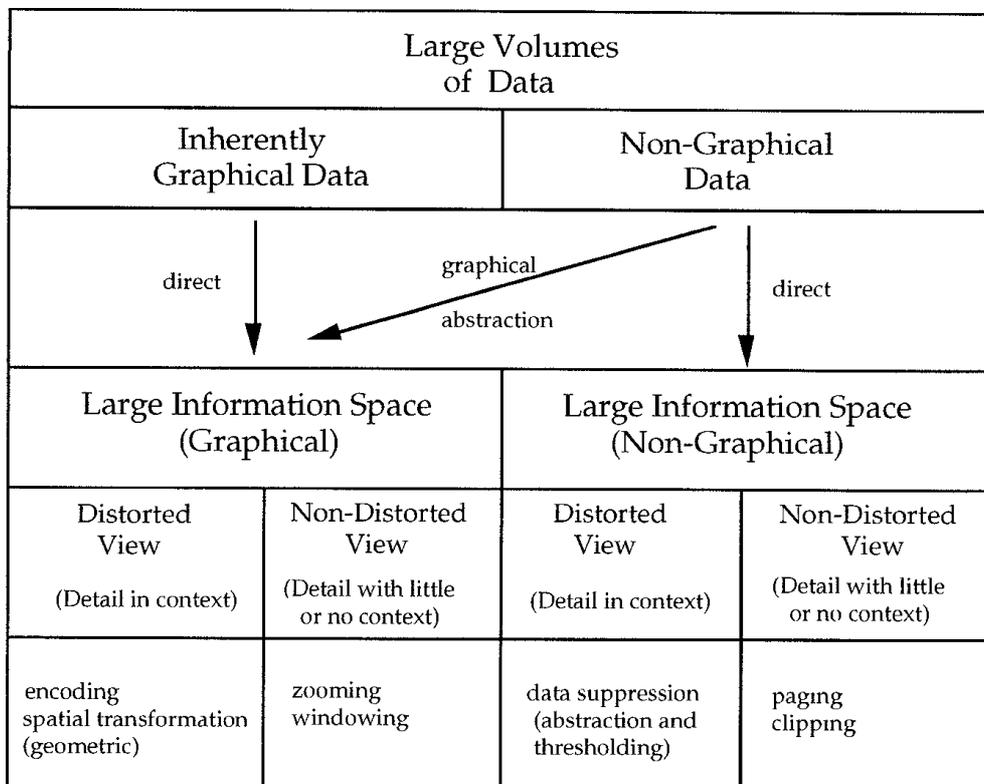


Figure 5.1 : Taxonomie proposée par Leung.

La méthode utilisée pour transformer la représentation semble être un critère de classification bien adapté à l'exploration hyper-documentaire. Par exemple, l'utilisateur peut visualiser une représentation globale pour *comprendre* l'organisation des documents, puis passer à une vue locale montrant le détail de certains documents afin de *trouver* celui qui l'intéresse le plus. Le passage du global au local peut être réalisé de plusieurs manières, notamment au moyen d'une déformation de l'affichage ou d'un zoom.

Taxonomie de Chuah et Roth

Chuah et Roth [Chuah96] classent les systèmes de visualisation en fonction d'un ensemble de primitives d'interaction. Ils présentent une classification fonctionnelle de ces primitives et un ensemble de règles permettant de les composer. Cette taxonomie, en plus de fournir un cadre d'étude pour la comparaison des systèmes existants, est utilisée par les auteurs comme un guide pour la conception d'un toolkit dédié à la construction d'interfaces de visualisation par la composition de primitives

d'interaction. Ces travaux reposent sur une étude de Foley [Foley90], concernant les interfaces utilisateurs, qui décrit trois niveaux de conception d'interfaces. Le *niveau lexical* fait référence à la manière dont les primitives d'entrées et de sorties dérivent des fonctions matérielles de base. Le *niveau syntaxique* consiste en un ensemble de règles par lesquelles des unités (basiques) d'entrées-sorties peuvent être composées pour former des séquences ordonnées d'entrées-sorties. Par exemple, des clics et des mouvements de la souris peuvent être utilisés pour spécifier le rectangle englobant une zone de focus (mais pas la fonction associée à cette zone). Le *niveau sémantique* définit une tâche comme la signification donnée à une séquence d'actions (par exemple, une tâche de sélection peut être définie par un enchaînement de clics souris, de déplacements d'ascenseurs, etc.). Chuah et Roth étendent le niveau sémantique de ce cadre d'étude en y incorporant les primitives d'interaction des systèmes de visualisation, nommées BVIs (Basic Visualization Interaction). Ce cadre d'étude permet également de définir comment composer les BVIS pour arriver aux fonctionnalités interactives fréquemment rencontrées dans les systèmes de visualisation (comme l'agrégation, le filtrage, le tri, l'affichage synchronisé, etc.).

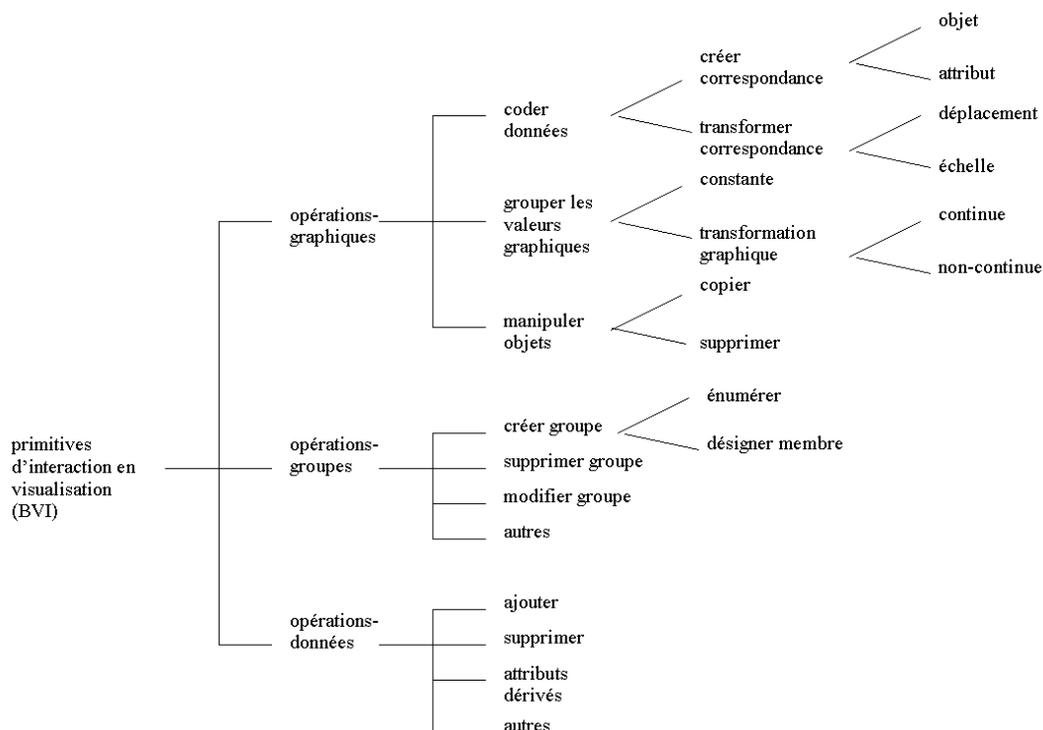


Figure 5.2 : Taxonomie BVI proposée par Chuah.

Chuah et Roth identifient trois classes de BVIs : les opérations graphiques qui permettent de changer l'apparence de l'espace d'affichage, les opérations de données qui permettent de manipuler les objets qui représentent les données dans l'espace

d'affichage, et les opérations de groupes qui permettent de créer et de manipuler des groupes d'objets (Figure 5.2).

Les sorties des primitives d'interaction peuvent affecter les différents états du système, identifiés au nombre de trois. L'état du graphique fait référence à tous les objets qui sont actuellement visualisés (les objets graphiques, les labels, etc.). L'état des données contient les informations sur les données qui sont actuellement dans le système (cardinalité, type de données, etc.). L'état du contrôle contient les informations internes générées pendant les opérations du système (abstraction des objets, propriétés globales, état de l'interaction, historique). Les opérations-graphiques affectent la représentation graphique des données (i.e. leur état graphique). Les opérations-groupes affectent l'état de contrôle et les opérations-données affectent l'état des données. La modification de l'état de contrôle et de l'état des données peut cependant engendrer des effets secondaires sur l'état graphique. Par exemple, la destruction d'une donnée entraîne la suppression de toutes les représentations graphiques de la donnée détruite, et par conséquent modifie l'état graphique de la représentation. Enfin, le modèle permet de composer les BVIs de trois manières. Dans une composition indépendante, les BVIs sont simultanément disponibles mais ne sont liées d'aucune façon. Les opérations peuvent être exécutées dans n'importe quel ordre ou en parallèle, et leurs effets s'appliquent de manière orthogonale. Dans une composition groupée, le déclenchement d'une BVI entraîne l'exécution de toutes les autres. Dans une composition chaînée, l'action d'une BVI particulière dépend de son prédécesseur dans la séquence.

Les primitives d'interactions identifiées par Chuah et Roth semblent fournir un moyen de décrire précisément, et de catégoriser, les opérations de bas niveau (focaliser sur une région d'intérêt, suivre un lien, etc.) qui entre en jeu dans un processus d'exploration.

Taxonomie de Chi

Les travaux de Chi [Chi98, Chi00] reposent sur l'analyse des opérateurs de traitements inhérents à chaque technique de visualisation. Selon Chi, ces opérateurs font partie intégrante du processus de visualisation qu'il représente au moyen du modèle Etat-Donnée (Data State Model) [Chi98]. C'est sur la base de ce modèle, schématisé à la Figure 5.3, que s'appuie son système de classification.

Ce modèle permet de décomposer chaque technique de visualisation en :

- ? quatre étapes par lesquelles transitent les données dans le processus de visualisation,
- ? trois types de transformations utilisées pour faire passer les données entre deux étapes,

? quatre types d'opérateurs intra-étape.

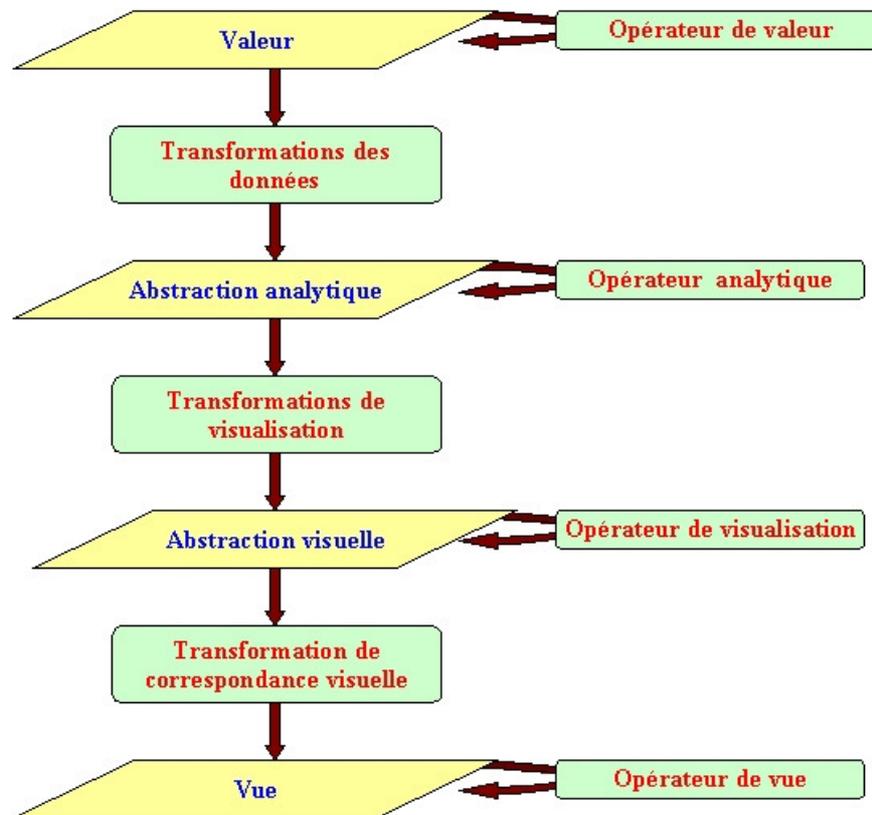


Figure 5.3 : Modélisation du processus de visualisation proposée par Chi.

Les étapes sur les données (Tableau 5.3) et les opérateurs de transformations (Tableau 5.4), structurent le processus de transformation des données en vues graphiques.

Le pipeline de visualisation décrit par ce modèle fournit un moyen de classier les opérateurs. Il permet de différencier les opérateurs sur les vues (rotation, zoom, etc.) et les opérateurs sur les données (tri, ajout, etc.), deux types d'opérateurs largement utilisés en visualisation d'information. Il met également en évidence un ensemble d'autres opérateurs qui permettent de décrire la manière dont sont générées les méta-données dans le processus d'analyse des données.

Cette étude décompose le processus de visualisation en deux espaces. L'espace des données qui permet au système de contrôler le traitement (extraction de méta-données) et le filtrage de l'information (réduction de l'espace). L'espace de la vue qui permet à l'utilisateur de contrôler les opérations appliquées à la représentation (zoom, changement de vue, etc.).

Etapas des données	Description
Valeur	Les données « brutes ».
Abstraction analytique	Des informations sur les données (méta-données).
Abstraction visuelle	Des informations visualisables à l'écran au moyen de techniques de visualisation.
Vue	Le produit final du processus de visualisation à partir duquel l'utilisateur voit et interprète les informations.

Tableau 5.3 : Etapes des données dans DSM (Chi).

Opérateurs de Transformation	Description
Tr. de données	Génère un ensemble d'abstractions analytiques à partir de la valeur des données (principalement par extraction).
Tr. de visualisation	Crée à partir d'une abstraction analytique des abstractions de visualisation (du contenu visualisable).
Tr. de correspondance visuelle	Présente une vue graphique à partir d'un format d'information visualisable.

Tableau 5.4 : Opérateurs de transformation dans DSM (Chi).

5.1.3 Les tâches

L'usage des données conditionne souvent la représentation de ces données. Cependant, peu de taxonomies existantes se basent sur l'analyse des tâches que les systèmes de visualisation permettent d'effectuer. La taxonomie de Shneiderman, décrite à la section 5.1.1, repose en partie sur la définition d'un ensemble de tâches

de bas niveaux (zoomer, filtrer, associer, extraire, etc.). Cette étude masque toutefois l'importance des tâches de haut niveau (trouver, comprendre, organiser, retrouver) dans le processus de conception d'un système de visualisation.

Taxonomie de Bruley

Selon Bruley [Bruley99], la manière de représenter des données dépend fortement des tâches que l'utilisateur souhaite effectuer sur ces données. Il introduit donc la notion de point de vue, un même ensemble de données pouvant être représenté par différents points de vues. Cette approche rejoint celle d'Ed Chi qui met en évidence la séparation entre l'espace des données et l'espace de représentation. L'idée de point de vue (une vision particulière) sur les données est, selon nous, un des points les plus importants de cette étude.

Sur la base des travaux de Shneiderman, Bruley définit donc une taxonomie induite par le point de vue sur les données, plutôt que par leur type. Il met en avant la manière de représenter l'organisation des données plutôt que leur organisation réelle. Bruley propose deux nouvelles catégories (le point de vue spatial et le point de vue non structuré), afin de séparer les données de type 1D, 2D, 3D et nD qui ont une géométrie, de celles qui n'en ont pas. Cette séparation est d'ailleurs à rapprocher de celle proposée par Leung sur les données graphiques et non graphiques.

Bruley déduit de son étude une démarche à suivre pour la conception de représentations graphiques qui est basée sur quatre éléments :

- ? *l'espace des informations* qui contient les données à représenter,
- ? *le (ou les) point(s) de vue* qui résulte(nt) de l'analyse de cet espace et de la nature de la tâche à réaliser,
- ? *l'espace géométrique de représentation* qui fixe la position des objets graphiques,
- ? *l'espace géométrique de construction* qui est la sous-partie de l'espace de représentation réellement affichée.

5.1.4 Synthèse

Le type des données, les traitements et les tâches sont des aspects complémentaires à prendre en compte lors de la conception d'un système de visualisation. Sur la base des travaux de Chi (section 5.1.2) [Chi98], Vernier propose de synthétiser ces différents aspects au moyen d'un cadre d'étude qui décompose le processus de visualisation en quatre étapes [Vernier01].

Le processus de visualisation selon Vernier

Le cadre d'étude de Vernier [Vernier01] peut être schématisé comme le montre la Figure 5.4.

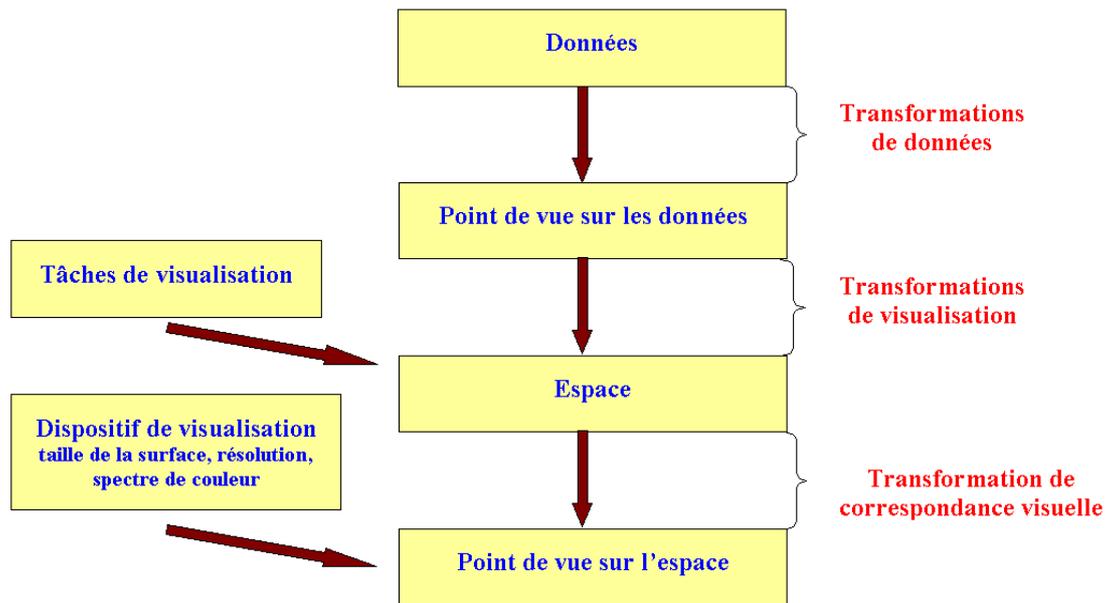


Figure 5.4 : Cadre d'étude du processus de visualisation proposé par Vernier.

La première et la troisième étapes (*données* et *Espace*) sont des états incontournables du processus de visualisation. Les deux autres sont des points de vues sur l'état qui les précède (i.e. des choix d'interprétation).

L'étape des *données* est en amont de la conception de l'interface. Elle regroupe les mécanismes de définition et de mise à disposition des données (numérisation, regroupement, indexation, etc.). Elle permet également de classer les types de données, comme le fait par exemple Shneiderman en différenciant sept catégories (1D, 2D, 3D, temporelles, multi-dimensionnelles, hiérarchiques, relationnelles) ou Leung qui sépare les données graphiques et non graphiques.

Le *point de vue sur les données* est une étape intermédiaire, relative à la structuration des données. Elle consiste à définir un point de vue sur la manière dont il est possible d'organiser les données. Par exemple, à partir d'un ensemble de pages Web, il est possible de construire un diagramme représentant leurs relations (un graphe), une visite guidée ou une liste de points de repères (des séquences) ou encore un historique de navigation (une hiérarchie). Ces différentes organisations peuvent également dépendre de certains critères. Par exemple, la liste des points de repères peut être ordonnée par le titre, par intérêt, etc. Cet état dépend fortement de la tâche à réaliser. Une liste de points de repères ou un diagramme sont plus utiles pour *trouver*

une page, alors qu'un historique permettra de *retrouver* plus facilement une page préalablement consultée. Les catégories de points de vue de Bruley appartiennent à cet état.

L'*espace* permet d'associer aux données un modèle spatial qui sera utilisé pour l'affichage. Par exemple, la représentation d'une hiérarchie nécessite un algorithme de placement qui calcule la position des nœuds et des liens. Cette étape du processus de visualisation dépend du point de vue sur les données. Le placement d'une hiérarchie ne se construit pas de la même manière que celui d'un graphe ou d'une liste. Cette étape doit également prendre en compte la nature de la tâche à réaliser, et de manière plus précise que lors de l'étape précédente.

Le *point de vue sur l'espace* correspond à la manière dont les utilisateurs vont percevoir la représentation. Dans un environnement tridimensionnel, le point de vue sur l'espace est défini par la position de la caméra. Celle-ci peut entraîner l'occlusion de certains éléments, des effets de perspective, etc. Les déformations sur lesquelles reposent la taxonomie de Leung, se trouvent aussi à cette étape.

Conclusion

Dans le processus de visualisation de Vernier, la notion de tâche intervient à l'étape du *point de vue sur les données* et à l'étape de *l'espace*. La description précise de la tâche à réaliser peut aider à identifier l'étape à laquelle il est le plus judicieux de la prendre en compte. La taxonomie de Shneiderman décrit un ensemble de tâches de bas niveau qui correspondent aux actions que l'utilisateur peut effectuer sur la représentation (zoomer, filtrer, détailler, etc.). La combinaison d'un ensemble de ces tâches permet de décrire comment réaliser une activité de haut niveau. Par exemple, trouver un document dans un hypermédia peut être réalisé en donnant une vue d'ensemble, puis en filtrant les nœuds affichés, puis en zoomant sur les nœuds restants. La taxonomie de Chuah fournit quant à elle le moyen de décrire les actions de bas niveau, comme zoomer ou filtrer, au moyen de primitives d'interaction.

Dans le cas de documents hypermédiés, le type des données est un aspect essentiel à prendre en compte qui intervient à deux niveaux. En effet, un hypermédia se définit par la *structure* de son réseau d'hyperliens et par la *nature* des données contenues dans les nœuds (images, textes, sons, etc.). A l'étape du point de vue sur les données, il est donc indispensable de prendre en compte l'organisation des données (comme le précise Vernier), mais également leur nature, afin de définir une représentation appropriée à la réalisation d'une tâche spécifique. Par exemple, il peut être plus utile de représenter des bookmarks de pages Web par des images réduites que par leur titre. En effet, la mise en page, les couleurs, la présence d'images ou encore de schémas sont autant d'éléments qui facilitent l'identification des pages. Dans le cas d'un ensemble de documents purement textuels, des images réduites ne facilitant pas

la lecture de leur contenu, elles ne permettront sans doute pas de différencier les documents. Dans ce cas, le titre (ou un résumé) est une information plus pertinente à donner aux utilisateurs.

La suite du chapitre présente un panorama des travaux réalisés en visualisation de l'information. Les sections 5.2 et 5.3 décrivent les techniques d'interaction utilisées pour explorer la représentation des données (leur organisation et leur contenu) que l'utilisateur visualise. Les sections 5.4, 5.5 et 5.6 décrivent différentes méthodes pour afficher des données organisées sous forme de listes, de matrices, de hiérarchies ou de graphes. Ces techniques fournissent un ensemble de solutions à la représentation d'index, de visites guidées, d'historiques de navigation, de bookmarks, de réseaux d'hyperliens, etc. Enfin, la section 5.7 présente des systèmes qui permettent aux utilisateurs d'organiser eux-mêmes la manière dont seront présentées les données à l'affichage, en fonction d'un contexte ou d'une activité spécifique. Shneiderman identifie ce type de système sous le nom de « Workspace » [Olive99].

5.2 Transformations de l'affichage

La représentation par laquelle les utilisateurs visualisent les données d'une base informationnelle est souvent amenée à évoluer au fur et à mesure de l'exploration. L'utilisateur peut en effet avoir besoin d'accéder à une autre partie de l'espace d'information (qui peut être éloignée de sa position courante), de visualiser le détails de certains éléments ou de les comparer, etc. autant d'activités qui nécessitent des modifications de l'affichage. Les distorsions et les zooms sont les techniques d'interactions les plus répandues pour appliquer des transformations visuelles à un espace d'affichage.

5.2.1 Distorsions

La principale caractéristique des techniques de distorsion est qu'elles permettent de modifier la représentation initiale afin de visualiser en détail une de ses parties tout en gardant à l'écran une représentation de son contexte. Ces techniques se regroupent en deux principales catégories, celles qui modifient la taille et/ou la position des éléments de la représentation [Furnas86, Sarkar92, Rao94] et celles qui déforment le support de visualisation afin d'obtenir un effet de perspective [Mackinlay91, Robertson93, Carpendale95, Keahey96].

La première implémentation d'une technique de distorsion a été réalisée par Spence en 1982 [Spence82]. De nombreuses techniques plus élaborées ont ensuite été proposées [Carpendale95, Carpendale97], sur la base d'un ensemble de principes mathématiques définis en 1978 par Kadmon et Shlomi [Kadmon78]. L'approche la connue a toutefois été proposée en 1986 par Furnas sous le nom de « vue fisheye » [Furnas86]. Après avoir présenté ces travaux et les principes sous-jacents, nous

présentons, comme exemple concret, le système « Table Lens » [Rao94] qui implémente une technique de distorsion pour faciliter la visualisation de tableau de grande taille.

5.2.1.1 L'affichage bifocal

L'affichage bifocal, à double foyers, (« The Bifocal Display ») [Spence82] combine dans un même espace d'affichage une vue détaillée encadrée de deux vues déformées, la déformation s'effectuant uniquement selon la dimension horizontale, comme schématisé à la Figure 5.5. Les objets qui se trouvent de chaque côté de la vue détaillée sont compressés uniformément dans la direction horizontale.

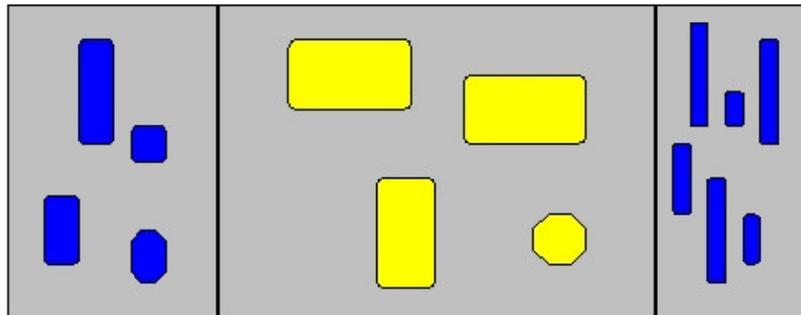


Figure 5.5 : Principe de l'affichage bifocal.

La Figure 5.6a représente une fonction de transformation et la Figure 5.6b montre la fonction de changement d'échelle correspondante. La Figure 5.6c montre quant à elle la déformation horizontale appliquée à une grille carrée. Cette technique a l'avantage d'être relativement simple à implémenter et de fournir une continuité spatiale entre les régions. Elle a toutefois le désavantage d'entraîner une discontinuité du facteur d'échelle sur les bords entre la vue détaillée et les vues déformées. Leung [Leung89] étend ce concept à un espace bidimensionnel dans une implémentation de la carte du métro londonien. La Figure 5.6d illustre les effets de la déformation appliquée aux axes vertical et horizontal. L'espace d'affichage est divisé en neuf régions (Figure 5.7). La région centrale correspond au focus et est mise en valeur. Les huit autres régions (le contexte) sont réduites afin de compenser l'espace alloué au focus. Le même facteur de réduction est utilisé dans les directions x et y. Enfin, il est intéressant de remarquer que les quatre régions se trouvant aux coins de la représentation ne sont pas déformées. Elles sont simplement réduites en taille, le changement d'échelle s'effectuant à la fois selon l'axe des x et l'axe des y.

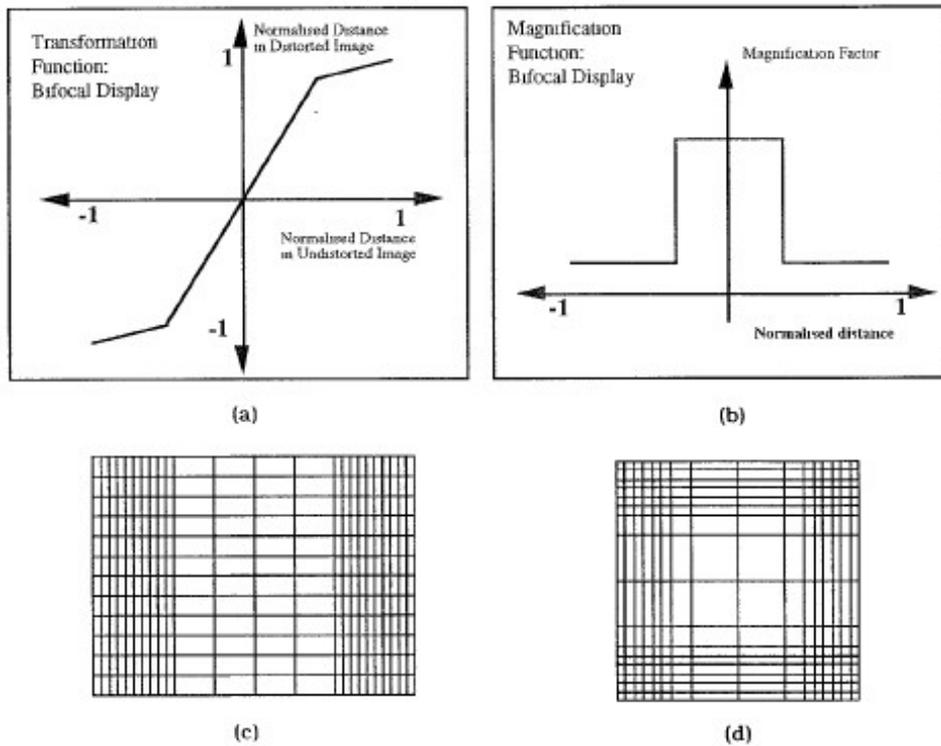


Figure 5.6 : L'affichage bifocal.

- (a) une fonction de transformation.
- (b) la fonction de changement d'échelle correspondante.
- (c) application de la technique à une dimension.
- (d) application de la technique à deux dimensions.

Demagnification in both X and Y dimensions	Demagnification in Y dimension	Demagnification in both X and Y dimensions
Demagnification in X dimension	Central 'Focus' Region no demagnification	Demagnification in X dimension
Demagnification in both X and Y dimensions	Demagnification in Y dimension	Demagnification in both X and Y dimensions

Figure 5.7 : Implémentation d'un affichage bifocal 2D.

5.2.1.2 Les projections poly-focales

La technique des projections poly-focales, proposée par Kadmon et Shlomi [Kadmon78] pour la représentation de données statistiques sur des cartes géographiques, fournit un cadre, aussi bien conceptuel que mathématique, à de nombreuses techniques de distorsions (et ce malgré le fait qu'elle n'ait pas été appliquée à une situation de type interactive).

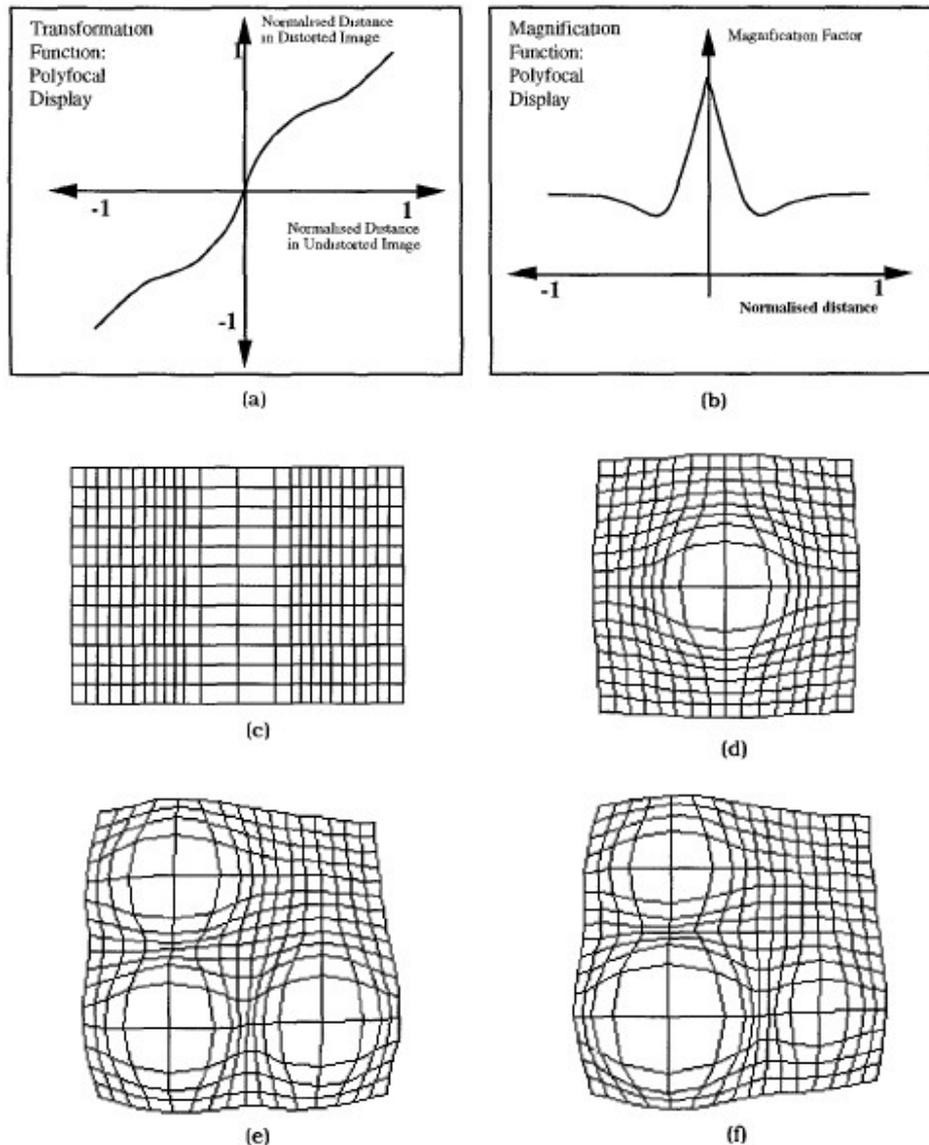


Figure 5.8 : Projection poly-focale

(a) une fonction de transformation typique. (b) fonction de changement d'échelle correspondante. (c) application de la projection à une dimension. (d) application de la projection à deux dimensions. (e) vue multi-focus de la projection avec les mêmes paramètres pour chaque focus. (f) vue multi-focus avec différents paramètres.

Le concept fondamental sur lequel repose une projection poly-focale (dans sa forme mono-dimensionnelle) peut être illustré par les fonctions de transformation et d'agrandissement présentées à la Figure 5.8 (a et b). Le plus haut sommet de la Figure 5.8b correspond au point de focus de l'affichage. La courbe de la fonction d'agrandissement dépend de deux ensembles de paramètres. Le premier contrôle l'agrandissement du point de focus et le second la modification du facteur d'agrandissement en fonction de la distance au point de focus. Les Figures Figure 5.8c et Figure 5.8d montrent les effets de cette technique appliquée à une et deux dimensions. Cette technique permet également l'affichage de plusieurs focus, chacun d'eux correspondant à un sommet de la fonction d'agrandissement (à un point « étiré » de la surface). Théoriquement, il n'y a aucune limitation sur le nombre de sommets. Cependant, ce nombre influe sur la durée du temps de calcul et sur la compréhensibilité de la représentation résultante. Les Figures Figure 5.8e et Figure 5.8f donnent des exemples d'affichages multi-focus, la première applique les mêmes paramètres à chaque focus, la seconde utilise des valeurs différentes. Cette technique peut être étendue, par exemple, en limitant les déformations à une ou plusieurs parties de l'affichage, ou en appliquant une fonction de réduction à la place de la fonction d'agrandissement, ou encore en combinant ces diverses possibilités.

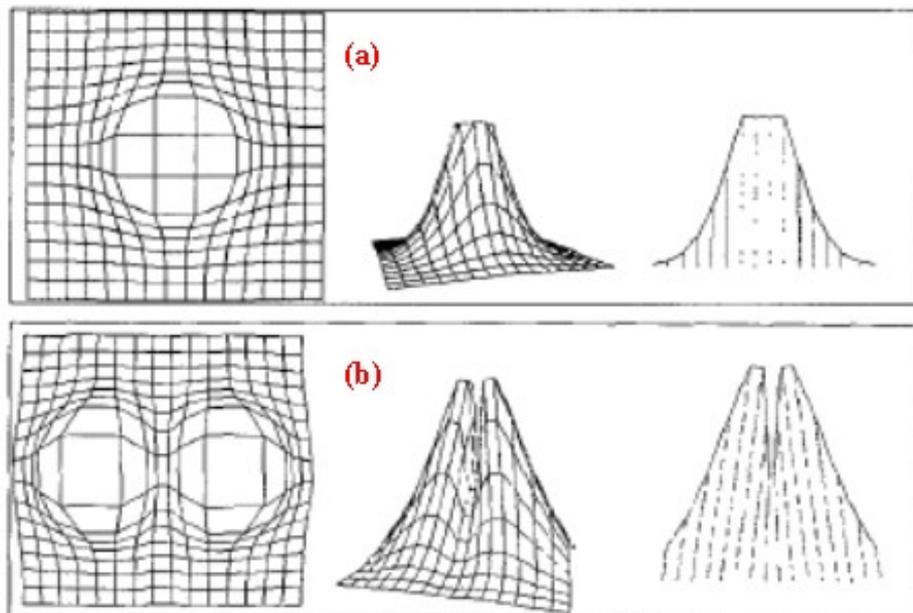


Figure 5.9 : Distorsions 3D de surfaces planes.

