



HAL
open science

Génération dynamique de présentations interactives en multimédia 3D, de données, pour les applications en ligne

Nicolas Bonnel

► **To cite this version:**

Nicolas Bonnel. Génération dynamique de présentations interactives en multimédia 3D, de données, pour les applications en ligne. Interface homme-machine [cs.HC]. Université Rennes 1, 2006. Français. NNT: . tel-00532641

HAL Id: tel-00532641

<https://theses.hal.science/tel-00532641>

Submitted on 4 Nov 2010

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.

N° d'ordre : 3362



THÈSE

à présenter devant

l'université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Nicolas BONNEL

Équipe industrielle : FRANCE TELECOM, DIVISION R&D
UNITÉ TECH/IRIS/IAM

Équipe universitaire : IRISA - ÉQUIPE TEXMEX

École doctorale : MATISSE

Composante universitaire : IFSIC

Titre de la thèse :

***Génération dynamique de présentations interactives
en multimédia 3D, de données, pour des applications en ligne***

À soutenir le 4 décembre 2006 devant la commission d'examen

Rapporteurs :	Yves	LECHEVALLIER	Directeur de recherche, INRIA Rocquencourt
	François	POULET	Enseignant/chercheur HDR, ESIEA Laval
Examineurs :	Olivier	RIDOUX	Professeur, université de Rennes 1
	Gilles	VENTURINI	Professeur, université François-Rabelais de Tours
	Max	CHEVALIER	Maître de conférences, université Paul Sabatier de Toulouse
Directrice :	Annie	MORIN	Maître de conférences HDR, université de Rennes 1
Co-encadrant :	Alexandre	COTARMANAC'H	Ingénieur, France Télécom

C'est en prévoyant l'avenir qu'on le change.

While techno-utopians believe that computing technology has steadily improved over four decades, the fact remains that it is too unreliable, too difficult to use, and too error prone. The demands and expectations of diverse users have grown faster than the quality of products.

Ben Shneiderman

Table des matières

Table des matières	1
Contributions	5
Notations	9
1 Restitution des résultats de recherche	17
1.1 Contexte de la RI textuelle	17
1.1.1 Différents systèmes de RI textuelle sur le Web	17
1.1.2 Différentes étapes d'un moteur de recherche	19
1.1.3 Interface de restitution	21
1.1.4 Évaluation de la qualité des résultats	23
1.2 Visualisation des résultats	24
1.2.1 Visualisation des attributs	27
1.2.1.1 Visualisations géométriques	27
1.2.1.2 Visualisations symboliques	27
1.2.1.3 Discussion	28
1.2.2 Visualisation des résultats sous la forme de classement(s)	28
1.2.2.1 Classement simple	28
1.2.2.2 Classement multiple	29
1.2.2.3 Classement personnalisé	32
1.2.3 Visualisation des relations entre les résultats	33
1.2.3.1 Visualisations basées sur la distribution des termes de la requête	34
1.2.3.2 Approches cartographiques basées sur la notion de graphes	37
1.2.3.3 Visualisations topographiques respectant la notion de proximité sémantique	39
1.2.3.4 Organisation basée sur une catégorisation hiérarchique des résultats	41
1.2.4 Nombre de dimensions	44
1.2.4.1 Visualisations 3D	44
1.2.4.2 2D Vs 3D	48
1.3 Méthodes d'interaction	51
1.4 Ergonomie	55

1.5	Discussion	57
2	Organisation des résultats	59
2.1	État de l'art sur l'organisation des résultats de recherche	60
2.1.1	<i>Clustering</i>	60
2.1.2	Catégories hiérarchiques à facettes	63
2.1.3	Discussion	63
2.2	Présentation de la méthode des cartes auto-organisatrices	65
2.2.1	Principe	65
2.2.2	Algorithme	65
2.2.2.1	Algorithme <i>stochastic SOM</i>	66
2.2.2.2	Algorithme <i>batch SOM</i>	67
2.2.2.3	Méthode des <i>k-means</i> ou k-moyennes	67
2.2.2.4	Fonction de voisinage	68
2.2.2.5	Initialisation	69
2.2.2.6	Critère de convergence	69
2.2.2.7	Complexité	70
2.2.2.8	Choix des distances	70
2.2.3	Étiquetage d'une carte auto-organisatrice	71
2.2.4	Extensions	72
2.2.5	Interprétation	77
2.3	Organisation des résultats	78
2.3.1	Pré-traitement des résultats	79
2.3.2	Auto-organisation des résultats	80
2.3.3	Post-traitement des résultats organisés	81
2.4	Évaluation de l'organisation	81
2.5	Discussion	82
3	Visualisation	85
3.1	Introduction	85
3.1.1	Définition des éléments à visualiser	85
3.1.2	Approche proposée	85
3.2	Métaphore 3D de la ville	86
3.2.1	Principe	86
3.2.2	Caractéristiques	86
3.2.2.1	Première version	87
3.2.2.2	Évolution	90
3.2.3	Navigation	94
3.2.3.1	Première version	94
3.2.3.2	Évolution	95
3.2.3.3	Améliorations	96
3.3	Discussion	97
3.3.1	Critères spécifiques aux métaphores 3D	97
3.3.2	Bilan	99