Dans le même ordre d'idée, Carpendale et al. [Carpendale95] proposent une technique qui repose sur l'utilisation de courbes gaussiennes pour transformer une surface plane 2D en une surface courbe 3D (métaphore du pliage). L'affichage correspond à une vue du dessus de la surface 3D résultante de la déformation, comme illustré à la Figure 5.9a. L'utilisateur peut modifier interactivement la

distorsion appliquée à la surface plane en « tirant » sur un point de la surface afin de l'agrandir (ce qui a pour effet d'ajouter une courbe gaussienne à la distorsion). L'apparence globale d'un affichage multi-focus peut être vue comme un paysage curviligne (Figure 5.9b).

Des extensions de ce travail prennent en compte d'autres facteurs comme l'ombrage, la perspective 3D. Enfin, cette technique a également été appliquée à la déformation de formes 3D [Carpendale97], comme illustré à la Figure 5.10.

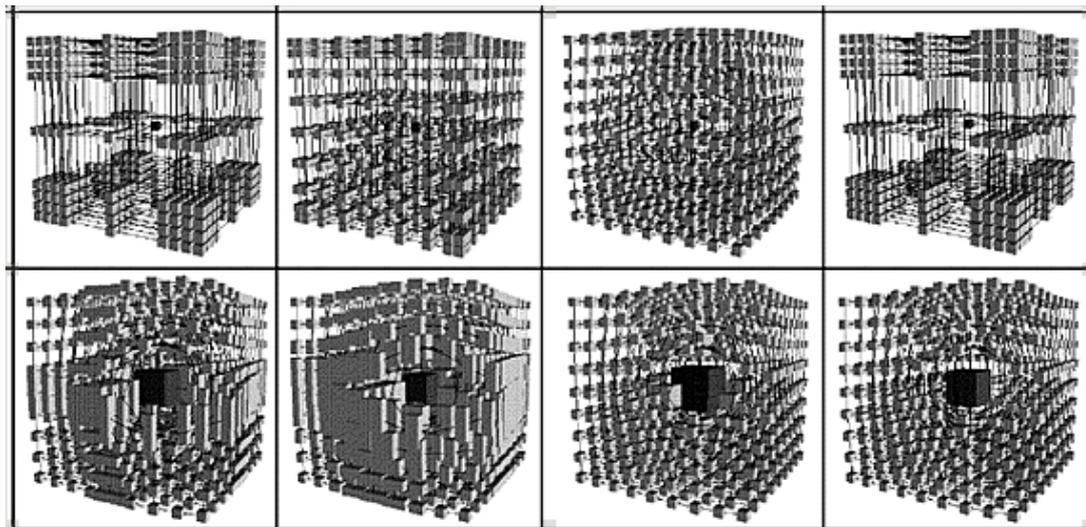


Figure 5.10 : Distorsions de formes 3D.

5.2.1.3 Les vues Fisheyes

La technique des vues fisheyes [Furnas86] est une des approches proposées les connues pour afficher dans une même vue l'information à plusieurs niveaux d'abstraction, en équilibrant détail local et contexte global. Plus formellement, une transformation fisheye repose sur une représentation structurée de l'espace d'information caractérisée par les deux conditions suivantes :

- ? chaque élément possède une importance intrinsèque appelée importance a priori IAP ,
- ? chaque couple d'éléments est séparée par une distance D qu'il est possible de calculer.

La transformation fisheye s'appuie alors sur un élément particulier, le point focal PF , et sur une fonction DI (le degré d'intérêt) afin de déterminer l'importance de chaque élément par rapport à ce focus. La formulation de la fonction DI proposée par Furnas est définie de la façon suivante :

« Soit x un élément quelconque de la structure P et y le focus de la vue fisheye, alors la fonction DI donnant le degré d'intérêt de x est définie par $DI(x, y) = IAP(x) \cdot D(x, y)$ »

La stratégie consiste à calculer le degré d'intérêt de chaque point de la représentation initiale et d'afficher tous ceux dont le degré d'intérêt est inférieur à un seuil donné.



Figure 5.11 : Vue initiale (issue de Sarkar et Brown (1992)).

Ce cadre général des transformations fisheye proposé par Furnas s'avère toutefois assez restreint. En exemple, on remarque qu'il ne prend pas en compte la taille (au sens graphique) ni la position des éléments de la vue fisheye par rapport à la vue initiale. Les systèmes qui ont recours aux vues fisheye peuvent être regroupés en deux catégories en fonction de la technique de transformation utilisée (filtrage ou déformation). La première stratégie utilise le seuillage pour restreindre l'affichage de l'information aux éléments pertinents ou intéressants et reste donc fidèle à la conception originale de Furnas. La seconde repose sur la métaphore de la lentille. Elle consiste à appliquer des déformations géométriques à la représentation afin d'obtenir des effets fisheye visuels. Au lieu de filtrer les éléments, cette stratégie utilise des déformations pour en altérer la position, la forme et la taille. Par exemple, Sarkar et Brown [Sarkar92] utilisent cette stratégie pour définir des vues fisheye de graphes, comme illustré par les Figures Figure 5.11 et Figure 5.12. Les éléments de la présentation ne sont pas seulement absents ou présents, ils ont une taille, une position et un niveau de détail.

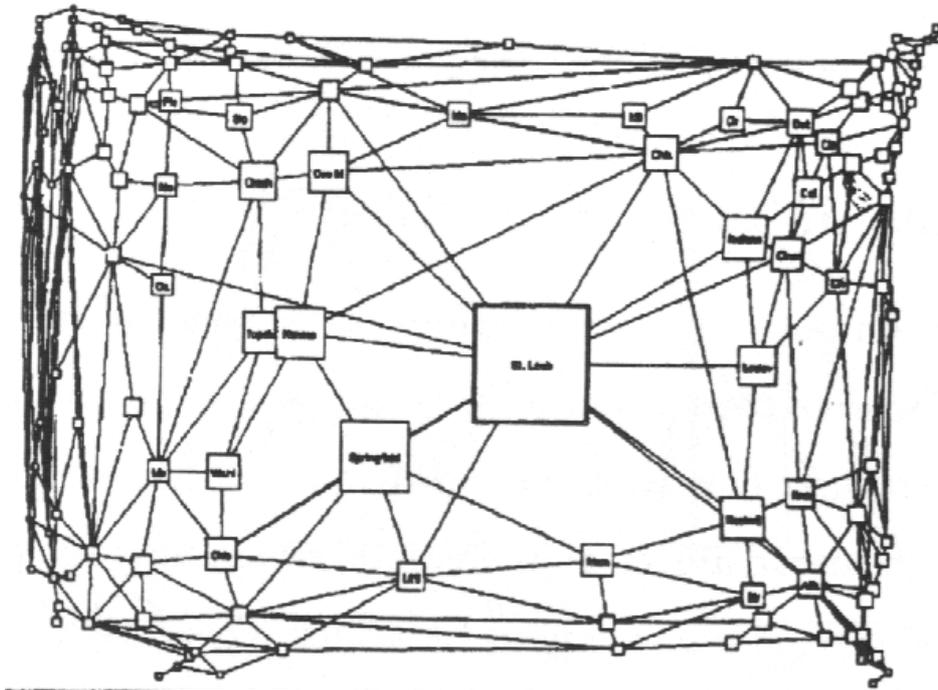


Figure 5.12 : Vue fisheye (issue de Sarkar et Brown (1992)).

La technique des vues fisheye est largement utilisée et peut s'appliquer à de nombreux cas. Elle n'est cependant pas dépourvue d'inconvénients. Le filtrage peut éliminer du contexte des éléments intéressants ou rendre difficile la compréhension des relations associant les divers éléments de l'espace informationnel. Les déformations peuvent entraîner des transformations importantes de l'affichage difficiles à comprendre et à interpréter, entraînant une surcharge cognitive pour les utilisateurs.

5.2.1.4 Le tableau optique

Le tableau optique (« The Table Lens ») [Rao94] propose une méthode pour visualiser, manipuler et donner du sens à des tableaux de grande taille. En fonction de son intérêt, l'utilisateur peut augmenter le niveau de détails de certaines cellules en appliquant des distorsions à la représentation. Il en résulte une représentation focus+contexte qui donne accès au contenu de certaines cellules tout en affichant l'ensemble du tableau. L'utilisateur peut focaliser son attention simultanément sur plusieurs parties du tableau, comme illustré à la Figure 5.13. Cette possibilité permet de maintenir le focus sur une zone d'intérêt tout en explorant d'autres portions du tableau, ce qui facilite la comparaison de régions distantes.

Figure 5.13 : Le tableau optique.

Les distorsions appliquées à la représentation s'effectuent soit selon l'axe horizontal soit selon l'axe vertical, et modifient en conséquence la géométrie des cellules. L'espace réservé à l'affichage des cellules est alloué indépendamment le long de chaque axe. La taille (hauteur ou largeur) des cellules qui appartiennent à une zone de focus augmente. Quant aux autres cellules (le contexte), elles se répartissent équitablement l'espace restant afin de rester visibles. Il résulte de ces distorsions plusieurs types de cellules. Les cellules « focales » qui appartiennent à une zone de focus le long des deux axes. Les cellules « semi-focales » qui appartiennent à une zone de focus seulement sur un des deux axes et les cellules « non-focales » (toutes les autres) qui font partie du contexte le long des deux axes.

L'utilisateur dispose de plusieurs possibilités pour contrôler une zone de focus :

- ? changer la taille d'une zone de focus sans modifier le nombre de cellules contenues dans cette zone,
- ? ajouter ou enlever des cellules dans une zone de focus sans modifier sa taille,
- ? ajouter ou enlever des cellules dans une zone de focus sans modifier la taille des cellules déjà présentes dans cette zone,
- ? déplacer une zone de focus.

5.2.2 Zoom

Le zoom est une méthode couramment utilisée dans le domaine de la visualisation d'information pour modifier la vision de l'utilisateur sur l'espace de représentation. Selon Furnas et Bederson [Furnas95] deux types de zoom sont à différencier.

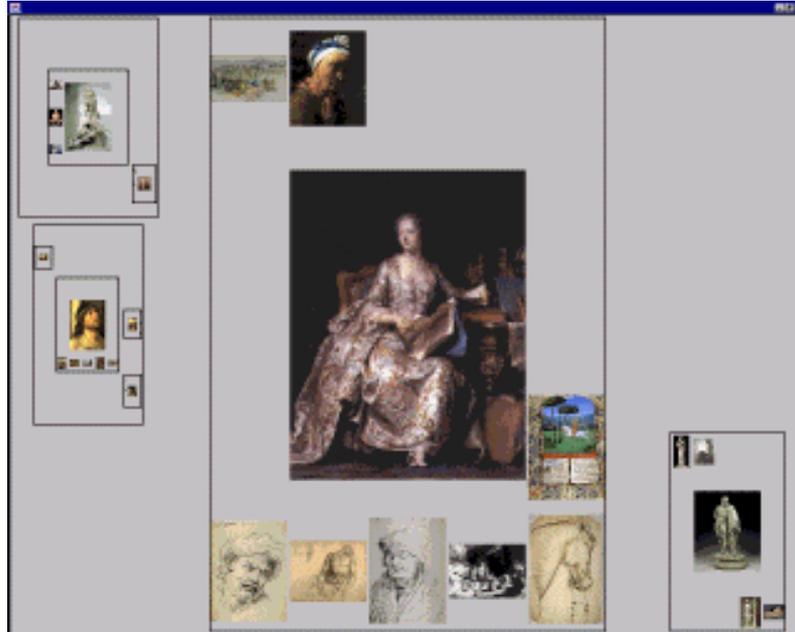


Figure 5.14 : Exemple de zoom géométrique appliqué à l'exploration d'une base d'images hiérarchique (Holmquist 1998).

Le *zoom géométrique* permet de changer la taille (agrandir ou réduire) d'un élément (ou d'une partie) de l'espace de visualisation, ce qui peut être réalisé de nombreuses manières, en allant du simple changement d'échelle dans un mode de représentation bitmap, à la composition d'une translation et d'une homothétie dans un mode vectoriel. Cette technique sert, par exemple, à afficher simultanément un ensemble de documents sous forme d'images réduites avec lesquelles l'utilisateur pourra directement interagir (Figure 5.14). Même si leur taille ne permet pas toujours de visualiser en détails leur contenu, ces images fournissent toutefois à l'utilisateur des repères visuels (les couleurs, les relations spatiales des éléments qui composent le document, etc.) qui en donnent une idée générale. Ces repères peuvent également aider à identifier des documents préalablement consultés. De telles représentations facilitent généralement la comparaison de documents et la consultation en contexte [Ginsburg96, Holmquist98, Robertson98, Björk00].

le *zoom sémantique* permet de modifier le niveau de détails et la représentation des éléments en fonction de leur taille. Le zoom entraîne l'apparition de nouvelles informations en plus ou à la place de la transformation géométrique. Plus la taille

d'un élément augmente, plus le nombre de détails augmente. Dans un système qui implémente le zoom sémantique, la taille des objets affichés dépend souvent de la distance qui les sépare de l'utilisateur. Ainsi, plus l'utilisateur se rapproche d'un objet, plus il visualise de détails sur cet objet. Par exemple, un document représenté par un point à une distance éloignée de l'utilisateur, peut se transformer, lors du zoom, en un rectangle plein contenant tout d'abord un identifiant, puis un titre, un résumé, puis se transformer en une page et enfin en le document lui-même, comme schématisé à la Figure 5.15. La principale difficulté technique du zoom sémantique ne concerne pas l'opération de zoom elle-même, mais plutôt l'assignation des niveaux de détails aux éléments de la représentation.

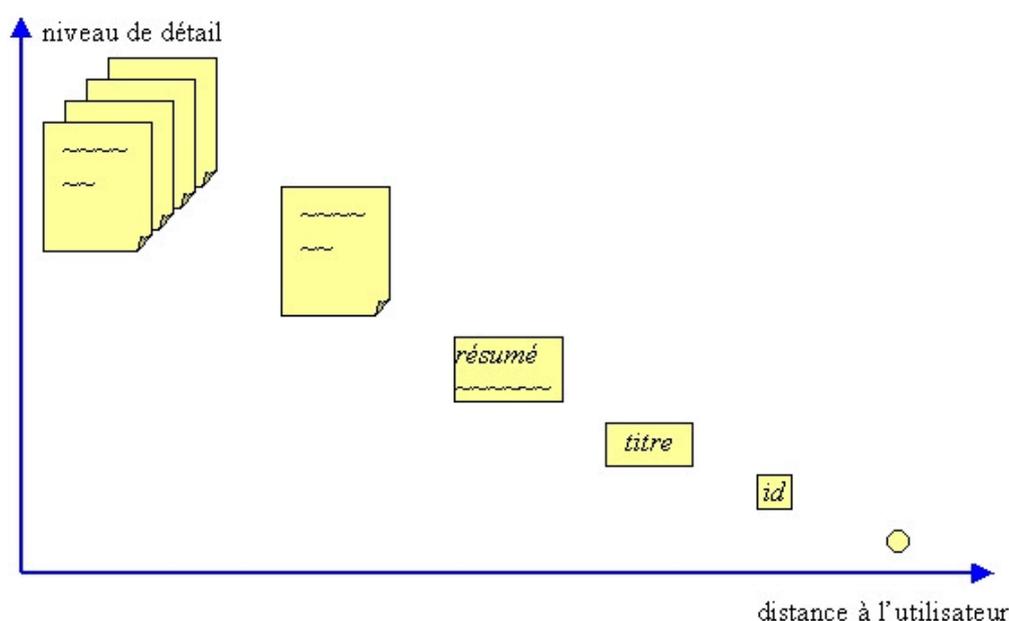


Figure 5.15 : Exemple schématique de zoom sémantique.

Les interfaces qui permettent de présenter et d'explorer des informations à différents niveaux de zoom, géométrique et/ou sémantique, portent le nom de ZUI (Zoomable User Interface).

5.2.2.1 Space-scale Diagrams : représentation des ZUIs

Furnas et Bederson introduisent le concept des « space-scale diagrams » afin de fournir un cadre d'analyse et de conception aux ZUIs [Furnas95]. Ces diagrammes font du facteur d'échelle une dimension explicite de la représentation pour que son rôle soit mis en évidence et plus simple à analyser. L'idée consiste à définir un espace abstrait en créant plusieurs copies de la représentation spatiale originale, une pour chaque niveau d'échelle, et de les empiler pour former une pyramide inversée.

La Figure 5.16b illustre ce principe de construction appliqué à une image bidimensionnelle schématisée par la Figure 5.16a. Les axes horizontaux représentent les dimensions spatiales de l'image et l'axe vertical le niveau d'agrandissement.

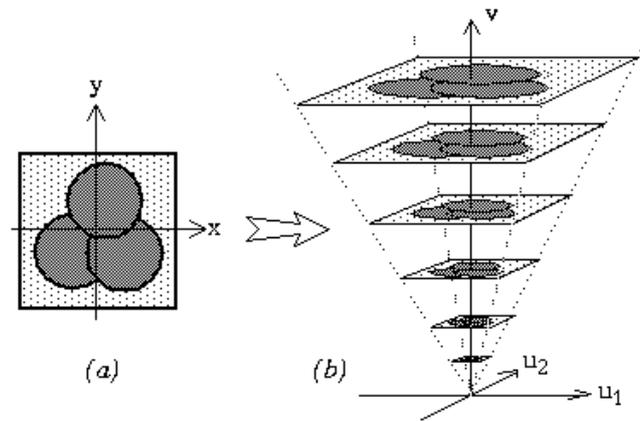


Figure 5.16 : Construction d'un « space-scale diagram » à partir d'une image 2D.

La fenêtre de visualisation est représentée comme un rectangle de taille fixe, parallèle au plan (u_1, u_2) , qui peut être déplacé dans l'espace 3D du diagramme (Figure 5.17). Ces déplacements permettent de décrire toutes les combinaisons de déplacements latéraux et de zoom (les opérations d'exploration) effectuées par l'utilisateur sur la surface 2D originale.

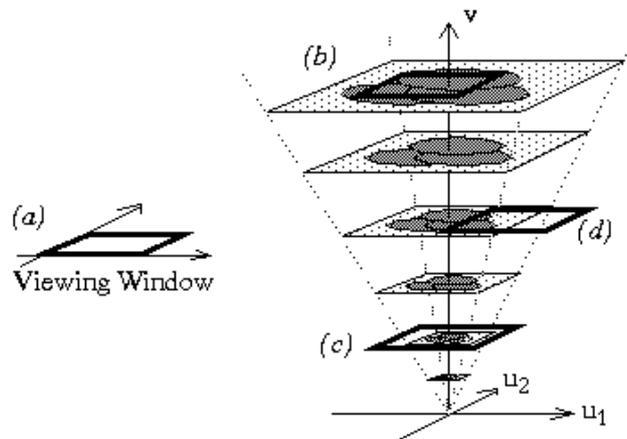


Figure 5.17 : Représentation de la fenêtre de visualisation utilisateur dans un « space-scale diagram ».

La Figure 5.18 donne (a) la description d'un déplacement latéral, (b) d'un zoom et (c) d'une combinaison de ces deux opérations. Ce type de description permet

d'analyser les trajectoires engendrées par ces opérations et ainsi de faciliter l'interaction nécessaire pour atteindre un point donné. En effet, dans une interface zoomable, le chemin le plus simple pour aller d'un point à un autre n'est pas toujours une ligne droite. Il est par exemple possible de réaliser un zoom arrière à partir du point d'origine jusqu'à avoir une vue d'ensemble qui montre les extrémités du parcours, puis de zoomer vers la destination.

Ce principe de représentation permet également de décrire le zoom sémantique. Au lieu d'empiler la même image dans la pyramide, son contenu dépend du facteur d'échelle, un zoom (i.e. un déplacement selon l'axe vertical) pouvant alors entraîner une modification sémantique des données affichées.

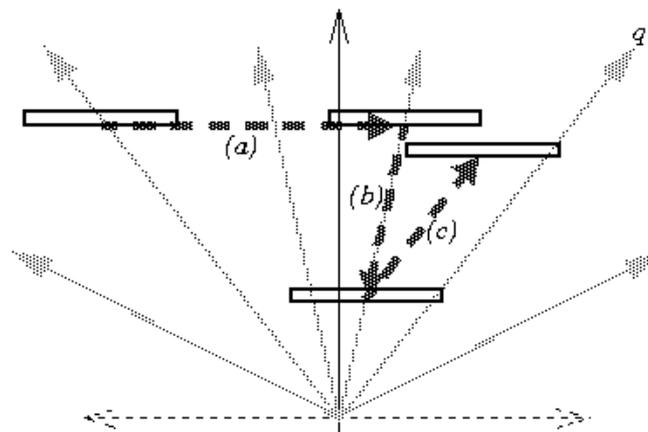


Figure 5.18 : Trajectoires d'exploration dans une interface zoomable.

5.2.2.2 Les interfaces zoomables d'exploration

Les ZUIs dédiées à l'exploration interactive d'espaces informationnels implémentent généralement le zoom sémantique. Les systèmes Pad++ [Bederson94, Bederson96] et Zomit [Pook00a, Pook00b] sont de bons exemples de telles ZUIs. L'accès aux données s'effectue par une succession de zooms et de défilements latéraux qui modifient l'ensemble de la représentation. Lorsqu'un zoom est appliqué, de nouveaux objets apparaissent et le détail de ceux déjà présents augmente. Les objets de grande taille passent derrière le point de vue de l'utilisateur, qui est en quelque sorte « immergé » dans la scène, et disparaissent. Le système Pad++ a entre autre été appliqué à l'exploration d'un système de gestion de fichiers (Figure 5.19) et à la représentation d'historiques de navigation Web [Hightower98].

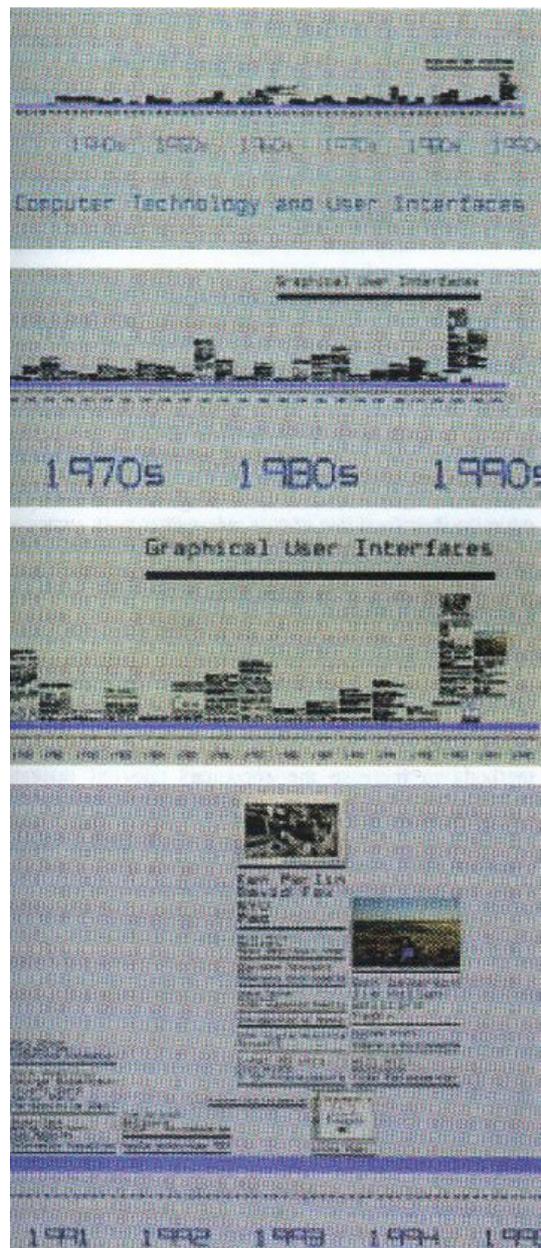


Figure 5.19 : Système Pad++, exploration d'une base documentaire.

Le système Zomit a quant-à-lui été appliqué à l'exploration d'une base de données génétiques (et d'une bibliothèque virtuelle), dont une illustration est donnée à la Figure 5.20. L'image de gauche correspond à la vue initiale de la représentation. L'utilisateur dispose alors d'une vue générale de l'ensemble des éléments de la base (les chromosomes) qui lui sert de point de départ à l'exploration. Une succession de zooms sur une zone particulière de la surface d'affichage permet d'obtenir de nouvelles informations sur les éléments de cette zone. Les images de droite et du dessous de la Figure 5.20 montrent deux vues qui résultent d'un zoom sémantique.

Le détail des chromosomes qui se trouvent dans la zone de focus apparaît au fur et à mesure du zoom. Afin d'éviter à l'utilisateur d'avoir à revenir à la vue initiale pour sélectionner une nouvelle zone de focus, ce dernier peut à tout moment se déplacer latéralement sur la surface de visualisation au moyen de la souris. Les objets gardent alors le même niveau de grossissement.

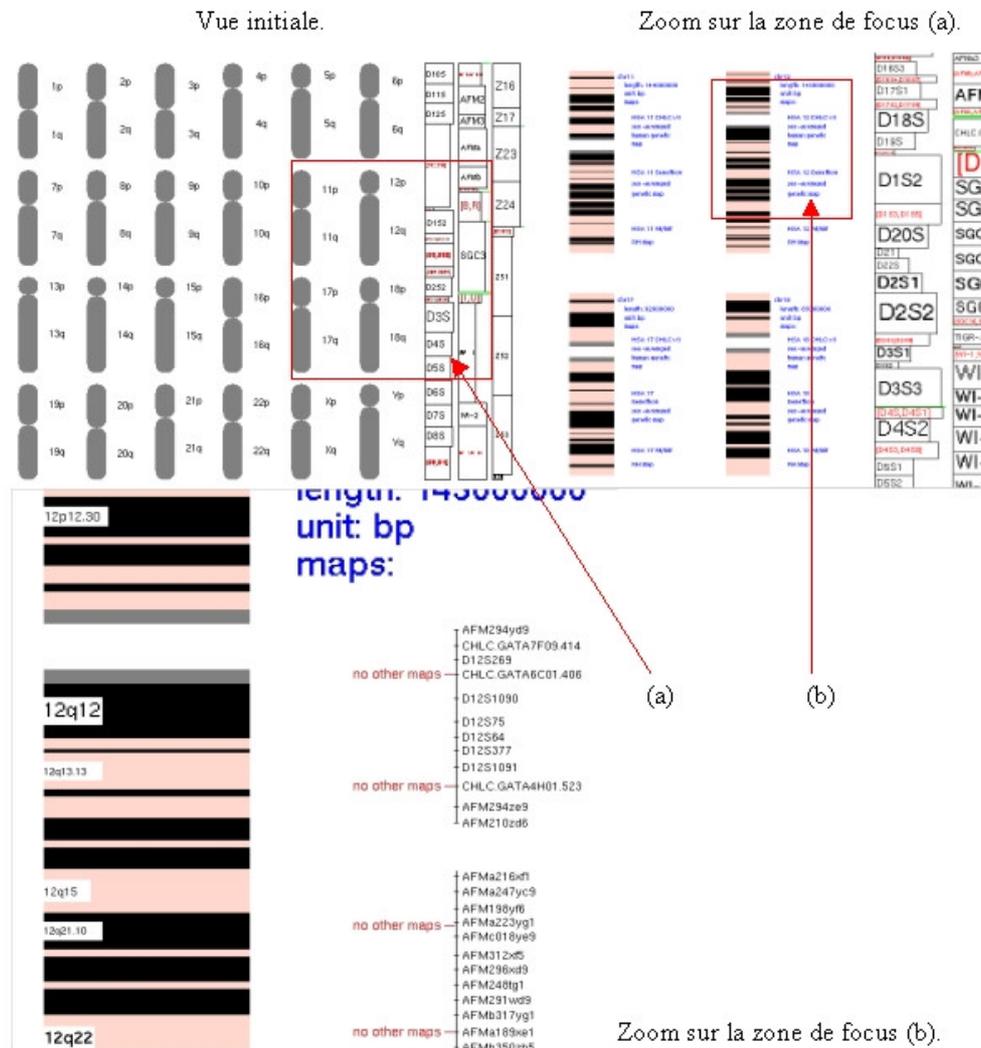


Figure 5.20 : Système Zomit, exploration d'une base de données génétiques.

Les utilisateurs de tels systèmes ont cependant souvent des difficultés à comprendre la structure de l'espace exploré ou à identifier et trouver un élément particulier. Les relations entre les divers éléments ne sont pas clairement définies, ce qui ne facilite pas la compréhension de leur positionnement respectif (comment et pourquoi tel objet est-il positionné par rapport à tel autre ?). Après plusieurs opérations de zoom, il peut être difficile pour l'utilisateur de savoir précisément à quel endroit il se trouve et comment faire pour atteindre rapidement une autre région de l'espace. De même,

la comparaison entre objets distants ne peut pas s'effectuer simplement (ce qui ne favorise pas les associations d'idées). Ces difficultés s'expliquent principalement par un manque de contexte. Enfin le zoom est global et non borné. Il s'applique à l'ensemble de l'espace documentaire, ce qui engendre de fréquentes opérations de mise à jour.

5.3 Autres paradigmes d'interaction

Mis à part les distorsions et les zooms, d'autres techniques d'interaction sont apparues afin de faciliter l'exploration d'espaces informationnels et la manipulation des données affichées. Parmi les plus intéressantes, on trouve les techniques du « filtrage dynamique » [Shneiderman94], des « lentilles magiques » [Bier93, Stone94] et des « excentric labels » [Fekete99]. Le « filtrage dynamique » permet de contrôler l'affichage des données au moyen de requêtes définies par manipulation directe. Une lentille magique est un interacteur que l'utilisateur positionne sur l'espace d'affichage pour modifier la partie qu'il recouvre ou pour effectuer un traitement sur les éléments de cette partie. La technique des « excentric labelling » fournit un moyen pour identifier ou obtenir des informations (de contexte, d'aide, etc.) sur les objets de la représentation.

5.3.1 Filtrage dynamique

La technique du filtrage dynamique [Shneiderman94] fournit une alternative visuelle aux mécanismes standard d'interrogation de bases de données. L'utilisateur définit les paramètres d'une requête de manière interactive au moyen de barres de défilements et de boutons de sélections (ce qui lui évite d'avoir à connaître et à utiliser un langage de requête spécifique tel que SQL). Ces interacteurs lui donnent un contrôle rapide, incrémental et réversible de la requête. Les résultats de la recherche sont donnés sous la forme d'une représentation visuelle. Cette représentation est dynamiquement mise à jour en fonction des actions de l'utilisateur sur les contrôles de l'interface, celui-ci modifiant ou affinant sa recherche selon ses besoins.

Le HomeFinder [Williamson92] et le FilmFinder [Ahlberg94b] sont deux des systèmes les plus connus qui implémentent la technique du filtrage dynamique. Le système HomeFinder est utilisé dans le cadre d'une base de données immobilière de Washington (Figure 5.21). Il permet aux utilisateurs d'identifier des habitations à vendre en fonction de critères comme le prix, le nombre de chambres et la distance au lieu de travail. Comme montré à la Figure 5.21, les barres de défilement utilisées pour paramétrer les requêtes permettent de définir des intervalles de valeurs. Chaque habitation qui répond aux critères de la sélection apparaît à l'écran sur une carte de Washington et sous la forme d'un point. Cette représentation graphique des données

offre un avantage quant à l'interprétation des résultats d'une recherche (identification des zones d'habitations, etc.).

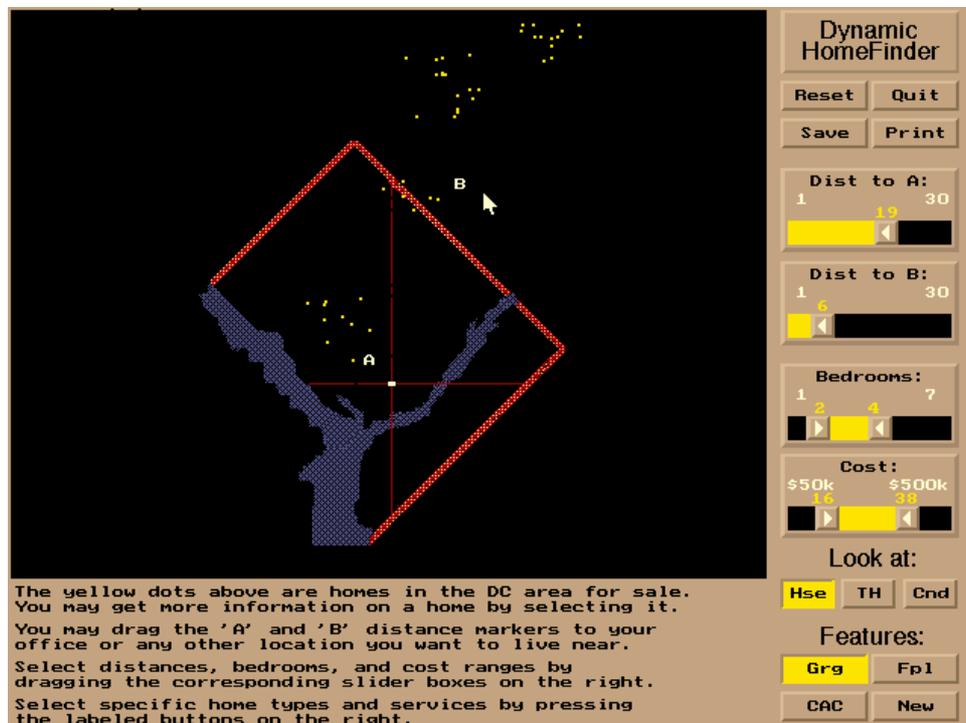


Figure 5.21 : Système HomeFinder.

Le système FilmFinder est conçu pour l'exploration d'une base de données cinématographiques. Il permet d'effectuer des recherches sur le titre des films, le nom des acteurs, la durée, etc. Certains critères comme la catégorie d'appartenance d'un film (comédie, aventure, etc.) sont à définir au moyen de boutons de sélection (Figure 5.22). Les résultats d'une requête correspondent à un ensemble de films affichés à l'écran sous la forme de rectangles dont la couleur identifie leur catégorie. Ces rectangles sont positionnés dans un plan selon leur mesure de popularité (sur l'axe des ordonnées) et leur date de sortie en salle (sur l'axe des abscisses). Ces deux axes sont paramétrables, il est ainsi possible de réduire l'affichage à des intervalles spécifiques. Seuls les éléments qui correspondent aux critères de la recherche et qui se trouvent dans les intervalles des deux axes sont montrés à l'utilisateur. Moins le nombre de ces éléments est important, plus leur taille augmente. Lorsque l'affichage comporte moins de 25 éléments, le titre du film apparaît automatiquement. L'utilisateur peut également cliquer sur un film pour obtenir plus d'information (Figure 5.22).

La technique du filtrage dynamique offre plusieurs avantages. Elle permet de passer simplement du global (l'affichage de l'ensemble des données) au local (le résultat d'une requête), et ainsi d'obtenir assez rapidement une idée du contenu de la base

explorée. La représentation graphique permet aux utilisateurs d'interpréter en contexte les résultats affichés (contrairement aux systèmes de requêtes traditionnels qui présentent souvent les résultats sous la forme d'une liste textuelle). Elle les aide également à comprendre les relations entre les données. L'utilisabilité de cette technique est toutefois contrainte par les temps de réponses du système, tant du point de vue de la résolution des requêtes que de l'affichage, ce qui nécessite un support logiciel et matériel non disponibles dans tous les environnements de gestion de base de données. Un autre inconvénient de cette technique est qu'elle est dépendante du domaine d'application. Elle nécessite la définition de critères de recherches particuliers et l'implémentation de représentations graphiques adaptées. Enfin, elle ne fournit pas une puissance d'expression équivalente à celle des langages de requêtes, souvent nécessaire pour l'interrogation de bases de données complexes.

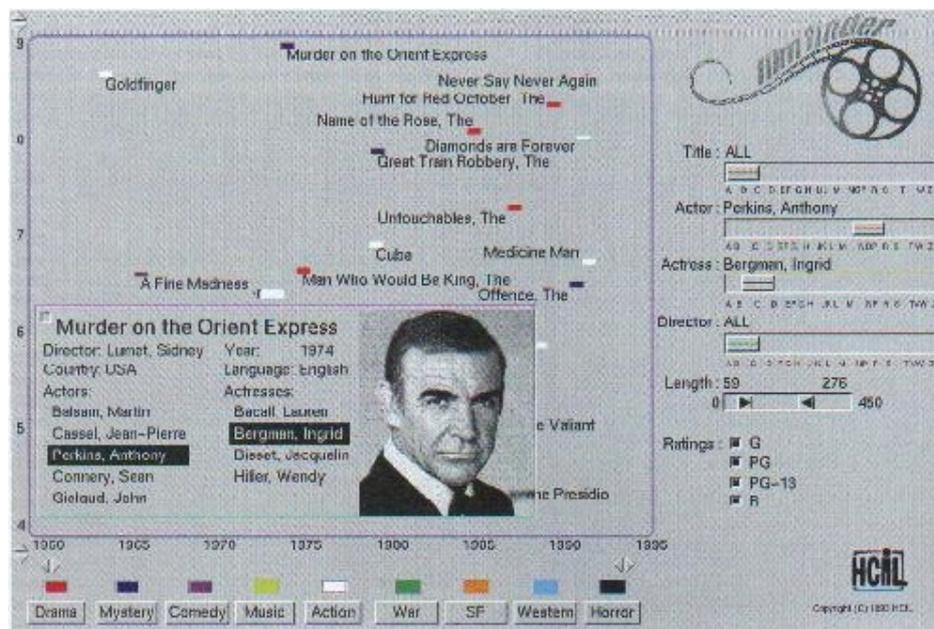


Figure 5.22 : Système FilmFinder.

La technique du filtrage dynamique peut être adaptée à l'exploration de données structurées. Elle a déjà été appliquée au cas des structures hiérarchiques [Kumar97]. Elle pourrait être étendue au cas d'hyperstructures et appliquée à la visualisation de documents hypermédias.

5.3.2 Lentilles magiques

Les lentilles magiques (« Magic Lenses / Toolglass Widgets ») [Bier93, Bier94, Stone94] sont des objets graphiques de forme quelconque (souvent rectangulaire) qui permettent soit d'appliquer une transformation visuelle soit d'effectuer un traitement

informatique sur les objets (ou les parties de l’affichage) qu’ils recouvrent. Elles peuvent être déplacées par manipulation directe au moyen de la souris via une technique classique de « drag and drop » (de la même manière qu’une loupe grossissante peut être déplacée, par exemple, sur la surface d’un journal). Une lentille magique peut servir à changer l’aspect d’un objet (la couleur, le contenu, etc.), à faire apparaître des informations sémantiques, à détailler une partie de l’affichage (comme illustré à la Figure 5.23b), à appliquer des requêtes booléennes [Eick94] ou à effectuer des calculs sur un ensemble de données (comme la somme des éléments d’une colonne de nombres dans un tableau). Dans l’exemple présenté à la Figure 5.23a, la lentille magique affiche la définition du mot sélectionné. La Figure 5.23c présente une lentille magique que l’utilisateur peut déplacer sur une carte du métro parisien. Lorsque la lentille passe sur un icône représentant un site touristique, elle affiche automatiquement la photographie correspondante.

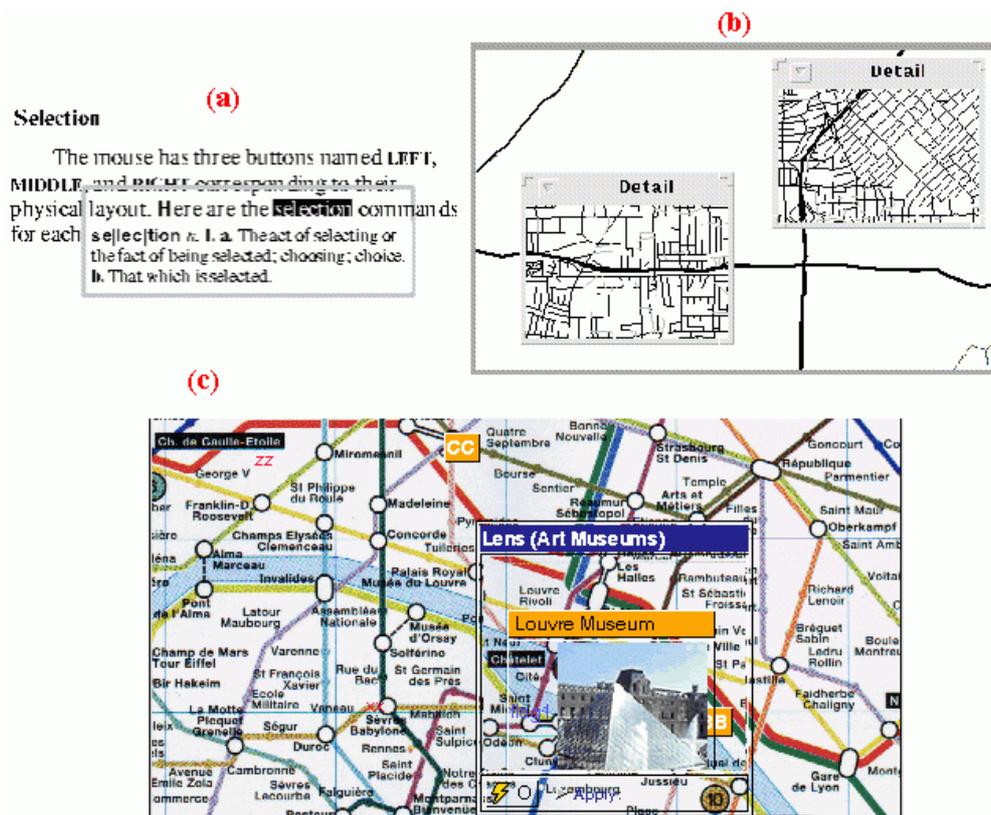


Figure 5.23 : Exemples de lentilles magiques.

Les transformations visuelles que les lentilles magiques réalisent sont locales. Le contexte des parties de l’affichage ou des objets auxquels elles s’appliquent est préservé, fournissant ainsi une vue focus+contexte (ce qui est d’autant plus marqué si la lentille est transparente).

Lorsque des lentilles magiques se chevauchent, leurs effets se combinent dans leur zone d'intersection. Les opérations sont appliquées de manière séquentielle, du bas vers le haut. La Figure 5.24 illustre ce procédé. Elle montre une carte sur laquelle se trouvent deux lentilles magiques. La première met en évidence les cours d'eau, la seconde les routes principales. La partie de la carte où elles se chevauchent présente ces deux caractéristiques.

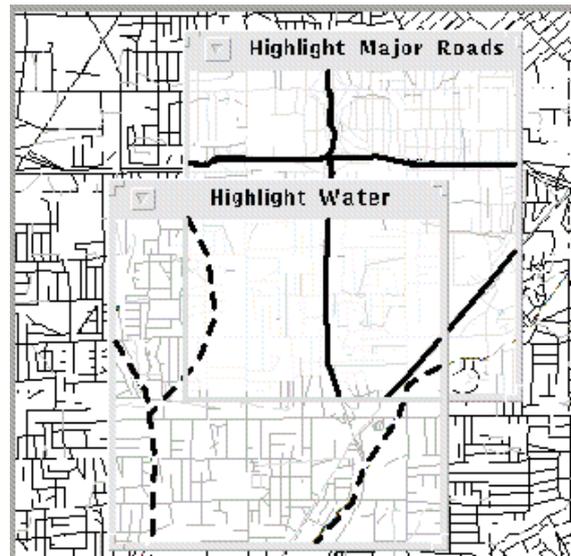


Figure 5.24 : Composition de deux lentilles magiques.

Les lentilles magiques peuvent être employées comme des dispositifs d'interaction « click-through ». Elles sont alors appelées des « toolglass widgets » et sont transparentes [Bier94]. La fonction associée à une telle lentille est appliquée à un objet lorsque l'utilisateur clique avec le curseur sur cet objet à travers la lentille. Dans l'exemple de la Figure 5.25, la fonction associée aux lentilles permet de changer la couleur des objets. L'utilisateur fait passer la couleur de l'élément se trouvant sous le curseur au vert en cliquant à travers la lentille dont le coin supérieur gauche est vert.

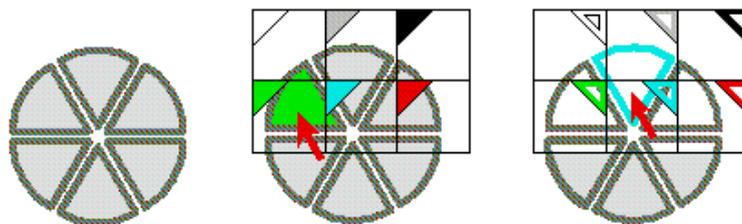


Figure 5.25 : Lentilles magiques « click-through » de changement de couleur.

Les lentilles magiques ont également été adaptées afin d'être utilisées dans des environnements 3D. Viega propose deux types d'implémentations présentées comme des artefacts de visualisation [Viega96]. La « lentille plate » est une surface plane qui applique l'opération associée à tous les objets de l'espace qui se trouve derrière elle par rapport au point de vue de l'utilisateur. La « lentille volumétrique » est un objet 3D (par exemple un cube) qui limite les effets de l'opération à l'intérieur de cet objet, comme illustré à la Figure 5.26.

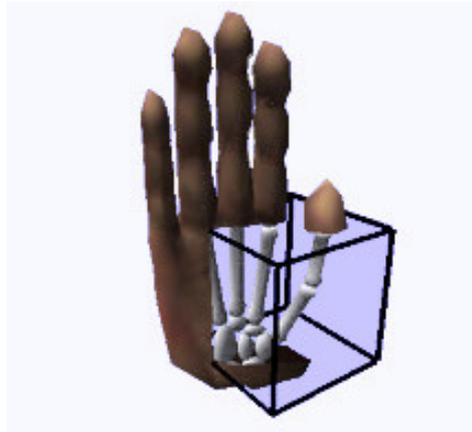


Figure 5.26 : Lentille volumétrique dans un environnement 3D.

Les lentilles magiques fournissent un paradigme d'interaction original, simple à comprendre car basé sur l'expérience des lentilles physiques (loupes, filtres photographiques, etc.), qui remplace le mode temporel des interfaces traditionnelles par un mode spatial. Les utilisateurs interagissent sur les objets de l'interface non plus en sélectionnant les actions, par exemple, dans des menus mais en déplaçant des lentilles sur ces objets.

5.3.3 Excentric labeling

Cette technique [Fekete99] offre un moyen rapide d'identifier un ensemble d'objets affichés à l'écran au moyen de labels textuels (des « tool tips »). Le principe consiste à montrer les labels de tous les objets se trouvant dans le voisinage du curseur (la zone de focus), comme illustré à la Figure 5.27. Ce voisinage correspond à un cercle rayon prédéfini centré sur la position du curseur. Les labels sont affichés sans chevauchement et alignés afin de faciliter la lecture rapide du texte qu'ils contiennent. Ils apparaissent lorsque le curseur reste immobile plus d'une seconde. Ils restent alors visibles et sont automatiquement mis à jour en fonction des déplacements du curseur. Ils disparaissent, soit lorsqu'un nouveau type d'interaction est enclenché (par exemple un clic souris), soit lorsque l'utilisateur déplace rapidement le curseur pour le faire sortir de la zone de focus. Les labels sont

positionnés de manière à être suffisamment éloignés du curseur afin de ne pas masquer les informations contenues dans la zone de focus. Ils sont encadrés par des rectangles de couleurs et connectés à leur objet au moyen de lignes de même couleur. Cette correspondance visuelle permet d'identifier les couples (label, objet) de manière non ambiguë. Le texte du label apparaît toujours en noir sur un fond blanc pour une meilleure lisibilité.

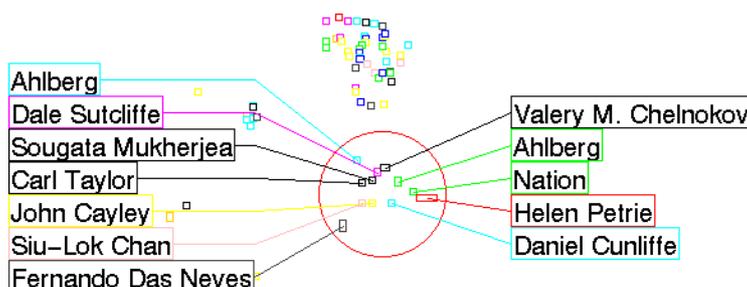


Figure 5.27 : Excentric labeling.

Cette technique propose un moyen efficace d'ajouter de l'information temporaire à une représentation. Cependant, elle peut fortement surcharger (et compliquer) l'affichage de représentations complexes qui font intervenir un grand nombre d'objets.

5.4 Structures linéaires et matricielles

Les séquences et les matrices sont des structures simples à représenter, à manipuler et à comprendre. Dans des environnements de type hypermédia, elles sont mises à profit de nombreuses manières. Elles sont utilisées pour présenter des visites guidées (section 4.1.3) [Halasz87, Marshall89], des indexes, des résultats de requêtes, des historiques de navigation, ou encore des bookmarks comme c'est par exemple le cas dans les navigateurs Web traditionnels ou dans des systèmes tels que Deckview [Ginsburg96], le WebBook et le Web Forager [Card96]. Le WebBook se base sur la métaphore du livre pour permettre aux utilisateurs de stocker et de parcourir des ensembles de pages Web de manière séquentielle. Le Web Forager permet de ranger des pages Web dans une matrice représentée à l'écran par les rayons d'une bibliothèque virtuelle (section 5.7.1). Les matrices fournissent également un moyen simple d'organiser des ensembles de documents. Une des deux dimensions (verticale ou horizontale) peut servir à regrouper les documents par catégories, l'autre à les classer selon un ordre séquentiel déterminé par un quelconque attribut (par exemple la date de dernière visite).

La manière de représenter une séquence ou une matrice influe toutefois sur la capacité des utilisateurs à interpréter les informations présentées. Il existe de

nombreuses méthodes pour représenter une séquence d'objets. La plus simple et la plus répandue consiste à afficher les objets les uns à la suite des autres dans un menu ou dans une fenêtre. Une telle fenêtre peut être munie d'une barre de défilement afin de permettre le parcours des listes de grande taille. Ce type de représentation s'avère toutefois limitée par le nombre d'éléments qu'il est possible de visualiser à la fois. L'accès à un élément éloigné dans la liste nécessite de faire défiler l'ensemble des éléments qui le précèdent. La comparaison entre éléments distants ne peut pas s'effectuer directement. Ces opérations requièrent souvent de nombreuses interactions. L'alphaslider [Ahlberg94a] fournit une solution à ce problème en proposant une technique de défilement à grain variable. Cette technique permet de réduire l'espace nécessaire à l'exploration et de diminuer le temps d'accès aux éléments. Plusieurs implémentations de l'alphaslider ont été réalisées. Dans l'exemple montré à la Figure 5.28, le curseur de la barre de défilement est divisé verticalement en trois parties. Le niveau de granularité du défilement dépend de la partie manipulée par l'utilisateur. Le niveau le plus fin permet de faire défiler tous les éléments. Le niveau le plus grossier permet de passer d'une zone de la liste à une autre sans afficher tous les éléments qui les séparent.

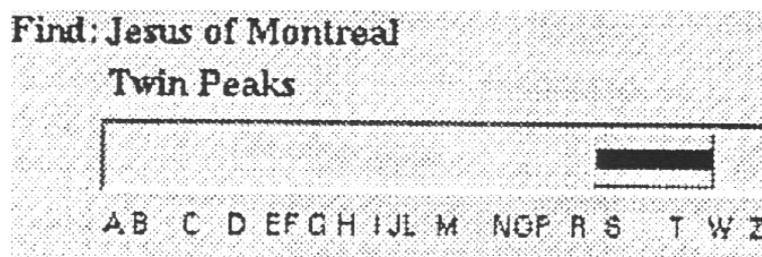


Figure 5.28 : L'alphaslider.

Pour afficher une matrice d'objets, les techniques les plus courantes utilisent une grille qui définit le marquage nécessaire afin que les alignements et les regroupements soient facilement repérables. Les règles de construction d'une telle grille veillent généralement au respect des proportions esthétiques et du format [Marcus92]. C'est par exemple le cas dans les systèmes Infogrid [Rao92] (Figure 5.29) et Labrador [Rao93], tous deux utilisés pour afficher des matrices de documents. Les techniques de distorsion et de zoom, présentées en début de chapitre, sont souvent mises à profit pour améliorer l'affichage et faciliter l'exploration de structures linéaires et matricielles. Parmi les systèmes qui adoptent ce type d'approche se trouvent le « tableau optique » [Rao94] (section 5.2.1.4), ou encore le « mur perspectif » [Mackinlay91] et « le document optique » [Robertson93], tous deux détaillés ci-dessous.

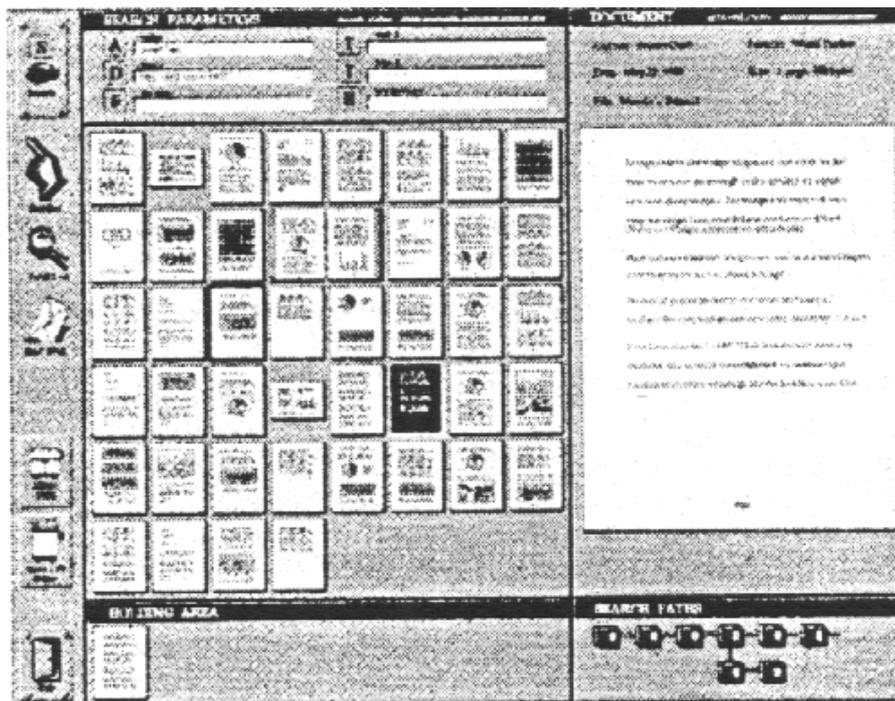


Figure 5.29 : Système Infogrid, affichage d'une matrice de documents.

5.4.1 Le mur perspectif

Le mur perspectif (« The Perspective Wall ») [Mackinlay91] est un dispositif de visualisation 3D qui met à profit l'effet de perspective pour intégrer naturellement contexte et détails dans une même représentation. Le mur perspectif est obtenu en transformant une surface 2D au moyen de la métaphore physique du pliage. Comme illustré à la Figure 5.30, il se compose d'un panneau central, pour visualiser les détails, et de deux autres panneaux, un de chaque côté, affichés en perspective pour visualiser le contexte.

Les objets affichés sur le mur ont tous le même niveau de grossissement sur le panneau central alors que leur taille diminue progressivement sur les panneaux en perspectives au fur et à mesure qu'ils s'éloignent de la partie centrale, et donc de l'utilisateur. La transition entre détails et contexte s'effectue ainsi naturellement par un effet de perspective.

L'utilisateur peut modifier son centre d'intérêt en déplaçant les deux plis de la surface 2D par manipulation directe. Pour visualiser en détails des objets affichés sur la partie gauche du mur, il doit déplacer la partie centrale vers la gauche. Ceci a pour effet de faire « glisser » tous les objets vers la droite. Une partie des objets préalablement affichés sur la partie gauche sont alors visualisés en détails, alors que d'autres préalablement affichés sur la partie centrale se retrouvent sur la partie droite du mur, faisant maintenant partie du contexte. L'utilisateur peut également

sélectionner directement un objet au moyen de la souris, cet objet étant alors positionné sur le panneau central. Un processus d'animation aide les utilisateurs à comprendre les divers changements de vue, réduisant du même coup la surcharge cognitive associée à l'analyse de ces transformations (processus de réassimilation). L'animation a également l'avantage de maintenir la relation entre les objets du détail et ceux du contexte (qui restent « à portée »). Le système permet également de modifier la proportion d'espace alloué au détail et au contexte en « étirant le mur comme une bande de caoutchouc ». Cette possibilité s'avère très importante lorsque la vue détaillée contient une importante quantité d'information.



Figure 5.30 : Le mur perspectif.

Le mur perspectif est utilisé pour la visualisation de données organisables en structures linéaires. Il est par exemple possible d'afficher sur le mur des documents en les classant linéairement soit par ordre alphabétique, soit par ordre chronologique (en fonction de la date de leur création ou de leur dernière modification). La dimension verticale du mur peut également être exploitée afin de visualiser d'autres caractéristiques de l'information, comme le type des documents ou leur propriétaire (Figure 5.30).

Cependant, les choix de conception du mur perspectif ne présentent pas que des avantages. La forme du mur entraîne une perte d'espace à l'écran. En effet, la hauteur des panneaux en perspective diminue en s'éloignant du centre, ce qui laisse aux quatre coins de l'écran de l'espace inutilisé (où il n'est donc pas possible d'afficher les données). Le mur perspectif ne permet de focaliser que sur une seule région de l'espace de données à la fois, ce qui empêche la comparaison d'éléments appartenant à des régions éloignées.

5.4.2 Le document optique

Le document optique (« The Document Lens ») [Robertson93] est un système qui affiche des matrices de documents et qui implémente une technique de distorsion pour faciliter la consultation en contexte. Les documents sont initialement affichés sur une surface de visualisation 2D, comme illustré à la Figure 5.31.

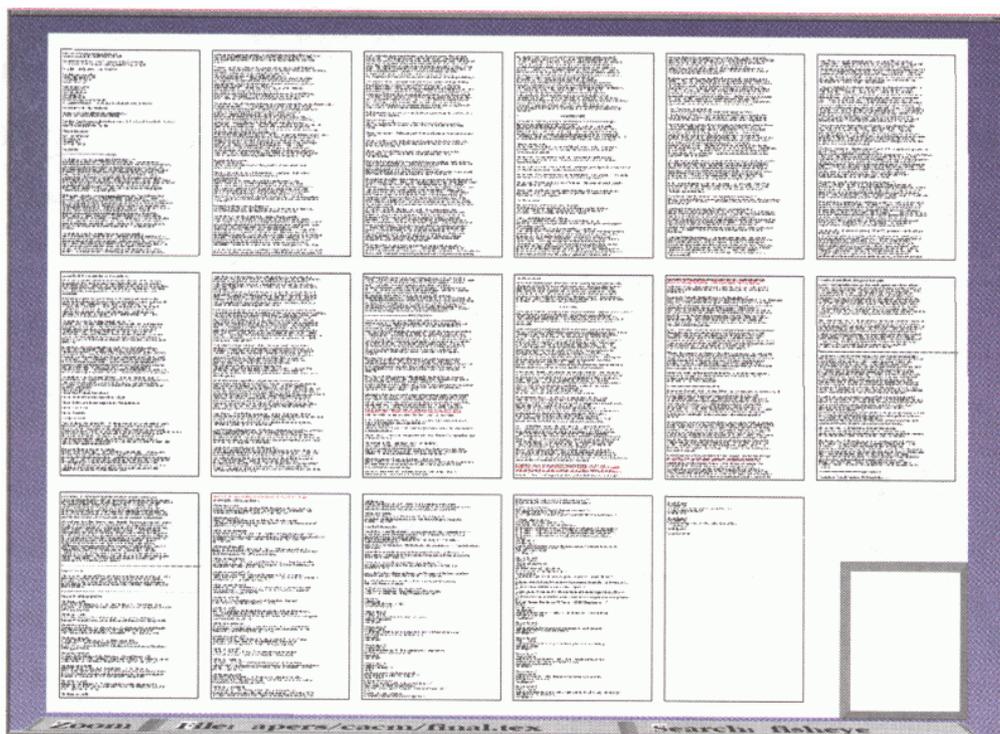


Figure 5.31 : Le document optique (vue initiale).

L'utilisateur peut augmenter le niveau de détail d'une partie de cette surface au moyen d'une lentille rectangulaire (métaphore de la loupe). La zone délimitée par les côtés de la lentille est magnifiée tandis que l'espace extérieur est étiré afin que son contenu reste entièrement visible. Il résulte de cette transformation une représentation 3D pyramidale qui fournit un affichage continu du contexte (Figure 5.32). Les éléments proches des bords extérieurs de la lentille sont plus facilement identifiables que ceux qui s'en trouvent éloignés (ex : le texte reste lisible). L'utilisateur déplace la lentille (le sommet de la pyramide) dans le plan (x, y) de la surface de visualisation au moyen de la souris et utilise le clavier pour la déplacer d'avant en arrière. Cette dernière possibilité permet de contrôler la taille de la lentille qui se modifie automatiquement en fonction de sa distance à l'utilisateur.

Un des inconvénients de cette technique réside dans le temps pris pour le rendu du texte en perspective 3D. A ce problème, les auteurs proposent trois solutions. La

première consiste à remplacer le texte par des segments de droite (peu gourmand en temps de calcul). La seconde consiste à utiliser des fontes vectorielles. Enfin, la troisième utilise un procédé de texture de polices de caractères qui fournit les résultats les plus satisfaisants au niveau de la qualité du rendu textuel. Cependant, les performances du système en temps de réponse sont fortement dégradées, ce qui rend difficile le traitement d'une quantité importante de texte. Un autre problème lié à l'affichage du texte vient du fait que peu de texte contextuel (en dehors de la lentille) reste lisible, surtout si la lentille est proche de l'utilisateur (de grande taille), ce qui rend difficile l'analyse du contexte global.

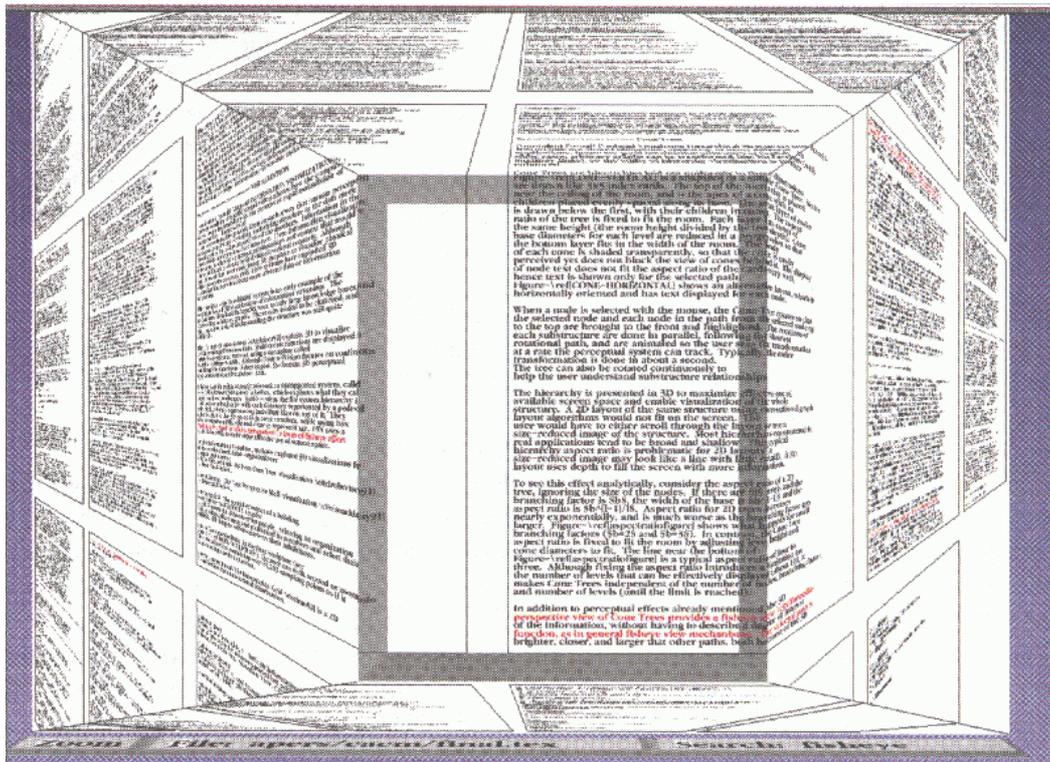


Figure 5.32 : Le document optique (vue focus+contexte pyramidale).

5.5 Structures hiérarchiques

Une hiérarchie est une structure qui se caractérise par la propriété suivante : chaque nœud a un parent unique excepté la racine qui n'en a pas. A chaque couple de nœuds est donc associé un chemin unique. Cette propriété implique un ordonnancement des éléments qui facilite la représentation de hiérarchies. Ce type de structure est assez simple à comprendre et les utilisateurs ont l'habitude de les manipuler. En effet, il existe une grande diversité de données hiérarchiques comme par exemple les taxonomies, les systèmes informatiques de gestion de fichiers, les organigrammes de sociétés, les généalogies, etc. Les tâches que les utilisateurs peuvent être amenés à

réaliser sur une hiérarchie incluent la recherche d'un nœud particulier, la visualisation d'un nœud dans son contexte, l'examen de l'ensemble de la structure, etc., ainsi que divers traitements sur le contenu d'un nœud ou de groupes de nœuds.

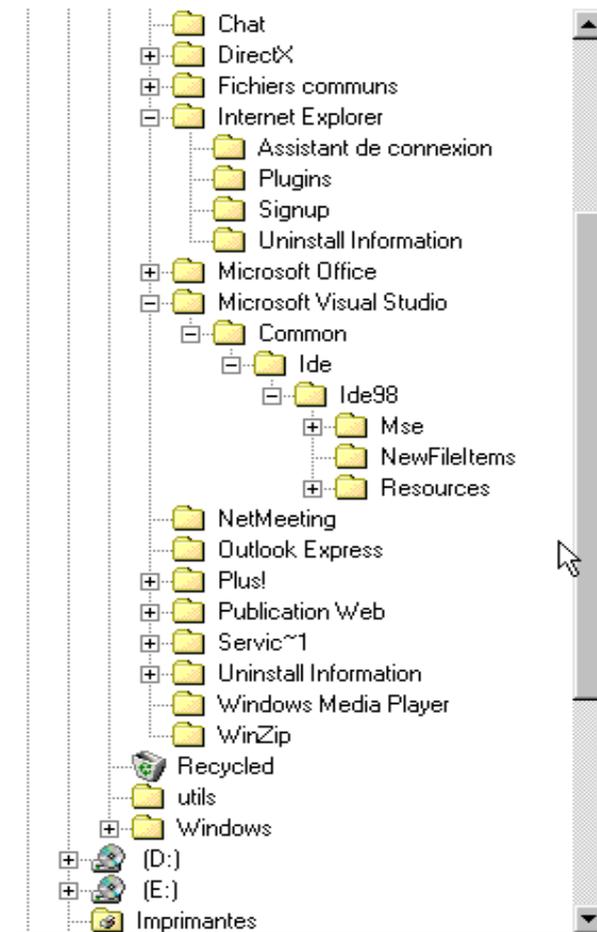


Figure 5.33 : Une arborescence de fichiers (Explorateur Windows).

L'approche traditionnelle consiste à représenter les nœuds de l'arbre par des icônes et les liens qui structurent les nœuds en hiérarchie par des segments de droites. Cette approche est couramment utilisée pour représenter l'arborescence d'un système de gestion de fichiers, comme illustré à la Figure 5.33, ou encore une arborescence de classes en programmation orientée objet. Les éléments de l'arbre sont positionnés verticalement sous forme de listes. La dimension horizontale sert juste à indiquer à quel niveau appartiennent les éléments (i.e. leur profondeur dans la hiérarchie).

De telles représentations imposent toutefois une limitation quant au nombre de nœuds qu'il est possible de montrer aux utilisateurs. Elles mettent l'accent sur les détails aux dépens de l'ensemble. Chaque nœud présente ses enfants sous la forme d'une liste verticale. Une grande partie de l'espace horizontal est gaspillé. Si l'arbre à

visualiser est trop grand pour être affiché dans l'espace écran, l'utilisateur doit faire défiler les éléments pour atteindre une partie de l'arbre éloignée de la racine. Il perd alors le contexte global.

Les techniques interactives de visualisation présentées dans cette section fournissent différentes réponses à ces problèmes. Le choix d'une technique dépend cependant de la tâche à effectuer et de la nature des données (i.e. de leurs propriétés). Le système Treemaps [Shneiderman92] repose sur une méthode d'encapsulation spatiale particulièrement bien adaptée à la représentation d'arbres dont les nœuds incluent une variable quantitative. Les arbres coniques [Robertson91] utilisent la troisième dimension pour gérer le fait que le nombre d'éléments d'une hiérarchie a tendance à s'accroître exponentiellement avec sa profondeur. Les arbres hyperboliques [Lamping96] sont des vues focus+contexte de hiérarchies basées sur l'utilisation d'une géométrie non-euclidienne.

5.5.1 Approche Pavage : le système Treemap

La méthode de représentation proposée par Shneiderman (et implémentée dans le système Treemap) [Shneiderman92], s'appuie sur le remplissage d'une surface rectangulaire pour faire apparaître la structure hiérarchique. La construction de la représentation débute par la définition des dimensions de la surface rectangulaire et par la donnée d'un arbre pondéré, le poids d'un nœud non terminal correspondant au cumul des poids associés à ses descendants. L'algorithme de construction attribue tout d'abord à la racine de la hiérarchie l'ensemble de la surface. Cet espace est ensuite subdivisé horizontalement en sous rectangles représentant chacun un des fils de la racine de telle sorte que leurs poids soient proportionnels à la surface occupée. Chaque sous rectangle est à son tour subdivisé verticalement en sous-rectangles correspondant à ses fils, et ainsi de suite. L'axe de subdivision change à chaque niveau de récursion afin de pouvoir différencier les différents niveaux de la hiérarchie. Le poids à associer aux nœuds de l'arbre est un facteur à paramétrer en fonction du type et des caractéristiques des objets sous-jacents. Il peut par exemple représenter la taille des fichiers dans un système de gestion de fichiers (Figure 5.34), ou encore l'importance relative des différentes collections d'une bibliothèque disposant d'un système de classification hiérarchique. Treemap offre également la possibilité aux utilisateurs de pouvoir zoomer sur un nœud particulier afin de remplacer l'affichage courant par le contenu de ce nœud. Le changement de vue s'accompagne d'un processus d'animation. Cependant aucune information contextuelle n'aide l'utilisateur à se localiser dans l'ensemble après avoir zoomé.

Cette technique ne respecte globalement aucune proportion géométrique particulière lors de la construction des rectangles. Cette particularité est à la source de deux inconvénients majeurs. Tout d'abord, il est fréquent d'obtenir des rectangles très fins, ce qui pose un problème quant à l'identification des rectangles apparaissant comme

un amas de lignes. Enfin, avoir des proportions très différentes d'un rectangle à l'autre ne facilite généralement pas la comparaison des surfaces respectives qu'ils occupent. En effet, la comparaison à l'œil nu de deux surfaces est assez facile quand les formes des objets à comparer sont similaires. Il n'en va cependant pas de même si les formes sont très différentes, comme par exemple entre une bande très fine et un carré. L'aspect intéressant de cette approche réside dans l'usage des surfaces comme moyen de véhiculer de l'information, une caractéristique assez rare dans les stratégies de présentations existantes.

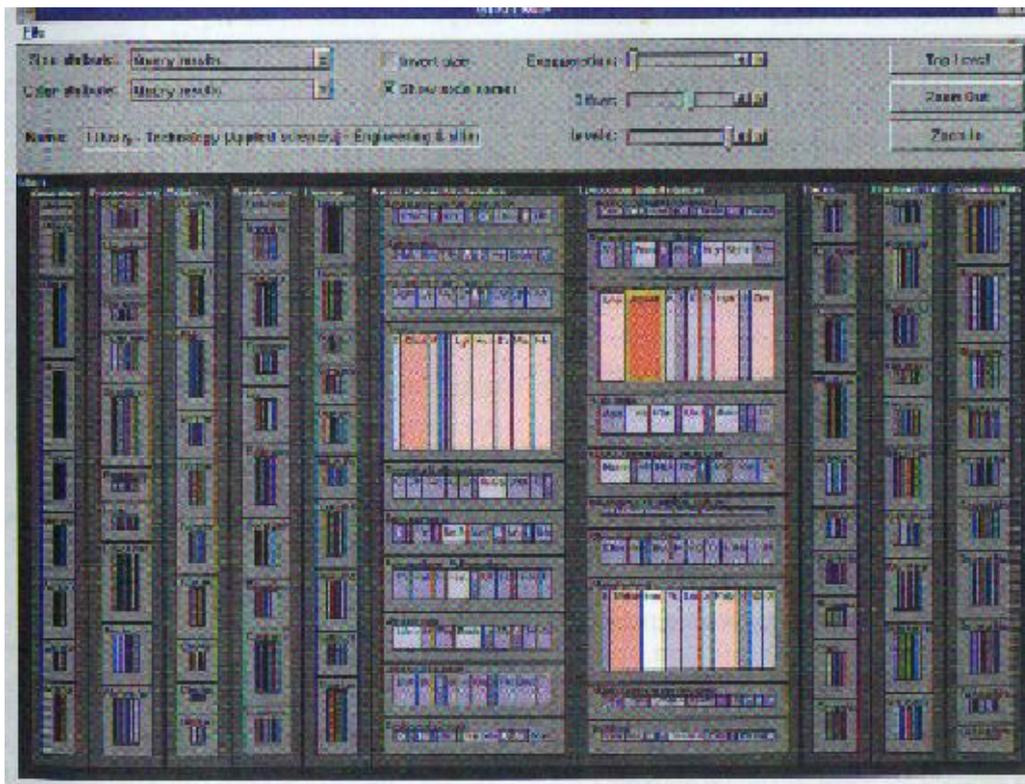


Figure 5.34 : Le système Treemap.

5.5.2 Approche 3D : les arbres coniques

Les arbres coniques (« The Cone Trees ») [Robertson91] sont utilisés pour représenter des structures d'information hiérarchiques sous formes de cônes affichés en 3D (Figure 5.35).