4	Prototype SmartWeb	101
4.1	Présentation générale	101
4.2	Architecture	102
4.2.1	Architecture d'un point de vue conceptuel	102
4.2.2	Architecture d'un point de vue implémentation	103
4.2.2.1	Partie serveur	103
4.2.2.2	Partie client	105
4.3	Caractéristiques du prototype	106
4.3.1	Langage X-VRML	106
4.3.1.1	Apports du X-VRML	106
4.3.1.2	Spécifications du X-VRML	107
4.3.1.3	Comparaison avec X3D et Java 3D	107
4.3.2	Personnalisation	109
4.4	Utilisation du prototype	109
4.4.1	Fonctionnement	109
4.4.2	Exemple	110
4.4.2.1	Spécifier sa requête	110
4.4.2.2	Utilisation de la métaphore de ville	110
4.4.2.3	Utilisation de la métaphore de la galerie	113
4.5	Applications	115
5	Évaluation	121
5.1	Première évaluation	121
5.1.1	Contexte	121
5.1.2	Protocole	122
5.1.3	Résultats	122
5.1.3.1	Profil des utilisateurs	125
5.1.3.2	Évaluation du prototype SmartWeb	135
5.1.3.3	Analyse globale des résultats	142
5.1.3.4	Discussion	145
5.2	Définition d'un nouveau cadre d'évaluation	148
5.2.1	Motivations	148
5.2.2	Évaluation d'une IUI	150
5.2.3	Critères d'évaluation	151
5.2.3.1	Critères liés à l'utilisateur	152
5.2.3.2	Critères liés à la tâche de l'utilisateur	152
5.2.3.3	Critères liés au système	153
5.2.4	La phase d'évaluation	157
5.2.4.1	Données	157
5.2.4.2	Fonctionnalités des IUI	157
5.2.4.3	Jeux de requêtes	158
5.2.4.4	Les évaluateurs	158
5.2.5	Conclusion	158
5.3	Discussion	159

Glossaire	171
A Compléments sur les méthodes de visualisation	173
A.1 Techniques basées sur la visualisation des attributs	173
A.1.1 Visualisations géométriques	173
A.1.2 Visualisations symboliques	175
A.1.2.1 Techniques d'iconification	175
A.1.2.2 Techniques de pixelisation	178
A.2 Visualisations hiérarchiques	180
A.2.1 Visualisations 2D	180
A.2.2 Visualisations 3D	181
B Algorithme <i>batch</i> SOM	185
C Informations sur l'implémentation du prototype	187
C.1 Base de données	187
C.2 Exemple de fichier X-VRML	190
D Résultats de l'ACM	197
Références	223
Liste des figures	225
Liste des tableaux	229
Liste des algorithmes	231
Index	233

Contributions

Sujet de la thèse

Cette thèse s'est déroulée dans le cadre d'une convention industrielle de formation par la recherche (CIFRE). Le sujet de thèse a été proposé par l'entreprise d'accueil, France Télécom, Division R&D. Nous rappelons le sujet initial de la thèse avant d'en proposer, avec le recul obtenu à l'issue de la thèse, une brève interprétation.

SUJET INITIAL

Génération dynamique de présentations interactives en multimédia 3D, de données, pour des applications en ligne.

De nombreux travaux ont été réalisés sur la visualisation de données et mettent en lumière la nécessité :

- de créer des métaphores de visualisation adaptées à la fois à l'utilisateur, à la tâche à réaliser (recherche sur le Web, indexation, tri de données, visualisation de l'état de systèmes complexes) et aux données ;
- de construire dynamiquement des présentations multimédia suivant ces métaphores ;
- d'agréger des contenus de types divers : texte, image, son, vidéo ;
- de construire dynamiquement les interactions que peut avoir l'utilisateur avec la présentation fournie en fonction de la tâche à réaliser. Cette interactivité peut aller jusqu'aux problématiques dites collaboratives.

Les présentations 3D permettent non seulement l'agrégation de plusieurs médias, mais aussi d'utiliser une représentation spatiale plus riche. Néanmoins, les coûts de production de tels contenus sont élevés ce qui impose le recours à des méthodes automatiques.

Cependant, la génération automatique de contenus 3D et multimédia, ne permet pas aujourd'hui de répondre à toutes les attentes exprimées ci-dessus. En effet, il ne s'agit pas seulement de construire une présentation statique mais d'y associer des comportements, ainsi que des outils permettant de re-générer, en fonction des choix de

l'utilisateur, un nouveau format de présentation (*e.g.*, dans le cas de la recherche de documents sur le Web, cela pourrait impliquer d'insérer des méthodes semi-automatiques de filtrage de documents, ou bien de réaliser de façon collaborative une recherche sur le Web).

Cela recouvre donc l'étude des liens entre les données et leur structurations possibles, l'utilisateur et la tâche à réaliser, la présentation et ses interactions. Dans d'autres termes, il est nécessaire d'étudier des algorithmes et des langages :

- pour exprimer efficacement les métaphores de présentation 3D et multimédia ;
- pour les lier aux sources de données pour former une présentation 3D et multimédia ;
- pour introduire des outils intelligents permettant d'aider de façon efficace l'utilisateur dans sa tâche ;
- pour permettre un travail collaboratif avec d'autres utilisateurs.