La racine de la hiérarchie est positionnée en haut de l'espace d'affichage et correspond au sommet d'un cône. Ses enfants sont ensuite placés sur la base de ce cône de manière à y être régulièrement espacés. Chacun de ces nœuds devient alors à son tour le sommet d'un cône 3D, le tout formant diverses couches de cônes. Toutes ces couches ont la même hauteur, celle-ci étant calculée pour faire tenir l'arbre dans

l'espace d'affichage (la hauteur de l'espace d'affichage divisée par la profondeur de l'arbre). Tous les cônes d'une même couche ont une base de diamètre identique. Le diamètre associé aux diverses couches diminue cependant progressivement avec l'augmentation de la profondeur de l'arbre afin de faire tenir la couche du bas dans la largeur de l'espace d'affichage. Le corps de chaque cône est affiché en transparence afin d'éviter les problèmes d'occlusion (pour ne pas bloquer la vue des cônes se trouvant derrière). Enfin, des techniques d'ombrage sont utilisées afin d'apporter des renseignements supplémentaires sur la disposition des éléments de l'arbre.

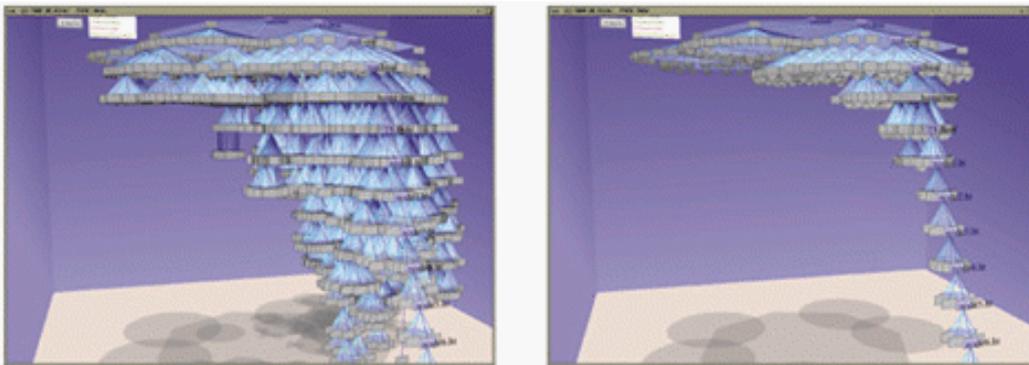


Figure 5.35 : Les arbres coniques.

L'accès à l'information s'effectue simplement par sélection des nœuds au moyen de la souris. La sélection d'un nœud entraîne la rotation de tous les cônes qui contiennent les nœuds du chemin reliant la racine au nœud sélectionné. Ce dernier et tous ses ancêtres sont alors ramenés au premier plan et mis en évidence par un changement de couleur. Les rotations de chaque sous-structure s'effectuent en parallèle et suivent le plus court chemin. Ces rotations parfois assez complexes peuvent provoquer une transformation de l'ensemble de l'arbre conique. Une animation permet aux utilisateurs de suivre ces rotations (habituellement, une telle transformation nécessite environ une seconde) et d'assimiler la nouvelle représentation sans effort cognitif supplémentaire. Afin de mieux comprendre les relations existant entre les sous-structures de l'arbre, l'utilisateur a également la possibilité d'appliquer des rotations par manipulation directe. Les rotations sont toutefois difficiles à suivre dans des arbres équilibrés à cause de l'apparence uniforme des sous-structures. Il apparaît donc clairement que cette technique de représentation s'adapte mieux à l'exploration de hiérarchies déséquilibrées. Cependant, en pratique, la majorité des hiérarchies sont larges, peu profondes et déséquilibrées. L'utilisateur peut également lancer une recherche automatique sur les données sous-jacentes à l'arbre. Aux nœuds qui contiennent l'information recherchée est alors ajoutée une barre de couleur dont la taille est proportionnelle au score de la recherche.

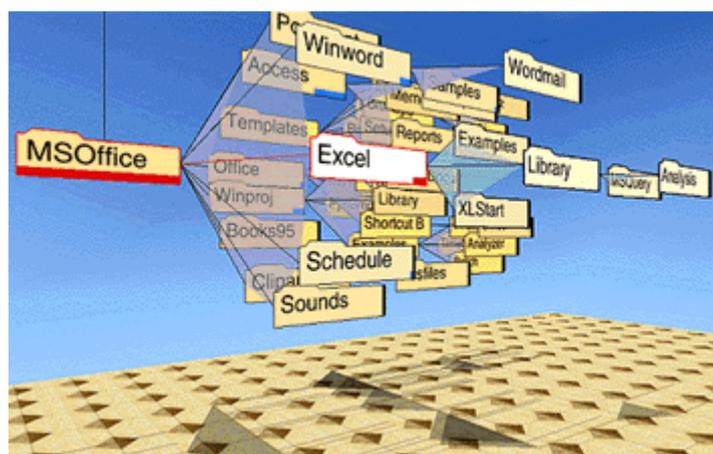


Figure 5.36 : Les arbres coniques (visualisation horizontale).

Comme première application, les auteurs ont utilisé la technique des arbres coniques pour représenter un système de fichiers UNIX. Cette technique a permis l’affichage de l’ensemble d’une hiérarchie composée d’environ 600 répertoires et 10000 fichiers, ce qui s’avère difficile à réaliser avec une représentation 2D. Les arbres coniques peuvent également être employés pour l’exploration de classes en programmation orientée objet, de bibliothèques ou de livres, etc. Selon les auteurs, l’encombrement de l’affichage réduit toutefois l’efficacité de la représentation pour des hiérarchies contenant plus de 1000 nœuds, plus de 10 couches ou plus de 30 fils par nœud.

5.5.3 Approche géométrique : les arbres hyperboliques

Lamping et Rao propose une technique focus+contexte basée sur la géométrie hyperbolique pour visualiser et représenter des structures hiérarchiques [Lamping96].

La géométrie hyperbolique est une géométrie non Euclidienne. La principale différence entre ces deux types de géométrie peut s’énoncer comme suit : dans le plan Euclidien, si une ligne L ne passe pas par un point P alors il existe une et une seule ligne qui passe par ce point P et qui est parallèle à la ligne L (i.e. pas d’intersection et coplanaire), au contraire du plan hyperbolique où il existe plusieurs parallèles de ce type. Une des propriétés intéressantes du plan hyperbolique est que le plan lui-même s’élargit de manière exponentielle par rapport au centre. En d’autres termes, l’espace disponible s’accroît exponentiellement lorsque la distance au centre augmente. Cette propriété fournit une bonne base pour le positionnement de hiérarchies car ces structures ont tendance à s’élargir exponentiellement avec l’accroissement de leur profondeur. Le plan hyperbolique offre la possibilité de positionner les éléments d’une hiérarchie de telle sorte que la distance entre parents, enfants et éléments d’un même niveau soit approximativement la même en n’importe quel point de l’espace.

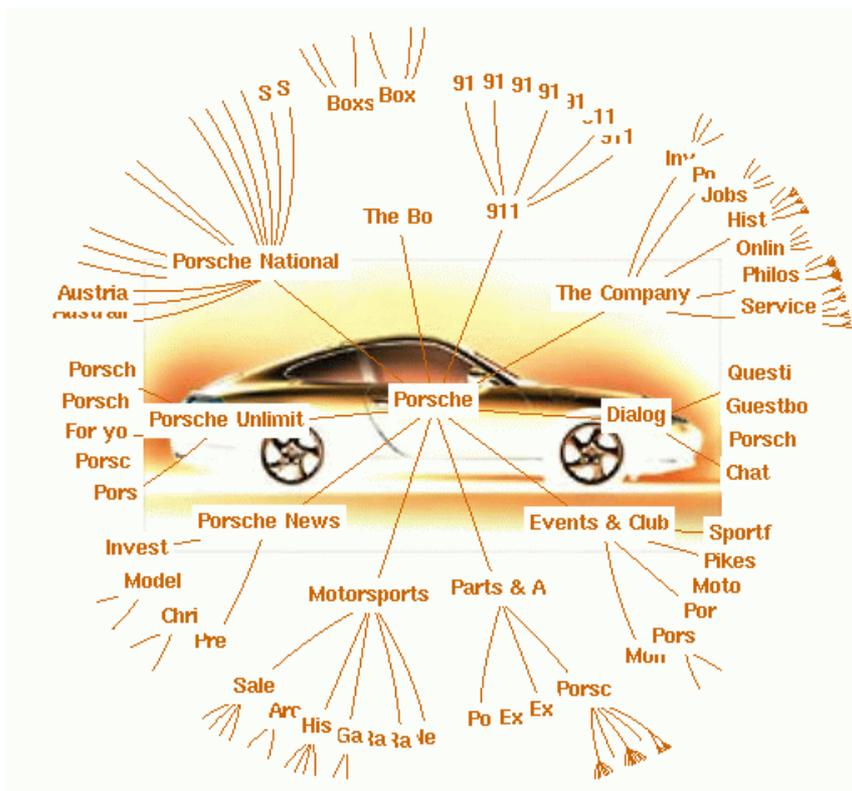


Figure 5.37: Affichage hyperbolique 2D d'une hiérarchie.

Dans le système proposé par Lamping et Rao (« the Hyperbolic Browser ») [Lamping96], le plan hyperbolique est projeté sur un disque bidimensionnel au moyen du modèle de Poincaré. Une hiérarchie positionnée dans le plan hyperbolique est alors représentée comme le montre la Figure 5.37. Le centre du disque correspond au point de focus, c'est la racine de la hiérarchie qui y est initialement positionnée. Les autres niveaux de la hiérarchie sont placés autour de la racine de manière à former un affichage concentrique.

La projection du plan hyperbolique sur le disque crée un effet de perspective sur les bords du disque. L'espace alloué à un nœud diminue de manière continue en fonction de sa distance au point de focus. Ce type de représentation peut être vue comme une variante des vues fisheyes. Une part plus importante de l'affichage est allouée à une partie de la représentation (le point de focus) tout en l'englobant dans le contexte de la hiérarchie. L'utilisateur peut à tout moment changer de focus. Il lui suffit de déplacer interactivement un nœud et de le mettre au centre du disque au moyen de la souris. L'ensemble de la représentation se transforme en conséquence. Le même paradigme d'interaction est utilisé pour l'exploration de la hiérarchie. Ces déplacements s'accompagnent tous d'un processus d'animation afin de faciliter la compréhension des modifications de l'affichage.

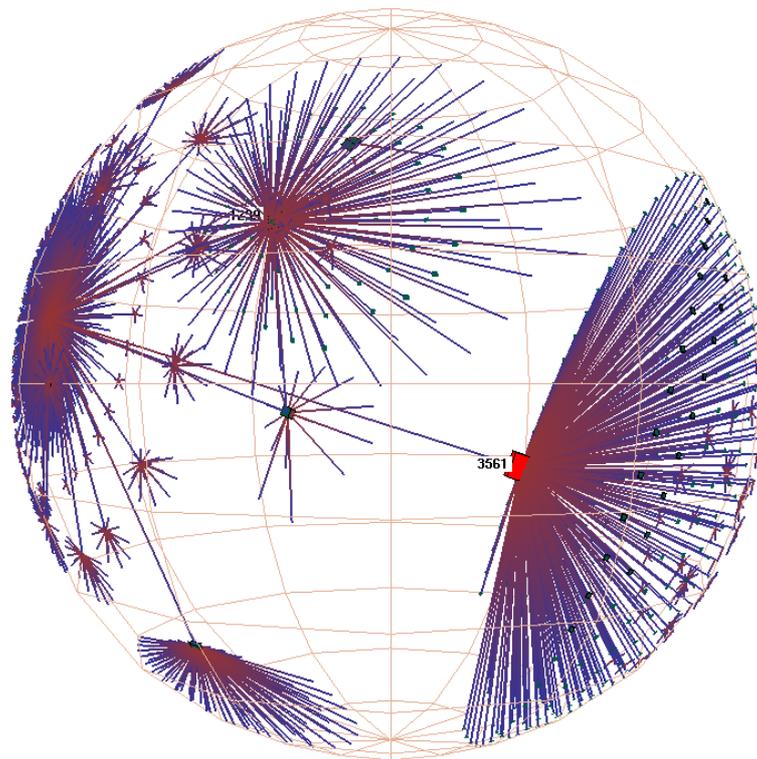


Figure 5.38 : Affichage hyperbolique 3D d'une hiérarchie.

Comme dans le cas des vues fisheyes, les transformations peuvent toutefois devenir une source de confusion. Par exemple, il n'est pas toujours évident, lorsque les nœuds changent de place, de déterminer à quel niveau de la hiérarchie ils appartiennent. Une autre particularité de ce type d'approche est que la représentation est construite une fois pour toute. La totalité de la hiérarchie est positionnée dans le plan hyperbolique dès le départ, le disque de projection permet seulement d'en visualiser un sous-ensemble qui évolue au cours de l'exploration. Il est ainsi impossible de rajouter de nouveaux nœuds dynamiquement sans recalculer l'ensemble de la disposition des éléments dans le plan hyperbolique.

Plusieurs implémentations des arbres hyperboliques ont été proposées, y compris en trois dimensions. Par exemple, la technique mise en œuvre par Munzner consiste à projeter le plan hyperbolique sur la surface d'une sphère, comme illustré à la Figure 5.38 [Munzner97]. Même si ce type de représentation permet d'augmenter le nombre d'éléments visualisés simultanément, il semble que l'amas de nœuds et de liens résultant complique l'exploration.

5.6 Structures de graphes

Les diagrammes de graphes sont souvent utilisés pour représenter des données organisées au moyen de nœuds et de liens, comme c'est par exemple le cas sur le Web et plus généralement dans tout type d'environnement hypermédia. La complexité inhérente à la structuration arbitraire de ces réseaux rend difficile leur représentation, contrairement par exemple aux hiérarchies ou aux matrices dont la structure même fournit un support à la disposition spatiale de leurs éléments.

La taille d'un graphe est un facteur source de problèmes de représentation. L'affichage d'un nombre élevé d'éléments peut compromettre la visibilité et l'utilisabilité de la représentation [Eades90, Battista99, Cruz01]. Face au nombre de nœuds et de liens affichés, l'utilisateur peut, par exemple, avoir des difficultés à identifier ou à trouver un nœud particulier. Ces deux critères (visibilité et utilisabilité) sont pourtant des facteurs clés en visualisation de l'information, la représentation étant utilisée comme un support à la manipulation des données. L'affichage d'un graphe dans sa totalité peut toutefois donner une indication sur la structure d'ensemble ou sur la localisation d'un élément par rapport à l'ensemble.

De l'avis de nombreux chercheurs, la méthode la plus simple pour visualiser un graphe de grande taille consiste à le transformer en une structure plus simple à comprendre et à afficher (par exemple une hiérarchie) [Akscyn88, Conklin88, Rivlin94], en utilisant des techniques comme celles décrites à la section 4.4.2.

Les systèmes qui permettent de visualiser et de manipuler des structures de graphes non modifiées implémentent le plus souvent un algorithme de placement de type FDP (Force Directed Placement) qui simule un système physique de forces [Quinn79]. Plusieurs variantes de ce type d'algorithme ont été proposées [Eades84, Fruchterman91, Young96]. Sous sa forme la plus simple, les nœuds exercent une force répulsive et les liens une force attractive entre les objets qu'ils connectent afin de maintenir un équilibre. Cette force attractive est typiquement proportionnelle à la longueur du lien et simule le comportement d'un ressort. Le positionnement initial du graphe représente le système de forces dans un état d'énergie maximum. Après un certain nombre d'itérations, l'algorithme de placement a organisé les nœuds de manière à atteindre un état d'énergie minimum. Les résultats fournis sont généralement satisfaisants d'un point de vue esthétique. Le positionnement des nœuds est optimisé en fonction de la longueur des liens, de l'espacement minimum requis entre deux nœuds, etc. Ceci ne signifie pas pour autant que la représentation du graphe est compréhensible ou utilisable.

Les techniques de type FDP ont également été adaptées aux espaces de représentation 3D [Chalmers95, Hendley95, Wood95]. Par exemple, le système Hyperspace propose une implémentation pour visualiser des réseaux hypertextes. Ce système a été appliqué au cas du Web. Comme illustré à la Figure 5.39, les pages

Web sont représentées par des nœuds sphériques et les liens par des arcs. L'utilisateur peut changer son angle de vue et manipuler les objets de la représentation. La sélection d'un nœud entraîne le chargement de la page Web correspondante dans le navigateur Web Mosaic. Hyperspace utilise également l'API de Mosaic en phase de création du graphe pour accéder aux pages HTML distantes. Il est possible d'associer aux nœuds des textures ou des couleurs afin d'aider les utilisateurs à les identifier, ainsi que des attributs (comme par exemple un URL) au moyen de labels. Cependant, selon les auteurs, l'affichage des labels perturbe souvent les utilisateurs en cachant des objets de la représentation. L'algorithme de placement FDP utilisé est dynamique. La modification de la structure (ajout ou suppression de nœuds) entraîne une réorganisation générale de la représentation. Ceci engendre souvent une surcharge cognitive, les utilisateurs devant réinterpréter l'affichage à chaque modification.

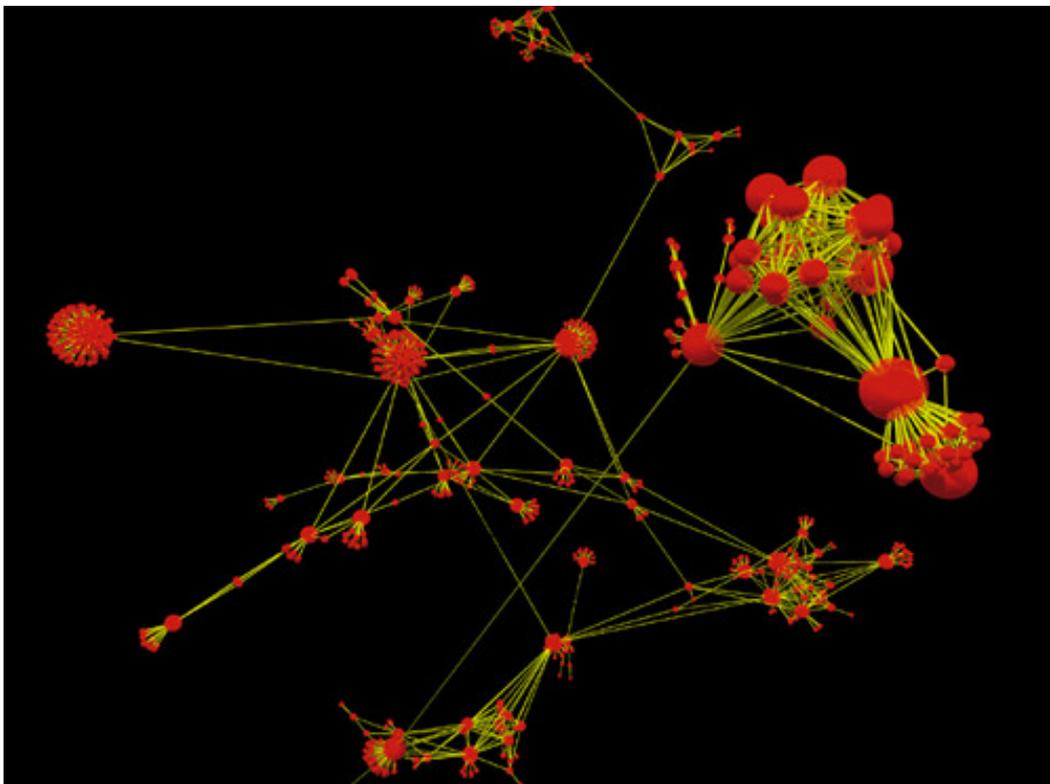


Figure 5.39 : Système Narcissus.

L'utilisation d'une dimension supplémentaire ne fournit pas une solution aux problèmes d'utilisabilité et de visibilité mais permet de réduire les limitations de place imposées par les surfaces bidimensionnelles.

Concernant la visualisation de graphe, il semble que l'efficacité avec laquelle les utilisateurs vont explorer et analyser les données dépend moins de la technique de

placement utilisée que des mécanismes d'interactions disponibles (animation, distorsion, zoom, etc.). SemNet est un des premiers systèmes à avoir pris en compte les aspects liés à l'interaction [Fairchild88].

5.7 Espaces informationnels de travail

Le concept d'espace informationnel de travail (« information workspace ») a été introduit par Card, Robertson et Mackinlay [Card91] pour désigner un espace où les utilisateurs peuvent rassembler, visualiser, organiser et travailler avec un ensemble de données extraites d'une ou plusieurs bases informationnelles. L'idée principale repose sur le fait que les utilisateurs ont besoin d'espaces de travail personnalisés pour gérer leurs différentes activités.

La présentation des données sous forme de listes, de hiérarchies et d'icônes est à la base des premières techniques graphiques utilisées pour gérer interactivement l'organisation de collections de données. Le gestionnaire de fichiers, popularisé par le Macintosh d'Apple puis par le système Windows de Microsoft, permet de montrer la structuration physique des données (sur disque) et de la modifier interactivement. Dans ces environnements WIMP classiques, les utilisateurs peuvent également positionner des icônes sur le bureau de travail afin de pouvoir accéder plus facilement aux données qu'elles référencent. Le nombre d'icônes qu'il est possible d'afficher simultanément reste cependant limité. De plus, elles donnent peu d'information sur le contenu des données, ce qui peut entraîner de nombreuses erreurs de sélection. Mander, Salomon et Wong augmentent la métaphore des objets positionnés sur le bureau de travail en introduisant le concept de piles [Mander92]. Une pile permet de regrouper un ensemble de données et de les représenter à l'écran au moyen d'un objet unique. Ce procédé fournit un moyen simple pour mieux structurer l'organisation des données et pour économiser l'espace écran.

« L'Information Visualizer » est un des premiers systèmes à montrer comment améliorer l'utilisabilité d'un espace informationnel de travail au moyen de techniques de visualisation [Robertson93]. Ce système divise l'espace de écran en plusieurs « pièces » (Figure 5.40). Chacune d'elles est en charge de la représentation d'un certain type de données (linéaire, hiérarchique, géographique, etc.) avec lesquelles l'utilisateur peut interagir. Ce système inclut des mécanismes pour la navigation 3D, pour la manipulation d'objets, pour la construction de représentations zoomables et de vues focus+contexte [Robertson89, Mackinlay90]. Le « mur perspectif » [Mackinlay91], le « document optique » [Robertson93] et les « arbres coniques » [Robertson91] (section 5.5.2) sont des systèmes qui font partie de « l'Information Visualizer ».

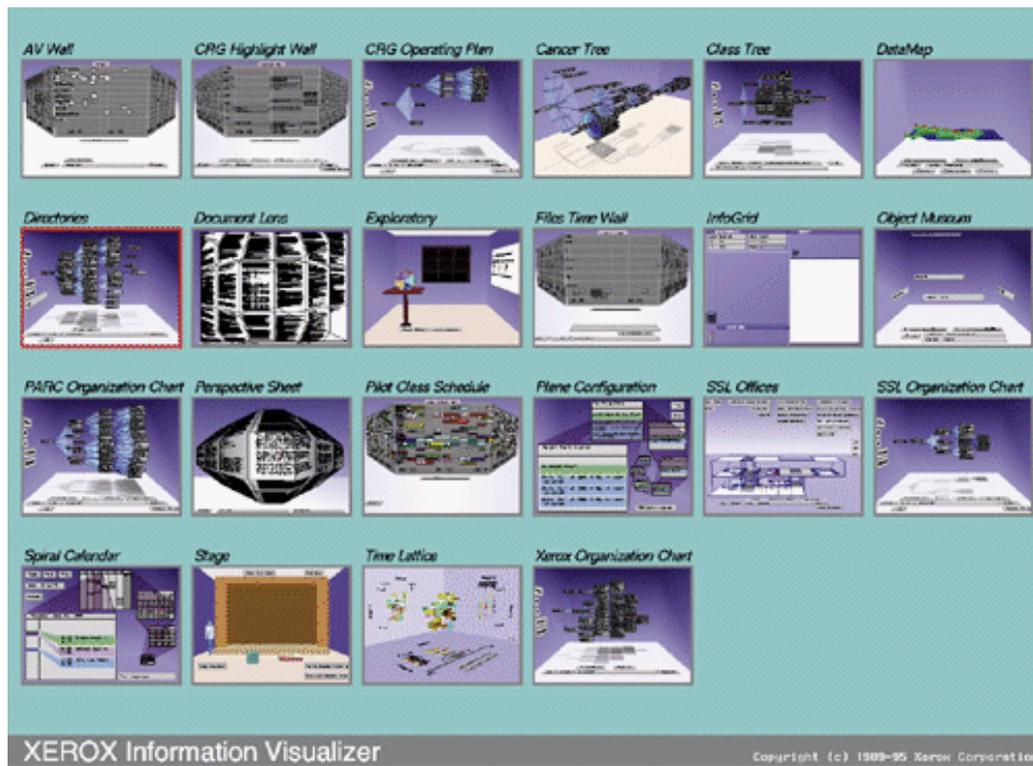


Figure 5.40 : L'Information Visualizer.

Les espaces de bookmarks, principalement développés pour des environnements de type hypermédia et souvent plus particulièrement pour le Web, sont de bons exemples d'espaces personnels d'information. Parmi les systèmes les plus connus se trouvent le « WebBook », le « Web Forager » [Card96] et le « Data Mountain » [Robertson98].

5.7.1 Les WebBooks et le Web Forager

Un WebBook [Card96] est un livre 3D interactif de pages Web, comme illustré à la Figure 5.41. L'idée consiste à regrouper et stocker des pages Web au sein d'une même entité (un livre) afin de donner du sens à cet ensemble de pages en fonction d'un contexte ou de l'intérêt d'un utilisateur. Chaque WebBook permet de consulter les pages qu'il contient et d'interagir avec leur contenu.

Les liens des pages Web sont représentés par deux couleurs afin de permettre aux utilisateurs de faire la différence entre une référence à une autre page du livre (lien rouge) et une référence à un document extérieur (lien bleu). Lorsque l'utilisateur clique sur un lien rouge, les pages du livre tournent, via un processus d'animation, jusqu'à atteindre la destination du lien. L'activation d'un lien de couleur bleue entraîne la fermeture du WebBook courant et la recherche de la page pointée. Si cette

page appartient à un autre WebBook, il est ouvert à la page désirée. Si elle n'est pas présente dans le système, elle est alors téléchargée et affichée comme une entité individuelle dans l'espace de travail de l'utilisateur (voir le Web Forager).



Figure 5.41 : Un WebBook.

Comme avec un livre traditionnel, l'utilisateur dispose de plusieurs possibilités pour parcourir les pages. Cliquer sur la partie gauche ou droite d'une page (en dehors d'un lien) permet de passer à la page précédente ou suivante. L'utilisateur peut également feuilleter (Figure 5.42) un WebBook pour avoir un aperçu de son contenu, ou pour accéder plus rapidement à un passage particulier. Ces mécanismes de navigation sont associés à un processus d'animation pour renforcer l'effet de consistance du livre (le texte et les images restent toujours visibles lorsque tournent les pages du livre) et fournir une indication claire des relations existant entre les pages.

Selon les auteurs, l'utilisation d'un objet familier (un livre) permet de se concentrer sur la manipulation des documents (organisation, consultation, etc.) et non pas sur l'interaction (les moyens de manipulation). La métaphore du livre est ici utilisée pour transposer des mécanismes courants d'accès à l'information dans le monde informatique sans augmenter la charge cognitive des utilisateurs. De plus la manipulation d'un livre véhicule des informations que les utilisateurs ont l'habitude d'interpréter (par exemple, l'action de tourner les pages donne des indications sur les relations entre ces pages, sur la direction de parcours du livre, sur sa taille et son contenu, etc.). Le WebBook tire avantage des avancées graphiques et de la puissance des processeurs pour être le plus proche possible d'une simulation réaliste d'un livre, tout en proposant des fonctionnalités étendues comme la gestion de liens hypertextes ou de modifications de la représentation (taille, fonte, etc.).

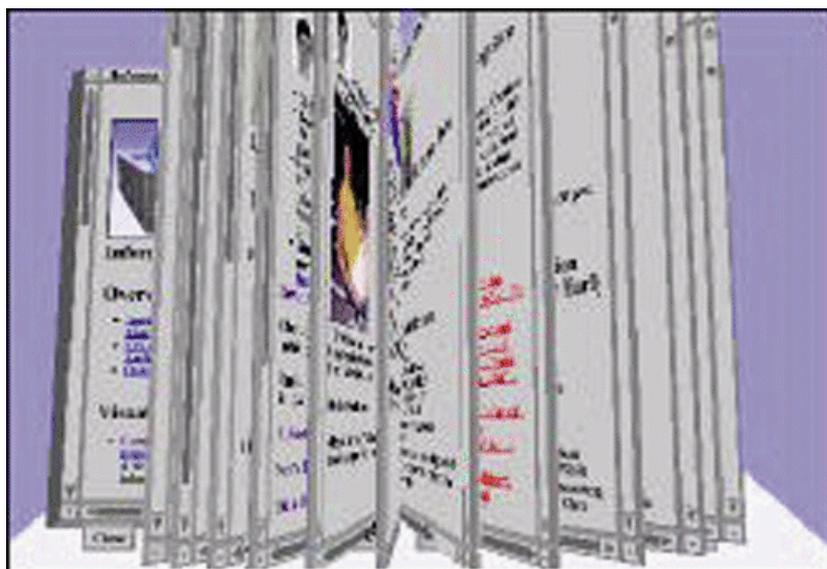


Figure 5.42 : Feuilletage d'un WebBook.

La compacité des livres représente un avantage du point de vue du stockage et de l'affichage. La séquentialité des pages entraîne quant à elle certains problèmes du point de vue de la navigation hypertextuelle. Un WebBook ne montre que deux pages à la fois et il est difficile de comparer des pages éloignées. Comme solution à ce problème, il est offert aux utilisateurs la possibilité « d'explorer » le livre afin de positionner les pages à plat (sous forme matricielle) et pouvoir ainsi en disposer simultanément. Les auteurs ont intégré à leur environnement le système Document Lens [Robertson93] (section 5.4.2) afin de se déplacer, zoomer et consulter les pages ainsi positionnées, tout en gardant un affichage focus+contexte du livre (Figure 5.43). Un processus d'animation permet de passer de la représentation du livre à la représentation matricielle utilisée par le document Lens (et vice versa) afin de ne pas perturber les utilisateurs par des modifications « brutales » de l'affichage.

Le Web Forager [Card96] est un espace virtuel 3D dédié au stockage, à la manipulation et à la consultation d'un ensemble de WebBooks et de pages Web individuelles (Figure 5.44). Il repose sur la métaphore de la bibliothèque où les documents sont organisés en collections, rangés dans des rayons et accessibles pour consultation.

Lorsque l'utilisateur clique sur un lien d'une page, la page de destination est automatiquement chargée dans le Web Forager (si elle n'en fait pas déjà partie) et affichée au premier plan à une taille suffisamment importante pour être lisible (c'est le point de focus). Une fois chargées, les pages Web sont stockées localement afin de ne pas gêner la fluidité de l'interaction par des temps d'accès trop longs. Chaque page peut être ajoutée à un WebBook par manipulation directe.

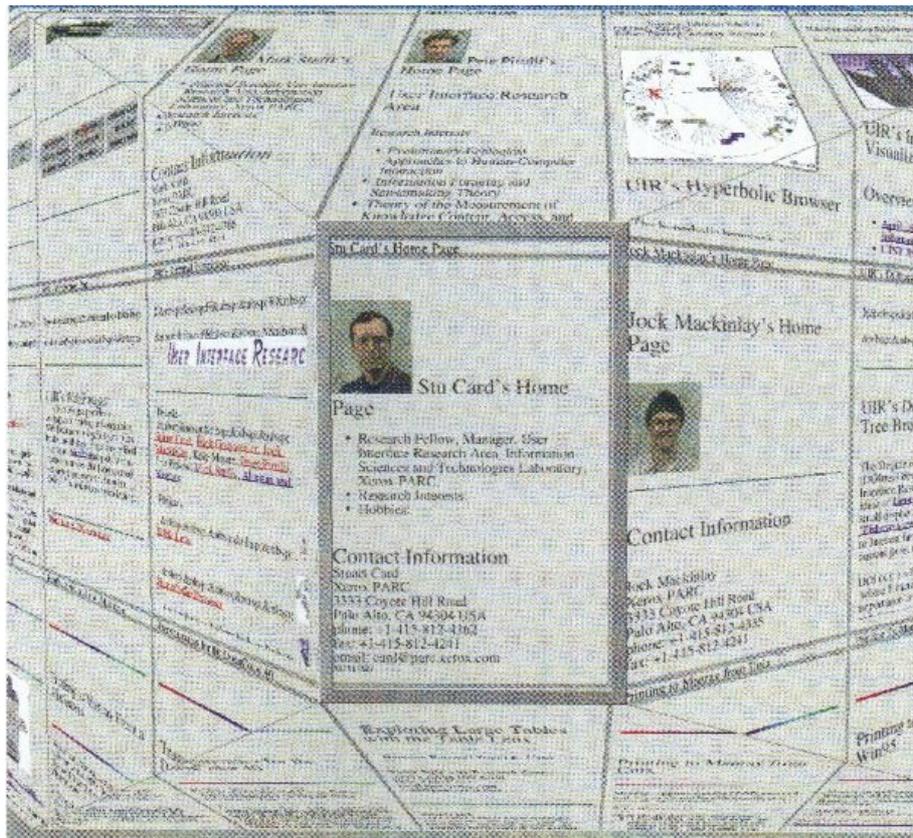


Figure 5.43 : Vue d'un WebBook avec le système Document Lens.

L'espace du Web Forager se divise principalement en trois niveaux du point de vue de l'organisation et de l'interaction (Figure 5.44).

1. Un point de focus qui montre un élément de la représentation (une page, un livre ou un livre ouvert) à sa taille maximum afin de pouvoir le consulter et interagir avec son contenu.
2. Un « espace mémoire immédiat » (le bureau), qui permet à l'utilisateur de placer et de garder à proximité les pages et les livres utilisés mais qui ne sont pas actuellement dans la zone de focus. C'est l'espace contextuel du Web Forager. Les objets de cet espace peuvent être déplacés interactivement sur les axes X, Y et Z. Plus ils s'éloignent de l'utilisateur, plus leur taille diminue. Selon les auteurs, il est possible de positionner dans cet espace jusqu'à 30 pages sans occlusion, et environ une centaine dans le cas contraire (sans compter le contenu des livres).
3. Un espace de stockage (une bibliothèque) pour ranger et classer les documents qui ne sont pas actuellement utilisés. Dans la vue principale, la bibliothèque est affichée en diagonale (par rapport au point de vue de l'utilisateur) afin d'économiser l'espace écran (en faveur du focus et du contexte) tout en montrant assez d'information pour pouvoir reconnaître les livres et les pages stockés. Cliquer sur un

des objets de la bibliothèque entraîne un déplacement de cet objet jusqu'à la zone de focus par un processus d'animation. Le document qui avait le focus se retrouve quant à lui dans la zone de stockage immédiate (le contexte). Si l'utilisateur souhaite examiner plus attentivement les éléments avant de faire son choix, il lui suffit de cliquer sur la bibliothèque pour se retrouver en face de celle-ci. L'ensemble de la représentation est alors modifiée de telle sorte que l'utilisateur garde la même vue sur les zone de focus et de contexte. Un clic souris sur une page ou un livre engendre la mise en focus de cet élément et le retour à la position initiale.



Figure 5.44 : Le Web Forager.

Le Web Forager peut être vu comme un espace de bookmarks qui permet à chaque utilisateur de donner un sens personnel à la représentation. L'utilisateur organise interactivement un ensemble de documents et se retrouve à chaque utilisation dans un environnement familier qu'il fait évoluer en fonction de ses besoins ou de ses centres d'intérêts.

5.7.2 Le Data Mountain

Le système « Data Mountain » [Robertson98] permet de stocker et d'organiser de manière interactive des documents jugés intéressants en vue d'une utilisation future. Il est utilisé comme une alternative aux mécanismes de bookmark de pages Web proposés par les navigateurs courants (Microsoft Explorer, Netscape Navigator). Avec le Data Mountain, les pages Web sont représentées sous forme d'images réduites et placées sur un plan incliné, comme illustré à la Figure 5.45.

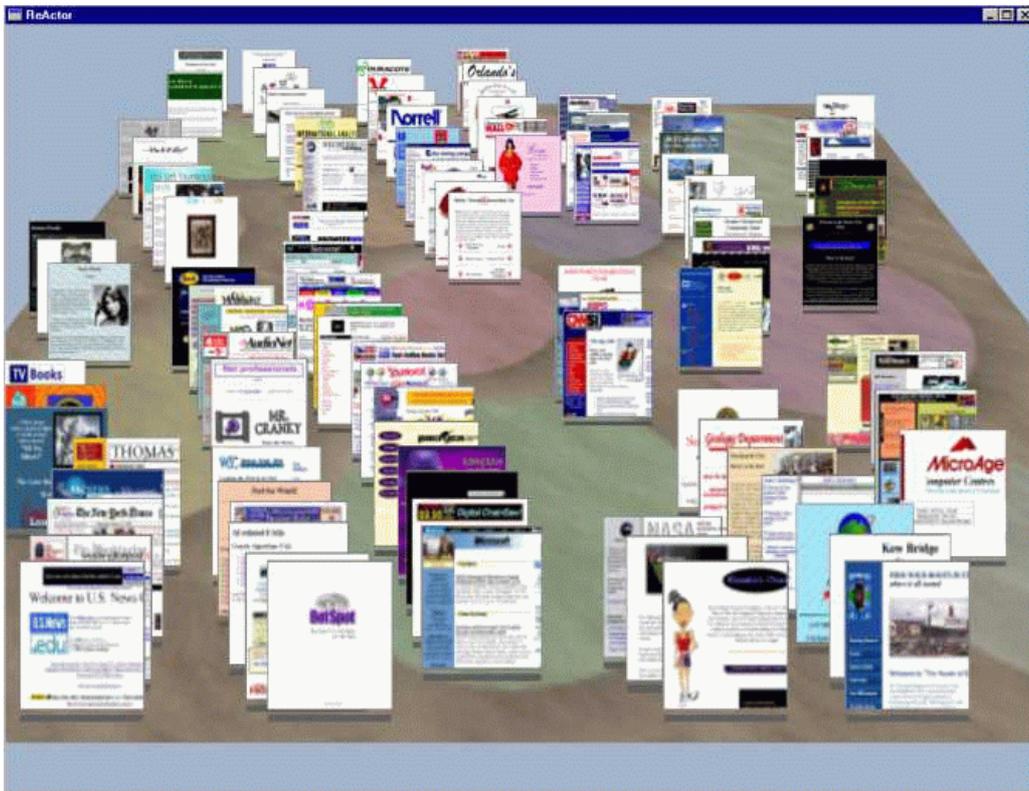


Figure 5.45 : Le Data Mountain.

Chaque page peut être déplacée par manipulation directe au moyen de la souris (par une technique classique de drag and drop) et positionnée manuellement à n'importe quel endroit de la surface. Une page reste toujours visible lors de son déplacement, ce qui aide l'utilisateur à la situer par rapport à son entourage et à la placer à la position souhaitée (les pages qui se trouvent sur son chemin sont temporairement décalées lors de son passage). Chaque utilisateur donne un sens à la représentation en organisant lui-même l'espace et en le faisant évoluer selon son activité. Il est par exemple possible de regrouper les pages par localisation spatiale. La texture de la surface contient également des marques passives (des cercles de couleurs) qui peuvent être utilisées (ou non) comme des aides au regroupement. L'effet de perspective 3D, dû à l'inclinaison de la surface, permet d'afficher un plus grand nombre de pages dans l'espace écran (en comparaison avec une surface 2D). La taille d'une page est automatiquement définie en fonction de sa position par rapport au point de vue de l'utilisateur. Afin qu'une page ne soit jamais entièrement cachée par une autre, le système impose une distance minimale entre deux pages, et ce même pendant les déplacements. Lorsque l'utilisateur met une page à la même position qu'une autre, cette dernière est automatiquement déplacée pour respecter la distance minimale. Si besoin, le décalage se propage au reste de la représentation. Ces déplacements implicites peuvent toutefois perturber l'organisation imposée par l'utilisateur (une page qui change de groupe, etc.). Le Data Mountain ne propose pas

de mécanismes pour définir explicitement des groupes de pages, ni pour les manipuler. Enfin, il ne dispose que d'une seule surface de visualisation, ce qui limite le nombre de pages qu'il est raisonnablement possible d'afficher.

Le Data Mountain utilise les capacités mémorielles humaines pour aider à retrouver plus facilement des pages qui ont été préalablement jugées intéressantes. L'emploi d'images réduites stimule la mémoire visuelle de l'utilisateur qui reconnaît plus facilement une page par la disposition spatiale de ses éléments, par les couleurs employées, etc. Afin de renforcer ce processus d'identification un label est associé à chaque page pour en montrer le titre. Ce label se présente comme une bulle d'aide qui apparaît lorsque la souris passe sur la page (le délai d'apparition est nul afin de ne pas ralentir la recherche). Le Data Mountain tire également avantage de la mémoire spatiale de l'utilisateur (la capacité de se rappeler ou l'on met quelque chose) en le laissant organiser manuellement les pages sur la surface de visualisation. La perspective, l'occlusion et la texture de la surface sont autant d'éléments qui facilitent la cognition spatiale. L'utilisateur se base sur sa capacité à reconnaître des relations spatiale 3D pour placer et retrouver les pages. Lors de la recherche d'une page, l'utilisateur concentre tout d'abord son attention sur la région ou il pense l'avoir placée, et s'aide ensuite des images réduites et/ou des labels associés pour l'identifier. Une fois repérée, la page peut être sélectionnée via un clic souris pour consultation. Elle est alors affichée au premier plan à une résolution suffisamment grande pour être lisible.

Le Data Mountain permet donc aux utilisateurs de tirer avantage d'un environnement 3D (meilleure utilisation de l'espace écran, relations spatiales perçues à faible surcharge cognitive) tout en manipulant ses éléments au moyen de techniques simples d'interaction 2D. Les auteurs ont montré qu'il permet de retrouver les pages plus rapidement que dans un système de bookmark traditionnel, et ce même après une longue période de temps [Czerwinski99].

6. Trouver l'information

Trouver un document dans un espace informationnel peut s'effectuer soit par formulation de requêtes soit par exploration interactive via un système de navigation. Comme aide à l'exploration, nous proposons un système mêlant l'utilisation de vues globales et de vues locales.

Les vues globales donnent une idée sur l'organisation de l'espace documentaire et fournissent des repères visuels favorisant la navigation. Elles sont utilisées pour survoler un nombre important de documents afin d'effectuer une exploration rapide, trouver des zones d'intérêts ou localiser des composants particuliers.

Les vues locales permettent de visualiser plus en détail des sous-ensembles de l'espace documentaire sélectionnés à partir d'une vue globale. Elles montrent l'organisation d'un petit nombre de documents et donnent une idée de leur contenu tout en rendant possible une lecture détaillée. Certains de ces documents sont visualisés en détail alors que d'autres, qui composent leur contexte, sont représentés de manière plus grossière. Il est ainsi possible de comparer plusieurs documents et d'en comprendre les relations, ce qui favorise les associations d'idées.

6.1 Association vues globales / vues locales

Dans une vue globale, l'affichage est géré de manière à représenter le plus grand sous-ensemble possible de l'espace documentaire, sans pour autant encombrer l'écran. Des représentations comme les arbres hyperboliques [Lamping96], les arbres coniques [Robertson91] ou le mur perspectif [Mackinlay91] en sont de bons exemples (section 5.5.3, 5.5.2, 5.4.1). Cependant, afficher un nombre important de documents contraint à les représenter soit par des labels soit par des icônes qui donnent peu d'information sur leur contenu. De ce fait, l'utilisateur est souvent amené à ouvrir plusieurs documents avant d'en trouver un qui s'accorde à ses centres d'intérêts.

Inversement, les vues locales donnent un aperçu sur le contenu de plusieurs documents. Elles fournissent une information visuelle qui aide l'utilisateur à choisir le document à consulter. Des techniques de zoom ou des vues focus+contexte peuvent être efficacement utilisées pour visualiser le détail d'un document tout en donnant une idée générale de son contexte. On peut par exemple citer le système Pad++ [Bederson96] qui utilise le zoom pour afficher simultanément plusieurs documents à des tailles variables (section 5.2.2.2). Les transformations Fisheye [Furnas86], utilisées pour augmenter le nombre de détails sur une zone d'intérêt par distorsion de la représentation, sont quant à elles une bonne illustration des vues

focus+contexte (section 5.2.1.3). A partir d'une vue locale, il est cependant difficile de comprendre la place des quelques documents visualisés par rapport à l'ensemble de l'espace informationnel, ainsi que leurs relations avec les autres documents. Ceci pose des problèmes de localisation qui compliquent la navigation.

C'est pour pallier leurs désavantages respectifs que nous proposons un environnement où ces deux types de représentations sont fortement connectés. Le passage du global au local (et respectivement) s'effectue au cours de la navigation en mêlant exploration générale et accès aux détails.

6.2 Exemples de vues globales

Représenter l'organisation d'un espace documentaire peut aider à réduire les problèmes de désorientation qui nuisent à la fluidité de la navigation. Certains systèmes [Halasz87, Yankelovich88] utilisent des vues d'ensembles ou des cartes pour supporter le processus de navigation. Cependant, une représentation complète se révèle souvent trop complexe et de trop grande taille pour être efficacement utilisable. Ceci est d'autant plus marqué lorsque les documents sont organisés au moyen d'un réseau de liens hypertextes, la topologie de ce réseau correspondant souvent à un graphe quelconque. La méthode que nous proposons consiste à afficher une sous-structure du réseau de liens (matrice, hiérarchie, etc.), assez simple à représenter et choisie en fonction de la nature de la base informationnelle, pour donner une idée rapide de la répartition des documents et en faciliter ainsi l'exploration. Dans notre environnement, une vue globale correspond donc à une représentation structurelle « simplifiée » de l'organisation des documents.

Afin d'illustrer ces idées, nous donnons ci-dessous deux exemples de vues globales, le premier appliqué à notre hyperbase littéraire dans le cadre du projet Philectre (section 2.1.3), le second appliqué au cas du Web.

6.2.1 Représentation perspective

Dans le cas de notre espace hypermédia littéraire, nous proposons une représentation matricielle de type « focus/contexte » inspirée du modèle du mur perspectif [Mackinlay91] (section 5.4.1). Ce modèle simule un effet de perspective pour mettre en évidence les objets qui sont au premier plan en réduisant d'autant plus la taille des autres objets qu'ils sont éloignés de ce premier plan. L'objectif recherché est d'attribuer davantage de place aux objets sur lesquels on est actuellement en train de travailler (le « focus ») tout en donnant une idée du « contexte » environnant.

La vue structurelle proposée tire parti de la structure quasi matricielle sous-jacente aux œuvres littéraires numérisées. Matériellement, celles-ci sont en effet initialement composées d'une succession de pages qui peuvent éventuellement comporter

plusieurs variantes ou plusieurs versions. D'autres documents (notes, commentaires, documents externes, etc.) peuvent bien sur être ensuite superposés à cette structure de base. Ce canevas élémentaire offre cependant l'avantage de permettre de structurer l'hyperespace documentaire de manière relativement intuitive car fondée sur la réalité matérielle des sources. Notre vue structurelle repose donc sur une double représentation perspective où les icônes représentent les documents sources (les fac-similés numériques de manuscrits d'œuvres littéraires) qui sont disposés horizontalement selon l'ordre des pages et verticalement selon l'ordre des versions ou des variantes de ces pages (Figure 6.1). Par correspondance, les icônes référencent également les fichiers XML d'annotation et de transcription qui sont associés aux fac-similés numériques (section 2.3).

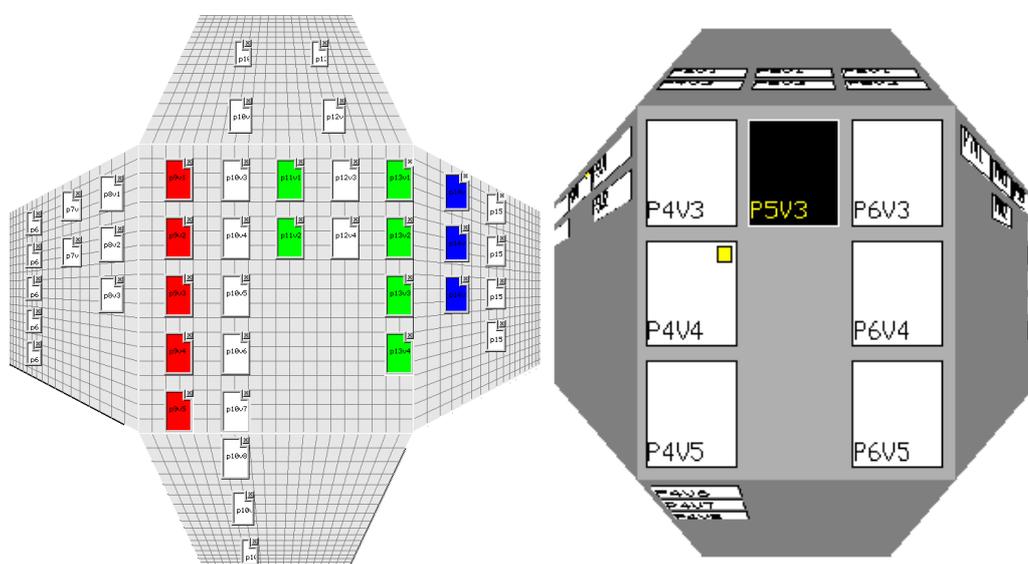


Figure 6.1 : Vue structurelle perspective (version 2D et 3D),
représentation matricielle des pages et des versions.

La surface de visualisation se compose de quatre panneaux en perspective qui entourent une partie centrale (ce qui la rend symétrique par rotation de 90°). La partie centrale est utilisée comme zone de focus alors que les quatre autres panneaux, affichés en perspective, forment son contexte. Il est également possible de faire varier le niveau de détail de la partie centrale, le zoom sur les icônes s'ajustant alors automatiquement, comme montré à la Figure 6.2.

L'exploration de l'espace documentaire s'effectue en faisant « glisser » horizontalement ou verticalement la surface de représentation de manière à approcher ou à éloigner les icônes de la zone de focus située au centre. Cette interaction se fait simplement en cliquant sur des boutons de type « points cardinaux » qui font glisser la surface de représentation dans la direction correspondante.

Enfin, cette vue structurelle est synchronisée avec le système d'annotation et de transcription présenté à la section 2.3 afin de permettre à l'utilisateur de passer de l'exploration (niveau global) à la consultation (niveau local), et vice versa, en contexte et de manière progressive. Le fac-similé numérique (d'un manuscrit) et le document XML associé sont automatiquement chargés et affichés dans la fenêtre principale du système (Figure 2.6) lors de la sélection de l'icône correspondante dans la vue structurelle. Tout changement dans l'une des deux vues entraîne la mise à jour adéquate de l'autre représentation de manière que cette correspondance reste toujours vérifiée. Cette double représentation présente certaines propriétés intéressantes. D'abord elle permet d'afficher simultanément le focus/contenu et le contexte local et global des documents sources qui compose l'hypermédia. Elle offre donc une vue d'ensemble basée sur une structuration assez simple, qui permet non seulement de donner une idée rapide de la répartition des composants élémentaires de l'hypermédia mais aussi d'accéder directement et rapidement à leur contenu en cliquant sur les icônes correspondantes. Ce type d'accès facilite le « butinage » dans la mesure où il évite d'avoir à passer par des menus ou des listes interminables d'items aux noms généralement cryptiques (on pourra d'ailleurs noter à ce propos que la vue structurelle proposée est en fait une sorte de liste, mais respectant une structure bidirectionnelle et offrant une représentation déformée visant à mettre en évidence la zone d'intérêt).

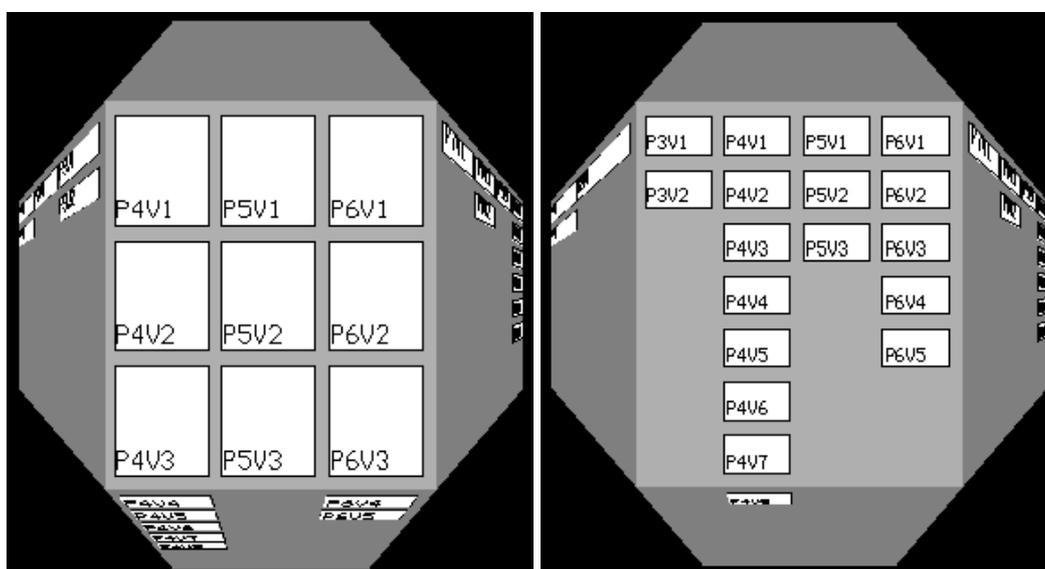


Figure 6.2 : Modularité du niveau de détail.

Dans la version 3D de l'outil (Figure 6.2 et Figure 6.3), développée avec la librairie OpenGL, le modèle original du mur a été étendu pour représenter les hyperliens qui interconnectent les sources de l'hypermédia. Afin d'éviter de surcharger l'affichage

et de le rendre ainsi inopérant, seuls sont représentés les liens partant d'une icône préalablement sélectionnée et allant vers d'autres composants.

Comme illustré à la Figure 6.3, les liens sont affichés sur la face arrière de notre vue structurelle. Il est possible de faire bouger la représentation dans l'espace pour faire apparaître diverses parties de la face avant ou arrière avec le niveau de détail souhaité. Ce déplacement spatial s'effectue au moyen d'un potentiomètre qui permet de changer interactivement la position du point de vue, et ce de manière progressive afin de ne pas perturber la perception de la représentation. Ce type d'interaction, qui est assez classique pour un utilisateur averti de systèmes de représentation tridimensionnels, peut cependant sembler moins intuitif pour d'autres types de publics. Nous envisageons donc d'adopter une autre représentation des hyperliens qui faciliterait leur manipulation en évitant des interactions 3D.

Les liens sont représentés par une droite qui part à la perpendiculaire du document source et qui se subdivise ensuite pour relier les destinations des différents liens. Cette disposition fournit une solution à l'enchevêtrement visuel des liens. Un lien supplémentaire, affiché en pointillé, indique le dernier emplacement de l'utilisateur, lui permettant de connaître la direction d'où il vient. Ce repère visuel fournit une sorte d'historique de navigation.

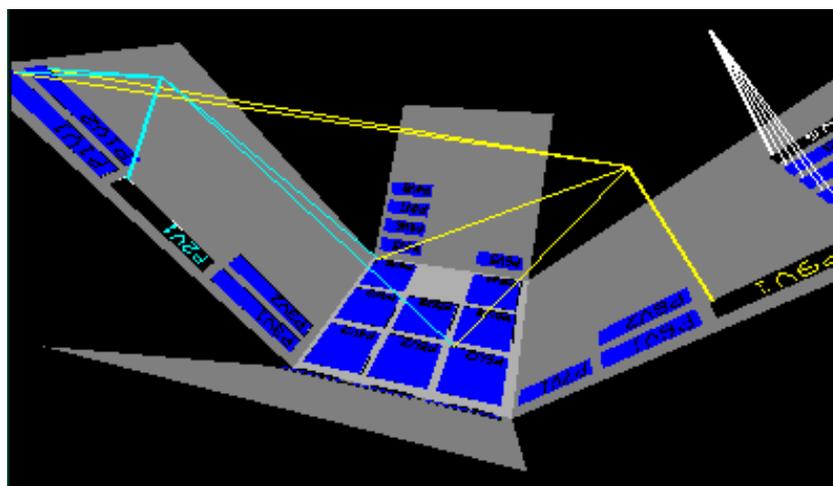


Figure 6.3 : Les hyperliens sont affichés sur la face arrière.

Le même principe pourrait être appliqué à la visualisation d'autres espaces de données, comme par exemple dans le cas d'une bibliothèque numérique où l'organisation des documents peut être représentée par des structures matricielles du type auteurs/années, auteurs/thèmes, etc. Dans le cas du Web, le mur pourrait servir à visualiser des bookmarks de pages HTML, chaque ligne regroupant les pages associées à un thème particulier.

6.2.2 Représentation concentrique hiérarchique

Dans le cas de la navigation Web, nous proposons une représentation structurelle de type hiérarchique inspirée du modèle des arbres hyperboliques développé par Xerox [Lamping96] (section 5.5.3).

Deux problèmes se dégagent du modèle original. Tout d'abord, les transformations hyperboliques résultantes de l'exploration entraînent des distorsions parfois complexes de la représentation. Il peut par exemple être difficile de savoir à quel niveau de la hiérarchie appartient un nœud spécifique. Le second problème vient du fait que la représentation est construite une fois pour toute. La totalité de la hiérarchie est positionnée dans le plan hyperbolique dès le départ, l'espace d'affichage permettant d'en visualiser un sous-ensemble qui évolue au cours de l'exploration. Il n'est ainsi pas possible de rajouter de nouveaux nœuds dynamiquement. Ceci pose problème pour le Web puisqu'il est illusoire de prendre en compte la totalité des documents. De plus, la navigation s'effectuant en cliquant sur les liens de proche en proche, l'utilisateur ne connaît pas à l'avance dans quelle partie du Web il va se retrouver.

Les systèmes que nous avons développés proposent deux variantes, basées sur des modèles 2D et 3D, de représentations hiérarchiques. Ils permettent de visualiser une arborescence de pages Web et de la faire évoluer dynamiquement au cours de l'exploration. Les URLs des pages sont recherchés de manière récursive à partir d'une page de focus. Cette recherche s'arrête lorsque la profondeur maximum d'exploration, préalablement définie par l'utilisateur, est atteinte. En parallèle, le système construit une représentation interne du graphe des hyperliens. Il extrait ensuite de ce graphe une structure arborescente, ayant pour racine la page de focus, et l'affiche à l'écran. Lors de la navigation, l'utilisateur peut changer de focus en sélectionnant un des nœuds affichés. Celui-ci devient le centre de la représentation. Il s'effectue alors une modification de la représentation interne et par conséquent de l'affichage. La recherche Web reprend, de nouvelles pages sont ajoutées alors que celles qui s'éloignent de la racine courante disparaissent.

6.2.2.1 Visualisation 2D

Cette représentation dispose l'arborescence des pages sur un disque. La racine est initialement positionnée au centre du disque. Les autres nœuds sont placés sur des cercles concentriques qui correspondent aux différents niveaux de l'arbre (Figure 6.4). L'espacement entre les divers niveaux de l'arbre peut être modifié interactivement pour changer le nombre de niveaux visualisés. Ceci permet soit d'aérer la représentation, soit d'augmenter le nombre de pages affichées.

A chaque nœud est associé un secteur angulaire qui détermine l'espace disponible pour la disposition de ses enfants. Un nœud est positionné sur son cercle et au centre de son secteur angulaire. La racine possède un secteur angulaire de 360°. Chaque nœud divise son secteur angulaire par le nombre de ses fils. Il résulte de cette opération la création de nouveaux secteurs qui sont affectés à chacun des fils. Ce processus se répète de manière récursive pour chaque niveau de la hiérarchie. Cet algorithme de placement est assez simple car la division de l'espace alloué à la descendance d'un nœud dépend seulement du nombre de ses fils. Une autre possibilité aurait été de prendre en compte le nombre de ses descendants sur plusieurs niveaux. Cependant notre représentation évoluant dynamiquement, le système ne connaît pas forcément le nombre total des descendants sur plusieurs niveaux au moment de l'affichage.

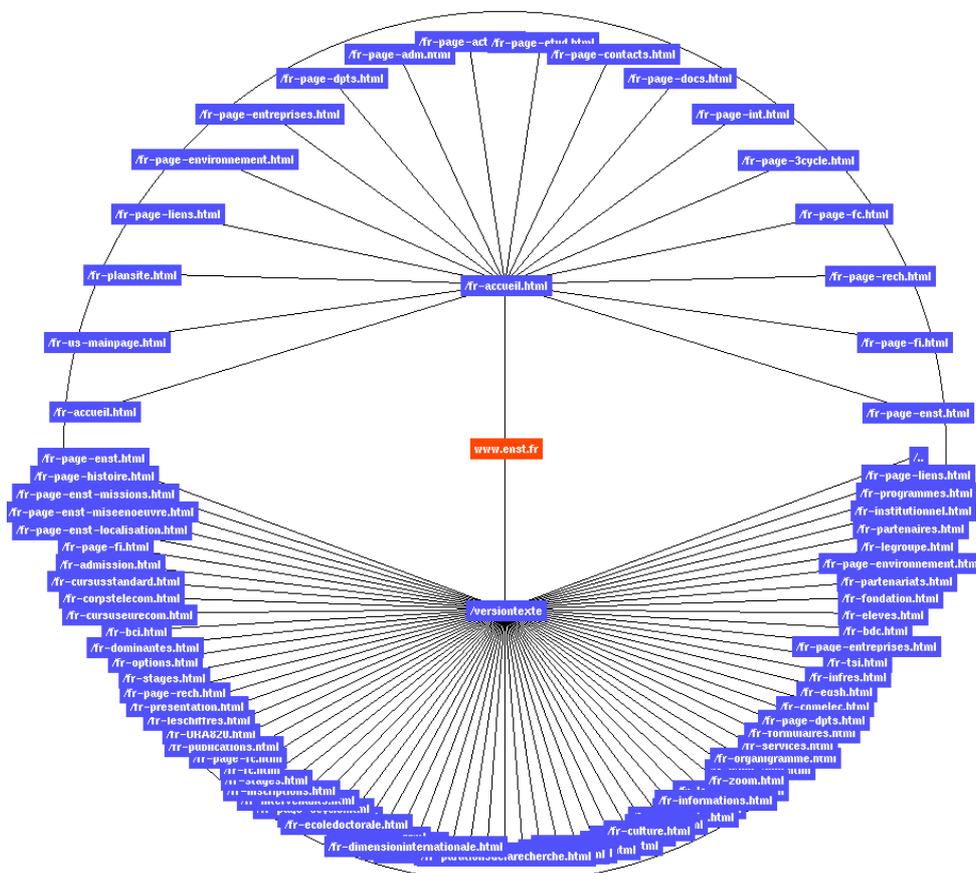


Figure 6.4 : Représentation concentrique (vue structurelle hiérarchique 2D).

Il résulte de ce choix de conception un affichage rapide qui évolue en temps réel au fur et à mesure de la navigation, alors même que le système effectue une mise à jour dynamique des pages à représenter. Cependant, le placement des nœuds fait apparaître un problème d'occlusion. Lorsque le nombre de nœuds affiché est

important, certains d’entre eux se chevauchent, ce qui empêche de lire le texte de leur label (titre ou URL de la page correspondante). Deux solutions sont proposées pour résoudre ce problème.

La première consiste à positionner la souris sur le nœud à visualiser. Celui-ci passe alors au premier plan et est agrandi afin de permettre la lecture du texte qu’il contient. Il reprend sa taille normale lorsque la souris est déplacée. Cette technique a l’avantage de ne pas modifier le positionnement des nœuds.

La seconde solution consiste à changer la position des nœuds. Ceci s’effectue en déplaçant interactivement un nœud à l’aide de la souris. Les couches circulaires de l’arbre sont déplacées en conséquence et la vue est globalement transformée (Figure 6.5). Les nœuds qui se rapprochent du centre s’espacent, alors que ceux qui s’en éloignent se resserrent. De plus des nœuds qui se trouvaient hors de l’espace d’affichage peuvent apparaître.

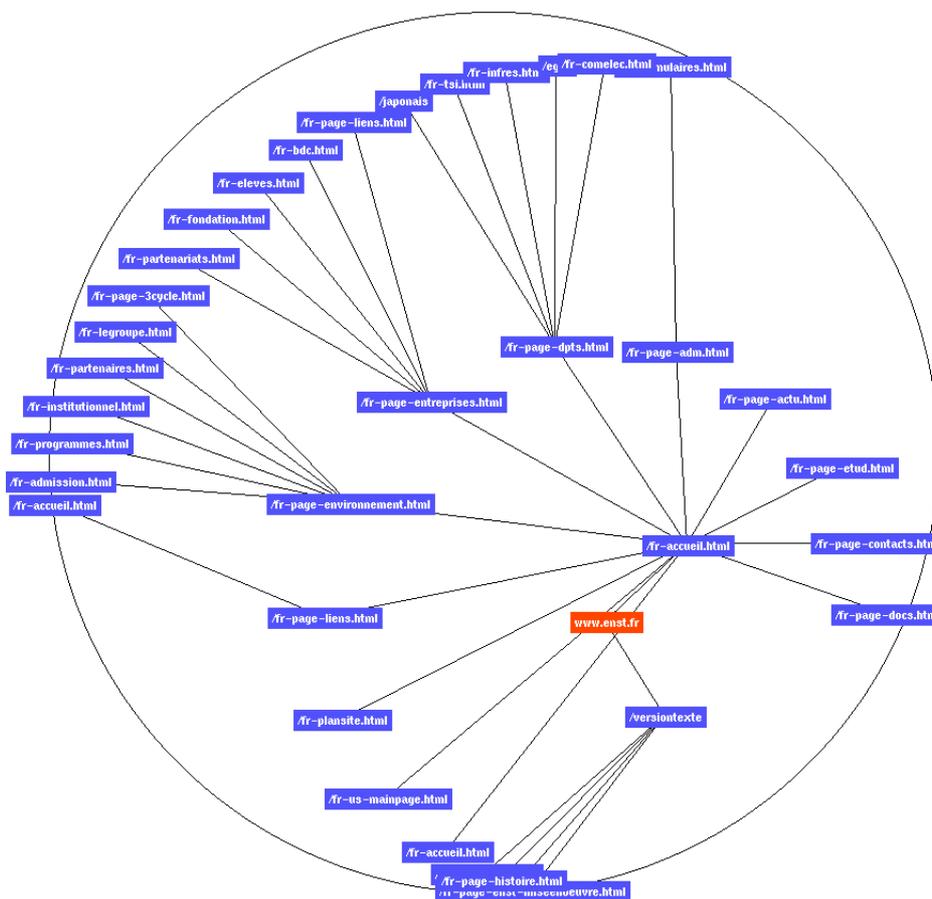


Figure 6.5 : Exploration de la hiérarchie.

Enfin, notre représentation permet d’afficher les hyperliens qui ne font pas partie de la structure hiérarchique (i.e. ceux qui forment le graphe initial du réseau hypertexte

les éléments des différents niveaux de l'arbre. Par un effet d'optique, un objet affiché sur l'arrière de la sphère pourrait être visualisé comme faisant partie d'un niveau de l'arbre affiché sur la face avant et ainsi fausser l'interprétation de la représentation.

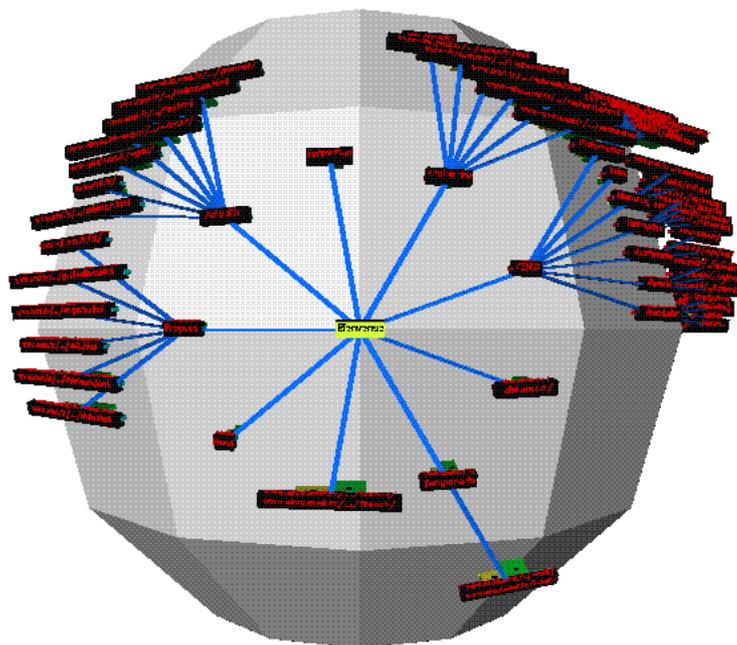
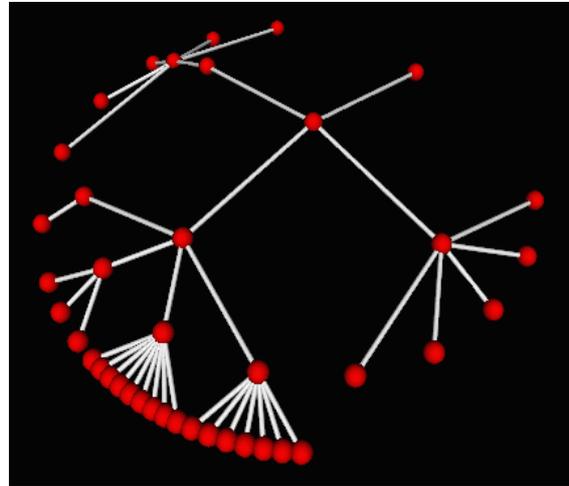


Figure 6.7 : Représentation sphérique perspective.

Pour l'instant le mode d'exploration proposé consiste simplement à se déplacer autour de la sphère. Ce mode ne semble toutefois pas intuitif à une grande majorité d'utilisateurs, la plupart n'ayant pas l'habitude des interfaces de représentation tridimensionnelles. Les observations effectuées montrent qu'ils passent beaucoup de temps à se positionner dans l'espace afin de trouver un point de vue qui leur convienne au détriment de l'exploration de l'arborescence des pages Web. De même

que le placement de l'utilisateur dans l'espace (qui est reconnu comme une activité non triviale [Eades98]), la manipulation des éléments de l'arbre pose des problèmes de par la difficulté à évaluer les distances relatives entre ces éléments.

6.3 ZoomTree : un système de vues locales

Dans notre environnement, l'accès aux documents s'effectue par un passage progressif du global au local. Les vues locales servent à affiner la recherche sur un sous-ensemble de la base documentaire trouvé à partir d'une vue globale. Elles fournissent des informations sur le contenu des documents et sur leur contexte qui aident à choisir les données à consulter.

6.3.1 Description

Notre système, appelé *ZoomTree*, permet d'afficher simultanément plusieurs documents de types hétérogènes à échelles variables ainsi que les liens existants entre ces documents. Comme illustration, nous avons appliqué ce système de visualisation à l'exploration interactive de deux types d'espaces documentaires hypermédia : le Web et l'hyperbase de données littéraires (fac-similés numériques, fichiers d'annotation / transcription et documents annexes) utilisée pour le projet Philectre (section 2.1.3). Le système de navigation repose sur l'organisation induite par les liens HTML dans le cas du Web, et sur celle induite par les liens XML dans le cas d'un hyperdocument littéraire.

Une des méthodes qui permet de réduire la désorientation consiste à fournir aux utilisateurs des clés qui les aident à mieux structurer l'information visualisée (comme par exemple des vues d'ensembles structurées) [Rouet96]. C'est à partir de cette idée qu'a été développé notre système qui présente à l'écran différentes parties de l'hyperbase affichées sous forme de vues hiérarchiques focus+contexte (Figure 6.8). *ZoomTree* tire avantage du fait que les structures hiérarchiques sont plus faciles à représenter et à comprendre que les graphes arbitraires, et que les utilisateurs ont l'habitude de manipuler de telles structures (tables des matières, systèmes d'exploration de fichiers, etc.). Chaque vue locale hiérarchique montre un petit groupe de documents « isolé » du réseau de liens afin d'être exploré en contexte. Chaque groupe représente une sous-partie de l'hyperespace exploré et se compose de quelques documents connectés entre eux par des hyperliens. Ces vues locales se matérialisent à l'écran sous forme de hiérarchies de fenêtres qui ont en charge l'affichage des documents (Figure 6.8) et que l'utilisateur peut déplacer ou redimensionner par manipulation directe au moyen de la souris. Les liens hiérarchiques (dessinés à l'écran) qui partent d'une fenêtre correspondent à des hyperliens qui partent du document contenu dans la fenêtre. Une fois que la racine de l'arbre (un élément de l'hyperdocument) a été choisie par l'utilisateur à partir d'une

vue globale (voir section précédente), *ZoomTree* construit automatiquement la structure hiérarchique en suivant les hyperliens rencontrés (chaque document n'est référencé qu'une seule fois dans la vue locale), la profondeur de l'arbre généré étant paramétrable par l'utilisateur.



Figure 6.8 : ZoomTree, un système de vues locales (appliqué à une hyperbase de fac-similés numériques).

Avantages

Ce type de représentation offre plusieurs avantages. Un premier point concerne l'accès aux données en contexte. Comme illustré à la Figure 6.8, plusieurs documents peuvent être visualisés en détail (des points de focus) alors que d'autres, formant leur contexte, sont affichés à taille réduite. Un point de focus correspond simplement à un document jugé intéressant par l'utilisateur et qui est affiché à un niveau de détails suffisamment important pour que son contenu soit lisible (des données textuelles) ou interprétable (des schémas, images, pictogrammes, etc.). La Figure 6.9 donne un exemple de vue locale qui contient un seul point de focus. Dans cet exemple les documents sont des pages Web, chacune d'elle est affichée une fenêtre de taille variable. La plus grande fenêtre a des dimensions suffisamment importantes pour permettre la lisibilité de la page Web qu'elle contient, c'est le point de focus. L'utilisateur dispose ainsi d'une représentation qui l'aide à comprendre la place d'un document particulier dans un ensemble plus important. Ceci facilite l'interprétation du document consulté, celui-ci étant rattaché à des données qui lui sont associées.

De par sa conception, notre système permet également de définir des représentations multi-focus, comme le montre la Figure 6.10. Ceci peut favoriser les associations d'idées en permettant la comparaison entre différents documents.

Enfin, l'information visuelle fournie par la représentation (mise en page des documents, nombre et forme de leurs composants, etc.) peut aider l'utilisateur à effectuer ses choix de navigation, lui évitant ainsi de fréquentes erreurs de parcours comme c'est par exemple le cas lors d'une exploration Web via un navigateur classique (utilisation fréquente du retour arrière).



Figure 6.9 : Vue locale focus+contexte de plusieurs pages Web.
Un seul point de focus.