Les travaux s'appuient sur les langages et outils du Web (XML, *Topic Maps*), les langages pour l'infographie 3D interactive ainsi que les méthodes basées sur l'intelligence artificielle (IA).

Tout d'abord, certaines précisions doivent être apportées à cette version initiale du sujet. Notamment l'aspect multimédia n'est pas pleinement considéré. En effet, bien que les données traitées peuvent être de nature multimédia, seul le contenu textuel de ces données est utilisé dans leur traitement. Les autres médias sont donc uniquement considérés comme des attributs supplémentaires attachés aux données. Par ailleurs, cette thèse s'inscrit dans un cadre applicatif privilégié, à savoir la présentation des résultats issus d'une recherche sur le Web (*i.e.* les réponses d'un moteur de recherche). En ce sens, cette thèse aurait aussi pu être intitulée : « **Interfaces utilisateur d'information en 3D — application aux moteurs de recherche textuels sur le Web** »¹. L'objectif est donc ici de proposer des métaphores de visualisation adaptées à la restitution de résultats de recherche. Ces métaphores doivent bien entendu être interactives et générées dynamiquement. Ces aspects posent notamment plus de problèmes dans le cas de métaphores 3D. On souhaite donc aider au maximum l'utilisateur dans l'interprétation des résultats de recherche. Pour cela, il est aussi nécessaire de prendre en compte des méthodes ou algorithmes basés sur l'IA (afin de donner plus de sens aux résultats) et de s'intéresser aux langages (afin d'exprimer efficacement les métaphores de visualisation). L'aspect collaboratif mentionné à plusieurs reprises dans le sujet initial n'est pas traité dans cette thèse pour des raisons de temps. Cela ne remet absolument pas en cause cette thématique qui est particulièrement intéressante dans le cadre applicatif de recherches sur le Web. Quelques mots sur ce sujet apparaissent dans les perspectives.

¹3D Information User Interfaces — Application to Web Textual Search Engines

Contributions

Une énumération succincte des différentes contributions est donnée par la figure 1 dont le fond est un aperçu du prototype proposé dans le cadre de cette thèse. Plus de détails sur ces contributions sont donnés dans le chapitre de conclusion.



FIG. 1 – Aperçu des contributions

Liste des publications réalisées au cours de la thèse

1. NICOLAS BONNEL ET GURVAN LE GUERNIC, **Systeme de recherche de méthodes Java basé sur leur signature**, *Actes de la 4^e Manifestation des Jeunes Chercheurs Francophones en Sciences et Technologies de l'Information et de la Communication (MajecSTIC)*, à paraître, Lorient, France, Novembre 2006.
2. NICOLAS BONNEL ET MAX CHEVALIER, **Critères d'évaluation pour les interfaces des systèmes de recherche d'information**, *Actes de la 3^e Conférence en Recherche d'Information et Applications (CORIA)*, pages 109-114, Lyon, France, Mars 2006, ISBN 2-9520326-6-1.
http://hal.inria.fr/docs/00/06/31/97/PDF/Bonnel_CORIA06_web.pdf
3. NICOLAS BONNEL, VINCENT LEMAIRE, ALEXANDRE COTARMANAC'H ET ANNIE MORIN, **Effective Organization and Visualization of Web Search Results**,

Proceedings of the IASTED International Conference on Internet and Multimedia Systems and Applications (EuroIMSA), ACTA Press, pages 209-216, Innsbruck, Autriche, Février 2006, ISBN 0-88986-564-7.

http://hal.inria.fr/docs/00/06/31/99/PDF/Bonnel_EuroIMSA06_web.pdf

4. NICOLAS BONNEL ET MAX CHEVALIER, **Évaluation des Interfaces Utilisateur d'Information**, *Actes du 4^e Atelier Visualisation et Extraction de Connaissances (associé aux 6^e Journées Francophones "Extraction et Gestion des Connaissances" - EGC)*, pages 23-38, Lille, France, Janvier 2006.
http://hal.inria.fr/docs/00/06/31/98/PDF/Bonnel_EGC06_AtelierVisu_web.pdf
5. NICOLAS BONNEL, ALEXANDRE COTARMANAC'H ET ANNIE MORIN, **Visualisation 3D des résultats de recherche : quel avenir?**, *Créer, jouer, échanger : expériences de réseaux - Actes de la 8^e Conférence Hypermedias Hypertexts, Products, Tools and Methods (H²PTM)*, Hermes Science Publications, pages 325-339, Paris, France, Novembre 2005, ISBN 2-7462-1244-7.
http://hal.inria.fr/docs/00/05/61/13/PDF/Bonnel_H2PTM05_web.pdf
6. NICOLAS BONNEL ET FABIENNE MOREAU, **Quel avenir pour les moteurs de recherche?**, *Actes de la 3^e Manifestation des Jeunes Chercheurs Francophones en Sciences et Technologies de l'Information et de la Communication (MajecSTIC)*, pages 291-299, Rennes, France, Novembre 2005.
<http://hal.inria.fr/docs/00/04/35/61/PDF/49.pdf>
7. NICOLAS BONNEL, ALEXANDRE COTARMANAC'H ET ANNIE MORIN, **Meaning Metaphor for Visualizing Search Results**, *Proceedings of the 9th International Conference on Information Visualisation (IV)*, IEEE Computer Society, pages 467-472, Londres, Angleterre, Juillet 2005, ISBN 0-7695-2397-8.
http://hal.inria.fr/docs/00/05/61/10/PDF/Bonnel_IV05_web.pdf
8. NICOLAS BONNEL, ALEXANDRE COTARMANAC'H ET ANNIE MORIN, **Gestion et visualisation des résultats d'une requête**, *Actes du 3^e Atelier Visualisation et Extraction de Connaissances (associé aux 5^e Journées Francophones "Extraction et Gestion des Connaissances" - EGC)*, pages 37-47, Paris, France, Janvier 2005.
http://hal.inria.fr/docs/00/05/61/11/PDF/Bonnel_EGC05_AtelierVisu_web.pdf
9. NICOLAS BONNEL, ANNIE MORIN ET ALEXANDRE COTARMANAC'H, **Classification non supervisée et visualisation 3D de documents**, *Revue des Nouvelles Technologies de l'Information (RNTI-E-3), Numéro spécial sur les actes des 5^e Journées Francophones "Extraction et Gestion des Connaissances" (EGC)*, Cépaduès-éditions, 2 :557-562, Paris, France, Janvier 2005, ISBN 2-85428-677-4.
http://hal.inria.fr/docs/00/09/80/82/PDF/Bonnel_EGC05_web.pdf

Notations

Les notations **en gras** sont réservées aux vecteurs ou matrices, *i.e.* à des données multidimensionnelles.

$\alpha(t)$	taux d'apprentissage
$\text{BMU}(\mathbf{x}_i)$	indice de la classe dont le centre est le plus proche de la donnée \mathbf{x}_i
c	nombre de neurones (<i>i.e.</i> nombre de prototypes) ou nombre de <i>clusters</i>
$\delta_{\mathbf{k}}$	erreur quadratique calculée sur le neurone k
δ_{kj}	erreur quadratique calculée sur le neurone k et pour la variable j
ϵ	seuil de convergence
EQ	erreur quadratique
EQM	erreur quadratique moyenne
$h_{kk'}$	fonction de voisinage centrée sur le neurone k et évaluée pour le neurone k'
$i = 1 \dots n$	indice de document
idf	fréquence documentaire inverse (<i>inverse document frequency</i>)
$j = 1 \dots p$	indice de variable
$k = 1 \dots c$	indice de neurone ou de <i>cluster</i>
$\mathbf{m}_{\mathbf{k}}$	vecteur prototype du neurone k (ou représentant du <i>cluster</i> k)
m_{kj}	valeur de la j^{e} variable pour le k^{e} neurone
$\bar{\mathbf{m}}_{\mathbf{k}}$	centroïde de l'ensemble de Voronoi k (ou du <i>cluster</i> k)
mot_j	j^{e} mot du vocabulaire
n	nombre de documents du corpus
n_j	nombre de documents du corpus contenant le j^{e} mot du vocabulaire
p	nombre de variables (ou de mots dans le vocabulaire)
$\mathbf{r}_{\mathbf{k}}$	position du neurone k dans l'espace de sortie (sur la grille)
$\sigma(t)$	largeur de la fonction de voisinage h
t	discrétisation du temps (étapes ou époques)
tf	fréquence du terme (<i>term frequency</i>)
τ_1, τ_2	seuils
V_k	région de Voronoi autour du neurone (ou du <i>cluster</i>) k
w_i	poids du i^{e} document
W_k	somme des poids des documents appartenant à l'espace de Voronoi V_k
\mathbf{x}_i	vecteur représentant le i^{e} document du corpus
x_{ij}	valeur de la j^{e} variable pour le i^{e} document

Nous précisons aussi, dans cette partie consacrée aux notations, le contexte dans lequel sont utilisés certains termes particuliers dans le cadre de la thèse. En effet, l'emploi de ces termes ne correspond pas forcément à leur définition habituelle. Ainsi, nous utilisons les termes :

- **corpus** pour parler de l'ensemble des résultats issus d'une requête,
- **vocabulaire** pour parler de l'ensemble des variables ou mots sélectionnés pour représenter le corpus,
- **métaphore** pour parler de la réalisation d'une association entre des paramètres graphiques de la présentation et des informations sur les documents indexés.

Introduction

INFOBÉSITÉ (nom féminin, langue générale, 1996, Québec) :
état résultant d'une sursaturation d'informations.