Discussion

D'autres systèmes, comme *PadPrints* [Hightower98] et *WebTOC* [Nation97], utilisent également des vues locales hiérarchiques pour faciliter l'exploration. *PadPrints* implémente un mécanisme d'historique de navigation Web. Chaque page Web est représentée par une image. L'utilisateur garde une représentation visuelle des pages consultées, ce qui l'aide à savoir d'où il vient et quel chemin il a suivi. Les retours en arrière peuvent s'effectuer simplement. Cependant, à la différence de

ZoomTree, l'utilisateur ne dispose pas d'information l'aidant à effectuer de nouveaux choix de navigation. De même, il semble également difficile de comparer des documents distants en termes de liens. *WebTOC* génère une table des matières hiérarchique (contenant des statistiques graphiques additionnelles) comme une aide à l'exploration d'un site Web. Les utilisateurs disposent d'une vue d'ensemble sur le contenu des documents mais n'ont aucune représentation visuelle de ce contenu (images réduites, etc.). De plus, la possibilité de comparer des documents (qui peuvent être distants en termes d'hyperliens) en vue d'associer des idées (au moyen de représentations multi-focus) ne semble pas avoir été prise en compte dans cette étude.

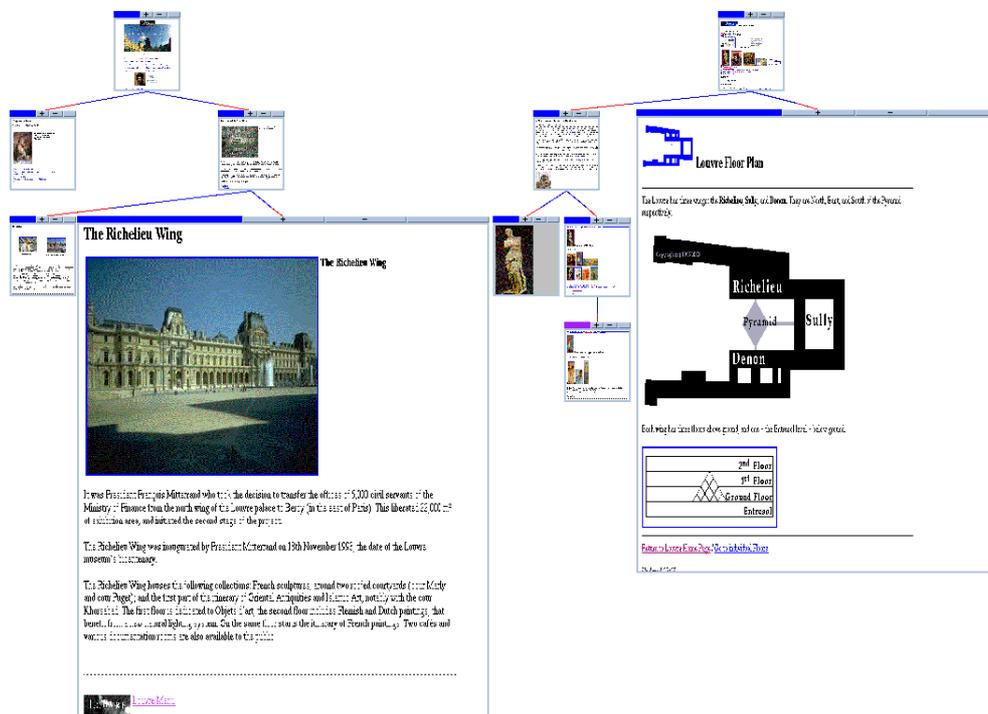


Figure 6.10 : Une représentation multi-focus.

Les vues fisheyes [Furnas86] (section 5.2.1.3) permettent de créer des représentations multi-focus [Shipman99]. La méthode consiste à augmenter l'espace alloué aux objets d'intérêts tout en diminuant la place allouée aux autres éléments de la représentation, l'ensemble des objets restant visibles. Cependant, les vues fisheyes multi-focus sont souvent difficile à comprendre à cause des distorsions spatiales qu'elles mettent en œuvre. De plus, les interactions nécessaires à la définition des paramètres de transformations sont souvent assez complexes. *Pad++* [Bederson96] (section 5.2.2.2), une interface zoomable pour la navigation, supporte le multi-focus au moyen de portails, chaque portail étant une sorte de fenêtre en charge de l'affichage d'une zone de focus. Comme le système ne permet pas la gestion automatique de ces portails, l'utilisateur doit créer, détruire et arranger manuellement

l'ensemble des portails, et ceci au détriment de l'exploration. Il ne semble pas simple de comprendre les relations entre les différentes vues proposées par les portails ni de les comparer.

ZoomTree supporte le multi-focus via un schéma d'interaction cohérent qui unifie la réalisation d'un ensemble de tâches liées à la gestion de la représentation.

6.3.2 Interaction

Le modèle d'interaction proposé, basé sur un paradigme de *zoom*, permet la manipulation rapide et intuitive des éléments affichés dans la représentation. Ceci évite aux utilisateurs d'avoir à se concentrer sur l'interaction au dépend de leurs tâches principales, à savoir la recherche d'information par navigation et la consultation des documents. Dans *ZoomTree*, le *zoom* est utilisé pour agir sur la représentation du contenu des documents et pour organiser l'espace d'affichage (gestion du focus et du contexte, manipulation de groupes de documents, etc.).

Redimensionnement des fenêtres et zoom de leur contenu

Les utilisateurs disposent de deux méthodes pour modifier la taille d'une fenêtre (et par conséquent le niveau de *zoom*). La première, le *zoom direct*, permet de redimensionner une fenêtre par manipulation directe en tirant dessus au moyen de la souris. La seconde, le *zoom quantique*, permet de changer simultanément la largeur et la hauteur d'une fenêtre de valeurs préalablement définies (paramétrables par l'utilisateur) en cliquant sur un bouton. Le *zoom direct* offre une plus grande précision mais nécessite une interaction un peu plus complexe tandis que le *zoom quantique* est plus rapide mais plus approximatif. Le choix de l'une ou l'autre de ces méthodes est laissé à l'utilisateur qui déterminera la plus appropriée en fonction de la tâche à réaliser, du niveau de précision désiré, etc.

Changer interactivement la taille d'une fenêtre induit une opération de *zoom* sur le document qu'elle contient. Tous les éléments du document (texte, images, etc.) sont redimensionnés tout en respectant la mise en forme initiale : ils sont toujours disposés de la même manière les uns par rapport aux autres. La Figure 6.11 illustre ce principe dans le cas d'une page HTML. On peut remarquer que les proportions des images et des tableaux sont respectées et que la taille du texte s'adapte en fonction des dimensions de la fenêtre. Pour ce faire, nous avons choisi une implémentation qui favorise la rapidité d'exécution ainsi que la lisibilité du texte. Le changement d'échelle des images se fait selon des valeurs fractionnelles par suppression ou addition de pixels, sans calcul d'interpolation coûteux en temps. Le changement d'échelle du texte s'effectue en employant un ensemble pré-déterminé de polices fixes qui correspondent aux différentes tailles souhaitées. Ces polices sont calculées une fois pour toutes et leur forme a été spécifiquement étudiée pour rester

lisible même à une échelle relativement petite. Le zoom n'est donc pas parfaitement continu mais respecte la disposition spatiale, qui reste cohérente, des différents éléments du document.

Ce comportement diffère de ceux rencontrés par exemple dans les navigateurs Web. En effet, changer la taille de la fenêtre d'un navigateur courant (comme Netscape Navigator) entraîne une réorganisation spatiale des éléments contenus dans la page Web visualisée. En comparaison, notre choix de conception offre plusieurs avantages. Les utilisateurs ne sont pas perturbés par des changements fréquents de mise en forme dus aux changements de taille. Les documents affichés à petite taille, comme résultat d'une opération de zoom, restent visuellement identifiables (en grande partie grâce à la préservation de leur mise en page), l'utilisateur se basant sur sa mémoire visuelle pour les reconnaître.

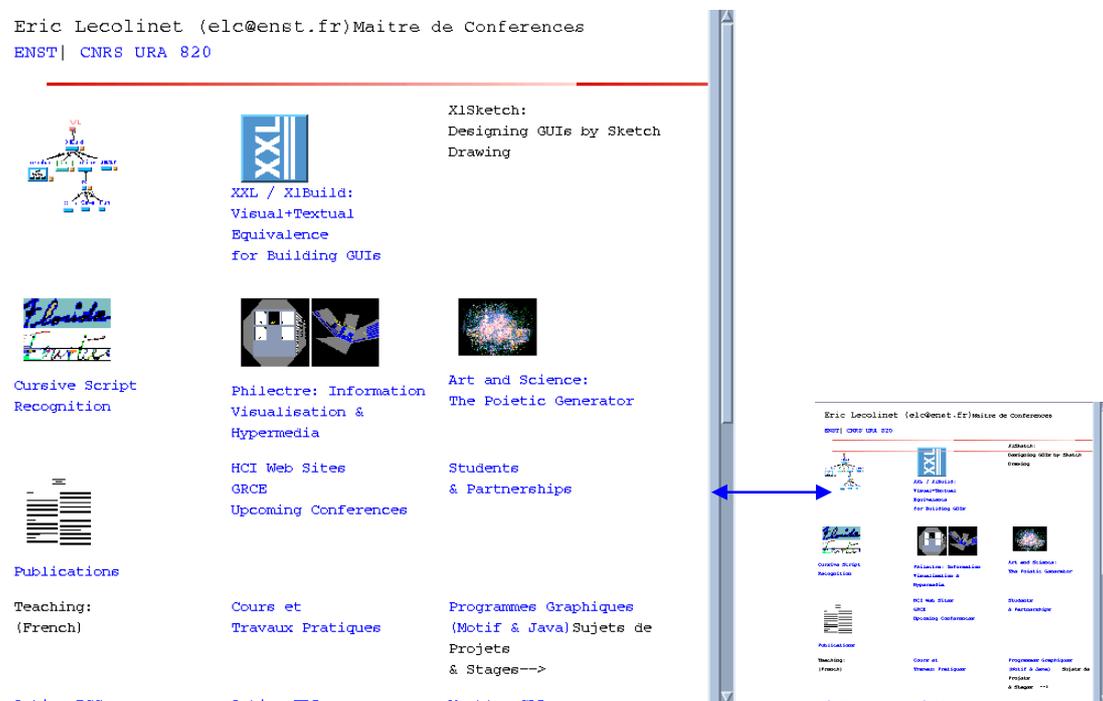


Figure 6.11 : Changement de taille proportionnel d'une page Web.
La mise en page est respectée.

L'interaction proposée unifie deux types d'opérations en une action : le zoom sert à gérer le niveau de détail des documents et la répartition spatiale entre les *fenêtres documents* de la représentation. Il est local et s'applique directement à ces éléments. Le zoom ne sert pas à se déplacer dans l'espace documentaire, comme c'est par exemple le cas dans les systèmes *Pad++* [Bederson96] ou *Zomit* [Pook00a] (section 5.2.2.2). Avec de tels systèmes, l'accès aux données s'effectue par une succession de zooms qui modifient l'ensemble de la représentation (le zoom est global et non

borné). Cependant, il s'avère souvent difficile de comprendre la structure de l'espace, de trouver un élément particulier ou de comparer des objets distants.

Zooms collectifs

Dans *ZoomTree*, les zooms interactifs permettent également de retailer/manipuler des ensembles de documents (et les fenêtres correspondantes). Il est ainsi possible de redimensionner un groupe de fenêtres en une seule interaction. Cette possibilité évite à l'utilisateur d'avoir à définir la répartition entre focus et contexte en désignant explicitement la taille de chaque fenêtre, ce qui pourrait s'avérer fastidieux. C'est également un moyen de simplifier les opérations d'organisation spatiale des fenêtres.

Plusieurs techniques de zooms « collectifs » sont ainsi proposées à l'utilisateur pour la manipulation des groupes de « fenêtres documents », parmi lesquelles le *zoom hiérarchique* et le *zoom contextuel*.

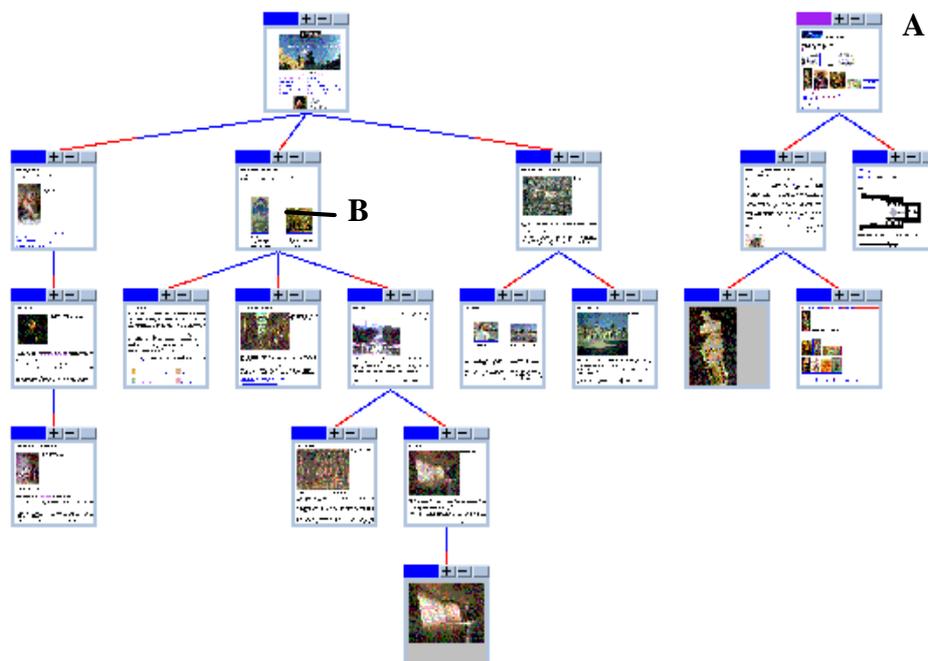


Figure 6.12 : Vue initiale des Figures Figure 6.13 et Figure 6.14.

Le *zoom hiérarchique* permet, en retillant une fenêtre donnée, de modifier le niveau de zoom de cette fenêtre et de tous ses descendants. Cette opération préserve la différence initiale de taille (le ratio) entre les fenêtres manipulées. Il est ainsi possible, par exemple, d'augmenter ou de diminuer en une seule interaction la taille d'un groupe de documents. Ceci est utile pour focaliser sur une zone d'intérêt ou pour garder, à taille réduite et de manière identifiable, des documents déjà consultés

dont on pourrait encore avoir besoin. La Figure 6.13 montre le résultat d'un zoom hiérarchique appliqué au sous-ensemble A de la Figure 6.12.

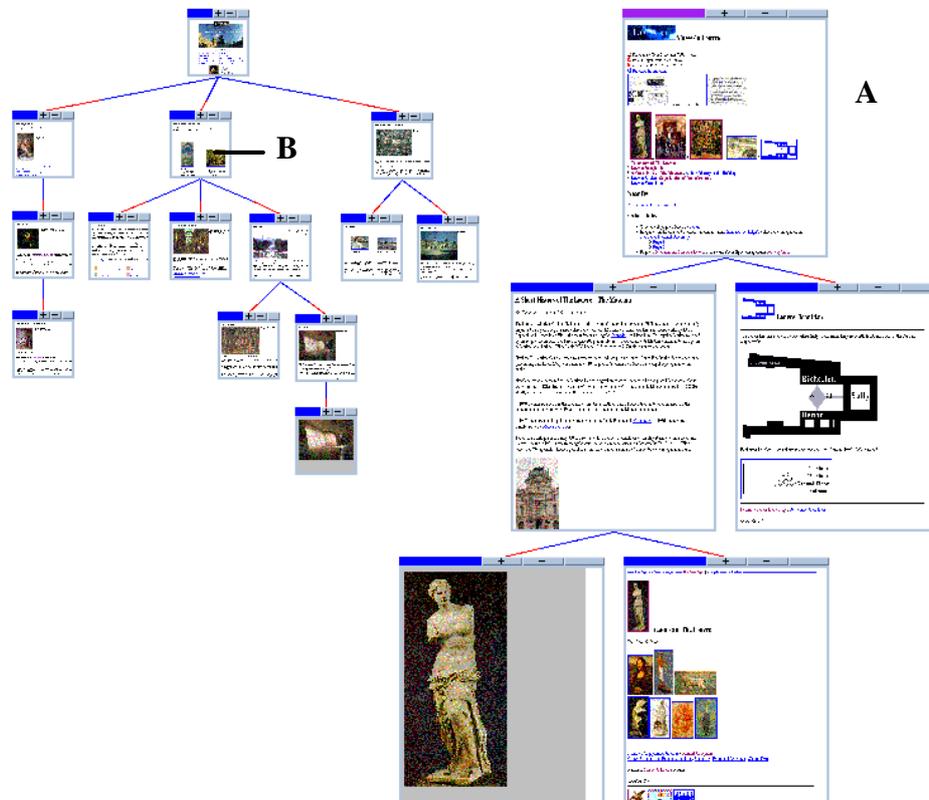


Figure 6.13 : Zoom Hiérarchique sur la partie A de la Figure 6.12.

Le *zoom contextuel* permet de retailler tous les documents en fonction de leur relation avec la fenêtre sélectionnée. Toutes les fenêtres qui sont directement ou indirectement liées à la sélection (ses ascendants et ses descendants) sont redimensionnées de manière proportionnelle à leur distance (en terme de liens) au document sélectionné. Les autres fenêtres sont retaillées dans le sens inverse. La Figure 6.14 présente une transformation résultante d'un zoom contextuel. La fenêtre B (la sélection) est retaillée et agrandie de manière à ce que son contenu soit lisible. Les documents qui figurent dans son contexte (ceux qui sont liés à B dans cette représentation) sont agrandis de manière progressive alors que la taille des autres documents (ceux appartenant à la partie A) est réduite. Il est ainsi possible de focaliser sur un document et d'en faire apparaître le contexte en une interaction tout en diminuant la taille de ce qui ne nous intéresse plus pour le moment. De plus l'utilisateur garde un contrôle explicite sur la place allouée à cet ensemble de documents.

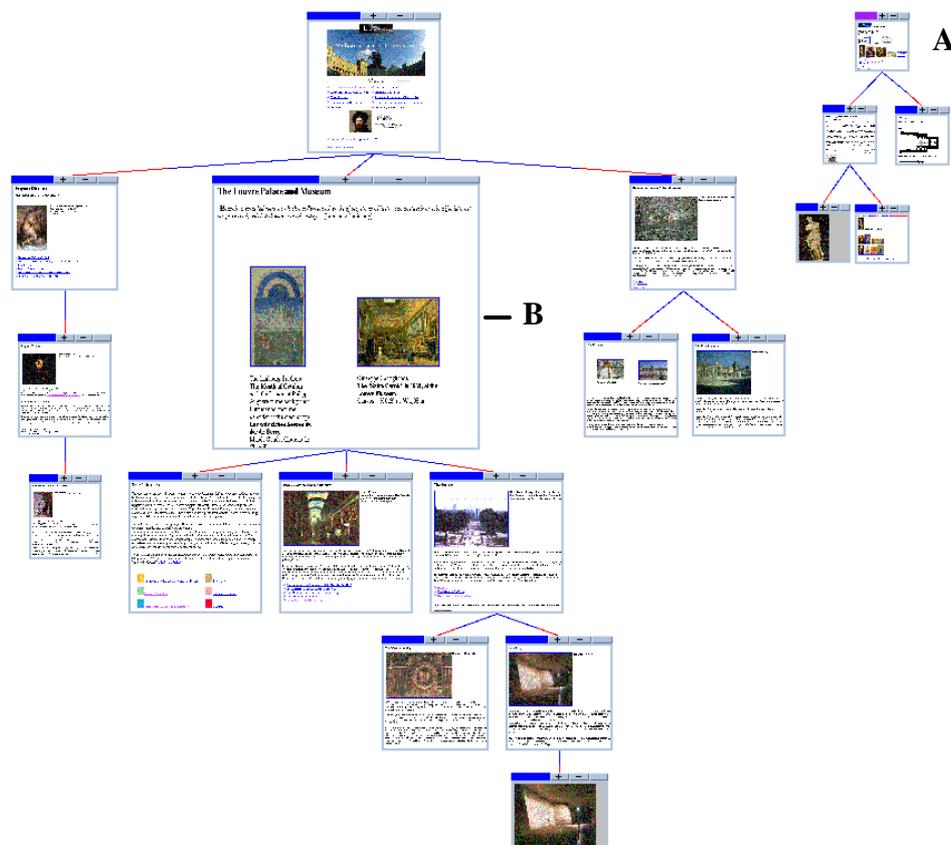


Figure 6.14 : Zoom contextuel à partir de B sur la Figure 6.12.

Autres fonctionnalités

ZoomTree dispose également d'un zoom global pour redimensionner toutes les fenêtres présentes dans l'espace de travail de manière uniforme. La largeur et la hauteur des fenêtres sont toutes deux augmentées ou diminuées d'une valeur définie par l'utilisateur lorsque ce dernier active des boutons d'incrément. Une autre fonctionnalité permet de remettre toutes les fenêtres de la représentation à la même taille (les documents qu'elles contiennent sont alors tous visualisés avec le même niveau de détails). Cette taille est soit spécifiée par l'utilisateur soit déterminée automatiquement par un calcul de moyenne. Cette fonctionnalité fournit un moyen rapide d'uniformiser l'affichage qui peut alors servir de point de départ à une nouvelle exploration.

Dans la même optique, et ce afin de préserver la cohérence du paradigme d'interaction proposé, le zoom est également identifié à une opération de fermeture. Lorsqu'une fenêtre approche d'une taille minimum, prédéfinie dans le système mais modifiable par l'utilisateur, l'arbre, ayant pour racine cette fenêtre, est automatiquement « fermé ». La fenêtre elle-même reste affichée à l'écran (ainsi que le document qu'elle contient) à sa taille minimum. Par contre, toutes ses descendantes

hiérarchiques disparaissent de l'écran. La Figure 6.16 montre le résultat d'une telle opération appliquée à la fenêtre A de la Figure 6.15. Un artefact de couleur permet aux utilisateurs d'identifier les fenêtres qui ont été fermées. Cette technique fournit un moyen efficace de gérer l'espace écran. Les documents dont l'utilisateur n'a plus besoin à un moment donné peuvent temporairement disparaître de l'affichage au profit d'autres documents jugés plus intéressants. L'utilisateur peut cependant les faire réapparaître à tout moment en agrandissant la fenêtre qui les encapsule (la racine de l'arbre fermé, la fenêtre A dans l'exemple de la Figure 6.16).

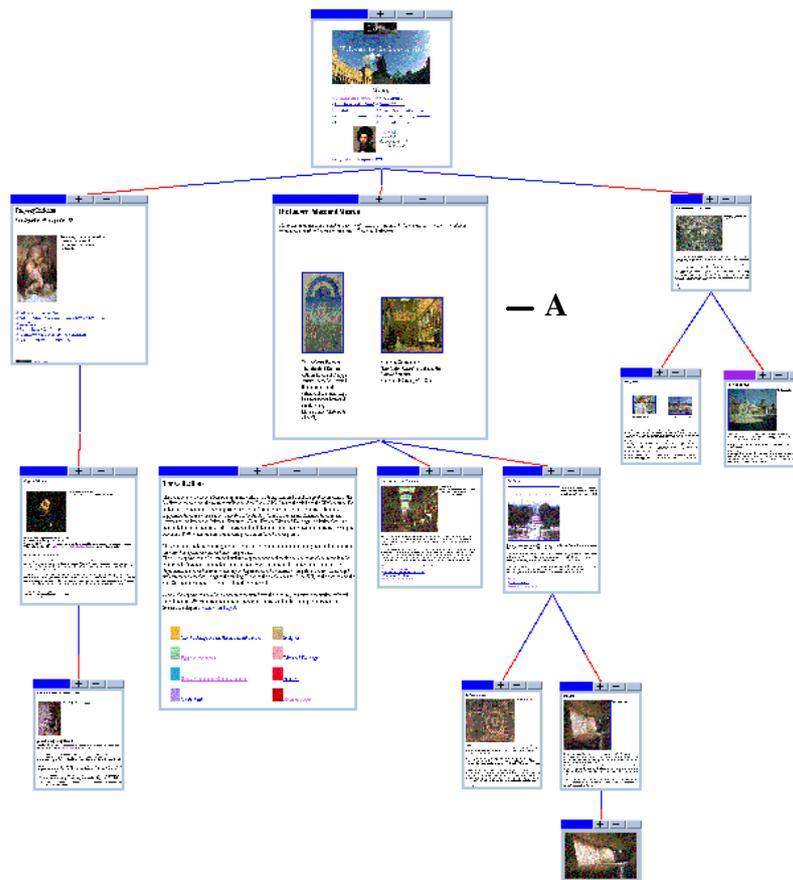


Figure 6.15 : Vue initiale de la Figure 6.16 (avant fermeture).

Les changements de taille et les fermetures entraînent une restructuration automatique de la représentation. L'affichage se transforme en conséquence : les fenêtres sont repositionnées et/ou retaillées. Si les changements s'avèrent être trop complexes, un processus d'animation est automatiquement déclenché pour permettre de comprendre la réorganisation. L'animation permet aux utilisateurs de suivre la transformation pas à pas et ainsi de réduire la surcharge cognitive nécessaire à la compréhension de la nouvelle disposition des fenêtres.

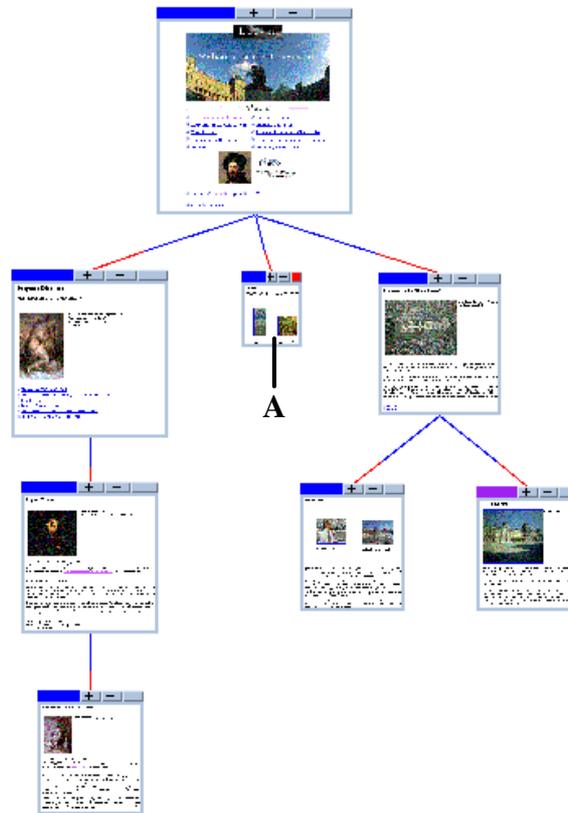


Figure 6.16 : Résultat d'une opération de fermeture appliquée à la fenêtre-document A de la Figure 6.15.

6.4 Conclusion

Dans ce chapitre, nous avons montré comment aider les utilisateurs à explorer un espace de données hypermédias en stimulant leurs capacités perceptives. Le système de navigation développé met en avant trois types de procédés.

Le premier consiste à représenter les données au moyen d'une structure plus simple à afficher (principalement hiérarchique dans notre cas), à comprendre et à manipuler que le graphe initial du réseau d'hyperliens. Ceci revient, selon la terminologie de Vernier [Vernier01] (section 5.1.4), à adopter un point de vue sur les données.

Le second procédé repose sur un fort couplage entre vues globales et vues locales qui facilite la transition entre le survol rapide d'un ensemble de données et l'accès au détails en vue d'une consultation. L'idée consiste à faciliter la navigation contextuelle afin de réduire la désorientation.

Le troisième procédé consiste à fournir aux utilisateurs un schéma d'interaction uniforme qui permet à la fois d'organiser l'espace d'affichage (gestion du focus et du contexte, manipulation de groupes de documents, etc.) et de contrôler le niveau de

détail des données visualisées. Un tel procédé évite aux utilisateurs d'avoir à se concentrer sur l'interaction au détriment de leur tâche principale. Dans notre système (*ZoomTree*), ce schéma est implémenté au moyen de techniques de zoom.

7. Retrouver l'information

Dans un espace informationnel fréquemment exploré, il est souvent utile de garder une « trace » des documents jugés intéressants afin de pouvoir les retrouver rapidement. Lors d'une consultation ultérieure, l'utilisateur dispose d'un accès direct aux documents au moyen de ces traces et n'a donc pas besoin d'effectuer une nouvelle recherche (par navigation ou requêtes) sur l'ensemble de la base informationnelle. Cette possibilité nécessite toutefois la mise en place de mécanismes pour stocker, représenter et, surtout, organiser ces traces. L'organisation est une notion clé car elle influe fortement sur l'efficacité avec laquelle les utilisateurs vont retrouver leurs documents. Card, Robertson et Mackinlay introduisent le concept d'espace de travail informationnel (« information workspace », section 5.7) pour désigner les environnements dans lesquels les documents d'intérêts peuvent être comparés, manipulés et stockés en vue d'une utilisation future [Card91]. Les bureaux de travail de type iconique, les mécanismes de bookmarks des navigateurs Web traditionnels (section 4.3.2) et des systèmes comme le Web Forager [Card96] (section 5.7.1) ou le Data Mountain [Robertson98] (section 5.7.1) en sont de bons exemples. Comme alternative, nous proposons un système de visualisation dédié à la gestion de bookmarks dont la conception repose sur deux principes. Le premier consiste à permettre aux utilisateurs d'organiser les documents en mêlant structuration physique et structuration sémantique. Le second s'appuie sur l'aptitude des utilisateurs à mettre à profit leurs capacités mémorielles afin de retrouver les documents. Ce système de bookmarks a été appliqué au cas de l'exploration du Web et de notre base de fac-similés numériques (section 2.1.2).

La définition de notre système prend également en compte les recommandations générales de conception proposées par Abrams, Baecker et Chignell à partir d'une étude comportementale sur l'utilisation de bookmarks Web [Abrams98]. Ces recommandations sont au nombre de quatre et s'énoncent comme suit :

1. les bookmarks doivent être simple à organiser,
2. des techniques de visualisation sont nécessaires pour fournir des vues d'ensembles compréhensibles de grands ensembles de bookmarks,
3. des représentations « riches » des pages Web sont requises, de nombreux utilisateurs remarquent que le titre des sites n'est pas un descripteur suffisamment informant du contenu,
4. les outils de gestion de bookmarks doivent être couplés aux navigateurs Web.

7.1 Mémoire spatiale et visuelle

Les choix de conception de notre système reposent sur des observations qui viennent de la littérature sur l'étude de la mémoire. Ces observations nous ont permis d'identifier un ensemble de clés qui permettent d'aider les utilisateurs à retrouver plus facilement leurs documents grâce à une stimulation de leur mémoire visuelle et spatiale.

7.1.1 Localisation spatiale

De nombreux travaux ont été réalisés afin d'étudier le rôle de la mémoire spatiale sur notre habilité à naviguer et à retrouver l'information dans des environnements virtuels [Jones86, Darken93, Darken96, Ruddle97, Waller98]. Darken et Silber préconisent l'acquisition d'une carte mentale de l'espace pour faciliter l'exploration [Darken96]. Ils proposent plusieurs principes de conception à suivre pour favoriser cette acquisition. Ces principes ont été validés de manière empirique et incluent l'utilisation de repères directionnels, de chemins, de frontières et de cartes. Ils ont également observé que si l'espace n'est pas divisé en utilisant ces principes d'organisation, alors les utilisateurs lui imposent leur propre organisation conceptuelle. Cette observation a été validée par de nombreuses études qui ont également mis en évidence que l'organisation spatiale imposée par les utilisateurs évolue généralement au cours du temps en fonction de leurs besoins [Franklin92, Montello93, Ruddle97].

La plupart des systèmes décrits au chapitre 5 utilisent des principes de cognition spatiale, de manière intentionnelle ou non. Les systèmes qui mettent en œuvre des mécanismes de positionnement automatique aident les utilisateurs à reconnaître et à comprendre les relations spatiales entre les objets de l'affichage (en 2D et en 3D). C'est par exemple le cas du système Treemap [Johnson91] qui affiche des vues d'ensembles hiérarchiques sur des collections de documents (section 5.5.1), ou encore du système SemNet [Fairchild88] qui affiche une représentation 3D d'un graphe de documents. D'autres systèmes, comme Workspace [Ballay94], le Web Forager [Card96] (section 5.7.1) ou encore le Data Mountain [Robertson98] (section 5.7.2) autorisent les utilisateurs à positionner eux-mêmes les documents afin de les aider à les retrouver ultérieurement. Le fait d'organiser les objets « à la main » permet de stimuler la mémoire spatiale [Czerwinski99].

7.1.2 Organisation

De la littérature sur le fonctionnement de la mémoire, il apparaît également que les gens ont une inclination naturelle à regrouper les informations dans leur mémoire lorsqu'ils peuvent le faire, et que cette organisation influe fortement sur la manière

dont ils se remémorent ces informations. Dans une de ses expérimentations, Bousfield [Bousfield53] demande à un ensemble de sujets de réécrire une liste de soixante noms appartenant à quatre catégories distinctes (animaux, personnages, métiers, végétaux) et présentés dans un ordre aléatoire. Les sujets ont une forte tendance à réécrire les noms de la liste par catégories, mettant en évidence ce que Tulving appelle l'organisation subjective (en opposition à l'organisation imposée par les expérimentateurs ou par un quelconque système informatique) [Tulving62]. Pour Mandler, tout type d'organisation est une aide mnémonique.

« memory and organisation are not only correlated, but organization is a necessary condition for memory... all organizations are mnemonic devices » [Mandler67].

Bower montre que le fait d'organiser les informations permet d'améliorer leur mémorisation à long terme [Bower69]. Pour cela, il demande à deux groupes de sujets d'apprendre un ensemble de mots présentés sous la forme d'une hiérarchie. Ces mots sont classés de manière compréhensible pour le premier groupe, et de manière aléatoire pour le second groupe. Sur quatre essais de mémorisation, le premier groupe réalise un score d'exactitude de 100%, alors que le second n'atteint que 62%.

Ces travaux montrent qu'il est important de permettre aux utilisateurs de définir et de gérer leur propre organisation des documents. Cette organisation sémantique peut être supportée par l'utilisation de structures physiques comme les listes, les hiérarchies, les graphes, etc. Il est intéressant de remarquer que les caractéristiques des hiérarchies facilitent les procédés de classification. De plus, elles permettent de décrire la structure physique des données sur disque. Nous faisons également l'hypothèse, sur la base des observations faites sur la mémoire spatiale, que l'organisation spatiale des documents fournit une aide mnémonique importante aux utilisateurs.

7.1.3 Représentation des documents

En plus des repères de localisation spatiale et d'organisation sémantique, il s'avère important de donner aux utilisateurs des repères concernant le contenu des documents afin de les identifier plus simplement. Les procédures mnémoniques classiques mettent en évidence l'utilisation d'images et de labels textuels pour faciliter la mémorisation de l'information [Jones86, Patten90, Czerwinski99]. Des études sur les bookmarks Web [Robertson98, Czerwinski99] ont montré que les utilisateurs préfèrent que les pages Web soient représentées sous forme d'images réduites. Cependant la pertinence de ce type de représentation dépend fortement du contenu du document et de sa mise en page. Il peut être difficile, voire impossible, de différencier des images réduites de documents qui ont une mise en page similaire. Dans ce cas, l'utilisation d'un label textuel (qui donne par exemple le titre du

document) peut aider à faire la différence. Selon Underwood, la multiplication des clés mémorielles est un avantage [Underwood60]. Plus le nombre de clés fournit à l'utilisateur pour stimuler sa mémoire est important, plus la probabilité qu'il retrouve l'information est élevée.

7.2 Tables de travail 2D

7.2.1 Description

Le système de bookmarks que nous avons développé est constitué d'un ensemble de surfaces de visualisation bidimensionnelles organisées de manière hiérarchique. Sur chacune de ces surfaces, appelées *tables de travail*, l'utilisateur peut stocker et organiser une représentation iconique des documents (Figure 7.1). Ces icônes permettent de garder une trace des documents préalablement trouvés et jugés intéressants lors d'une phase d'exploration de la base informationnelle. Un simple clic souris sur l'une de ces icônes entraîne le chargement du document correspondant dans un logiciel qui permet l'affichage de ce type de document. A titre d'illustration, nous avons connecté notre système de bookmarks au navigateur Web Netscape, ce qui permet de traiter les deux types de documents utilisés dans nos exemples, à savoir les pages HTML et les fac-similés numériques (des images bitmaps). Comme illustré à la Figure 7.1, chaque table de travail peut être vue comme un espace visuel de bookmarks.

Une icône est une image réduite du document qu'elle référence. Cette image est créée au moment de la consultation du document par capture d'écran. Une fois l'icône créée, l'utilisateur la positionne sur la surface de visualisation à l'endroit et à la taille désirée. Chaque icône peut être déplacée interactivement par « drag and drop ». Leur taille peut également être modifiée par manipulation directe en « tirant dessus » au moyen de la souris, ce qui entraîne un zoom de l'image. Ceci permet de mettre en correspondance le niveau de détail d'une icône et l'intérêt porté au document correspondant à un moment donné, un objet plus grand attirant plus facilement l'attention. De plus, l'utilisateur peut regrouper, en terme de localisation spatiale, des documents en fonction de son propre modèle cognitif des données.

L'utilisateur définit ainsi interactivement sa propre organisation conceptuelle de l'espace informationnel, celle-ci pouvant évoluer selon son activité. Il établit des schémas de classification et de relation entre les documents en se basant sur des propriétés visuelles et spatiales. A chaque exploration, l'utilisateur se retrouve dans un environnement familier. Ceci lui évite d'avoir à reconstruire un modèle mental de l'hyperespace documentaire : où se trouvent les documents intéressants ? comment faire pour y accéder ? etc., réduisant du même coup la surcharge cognitive associée à ces différentes tâches.