<http://ocag.qc.ca/terminologie/>

La quantité de données rassemblées dans les bases de données augmente sévèrement chaque année. Une étude tente ainsi d'estimer la quantité de nouvelles informations créées chaque année [Lyman et Varian 2003]. Cette étude sur le stockage et les flux d'informations prend comme cadre l'année 2002 afin d'estimer la taille annuelle de la quantité de nouvelles informations contenues dans les supports de stockage (papier, film, magnétique et optique). Elle estime qu'environ 5 exa-octets² de nouvelles informations sont produites dans le monde, en 2002, par ces quatre supports de stockages. Cela correspond à presque 800 méga-octets (Mo) d'informations enregistrées produites par personne chaque année. La grande majorité de ces nouvelles informations, soit 92%, est stockée sur des supports magnétiques (le plus souvent, il s'agit de disques durs). Ainsi, en se comparant à la même étude réalisée trois ans auparavant [Lyman et Varian 2000], nous pouvons estimer que la quantité de nouvelles informations stockées sur ces quatre supports a plus que doublée dans les trois dernières années. Le Web, où le nombre de données disponibles explose également, n'échappe pas à cette augmentation [Gulli et Signorini 2005]. Cependant, étant donné l'absence de consensus sur le comptage entre les différentes études concernant la quantité de pages référencées par les moteurs de recherche, nous ne prendrons pas le risque de citer un chiffre. Il semble toutefois que les moteurs de recherche les plus populaires (*e.g. Google*) aient déjà largement dépassé les 20 milliards de pages. Mais cette course à la taille des index ne semble plus la principale priorité des moteurs de recherche qui ne communiquent plus sur ce point (le nombre de pages référencées n'est plus mentionné depuis septembre 2005). Cette disparition soudaine de la taille de l'index s'explique en partie par l'absence de consensus pour le comptage des pages mentionnée précédemment. Cette explosion des index fait inévitablement augmenter le nombre de réponses aux requêtes de l'utilisateur et remet d'actualité le problème d'identification des résultats pertinents parmi les réponses des moteurs. En effet, le simple classement linéaire selon un critère de pertinence calculé par le système (*e.g. le PageRank* [Brin et Page 1998] utilisé par *Google*) semble atteindre ses limites. Mais, malgré les faiblesses actuelles des moteurs de recherche [Bonnell et Moreau 2005], il ne faut

²1 exa-octet (Eo) = 10⁹ giga-octets (Go) = 2⁶⁰ octets

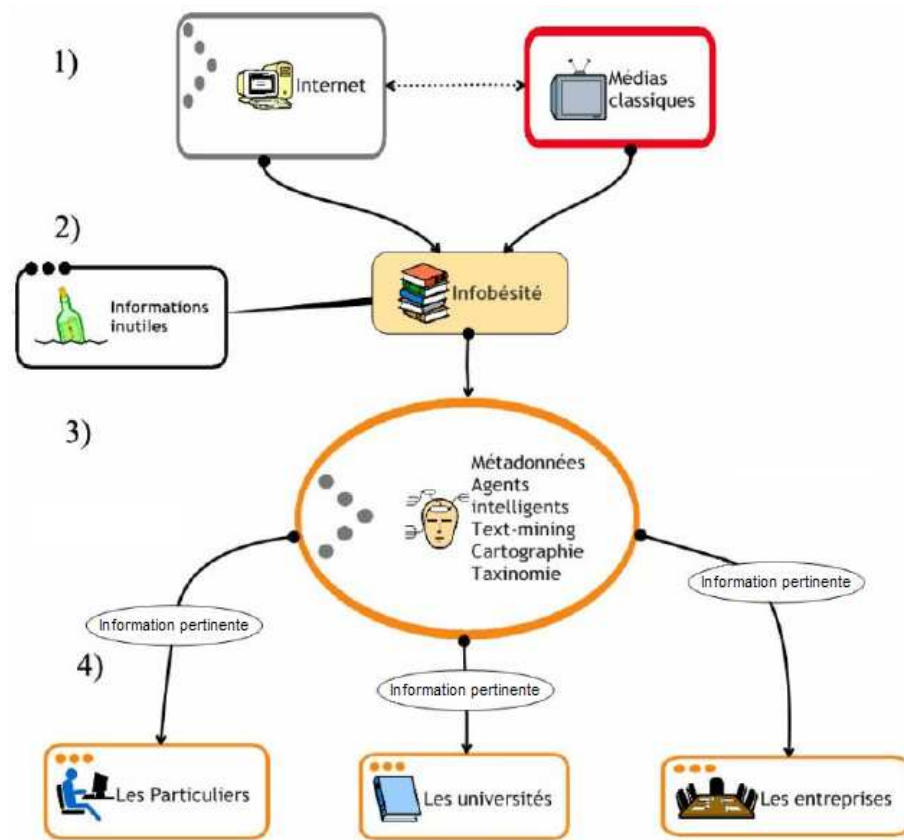
pas oublier que, lorsque les utilisateurs ont une nouvelle tâche à accomplir sur le Web, ils démarrent — 88% du temps — par l'utilisation d'un moteur de recherche [Nielsen 2004].