Pour retrouver les documents avec plus d'efficacité, les utilisateurs s'appuient sur leurs capacités mémorielles. La localisation et l'organisation spatiale des icônes stimulent leur mémoire spatiale. La représentation sous forme d'images réduites stimule leur mémoire visuelle. Afin de permettre aux utilisateurs d'identifier des documents représentés par des images réduites difficilement différenciables, le système offre également la possibilité d'associer à chacune de ces images un label textuel. Ce label peut contenir le titre du document ou toute autre information sémantique jugée pertinente par l'utilisateur.

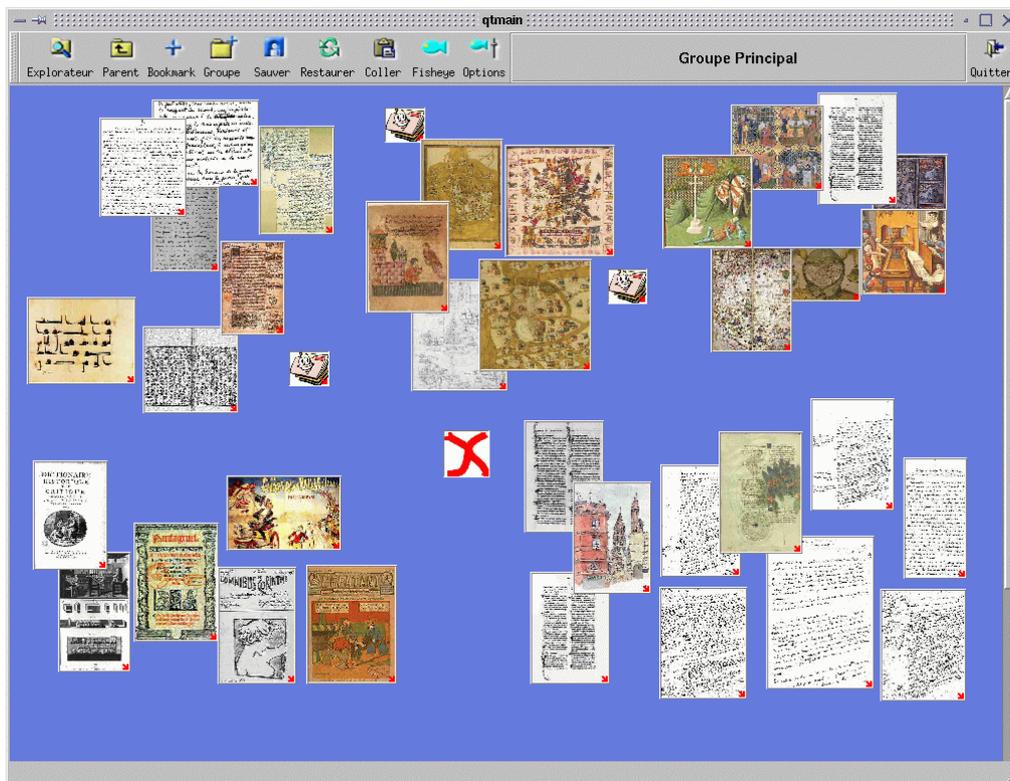
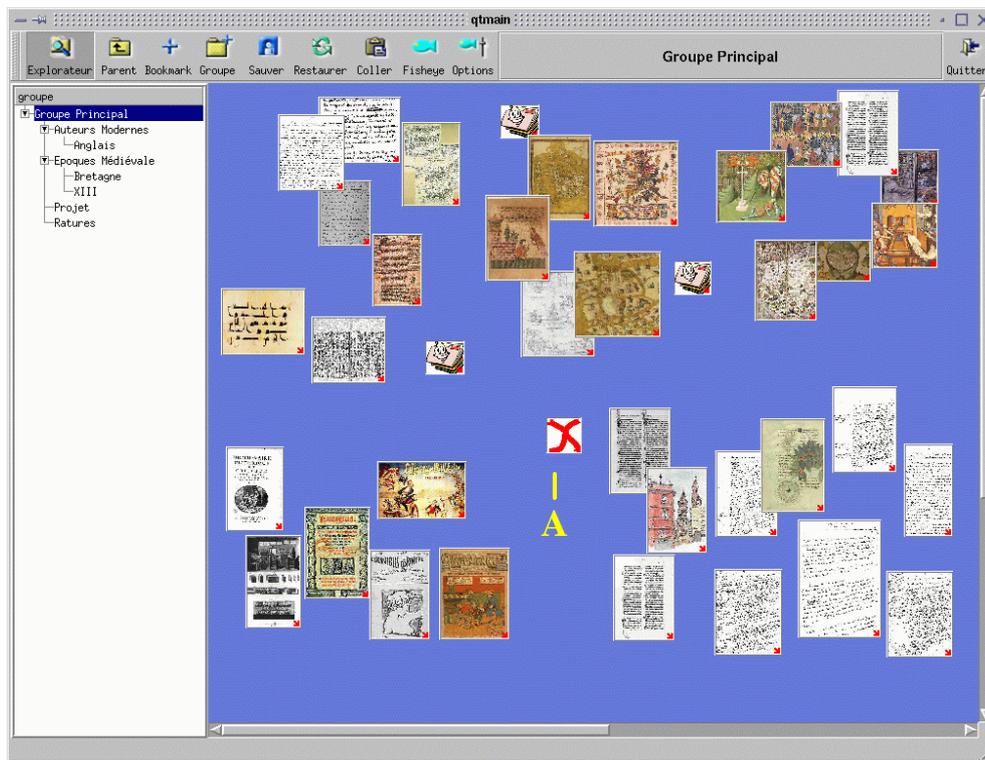


Figure 7.1 : Une table de travail 2D.
Organisation logique et personnalisée des documents.

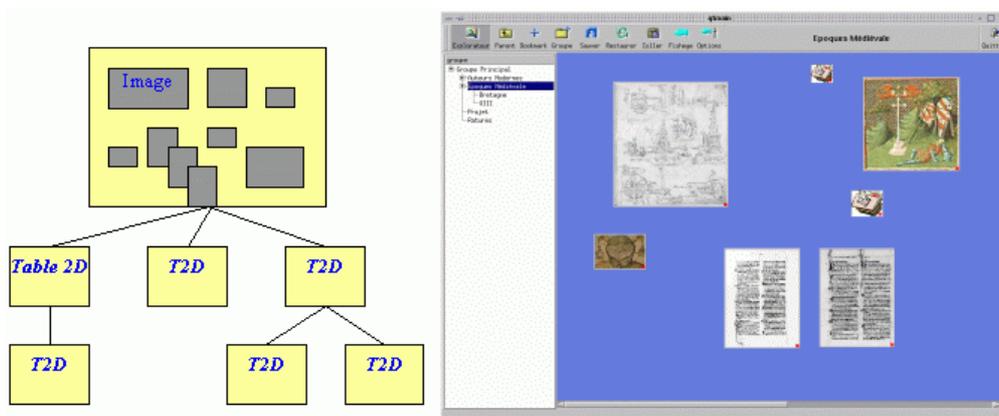
7.2.2 Structuration hiérarchique

Afin de ne pas être limité à une seule table de travail, notre système permet aux utilisateurs de créer et de gérer une hiérarchie de telles surfaces. Ce choix de conception résulte de plusieurs observations.

Les utilisateurs ont remarqué qu'à partir d'un trop grand nombre d'icônes affichées sur une table de travail, il devient difficile d'identifier leurs relations spatiales. Dans un tel cas, il semble également que les utilisateurs passent beaucoup de temps à définir l'organisation qui leur convient.



(a) La table de travail racine (niveau un de la hiérarchie).



(b) Représentation schématique. (c) Une table de niveau deux, « fille » de la racine.

Figure 7.2 : Organisation hiérarchique des tables de travail.

L'arborescence des tables de travail est affichée dans l'explorateur à droite.

A partir de ces remarques, associées aux études empiriques qui ont montré que les utilisateurs ont tendance à agir de manière répétitive avec de petits ensembles de documents [Card84, Henderson86], nous avons fait l'hypothèse qu'il est préférable de fournir aux utilisateurs plusieurs tables de travail. Chaque table de travail peut

ainsi être dédiée à une tâche spécifique (la constitution d'un dossier de presse, d'un mémoire de thèse, etc.), à un sujet donné (la visualisation d'information, l'empire romain, etc.) ou encore à un espace documentaire particulier (le Web, une base littéraire, etc.). Les tables de travail sont organisées de manière hiérarchique (Figure 7.2), mais la gestion de cette organisation est sous le contrôle de l'utilisateur. Le système propose donc un double niveau d'organisation sémantique : le premier au niveau de la hiérarchie, le second au niveau d'une table de travail, les documents pouvant être regroupés par localisation spatiale.

Le choix de ce type de structure s'explique par le fait que les hiérarchies sont simples à représenter et à comprendre, et que les utilisateurs ont l'habitude de les manipuler (tables des matières, systèmes d'exploration de fichiers, etc.). De plus, ce type d'organisation permet de mettre en correspondance le stockage physique des images réduites et leur organisation sémantique dans le système de bookmarks. En effet, chaque table de travail de la hiérarchie correspond à un dossier du système de gestion de fichiers. Toute modification interactive de la hiérarchie (création ou suppression d'une table, d'une icône, etc.) effectuée par l'utilisateur entraîne une mise à jour automatique du système de gestion de fichiers.

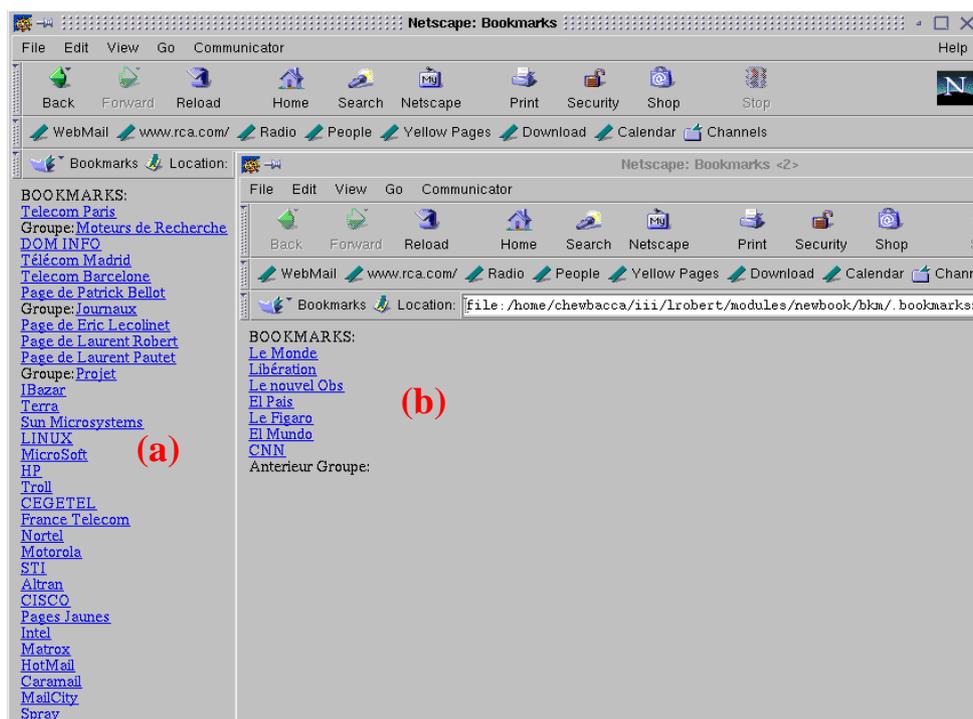


Figure 7.3 : Compatibilité des bookmarks entre les tables de travail et les navigateurs Web classiques.

- (a) la liste qui correspond à la table racine (niveau un de la hiérarchie). (b) la liste qui correspond à une table de niveau deux, « fille » de la racine.

Codage des données

Les données qui permettent de décrire la hiérarchie des tables de travail et le positionnement spatial des icônes sur chacune de ces tables sont codées en HTML. Chaque dossier du système de fichier associé à une table de travail contient un fichier HTML qui décrit le contenu de cette table. Cette approche assure la compatibilité entre notre système et les navigateurs Web traditionnels, ce qui favorise entre autres l'échange de bookmarks. Dans le navigateur Netscape, nos bookmarks sont affichés sous forme de listes (Figure 7.3). Chaque liste correspond à une table de travail et contient des liens hypertextes qui pointent vers ses enfants dans la hiérarchie des tables de travail.

7.2.3 Navigation

Nous avons développé plusieurs techniques afin de permettre aux utilisateurs de naviguer dans de leur espace hiérarchique de bookmarks. Ces techniques reposent sur le principe d'association entre vues globales et vues locales décrit au chapitre 6.1.

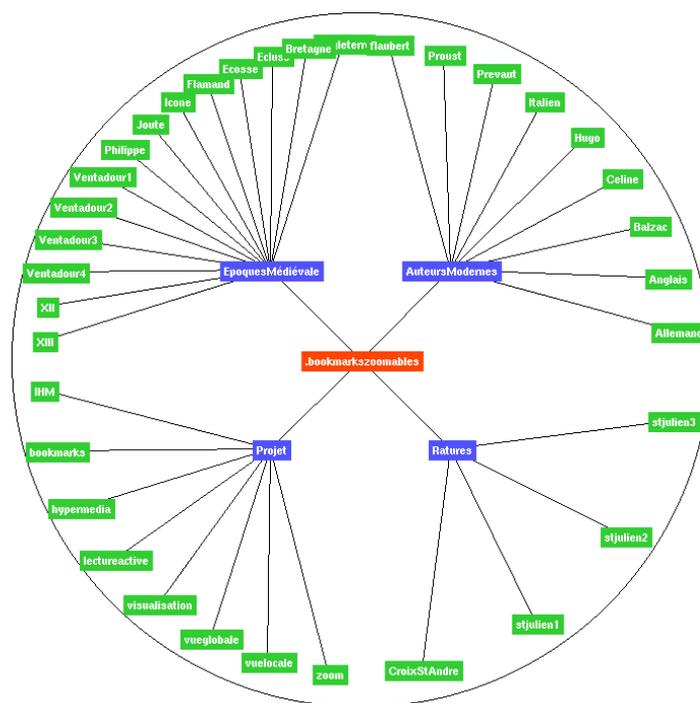


Figure 7.4 : Une vue globale concentrique de la hiérarchie des tables de travail.

Une vue globale hiérarchique, conçue sur le modèle classique trouvé dans la plupart des explorateurs de fichiers, a été ajouté à notre espace de bookmarks (Figure 7.2). Cette vue permet aux utilisateurs de visualiser l'organisation des tables de travail

dans son ensemble. L'utilisateur peut accéder à n'importe laquelle de ces tables par un simple clic souris sur l'élément correspondant dans la vue hiérarchique. Cet élément change alors de couleur, ce qui facilite la localisation de la table courante dans l'ensemble de la hiérarchie, renforçant du même coup l'association entre vue globale et vue locale. Comme alternative à cette vue hiérarchique classique, nous avons également connecté notre système de bookmarks à la vue concentrique présentée à la section 6.2.1 afin d'améliorer l'utilisation de l'espace écran, comme illustré à la Figure 7.4.

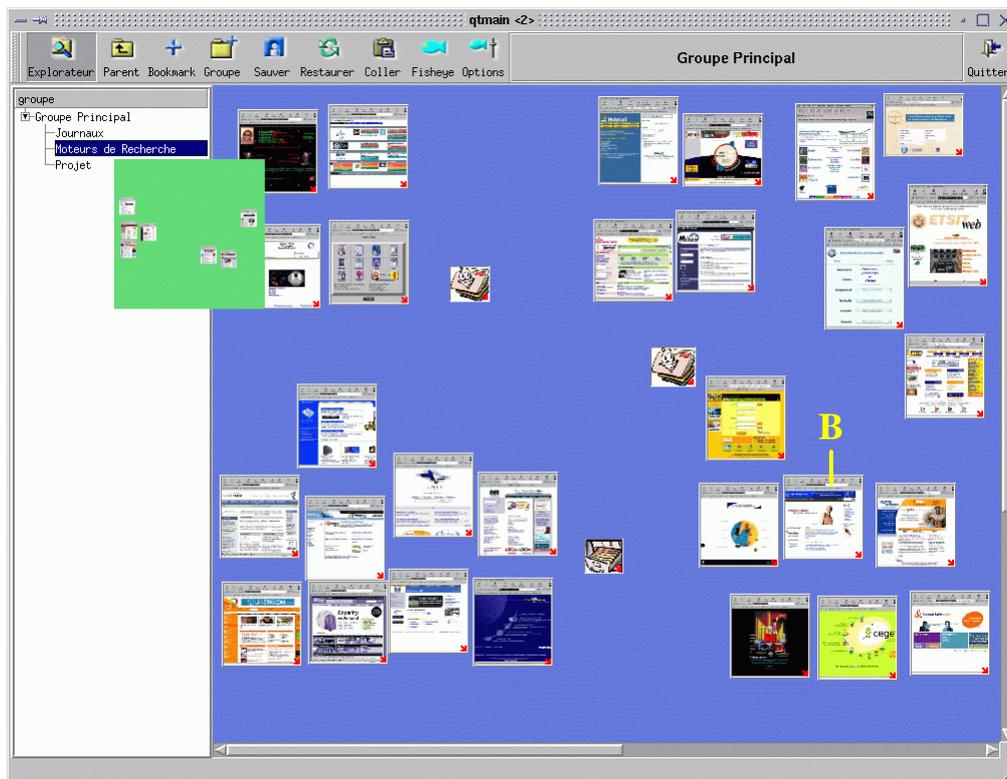


Figure 7.5 : Un aperçu sur une table de travail à partir de la vue globale.

En plus des documents, chaque table de travail contient des icônes qui font référence à ses enfants dans la hiérarchie (d'autres tables de travail). Ces icônes fournissent un moyen simple d'indexer des groupes de documents à partir d'une table. Les images qui leur sont associées peuvent aider les utilisateurs à se rappeler du contenu des tables correspondantes. Par exemple, l'icône A de la Figure 7.2a correspond à une table de travail où sont stockées des images de manuscrits contenant des ratures. Un simple clic souris sur une telle icône permet d'accéder à la table de travail correspondante. Afin de compléter ce mode local de navigation, un bouton situé dans la barre d'outils permet à tout moment d'accéder au « parent hiérarchique » de la table courante. Suite à un tel déplacement, l'icône qui référence la table d'où vient l'utilisateur clignote pendant environ une seconde, ce qui facilite la mise en

correspondance de cette table et de cette icône. L'utilisateur visualise de manière explicite le chemin emprunté, réduisant du même coup sa charge cognitive.

Des aperçus sur les tables de travail

Afin de faciliter l'identification des tables de travail et ainsi réduire les erreurs de navigation, l'utilisateur a la possibilité de visualiser un aperçu du contenu d'une table avant de la charger et de l'afficher. Un aperçu se matérialise à l'écran sous la forme d'une représentation réduite de la table. Il peut être affiché à partir de la vue globale hiérarchique par un simple clic souris sur une de ses entrées (Figure 7.5). L'utilisateur peut également obtenir un aperçu en cliquant sur une icône qui référence une table de travail. Dans ce cas, l'aperçu remplace l'icône désignée, comme illustré à la Figure 7.6. Cette transformation entraîne une restructuration de l'affichage afin que l'aperçu ne recouvre pas les icônes qui l'entourent. L'algorithme utilisé modifie le positionnement de l'ensemble des icônes en respectant leurs relations spatiales, ce qui offre l'avantage de préserver les repères organisationnels de l'utilisateur. Ce dernier dispose ainsi d'une vue focus+contexte qui semble l'aider à identifier la table représentée par l'aperçu. Comme le montre la Figure 7.7, plusieurs aperçus peuvent être affichés simultanément. Les vues multi-focus résultantes permettent de comparer plusieurs tables avant d'en sélectionner une.

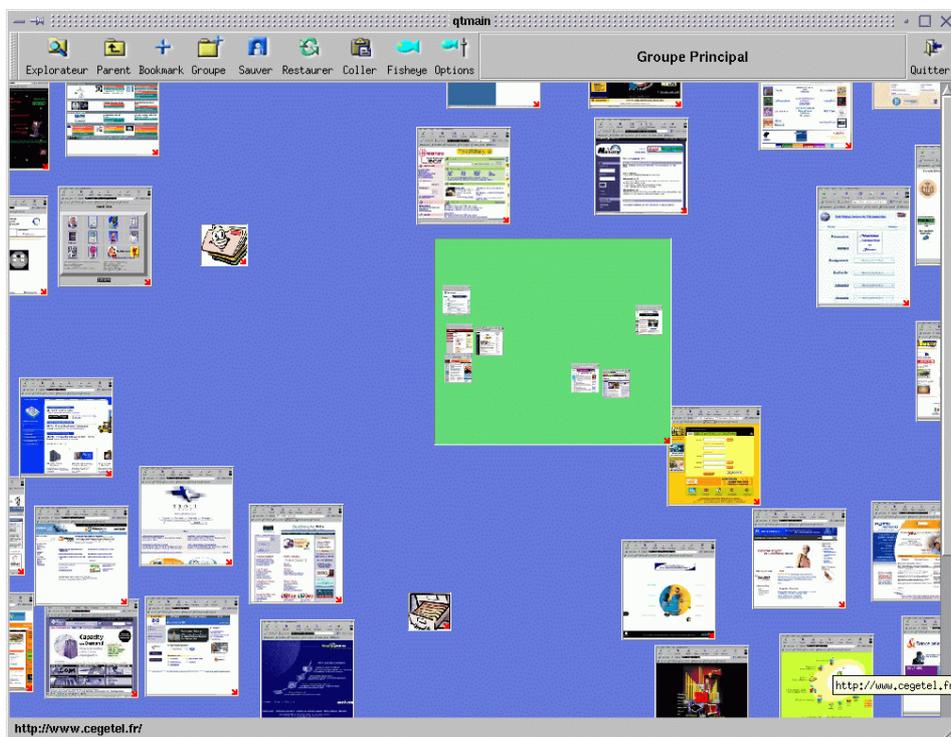


Figure 7.6 : Un aperçu affiché dans une table de travail.

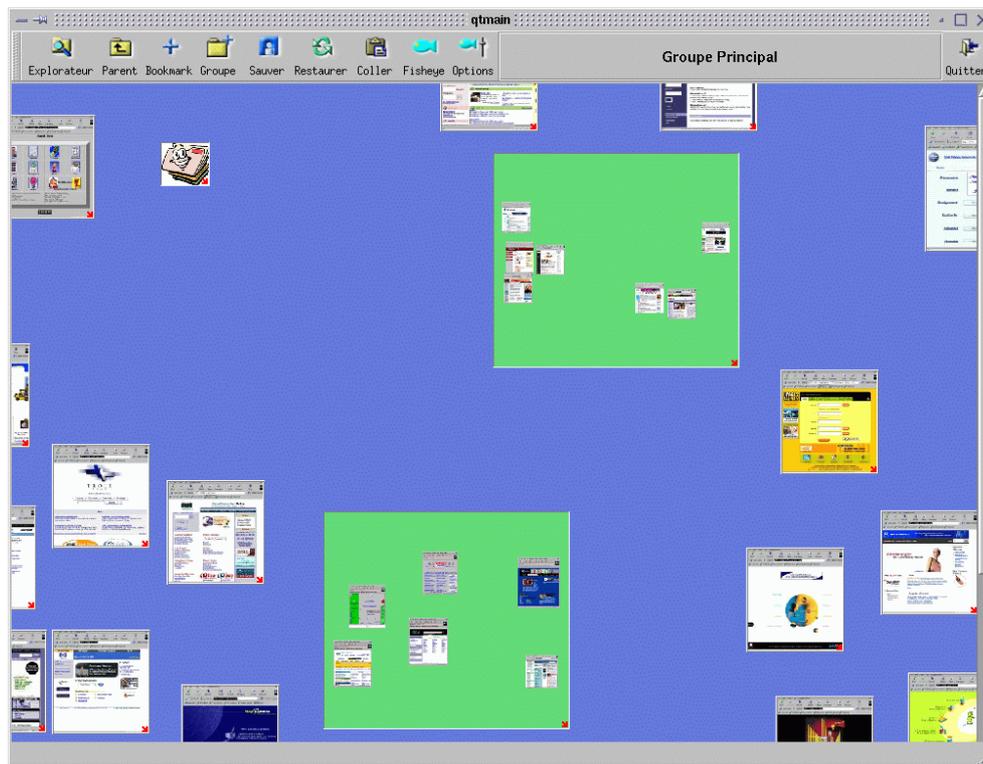


Figure 7.7 : Une vue multi-focus d'aperçus.

Autre Fonctionnalité

Enfin, le système dispose d'une autre fonctionnalité qui permet de focaliser sur une zone d'intérêt en une seule interaction au moyen d'une transformation Fisheye (voir section 5.2.1.3).

Une opération suffit à changer la taille et la position des documents affichés en fonction de l'intérêt de l'utilisateur. Ceci lui évite d'avoir à effectuer de trop nombreuses interactions. L'utilisateur clique à un endroit de la surface d'affichage. Le point désigné devient le centre de la vue Fisheye. Les éléments proches de ce point sont agrandis alors que ceux plus éloignés voient diminuer leur taille. La répartition de l'espace alloué à chaque élément est modifiée, chacun d'eux reste cependant affiché sur la surface. La Figure 7.8 montre le résultat d'une telle opération appliquée à la Figure 7.5 (la zone de focus est repéré par la lettre B).



Figure 7.8 : Résultat d'une transformation Fisheye sur la Figure 7.5.
Le point de focus est repéré par la lettre B.

7.3 Conclusion

Le système présenté dans ce chapitre permet aux utilisateurs d'*organiser* et de *retrouver* des documents jugés intéressants comme ils ont l'habitude de le faire dans le monde « réel ». Les utilisateurs organisent interactivement « à la main » ces documents en se basant sur des propriétés visuelles et spatiales (comme lors du rangement de documents papiers sur un bureau). Ils définissent ainsi leur propre organisation conceptuelle de l'espace informationnel. Ce travail d'organisation manuelle est ensuite mis à profit lors de nouvelles explorations, les utilisateurs s'appuyant sur leurs capacités mémorielles pour *retrouver* les documents avec plus d'efficacité. Par ce rapprochement du « réel » et du « virtuel », le système proposé contribue à une intégration « forte » du document à l'environnement informatique.

IV. Conclusion et perspectives

La mise à disposition de documents électroniques ne pose aujourd'hui plus réellement problème. Néanmoins, il s'avère que les utilisateurs ont souvent des difficultés à travailler avec ces documents de manière intuitive.

Nos travaux de recherche s'appuient sur la thèse que ce problème peut être résolu par une *intégration* « forte » du document à l'environnement informatique, c'est-à-dire par un outillage logiciel adapté permettant aux utilisateurs de mettre à profit les habitudes de travail qu'ils ont acquises dans le « monde réel ». L'idée consiste à rapprocher le réel du virtuel, au moyen de techniques d'interaction et de visualisation, afin de faciliter l'activité documentaire (la recherche, la lecture, l'analyse, la prise de notes, l'organisation, etc.).

Nous nous sommes plus particulièrement intéressé aux aspects liés à l'exploration d'espaces hypermédias, qui nécessite d'aider les utilisateurs à *trouver*, à *comprendre*, à *organiser* et à *retrouver* les documents mis à disposition. Deux approches complémentaires ont été proposées pour faciliter la réalisation ces tâches.

La première approche repose sur la métaphore du document papier. Elle montre comment simuler les principes de *lecture active* au moyen de techniques hypertextes d'annotation. Le système développé permet au lecteur d'enrichir interactivement les documents qu'il est en train de lire par un ensemble de signes graphiques (surlignages, cerclages, etc.) et d'informations textuelles (notes, commentaires, etc.). Le lecteur construit ainsi son propre système de repérage qui l'aide à *comprendre* et à *réutiliser* les documents. Nous avons montré comment associer ces méta-données « ajoutées » (que nous nommons, d'une manière générique, *annotations*) et les documents au moyen de liens hypertextes, et comment les coder dans un format textuel normalisé (dérivé de XML) afin de pouvoir les exploiter informatiquement. Ce couplage liens hypertextes / codage XML fournit une solution à l'accès, par indexation, à des documents difficilement exploitables par traitement automatique. Les utilisateurs ont la possibilité de *trouver* et de *retrouver* les documents au moyen d'une recherche par le contenu via les annotations XML.

La seconde approche montre comment tirer partie des *capacités perceptives et mémorielles humaines* pour faciliter l'exploration.

Nous proposons tout d'abord un système interactif de navigation pour aider à *trouver* les documents. La méthode implémentée repose sur un fort couplage entre vues globales et vues locales afin de réduire le phénomène de désorientation. Les vues globales montrent l'organisation de l'espace d'information et en permettent un survol rapide. Elles sont utilisées pour *trouver* des zones d'intérêts. Les vues locales

présentent le contenu de plusieurs documents en contexte afin d'aider l'utilisateur à choisir celui qu'il va consulter.

Nous proposons ensuite un système de gestion de bookmarks pour aider à *organiser* et à *retrouver* des documents jugés intéressants. L'idée consiste à permettre aux utilisateurs d'organiser ses bookmarks de la même manière qu'il le ferait avec un ensemble de documents papiers sur un bureau. L'utilisateur crée ainsi sa propre organisation logique. Il établit des schémas de classification et de relation entre les bookmarks en se basant sur des propriétés visuelles et spatiales. Pour *retrouver* les documents avec plus d'efficacité, il peut alors faire appel à sa mémoire visuelle et spatiale.

Comme nous l'avons soutenu dans cette thèse, l'activité documentaire en milieu informatique peut être améliorée par un outillage permettant de prendre en compte les habitudes de travail des utilisateurs. A ce jour, on s'aperçoit toutefois que peu d'implémentations ont été réalisées. Ces dernières nécessitent souvent d'importants développements logiciels qui font appel à des domaines de compétences variés (interaction homme-machine, visualisation d'information, hypermédia, structuration de documents, etc.). Face à ces difficultés, une perspective intéressante de notre travail consiste à voir comment intégrer les différents outils logiciels proposés à une boîte à outils graphiques.

Enfin, l'arrivée de nouveaux matériels de lecture comme les livres électroniques (e-books) [Schilit99, Harrison00] ouvre de nombreuses perspectives quant à l'utilisation des documents. Dans cette optique, il serait intéressant d'étudier dans quelle mesure les techniques proposées pourraient s'adapter à l'utilisation de ce type de matériels.

Index

A

ACM.....	28
activité	
documentaire	18
hyper-documentaire.....	19
affichage bifocal.....	122
ancre	27
animation.....	184
annotation.....	70
aperçu	196
arbre	
conique	150
hyperbolique.....	152
XML/Ubit.....	82
architecture	
fermée.....	50
ouverte.....	50
Aspen Movie Map.....	36
Augment.....	33

B

bookmarks	98, 158, 187, 190
-----------------	-------------------

C

cartes HTML	78
CERN	46
correspondance inverse	82

D

Data Mountain.....	162
dématérialisation	19
désorientation	87
Dexter.....	75
distorsion.....	121
Document Examiner.....	37
document optique	146
DTD.....	60
DTD TEI	62
Dynabook	35

E

espace informationnel de travail....	157
excentric labeling.....	141
exploration hyper-documentaire.....	19

F

fac-similé numérique	56
FDP	155
filtrage dynamique	136
FRESS.....	33

G

géométrie hyperbolique	152
Guide.....	41

H

HES.....	32
historique de navigation.....	95
HTML	47, 194
HTTP	47
hyperbase	26
HyperCard.....	45
hyperdocument	26
hypermédia	19, 25
hypertexte	25
HyperTies	36

I

indexation	91
Information Visualizer.....	157
interface zoomable d'exploration ..	133
Intermedia	39

K

KMS.....	34
----------	----

L

Lecture Active	54, 55
----------------------	--------

lentille magique	138		
lien.....	27		
livre électronique.....	54		
localisation spatiale	188		
<hr/>			
M			
MacWeb	100		
marquage	97		
marqueur interactif	66		
Memex.....	30		
mémoire spatiale et visuelle	188		
modèle hypertexte			
Amsterdam	49		
Dexter	48		
Mosaic	47		
mur perspectif.....	144		
<hr/>			
N			
navigation spatiale.....	103		
NLS	33		
noeud	26		
NoteCards.....	42		
<hr/>			
O			
organisation	188		
<hr/>			
P			
Pad++	133		
points de repères.....	92		
processus de visualisation	115, 118		
projections poly-focales	124		
projet			
LADIA	56		
Philectre.....	57		
<hr/>			
R			
recherche et d'impression	53		
représentation			
concentrique hiérarchique	170		
du contenu	189		
multi-focus	177		
perspective.....	166		
retour-arrière	95		
<hr/>			
		S	
		SGML	60
		space-scale diagrams	131
		structure	
		graphe	155
		hiérarchique	101, 147, 175, 191
		linéaire	142
<hr/>			
		T	
		tableau optique.....	128
		tables de travail 2D	190
		tâches exploration	19
		taxonomie des systèmes de	
		visualisation	
		Bruley	118
		Chi.....	115
		Chuah et Roth	113
		Leung	112
		Shneiderman	110
		Vernier	119
		transclusion	31
		transcription	57, 67
		transformation structurelle.....	101
		Treemap	149
		typage nœuds/liens.....	100
<hr/>			
		U	
		Ubit	80
		URL	47
<hr/>			
		V	
		visite guidée	93
		visualisation	109
		vues	
		fisheyes	126
		globales	165, 166, 194
		locales	165, 175, 194
<hr/>			
		W	
		Web Forager	160
		WebBook	158
		World Wide Web.....	46

X	
Xanadu	31
XLibris	54, 97
XML.....	47, 60
XSLT.....	77
<hr/>	
Z	
Zomit.....	133
zoom.....	130, 179
collectif	181
contextuel.....	182
géométrique	130
hiérarchique	181
sémantique	130
ZoomTree	175, 179
ZUI.....	133

Publications

- Robert L. & Lecolinet E. Digital Annotation and Exploration Techniques for Handling Image-based Hypermedia. IFIP Conference on Human-Computer Interaction (INTERACT'01). Tokyo, Japan, July 2001.
- Robert L. & Lecolinet E. Exploitation d'une base de fac-similés numériques : annotation, visualisation et accès. Numéro spécial « Interaction et Document » de la revue d'interaction Homme-Machine, 2001.
- Lecolinet L., Robert L. & Role F. Text-Image Coupling for Editing Literary Sources. Special issue of the Computers and the Humanities Journal on the topic of image-based humanities, 2001.
- Robert L. & Lecolinet E. Techniques d'interaction et de visualisation pour l'accès à des documents numérisés. Actes de la conférence Ergonomie et Interaction Homme-Machine (ERGO-IHM2000). Biarritz, France, Octobre 2000.
- Lecolinet E. & Robert L. Conception d'un poste d'édition et de lecture d'hypermédias littéraires. Numéro spécial « Numérisation et structuration des documents anciens » de la revue Document Numérique. Vol. 3, no. 1-2, pp. 103-115. Hermès, juin 1999.
- Gusnard de Ventabert (nom collectif). Représentation et exploitation électronique des documents anciens numérisés. Numéro spécial « Numérisation et structuration des documents anciens » de la revue Document Numérique. Vol. 3, no. 1-2, pp. 57-73. Hermès, juin 1999.
- Robert L. & Lecolinet E. Browsing Hyperdocuments with Multiple Focus+Context Views. ACM Conference on Hypertext and Hypermedia (HT'98), Pittsburgh, USA, June 1998.
- Lecolinet E., Likforman-Sulem L., Robert L., Role F. & Lebrave J-L. An Integrated Reading and Editing Environment for Scholarly Research on Literary Works and their Handwritten Sources. ACM Digital Libraries Conference (DL'98), Pittsburgh, PA, USA, June 1998.
- Robert L. & Lecolinet E. Couplage de représentations globales et locales pour l'exploration d'hyperdocuments. Actes des Dixièmes Journées Francophones sur l'Interaction Homme-Machine (IHM'98). Nantes, France, Septembre 1998.
- Robert L., Likforman-Sulem L. & Lecolinet E. Image and Text Coupling for Creating Electronic Books from Manuscripts. IEEE International Conference on Document Analysis and Recognition (ICDAR'97), Ulm, Germany, August 1997.

Likforman-Sulem L., Robert L., Lecolinet E. & Lebrave J-L. Edition hypertextuelle et consultation de manuscrits : le projet Philectre. *Revue Hypertextes et Hypermédias (H2PTM'97)*. Vol. 1, no. 2-3-4, pp. 299-310. Hermès, Paris, France, Septembre 1997.

Bibliographie

A

- [Abiteboul94] Abiteboul S., Hull R., Vianu V. *Foundations of Databases*. Addison-Wesley, 1994.
- [Abrams97] Abrams D. *Human Factors of Personal Web Information Spaces*. M.S. Thesis, University of Toronto, 1997.
- [Abrams98] Abrams D., Baecker R. & Chignell M. *Information Archiving with Bookmarks: Personal Web Space Construction and Organization*, CHI'98, pp. 41-48, 1998.
- [Adler72] Adler M.J & Van Doren C. *How to Read a Book*. Simon and Schuster, New York, NY, 1972.
- [Adobe97] Adobe System, *Adobe FrameMaker + SGML*, 1997.
- [Afnor91] AFNOR. *SGML-ODA: Présentation des concepts et comparaison fonctionnelle*. Danish Standards Association, AFNOR Technique, 1991.
- [Ahlberg94a] Ahlberg C. & Shneiderman B. *The Alphaslider: A Compact and Rapid Selector*. CHI'94, pp. 365-371, 1994.
- [Ahlberg94b] Ahlberg C. & Shneiderman B. *Visual Information Seeking Using the FilmFinder*. CHI'94, pp. 433, 1994
- [Akscyn88] Akscyn R.M., Donald L. & Yoder E. *KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations*. Communications of the ACM, 31(7), 1988.
- [Amann92] Amann B & Scholl M. *Gram: A Graph Data Model and Query Language*. ECHT'92, pp. 201-211, 1992.
- [André95] André J., Fekete J.D. & Richy H. *Mixed text/image processing of old documents*. Cahiers GuTenberg, (21), p. 75-85, 1995.
- [Andreessen93] Andreessen M. *NCSA Mosaic Technical Summary 2.1*. Available by anonymous ftp from <ftp.ncsa.uiuc.edu> as file /Web/Mosaic.ps.Z.
- [Ayers96] Ayers E. & Stasko J. *Using Graphic History in Browsing the World Wide Web*. 4th International World Wide Web Conference, pp. 11-14, 1996.
- [Azokly95] Azokly A & Ingold R. A. *Language for Generic Layout Description and its Use for Segmentation into Regions*. IEEE

Conference on Document Analysis and Recognition (ICDAR'95) Vol. 2, pp. 1123-1126, 1995.

B

- [Ballay94] Ballay J.M. *Designing Workscape: An Interdisciplinary Experience*. ACM Conference on Human Computer Interaction (CHI'94), pp. 10-15.
- [Balpe90] Balpe J.P. *Hyperdocuments, Hypertextes, Hypermédiat*. Eyrolles, 1990.
- [Battista94] Battista G.D., Eades P., Tamassia R. & Tollis I.G. *Annotated Bibliography on Graph Drawing*. Computational Geometry: Theory and Applications, vol. 4 (5), pp. 235-282, 1994.
- [Beer90] Beer C. & Kornatzky Y. *A Logical Query Language for Hypertext Systems*, ECHT'90, pp. 67-80, 1990.
- [Bederson94] Bederson B. & Hollan J.D. *Pad++ : a Zooming Graphical Interface for Exploring Alternate Interface Physics*. ACM Symposium on User Interface Software and Technology (UIST'94), pp. 17-26, 1994.
- [Bederson96] Bederson B., Hollan J.D., Perlin K., Meyer J., Bacon D. & Furnas G. *Pad++ : a Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics*. Journal of Visual Languages and Computing, 7:3-31, 1996.
- [Berge83] Berge. *Graphes*. Dunod, 1983.
- [Berners-Lee92] Berners-Lee T., Cailliau R., Groff J. & Pollermann B. *World Wide Web: The Information Universe*. Electronic Networking: Research, Applications and Policy, vol. 1 (2), pp. 52-58, 1992.
- [Berners-Lee94] Berners-Lee T., Cailliau R., Luotonen A., Nielsen H.F. & Secret A. *The World Wide Web*. Communications of the ACM, vol. 37 (8), pp. 76-82, 1994.
- [Bernstein88] Bernstein M. *The Bookmark and the Compass: Orientation Tools for Hypertext Users*. ACM SIGOIS Bulletin, 9(4):34-45, 1988
- [Bernstein91a] Bernstein M., Bolter J.D., Joyce M. & Mylonas E. *Architectures for volatile hypertext*. ACM Conference on Hypertext and Hypermedia (HT'91), pp. 243-261, 1991.
- [Bernstein91b] Bernstein M., Brown P.J., Frisse M., Glushko R., Landow G. & Zellweger P. *Structure, Navigation, and Hypertext: The Status of the Navigation Problem*. ACM Conference on Hypertext and Hypermedia (HT'91), pp. 363-366, 1991.

-
- [Bertin77] Bertin J. *La graphique et le traitement graphique de l'information*. Ed. Flammarion, 1997.
- [Bonhomme96] Bonhomme S., Quint V., Richy H., Roisin C. and Vatton I. *Manuel Thot*. Inria, Grenoble 1996.
<http://opera.inrialpes.fr/thot/doc/Thotman-F.html>.
- [Bieber89] Bieber M. & Kimbrough S.O. *On Generalizing the Concept of Hypertext*. Technical Report, Boston College Computer Science Department, 1989.
- [Bier93] Bier E., Stone M.C., Pier K., Buxton W. & DeRose T.D. *Toolglass and Magic Lenses: The See-Through Interface*. SIGGRAPH'93, pp. 73-80, 1993.
- [Bier94] Bier E., Stone M.C., Baudel M.C., Buxton W. & Fishkin K. *A Taxonomy of See-Through Tools*. CHI'94, pp. 358-364, 1993.
- [Björk00] Bjork S. *Hierarchical Flip Zooming: Enabling Parallel Explorations of Hierarchical Visualizations*. ACM Conference on Advanced Visual Interfaces (AVI2000), pp. 232-237, 2000.
- [Bousfield53] Bousfield W.A. *The Occurrence of Clustering in the Recall of Randomly Arranged Associates*. Journal of General Psychology, vol. 49, pp. 229-240, 1953.
- [Bower69] Bower G.H., Clark M.C., Lesgold A.M. & Winzenz D. *Hierarchical Retrieval Schemes in Recall of Categorical Word lists*. Journal of Verbal Learning and Verbal Behaviour, vol. 8, pp. 323-343, 1969.
- [Bozzi97] Bozzi A. & Calabretto S. *The Digital Library and Computational Philology*. European Conference on Digital Libraries, pp. 269-285, 1997.
- [Bozzi99] Bozzi A. *Pour un système de philologie numérique*. Numéro spécial « Numérisation et structuration des documents anciens » de la revue Document Numérique. Vol. 3 (1-2), pp. 57-73, pp. 93-101, Hermès, 1999.
- [Brown87] Brown P.J. *Turning Ideas into Products: The guide system*. ACM Conference on Hypertext and Hypermedia (HT'87), pp. 33-44.
- [Brown88] Brown P.J. *Linking and Searching within Hypertext*. *Electronic Publishing*, vol. 1 (1), pp. 183-191, 1988.
- [Bruley99] Bruley C. *Analyse des représentations graphiques de l'information, extension aux représentations tridimensionnelles*. Thèse de l'université Joseph Fourier, 1999.

-
- [Bush45] Bush V. *As We May Think*. Atlantic Monthly, vol. 176, pp. 101-108, 1945 – reproduit dans A. Goldberg, *A history of personal workstations*, ACM Press, New York, pp. 237-247, 1988.
-

C

- [Campell88] Campell B. & Goodman J.G. *HAM: A General Purpose Hypertext Abstract Machine*. Communications of the ACM, vol. 31(7), pp. 856-861, 1988.
- [Card84] Card S.K., Pavel M. & Farrell J.E. *Window-Based Computer Dialogues*. INTERACT'84, pp. 51-56, 1984.
- [Card91] Card S.K., Robertson G. & Mackinlay J.D. *The Information Visualizer, an Information Workspace*. ACM Conference on Computer Human Interaction (CHI'91), pp. 181-188, 1991.
- [Card96] Card S.K., Robertson G. & York W. *The WebBook and the WebForager : An Information Workspace for the World Wide Web*. ACM Conference on Computer Human Interaction (CHI'96), pp. 111-117.
- [Card99] S. K. Card S.K., Mackinlay J.D. & Shneiderman B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufman, 1999.
- [Carmody69] Carmody S., Gross T., Nelson T., Rice D. & Van Dam A. *Hypertext Editing System*. Conference in Computer Graphics, University of Illinois.
- [Carpendale95] Carpendale M.S.T., Cowperthwaite D.J. & Fracchia F.D. *Extending Distortion Viewing from 2D to 3D*. IEEE Computer Graphics and Applications, vol. 17 (4), pp. 42-51, 1995.
- [Carpendale97] Carpendale M.S.T., Cowperthwaite D.J., Storey M.A. & Fracchia F.D. *Exploring Distinct Aspects of the Distortion Viewing Paradigm*. Technical Report 97-08, School of Computing Science, Simon Fraser University, Canada, 1997.
- [Cerquiglini94] Cerquiglini B. & Lebrave J-L. *Philectre : un projet de recherche pluridisciplinaire en philologie électronique*. GIS sciences de la cognition, CNRS, 1994.
- [Chalmers95] Chalmers M. *Design Perspectives in Visualizing Complex information*. Third IFIP Visual Databases Conference, 1995.
- [Chi98] Chi E. & Riedl J. *An Operator Interaction Framework for Visualization Systems*. InfoVis'98, pp. 63-70, 1998.
- [Chi00] Chi E. & Riedl J. *A Taxonomy of Visualization Techniques using the Data State Reference Model*. IEEE, pp. 69-75, 2000.

-
- [Chuah96] Chuah M.C. & Roth S.F. *On the Semantics of Interactive Visualization*. IEEE Visualization (Vis'96), pp. 29-36, 1996.
- [CNRS95] CNRS. *Internet Professionnel*. Ouvrage collectif, éditions CNRS, 1995.
- [Conklin87] Conklin J. *Hypertext: an Introduction and Survey*. Computer, pp. 17-41, septembre 1987.
- [Conklin88] Conklin J. & Begeman M.L. *gIBIS: A Hypertext Tool for Exploratory Policy Discussion..* ACM Transactions Office Inf. Syst. Vol. 6 (4), pp. 303-331, 1988.
- [Crouch89] Crouch D.B., Crouch C.J. & Andreas G. *The Use of Cluster Hierarchies in Hypertext Information Retrieval*. ACM Conference on Hypertext and Hypermedia (HT'89), pp. 225-237, 1989.
- [Curiel98] Curiel J.M. & Radvansky G.A. *Mental organization of Maps*. Journal of Experimental Psychology: Learning, Memory and Cognition, vol. 24, pp. 202-214, 1998.
- [Czerwinski99] Czerwinski M.P., van Dantzich M., Robertson G. & Hoffman H. *The Contribution of Thumbnail Image, Mouse-Over Text and Spatial Location Memory to Web Page Retrieval in 3D*. IFIP Conference on Human-Computer Interaction (INTERACT'99), pp. 163-170, 1999.

D

- [Darken93] Darken R. & Sibert J.L. *A Toolset for Navigation in Virtual Environments*. ACM INTERCHI'93, pp. 157-165, 1993.
- [Darken96] Darken R. & Sibert J.L. *Navigating large virtual spaces*. International Journal of Human-Computer Interaction, vol. 8, pp. 49-72, 1996.
- [Davis92] Davis H. & al. *Towards an Integrated Information Environment with Open Hypermedia Systems*. ACM ECHT'92, pp. 181-191, 1992.
- [Davis94] Davis H. et al. *Light Hypermedia Link Service: A Study of Third Party Application Integration*. ACM ECHT'94, pp. 41-51, 1992.
- [DeRose89] DeRose S. J. *Expanding the Notion of Links*. ACM Conference on Hypertext and Hypermedia (HT'89), 1989.
- [DeRose95] DeRose S.J. & Durand D.G. *The TEI Hypertext Guidelines*. In Text Encoding Initiative: Background and Context, Kluwer Academic Publishers (ISBN 0-7923-3689-5), 1995.

-
- [DeRose98a] DeRose S.J. & Maler E. *XML Linking Language (XLink)*. World Wide Web Consortium Working Draft, March 1998.
- [DeRose98b] DeRose S.J. Maler E. (eds). *XML Pointer Language (XPointer)*. World Wide Web Consortium Working Draft, March 1998.
- [D'Ioro98] D'Ioro P., www.nietzsche.edu, communications orale, Genèses, 2^{ème} congrès international de critique génétique, Paris, Ecole Normale Supérieure, 1998.
-

E

- [Eades84] Eades P. *A Heuristic for Graph Drawing*. Congressus Numerantium, vol. 42, pp. 149-160, 1984.
- [Eades90] Eades P. & Sugiyama K. *How to Draw Directed Graph*. Information Processing, vol. 13 (4), pp. 424-434, 1990.
- [Eick94] Eick S.G. *Graphical Displaying Text*. Journal of Computational and Graphical Statistics, vol. 3, pp. 127-142, 1994.
- [Engelbart63] Engelbart D.C. *A Conceptual Framework for the Augmentation of man's Intellect*. In Vistas in Information Handling, vol. 1, pp. 1-29, Spartan Books, London.
- [Engelbart68] Engelbart D.C. & English W.K. *A Research Center for Augmenting Human Intellect*. AFIPS/ Fall Joint Computer Conference, vol. 33, Part I, pp. 395-410.
-

F

- [Fairchild88] Fairchild K.M, Poltrock S.E. & Furnas G.W. *Semnet: Three-dimensional Graphic Representations of Large Knowledge Bases*. In R. Guindon, editor, cognitive science and its applications for human computer interaction. Lawrence Erlbaum Associates, 1988.
- [Faure99] Faure C. *Preattentive Reading and Selective Attention for Document Image Analysis*. IEEE International Conference on Document Analysis and Recognition (ICDAR'99), pp. 577-580.
- [Fekete00] Fekete J-D. & Dufournaud N. *Compus: Visualization and Analysis of Structured Documents for Understanding Social Life in the 16th Century*. ACM Conference on Digital Libraries (DL'00), 2000.
- [Fekete99] Fekete J-D. & Plaisant C. *Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization*. ACM Conference on Computer Human Interaction (CHI'99), 1999.

-
- [Foley90] Foley J.D. & al. *Computer Graphics : Principles and Practice*. Addison-Wesley, 1990.
- [Foss89a] Foss C.L. *Tools for Reading and Browsing Hypertext*. Information Processing and Management, vol. 25, n°4, 1989, pp. 407-418.
- [Foss89b] Foss C.L. *Detecting lost Users: empirical Studies on Browsing Hypertext*. Technical Report 972, INRIA, France, Février 1989.
- [Franklin92] Franklin N, Tversky B. & Coon V. *Switching Points of View in Spatial Mental Models*. Memory and Cognition, vol. 20, pp. 507-518, 1992.
- [Frei92] Frei H.P. & Stieger D. *Making Use of Hypertext Links when Retrieving Information*. ACM ECHT'92, 1992.
- [Frisse88] Frisse M. *Searching for Information in a Hypertext medical Handbook*. Communications of the ACM, vol. 31(7), pp. 880-886, 1988.
- [Fruchterman91] Fruchterman T.M.J. & Reingold E.M. *Graph Drawing by Force-Directed Placement*. Software Practice and Experience, 1991.
- [Furnas86] Furnas G. *Generalized Fisheye Views*. ACM Conference on Computer Human Interaction (CHI'86), pp. 16-23, 1986.
- [Furnas95] Furnas G. & Bederson B.B. *Space-Scale Diagrams: Understanding Multiscale Interfaces*. CHI'95, pp. 234-241, 1995.
- [Furuta89] Furuta R. & Stotts P.D. *Object Structures in Paper Documents and Hypertexts*. Woodman'89, pp. 147-151.
- [Furuta90] Furuta R. & Stotts P.D. *A Functional meta-structure for Hypertext Model and Systems*. Electronic Publishing, vol. 3 (4), pp. 179-205, 1990.

G

- [Garg88] Garg P.K., *Abstraction Mechanisms in Hypertext*. Communications of the ACM, vol. 31 (7), pp. 862-870, 1988.
- [Garzotto91] Garzotto F. et al. *HDM : a Model for the Design of Hypertext Applications*. ACM Conference on Hypertext and Hypermedia (HT'91), pp. 313-328, 1991.
- [Ginsburg96] Ginsburg A., Marks J. & Shieber S. *A viewer for postscript documents*. ACM Symposium on User Interface Software and Technology (UIST'96), 1996.

-
- [Goodman87] Goodman D. *The Complete HyperCard Handbook*. Nw York, Bantam Books, Inc., septembre 1987.
- [Guinan92] Guinan C. & Smeaton A.F. *Information Retrieval from Hypertext Using Dynamically Planned Tours*. ACM ECHT'92, 1992.
- [Gusnard99] Gusnard de Ventabert (nom collectif). *Représentation et exploitation électronique des documents anciens numérisés*. Numéro spécial « Numérisation et structuration des documents anciens » de la revue Document Numérique. Vol. 3 (1-2), pp. 57-73. Hermès, juin 1999.
-

H

- [Halasz87] Halasz F.G., Moran T.P. & Trigg R.H. *Notecards in a Nutshell*. ACM Conference on Computer Human Interaction (CHI'87), pp. 45-52, 1987.
- [Halasz88] Halasz F.G. *Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems*. Communications of the ACM, vol. 31 (7), pp. 836-852, 1988.
- [Halasz90] Halasz F. & Schwartz M. *The Dexter Hypertext Reference Model*. in NIST Hypertext Standardization Workshop, 1990.
- [Halasz94] Halasz F. & Schwartz M. *The Dexter Hypertext Reference Model*. Communications of the ACM, vol. 37 (2), pp. 30-39, 1994.
- [Hardman94] Hardman L., Bulterman D.C & Van Rossum G. *The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model*. Communications of the ACM, vol. 37 (2), pp. 50-62, 1994.
- [Harrison96] Harrison B.L. & Vincente K.J. *An Experimental Evaluation of Transparent User Interface Tools and Information Content*. ACM Symposium on User Interface Software and Technology (UIST'96), pp. 391-398, 1996.
- [Harrison00] Harrison B.L. *E-Books and the Future of Reading*. IEEE Computer Graphics and Applications, 2000.
- [Henderson86] Henderson D.A. & Card S.K. *Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface*. ACM Transactions on Graphics, vol. 5 (3), pp. 211-243, 1986.
- [Hendley95] Hendley R.J, Wood A.M, Beale R. & Drew N.S. *Narcissus: Visualizing Information*. University Birmingham, 1995.

- [Hightower98] Hightower R.R., Ring L.T., Helfman J.I., Bederson B.B. & Hollan J.D. *Graphical Multiscale Web Histories : a Study of PadPrints*. ACM Conference on Hypertext and Hypermedia (HT'98), Pittsburgh, USA, 1998, pp. 58-65.
- [Hill94] Hill G. & Hall W. *Extending the Microcosm Model to a Distributed Environment*. ACM ECHT'94, pp. 32-40, 1994.
- [Holmquist98] Holmquist L.E. & Björk S. *A Hierarchical Focus+Context Method for Image Browsing*. ACM SIGGRAPH'98, Sketches and Applications, 1998.
- [Hu99] Hu S.C., Furuta R. & Urbina E. *An electronic edition of Don Quixote for humanities scholars*. Numéro spécial « Numérisation et structuration des documents anciens » de la revue Document Numérique. Vol. 3 (1-2), pp. 75-91. Hermès, 1999.
- [Hypercard89] HyperCard. *HyperCard Reference Manual*. Apple Computer Inc., Cupertino, CA, USA, 1989.
-

I

- [ISO86] Text and Office Systems-Standard Generalized Markup Language, ISO 8879, 1986.
-

J

- [Jones86] Jones W & Dumais S. *The Spatial Metaphor for User Interfaces : Experimental Tests of Reference by Location versus Names*. ACM Transactions on Office Inf. Syst. Vol. 4 (1), pp. 42-63, 1986
- [Johnson91] Johnson B. & Shneiderman B. *Space-filling Approach to the Visualization of Hierarchical Information Structures*. IEEE Visualization'91, pp. 284-291.
- [Johnson93] Johnson A. & Fotouhi F. *Automatic Touring in a Hypertext System*. IEEE Conference on Computers and Communications, pp. 524-530, 1993.
- [Johnson97] Johnson A. *Fax Trends. Office Equipment and Products*. pp. 25, July 1997.
-

K

- [Kacmar91] Kacmar C.J. & Leggett J.J. *PROXHY: A Process Oriented Extensible Hypertext Architecture*. ACM Transaction on Information Systems, vol. 9 (4), pp. 399-419, 1991.

-
- [Kadmon78] Kadmon N. & Shlomi E. *A Polyfocal Projection for Statistical Surfaces*. Cartograph, vol. 15 (1), pp. 36-41, 1978.
- [Kay77] Kay A. & Goldberg A. *Personal Dynamic Media*. Computer, vol. 10 (3), pp. 31-41.
- [Keahey96] Keahey T.A. & Robertson E.L. *Techniques for Non-Linear Magnification Transformations*. IEEE InfoVis'96, pp. 38-45, 1996.
- [Kumar97] Kumar H.P., Plaisant C. & Shneiderman B. *Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning*. International Journal of Human-Computer Studies, vol. 46 (1), pp. 103-124.
- [Kurtenbach97] Kurtenbach G., Fitzmaurice G., Baudel T. & Buxton B. The design of a GUI Paradigm Based on Tablets, Two-Hands, and Transparency. ACM Conference on Computer Human Interaction (CHI'97), pp. 35-42, 1997.
-

L

- [Lamping96] Lamping J. & Rao R. *The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies*. Journal of Visual Languages and Computing, 7:33-55, 1996.
- [Lecolinet98] Lecolinet E., Likforman-Sulem L., Robert L., Role F. & Lebrave J-L. *An Integrated Reading and Editing Environment for Scholarly Research on Literary Works and their Handwritten Sources*. ACM Conference on Digital Libraries (DL'98), pp 144-151, 1998.
- [Lecolinet99a] Lecolinet E. & Robert L. *Conception d'un poste d'édition et de lecture d'hypermédias littéraires*. Numéro spécial « Numérisation et structuration des documents anciens » de la revue Document Numérique. Vol. 3 (1-2), pp. 103-115. Hermès, juin 1999.
- [Lecolinet99b] Lecolinet E. *A Brick Construction Game Model for Creating Graphical User Interfaces: The Ubit Toolkit*. IFIP Conference on Human-Computer Interaction (INTERACT'99), pp. 510-518, 1999.
- [Lecolinet99c] Lecolinet E. *Réification et réplication dans les interfaces graphiques : le toolkit Ubit*. Onzièmes Journées Francophones sur l'Interaction Homme Machine (IHM'99). Page Web Ubit: <http://www.enst.fr/~elc/ubit>.
- [Lecolinet01] Lecolinet L., Robert L. & Role F. *Text-Image Coupling for Editing Literary Sources*. Special issue of the Computers and the

-
- Humanities Journal on the topic of image-based humanities, 2001.
- [Leggett90] Leggett J., Schnase J., Kacmar C.J. *Hypertext for learning*. Coord. D.H. Jonassen, H. Mandle, *Designing hypermedia for learning*, Springer, Berlin 1990, pp. 27-37.
- [Leung89] Leung Y.K. & Apperley M.D *Human Computer Interaction Techniques for Map-Based Diagrams*. Salvedy and Smith eds. *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, pp. 361-368, 1989.
- [Leung94] Leung Y.K. & Apperley M.D *A Review and Taxonomy of Distortion-Oriented Presentation Techniques*. ACM Transaction on Computer-Human Interaction, vol. 1 (2), pp. 126-160, 1994.
- [Likforman94] Likforman-Sulem L. & Faure C. *Extracting lines on handwritten documents by perceptual grouping*. C. Faure, P. Keuss, G. Lorette, A. Winter (Eds), « Advances in Handwriting and drawing: a multidisciplinary approach », p. 21-38, Europa Press, 1994.
- [Likforman97] Likforman-Sulem L., Robert L., Lecolinet E. & Lebrave J-L. *Edition hypertextuelle et consultation de manuscrits : le projet Philectre*. Revue Hypertextes et Hypermédias (H2PTM'97). Vol. 1 (2-3-4), pp. 299-310, Hermès, 1997.
- [Lucarella93] Lucarella D & al. *MORE : Multimedia Object Retrieval Environment*. ACM Conference on Hypertext and Hypermedia (HT'93), pp. 39-50, 1993.

M

- [Mackinlay90] Mackinlay J.D., Card S.K. & Robertson G.G. *Rapid Control Movement through a Virtual 3D Workspace*. SIGGRAPH'90, pp. 171-176, 1990.
- [Mackinlay91] Mackinlay J.D., Robertson G.G. & Card S.K. *Perspective Wall: Detail and Context Smoothly Integrated*. ACM Conference on Computer Human Interaction (CHI'91), pp. 173-179.
- [Mander92] Mander R., Salomon G & Wong Y.Y. *A Pile Metaphor for Supporting Casual Organization of Information*. CHI'92, pp. 627-634, 1992.
- [Mandler67] Mandler G. *Organization and Memory*. Spence eds. *The Psychology of Learning and Motivation*, vol. 1, Academic Press, pp. 327-372, 1967.

-
- [Marcus92] Marcus A. *Graphic Design for Electronic documents and user Interfaces*. Addison-Wesley, ACM Prss, 1992.
- [Marshall89] Marshall C. & Irish P. *Guided Tours and On-Line Presentations: How Authors Make Existing Hypertext intelligible for readers*. ACM Conference on Hypertext and Hypermedia (HT'89), pp. 15-26, 1989.
- [Marshall98] Marshall C.C. *Toward an Ecology oh Hypertext Annotation*. ACM Conference on Hypertext and Hypermedia (HT'98), pp. 40-57, 1998.
- [McCall90] McCall R.J. et al. *PHIDIAS: Integrating Cad Graphics into Dynamics Hypertext*. ECHT'90, pp. 152-165, 1990.
- [Montello93] Montello D.R & Pick H.L. *Integrating Knowledge of Vertically-Aligned Large-Scale Spaces*. Environment an Behavior, vol. 25, pp. 457-484.
- [Munzner97] Munzner T. H3: Laying out Large Directed Graphs in 3D Hyperbolic Space. IEEE InfoVis'97, pp. 85-91, 1997.

N

- [Nanard89] Nanard M & Nanard J. *MacWeb, un outil pour élaborer des documents*. ed. Bigre, Rennes, pp. 18-29, 1989.
- [Nanard91] Nanard M. *Les hypertextes et les hypermédias*. Ecole d'été EDF/CEA/INRIA sur les documents électroniques, juillet 1991.
- [Nation97] Nation D.A., Plaisant C., Marchionini G. & Komlodi A. *Visualizing websites using a hierarchical table of contents browser: WebTOC*. Conference on Human Factors and the Web, 1997.
- [Nelson65] Nelson T.H. *A File Structure for the Complex, the Changing and the Indeterminate*. Proceedings of the ACM National Conference, 1965.
- [Nelson72] Nelson T.H. *As We Will Think*. Proceedings of the Online 72 Conference, England, 1972.
- [Nelson87] Nelson T.H. *Literary Machines*. Mindful Press, Sausalito CA, 87.1 edition.
- [Nielsen90a] Nielsen J. *Hypertext and Hypermedia*. Academic Press, 1990.
- [Nielsen90b] Nielsen J. *The Art of Navigating through Hypertext*. Communications of the ACM, 33(3), 1990.

- [Nyce91] Nyce J.M. & Kahn P., editors. *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*. Academic Press, Boston MA.
-

O

- [O'Hara97] O'Hara K. & Sellen A. *A Comparison of Reading Paper and On-Line Documents*. ACM Conference on Computer Human Interaction (CHI'97), pp. 335-342, 1997.
- [Olive99] *OLIVE: On-Line Library of Information Visualization Environments*. <http://otal.umd.edu/Olive/1999>.
-

P

- [Parunak89] Van Dyke Parunak H. *Hypermedia Topologies and User Navigation*. ACM Conference on Hypertext and Hypermedia (HT'89), USA, 1989.
- [Patten90] Patten B.M. *The History of Memory Arts*. *Neurology*, 40, 346-252.
- [Pearl89] Pearl A. *Sun's Link Service: A Protocol for Open Linking*. ACM Conference on Hypertext and Hypermedia (HT'89), pp. 137-146.
- [Pook00a] Pook S., Lecolinet E., Vaysseix G. & Barillot E. *Context and Interaction in Zoomable User Interfaces*. ACM Conference on Advanced Visual Interfaces (AVI'2000), pp227-231, 2000.
- [Pook00b] Pook S., Lecolinet E., Vaysseix G. & Barillot E. *Des aides transparentes de navigation et un nouveau type de menu pour les interfaces zoomables*. Conférence Ergonomie et Interaction Homme-Machine (ERGO-IHM2000), pp. 170-177, 2000.
- [Price8] Price M.N., Golovchinsky G. & Schilit B.N. *Linking by Inking : Trailbalzing in a Paper-like Hypertext*. ACM Conference on Hypertext and Hypermedia (HT'98), pp. 293-294, 1998.
-

Q

- [Quinn79] Quinn N. & Breur M. *A Force Directed Component Placement Procedure for Printed Circuit Boards*. *IEEE Transactions on Circuits and Systems*, vol. 26 (6), pp. 377-388, 1979.

R

- [Rao92] Rao R. & Card S.K. *The Information Grid: A Framework for Information Retrieval-Centered Applications*. ACM UIST'92, pp. 22-32, 1992.
- [Rao93] Rao R., Russel D.M. & Card S.K. *System Component for Embedded Information Retrieval from Multiple Disparate Information Sources*. ACM UIST'93, 1993.
- [Rao94] Rao R. & Card S.K. *The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information*. ACM Conference on Computer Human Interaction (CHI'94), pp. 318-322, 1994.
- [Rao95] Rao R. & al. *Rich Interaction in the Digital Library*. Communications of the ACM, vol. 38 (4), pp. 29-39, 1995.
- [Rizk92] Rizk A. & Sauter L. *MultiCard : An Open Hypermedia System*. ACM ECHT'92, 1992.
- [Rivlin94] Rivlin E., Botafogo R. & Shneiderman B. *Navigating in Hyperspace : Designing a structured-based Toolbox*. Communications of the ACM, 37(2):87-96, 1994.
- [Robertson91] Robertson G.G., Mackinlay J.D. & Card S.K. *Cone Trees: Animated 3D Visualizations of Hierarchical Information*. ACM Conference on Computer Human Interaction (CHI'91), pp. 189-194, 1991.
- [Robertson93] Robertson G.G. & Mackinlay J.D. & Card S.K. *The Document Lens*. ACM Symposium on User Interface Software and Technology (UIST'93), pp. 101-108, 1993.
- [Robertson98] Robertson G., Czerwinski M., Larson K., Robbins D.C., Thiel D. & Van Dantzich M. *Data Mountain : Using Spatial Memory for Document Management*. ACM Symposium on User Interface Software and Technology (UIST'98), pp. 153-162, 1998.
- [Role91] Role F. *La norme SGML pour décrire la structure logique des documents*. Documentaliste-sciences de l'information, vol. 28 (4-5), pp. 187-192, 1991.
- [Role96] Role F. *Le codage informatique des appareils critiques : évaluations des recommandations de la Text Encoding Initiative*. Cahiers GUTemberg, vol. 24, pp. 154-165, 1996.
- [Rouet96] Rouet J. & Lovonen J. *Studying and Learning with Hypertext: Empirical Studies and their Implications*. Hypertext and Cognition, Erlbaum, 1996.

-
- [Ruddle97] Ruddle R.A., Payne S.J. & Jones D.M. *Navigating Buildings in Desktop virtual Environments: Experimental Investigations using Extended Navigational Experience*. *Experimental Psychology*, vol. 3, pp. 143-157, 197.
-

S

- [Salton68] Salton G. *Automatic Information Organization and Retrieval*. McGraw-Hill, New York, 1968.
- [Sandoval94] Sandoval V. *SGML: un outil pour la gestion électronique de documents*. Editions hermès, 1994.
- [Sandvad89] Sandvad E. *Hypertext in an Object-oriented Programming Environment*. Woodman'89, pp. 30-41, 1989.
- [Sarkar92] Sarkar M. & Brown M.H. *Graphical Fisheye Views of Graphs*. ACM Conference on Computer Human Interaction (CHI'92), pp. 83-91, 1992.
- [Schilit98a] Schilit B.N., Price M.N. & Golovchinsky G. *Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations*. ACM Conference on Computer Human Interaction (CHI'98), pp. 249-256, 1998.
- [Schilit98b] Schilit B.N., Price M.N. & Golovchinsky G. *Digital Library Information Appliances*. ACM Conference on Digital Libraries (DL'98), pp. 217-226, 1998.
- [Schilit99] Schilit B.N., Price M.N., Golovchinsky G., Tanaka K. & Marshall C.C. *As We May Read: The Reading Appliance Revolution*. *IEEE Computer*, vol. 32 (1), pP. 65-73, January 1999.
- [Shipman99] Shipman F.M., Marshall C. & LeMere J. *Beyond Location: Hypertext Workspaces and Non-linear Views*. ACM Conference on Hypertext and Hypermedia (HT'99), pp. 121-130.
- [Spence82] Spence R & Apperley M.D. *Database Navigation: an Office Environment for the Professional*. *Behav. Inf. Tech.*, vol. 1 (1), pp. 43-54, 1982.
- [Sperberg94] Sperberg-McQueen C.M. & Burnard L. *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. ACH-ACL-ALLC Text Encoding Initiative, 1994.
- [Shneiderman87] Shneiderman B. *User Interface Design for the Hyperties electronic Encyclopaedia*. ACM Conference on Hypertext and Hypermedia (HT'87), pp. 189-194, 1987

-
- [Shneiderman92] Shneiderman B. *Tree Visualization with Tree-Maps: 2D space-filling Approach*. ACM Transactions on Graphics, vol. 11(1), pp. 92-99, 1992.
- [Shneiderman94] Shneiderman B. *Dynamic Queries for Visual Information Seeking*. IEEE Software, vol. 11 (6), pp. 70-77, 1994.
- [Shneiderman96] Shneiderman B. *The Eyes Have It: A Task by Data Taxonomy for Information Visualizations*. Visual Languages Conference, pp. 336-343, 1992.
- [Stallman84] Stallman R.M. *Emacs: The Extensible, Customizable, Self-Documenting Display Editor*. Interactive Programming Environments, pp. 300-325, McGraw-Hill, New York, USA, 1984.
- [Stone94] Stone M.C., Fishkin K. & Bier E. *The Movable Filter as a User Interface Tool*. CHI'94, pp. 306-312, 1994.
- [Streitz92] Streitz N. *SEPIA: A Cooperative Hypermedia Authoring Environment*. ECHT'92, pp. 11-22, 1992.
-

T

- [Tales01] Blake N., Robinson P., Solopova E.. *The Canterbury Tales Project*. <http://www.shef.ac.uk/uni/projects/ctp/index.html,2001>.
- [Thorndyke82] Thorndyke P.W. & Hayes-Roth B. *Differences in Spatial Knowledge Acquired from Maps and Navigation*. Cognitive Psychology, vol. 14, pp. 560-589, 1982.
- [Tomek91] Tomek I., Kahn S. Muldner T., Nassar M., Navak G. & Proszynski P. *Hypermedia – Introduction and Survey*. Journal of Microcomputer Applications, vol. 14 (2), pp. 63-100.
- [Trigg88] Trigg R. *Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment*. ACM TOOIS, vol. 6, n°4, pp. 398-414, 1988.
- [Tulving62] Tulving E. *Subjective Organization in Free Recall of Unrelated Works*. Psychological Review, vol. 69. pp. 344-354, 1962.
- [Tweedie97] Tweedie L. *Characterizing Interactive Externalizations*. ACM Conference on Computer Human Interaction (CHI'97), pp. 375-382, 1997.
-

U

- [Underwood60] Underwood B.J. & Schultz R.W. *Meaningfulness and Verbal Learning*. Lippincott, 1960.

-
- [Utting89] Utting K. & Yankelovich N. *Context and Orientation in Hypermedia Network*. ACM Transaction on Information System. Volume 7, n°1, 1989, pp. 58-84.
-

V

- [Vernier01] Vernier F. *La multimodalité en sortie et son application à la visualisation de grandes quantités d'information*. Thèse de l'université Joseph Fourier, 2001.
- [Viega96] Viega J., Conway M.J. Williams G., & Pausch R. *3D Magic Lenses*. UIST'96, pp. 51-58, 1996.
- [Virbel93] Virbel J. *Reading and managing texts on the Bibliotheque de France station*. Text based computing in the Humanities, pp. 31-52, MIT Press, 1993.
-

W

- [Walker87] Walker J.H. *Document Examiner: Delivery Interface for Hypertext Documents*. ACM Conference on Hypertext and Hypermedia (HT'87), pp. 307-323, 1987.
- [Waller98] Waller D., Hunt E. & Knapp D. *The Transfer of Spatial Knowledge in Virtual Environment Training*. Presence, vol. 7, pp. 129-144, 1998.
- [Wiil92] Wiil U. & Leggett J.J. *Hyperform: Using Extensibility to develop Dynamic Open and Distributed Hypertext Systems*. ECHT'92, pp. 251-161, 1992.
- [Williamson92] Williamson C. & Shneiderman B. *The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System*. SIGIR'92, pp. 339-346, 1992.
- [Wood95] Wood A.M., Drew N.S., Beale R. & Hendley R.J. *Hyperspace: Web Browsing with Visualization*. Third International World Wide Web Conference, pp. 21-25, 1995.
- [W3C-DOM01] World Wide Web Consortium. *Document Object Model (DOM) Level 1 Specification*, W3C Recommendation.
<http://www.w3.org/TR/REC-DOM-Level-1>.
- [W3C-XML01] World Wide Web Consortium. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation.
<http://www.w3.org/XML>.
- [W3C-XSLT01] World Wide Web Consortium. *XSL Transformations (XSLT) Version 1.0*. W3C Recommendation.
<http://www.w3.org/TR/xslt>

Y

- [Yankelovich88] Yankelovich N. & al. *Intermedia: The Concept and the Construction of a Seamless Information Environment*. IEEE Computer, vol. 21 (1), pp. 81-96, 1988.
- [Young96] Young P. *Three Dimensional Information Visualization (Survey)*. Computer Science Technical Report, Centre of Software Maintenance, Dept. of Computer Science, University Durham, 1996.
-

Z

- [Zellweger89] Zellweger P. *Scripted Documents: A Hypermedia Path Mechanism*. ACM Conference on Hypertext and Hypermedia (HT'89), pp. 1-14, 1989.