C'est dans ce contexte de profusion d'informations que le terme d'infobésité³ est apparu il y a déjà quelques années. Ce néologisme, d'origine québécoise, désigne le surplus d'information dans lequel l'utilisateur est de plus en plus plongé. Afin de lutter contre cette surcharge de données, il faut alors réduire l'écart, sans cesse grandissant, entre la quantité d'informations disponibles et les outils qui permettent de transformer ces données en connaissances utiles. Ce processus est illustré sur la **figure 2** proposée par Jean-Paul Pinte, enseignant-chercheur à l'université catholique de Lille. Pourtant, d'après ce dernier, l'infobésité n'est pas si terrible. En effet, il espère que le milieu académique pourra contrebalancer ce phénomène, voire même en profiter, en inondant à son tour le Web de ses propres travaux. Cependant, cette solution n'a pas vocation à contrer la prolifération de l'infobésité, bien au contraire ! Mais elle a au moins le mérite de diriger les infophages vers des sources d'informations de meilleure qualité, en mesure d'améliorer leurs connaissances et leur efficacité. Le problème est alors que le public reste infobèse. En effet, la préoccupation majeure semble être de modifier la qualité des sources d'information (en influant sur le ratio et donc en ajoutant de nouveaux documents sur le Web), et non celle de modifier les comportements des utilisateurs.

Un des problèmes actuels est que l'information la plus facilement accessible par les moteurs de recherche n'est généralement pas la plus rigoureuse, tant en termes de pertinence que de qualité. Notons, par exemple, que les moteurs de recherche tels que *Google* ont la mauvaise habitude de placer au même degré d'importance un article scientifique tiré d'un site universitaire, la page promotionnelle d'un site Web d'entreprise et un texte d'opinion publié sur un blogue personnel. La rigueur et l'éthique varient sensiblement d'un cas à l'autre. Concernant l'infobésité, ainsi que les problèmes liés à la pertinence et à la qualité des résultats, Stephen Bell, bibliothécaire à l'université de Philadelphie et chercheur spécialisé dans les sciences de l'information, en impute la faute au moteur de recherche *Google* mais aussi au comportement des utilisateurs [Bell 2003, 2004, 2005].

Think of the library as the carving station that's been abandoned while the diners line up for greasy burgers and fries. Too many students are bringing a "supersize it" mentality to research, in effect asking librarians, "Can you Google-ize that for me?" (...) the term "infobesity", which nicely describes the outcome of Google-izing research : a junk-information diet, consisting of overwhelming amounts of low-quality material that is hard to digest and leads to research papers of equally low quality. (...) The challenge is getting students to move from infobesity to in-

³En anglais, cette notion, initialement désignée par l'expression « *information overload* », est désormais traduite par le terme « *infobesity* ».



- 1- Les principales sources d'information viennent de l'ensemble des médias, c'est-à-dire de la télé, de la radio, des journaux, etc., et bien sûr d'Internet. Ces deux grandes sources d'information se complètent souvent entre elles aujourd'hui par le jeu de l'hypertexte et donc le renvoi vers une ressource de la toile.
- 2- Face à cette massification de l'information, l'homme arrive vite à une surcharge informationnelle, qualifiée aujourd'hui d'infobésité. De cette masse d'information, il lui faut alors sélectionner et extraire l'essentiel.
- 3- Il existe différents moyens et méthodes pour sérier l'information, comme l'utilisation de métadonnées (données sur des données) pour décrire le contenu et la fonctionnalité de sources, les agents intelligents pour automatiser et filtrer l'information, l'exploration de texte (*text mining*) ou l'analyse de texte pour en retirer l'essentiel, la cartographie pour représenter de façon claire et synthétique de l'information sous forme de carte et, enfin, la taxinomie pour classer et répertorier l'information afin d'aller directement à l'essentiel.
- 4- Ces méthodes vont permettre d'extraire de la masse l'information pertinente pour toute personne qui recherche sur Internet; pour l'université dans le cadre d'une gestion de la connaissance; et pour l'entreprise, afin de réaliser des gains de temps dans la recherche d'une information stratégique dont elle a besoin (sur un produit, sur un concurrent, sur un marché, etc.).

FIG. 2 – L'homme face à l'infobésité [Pinte 2005]

fodieting. When confronted with aggregator databases and online library catalogs, students' eyes glaze over, not unlike those of a fast-food supersizer confronted with a nice, healthy salad plate. "Can't this be more like Google?" students ask. (...) isn't it more sensible to just make everything like Google? Perhaps. But we need to encourage our users to migrate from low-quality Web sites to the high-quality information we can provide. The trick is making the salad smell and taste like fast food, so they'll eat it. (...) Supersized search engines that imitate Google, producing piles of full-text articles, may initially be as satisfying as a candy bar. But empty calories — or citations — are not what the educated consumer wants.

Steven J. Bell, director of the library at Philadelphia University, USA.

Il existe plusieurs manières de lutter contre cette surcharge d'informations. Mais nous nous intéressons plus spécialement ici aux techniques et outils de visualisation qui sont une étape essentielle pour faire face à l'infobésité. En effet, dans un processus de recherche d'information sur le Web, les utilisateurs pensent souvent que le travail le plus dur est fait par les moteurs de recherche ou par les annuaires qui sont chargés de trouver les pages Web. Bien que cela soit partiellement vrai, une partie non moins importante du travail est faite par l'utilisateur qui doit décider quelle page est pertinente parmi l'énorme ensemble de résultats. Ainsi, la problématique de cette thèse concerne la restitution des résultats issus d'un système de recherche d'information (SRI) et plus particulièrement d'un moteur de recherche. Au delà de la simple visualisation des résultats de recherche, l'objectif est aussi d'analyser et de traiter ces résultats afin d'en extraire le maximum d'informations pour aider l'utilisateur dans sa tâche. Pour exploiter au maximum les résultats, il est nécessaire de s'intéresser à différentes thématiques. Cela explique que cette thèse s'articule autour de plusieurs domaines de recherche.

- Le premier domaine est celui des interfaces Homme-Machine (IHM). En effet, le point central de nos travaux concerne les interfaces utilisateur d'information dans le cas de recherches sur le Web. Différents aspects sont alors couverts tels que la visualisation, l'interaction avec l'utilisateur mais aussi la navigation dans l'espace des résultats. L'évaluation de l'utilisabilité de telles interfaces est également étudiée.
- Le second domaine concerne la catégorisation automatique (ou encore le *clustering*) de documents textuels. Cela passe par des techniques dont l'objectif est de fournir automatiquement des regroupements de documents permettant ainsi d'obtenir des informations utiles sur l'ensemble des résultats de recherche. Ces méthodes sont largement employées dans les communautés de la fouille de données (*data mining* et *web mining*) ou de l'extraction de connaissances (*knowledge discovery*). Dans nos travaux, l'analyse des documents textuels n'est abordée que d'un point de vue statistique. Néanmoins, nous précisons que nos travaux n'ont pas vocation à apporter leur contribution principale dans ce domaine.
- Le troisième domaine est celui de la recherche d'information (RI) textuelle, bien qu'il soit plus lié à notre cadre applicatif. Les données considérées sont donc des documents textuels, ce qui pose certains problèmes spécifiques. Notam-

ment l'appariement exact entre la requête de l'utilisateur et les documents de la base de données n'est pas forcément une solution adaptée, contrairement à d'autres types de données. En effet, la présence de synonymes dans les différentes langues témoignent de ce problème. Par ailleurs, bien que l'usage des moteurs de recherche ainsi que la quantité de données disponibles aient changé au cours des dernières années, les utilisateurs n'ont pas évolué dans leur manière d'exprimer leurs besoins. Ainsi, les requêtes soumises sont généralement courtes et composées de seulement quelques mots (généralement deux mots). Le défi est donc énorme : satisfaire les besoins d'un utilisateur à partir de seulement deux mots !

Le domaine du traitement automatique des langues (TAL) propose un ensemble de techniques et de méthodes qui s'avèrent utiles dans le traitement de données textuelles. Par exemple ces outils de TAL sont souvent utilisés afin d'améliorer la qualité des réponses issues des SRI. Cependant, dans le cadre de cette thèse, l'objectif n'est pas d'améliorer la qualité des résultats obtenus pour une requête donnée. Il s'agit plutôt d'aider l'utilisateur dans la compréhension et l'exploitation de ces résultats. Ainsi ce domaine n'est pas traité dans cette thèse, bien que certaines techniques de TAL pourraient être utilisées dans l'étape de catégorisation des résultats. L'approche multi-thématique proposée est particulièrement intéressante et a déjà été exploitée dans de précédentes thèses [Chevalier 2002, Mann 2002, Käki 2005].

Afin de préciser davantage le contexte de la thèse, il est essentiel de définir les données que nous souhaitons traiter. Il s'agit de documents textuels de type pages Web. Ces documents peuvent aussi contenir d'autres médias (*e.g.* des images, des vidéos ou encore des sources audio) qui ne sont pas exploités dans l'organisation des documents mais qui peuvent être utilisés comme des attributs descriptifs des documents lors de la visualisation. Une autre contrainte à définir est la quantité de données à traiter. En effet, le volume de données est un paramètre important qui influence fortement les choix d'organisation et de visualisation des résultats. Notre cadre applicatif étant la visualisation des résultats issus d'une requête, il semble alors cohérent d'étudier le comportement des utilisateurs. Une étude [iProspect 2006] révèle que 88% des utilisateurs ne dépassent pas les 3 premières pages de résultats. D'après cette étude, une solution basique consiste alors à ne considérer que les 30 premiers documents (10 résultats par page). Mais cela ne nous semble pas justifié étant donné que les résultats de l'étude citée sont uniquement basés sur une visualisation linéaire des résultats. Dans le cas de visualisations plus riches (2D ou 3D), l'utilisateur peut être tenté de dépasser le seuil des 30 premières réponses. Devant l'absence d'étude sur ce type d'interfaces, le nombre de documents à considérer dans cette thèse est fixé à une ou deux centaines.

La restitution des résultats de recherche implique de s'intéresser à différentes tâches dont les deux principales sont la visualisation et l'organisation des résultats. Ainsi, la thèse s'organise de la façon suivante. Le **chapitre 1** propose un état de l'art sur la restitution des résultats de recherche, qui est essentiellement présenté d'un point de vue visualisation (y compris les aspects navigation, interaction et ergonomie). En effet,

pour faciliter la lecture de ce manuscrit, l'état de l'art sur l'organisation est proposé au début du chapitre 2 qui se consacre à cette tâche. Dans ce premier chapitre, l'idée consiste à dresser un panorama des différentes méthodes employées, sans chercher à être exhaustif du point de vue des différentes implémentations de ces méthodes (*i.e.* des différents systèmes proposés). Nous y proposons aussi une taxonomie de ces différentes méthodes. Ensuite, le **chapitre 2** s'intéresse plus particulièrement à l'organisation des résultats, avec la contrainte d'utiliser des méthodes statistiques à la fois déterministes et non supervisées. L'objectif est de proposer une organisation efficace des résultats afin de permettre à l'utilisateur de retrouver le plus rapidement possible les documents pertinents à sa requête. Pour cela, il est nécessaire d'être capable de rassembler les résultats similaires (*clustering*) mais aussi de positionner les différents groupes de résultats les uns par rapport aux autres. Ainsi, l'approche proposée est basée sur les cartes auto-organisatrices. Le **chapitre 3** présente la métaphore proposée pour visualiser les résultats. Il s'agit d'une ville 3D virtuelle organisée par quartiers et dans laquelle les bâtiments représentent les résultats. Les aspects navigation et interaction sont aussi abordés dans ce chapitre. Il est donc maintenant possible de s'intéresser à la mise en œuvre des solutions proposées pour l'organisation des résultats (chapitre 2) et pour leur visualisation (chapitre 3). C'est l'objet du **chapitre 4** qui présente le prototype *SmartWeb* aussi bien du point de vue de l'architecture que de celui de l'implémentation. Le fonctionnement du prototype est également expliqué à travers un exemple typique d'utilisation. Puis, le **chapitre 5** propose une évaluation du prototype et donc des solutions proposées dans cette thèse. Une évaluation standard est d'abord proposée, puis nous en tirons des conclusions et nous proposons une nouvelle démarche d'évaluation afin de pallier aux problèmes rencontrés lors de l'exploitation des résultats de notre premier test utilisateur. Enfin nous terminons par un bilan des solutions proposées et en rappelant les différentes contributions de nos travaux. Une large place est également faite aux nombreuses perspectives applicatives et pistes de recherche qui découlent de ces travaux.

Chapitre 1

Restitution des résultats de recherche

Ce chapitre a pour objectif de dresser un état de l'art sur la restitution des résultats de recherche. Comme cela est déjà précisé en introduction, notre cadre applicatif concerne les outils de RI textuelle et les données que l'on souhaite visualiser sont des documents de type page Web. Ces deux points permettent de délimiter notre état de l'art, car il n'est bien sûr pas envisageable de traiter dans son ensemble le vaste domaine de la visualisation de données. Par ailleurs, l'expression « restitution des résultats » est suffisamment floue pour ne pas en voir les limites. Nous précisons donc que notre utilisation de cette expression regroupe les notions de visualisation, d'interaction, de navigation, d'ergonomie et d'organisation. Cependant, ce dernier point (l'organisation des résultats) n'est pas vraiment traité dans ce chapitre mais dans le chapitre suivant dédié à cette problématique.

1.1 Contexte de la RI textuelle

Il existe différentes tâches en RI. Les deux principales tâches consistent à :

- naviguer dans une collection de documents afin de trouver des informations pertinentes,
- rechercher des informations pertinentes par l'expression d'une requête.

Dans le cadre de cette thèse, nous nous intéressons essentiellement au deuxième type de tâche qui est celui auquel répondent les moteurs de recherche. Ces deux tâches ne sont toutefois pas totalement exclusives du fait qu'il est bien souvent nécessaire de naviguer dans les résultats de recherche obtenus en réponse à une requête, pour trouver des documents pertinents. Et ces résultats sont de plus en plus nombreux.

1.1.1 Différents systèmes de RI textuelle sur le Web

Il existe de nombreux portails pour accéder aux informations disponibles sur le Web. La plupart proposent un moteur de recherche en texte intégral sur les pages du

Web (*e.g. Google*). Il s'agit de l'approche la plus utilisée actuellement. Ce type de moteur de recherche nécessite que l'utilisateur exprime ses besoins sous la forme d'une requête qui est interprétée et exécutée. Le moteur cherche alors les documents qui sont les plus pertinents aux mots-clés de la requête et les présente à l'utilisateur. Les méthodes utilisées pour calculer la pertinence diffèrent selon les moteurs de recherche mais les facteurs généralement utilisés sont : le nombre d'occurrences du mot-clé dans le document, la position du mot-clé dans le document, la présence du mot-clé dans le titre du document, la conformité entre les mots-clés de la requête et ceux fournis par l'auteur du document, l'identification de la phrase... Les avantages de cette approche sont le grand nombre de documents indexés dans la base, le mécanisme de ré-indexation, la rapidité de l'interprétation de la requête et de la présentation des résultats. Par contre la pertinence des informations trouvées dépend d'algorithmes intégrés et reste donc en dehors du contrôle des utilisateurs. De plus, des requêtes trop vastes rendent un nombre trop important de documents dont l'analyse est impossible avec les méthodes de visualisation actuellement proposées.

Bien que les moteurs par requêtes soient la méthode la plus utilisée, il existe d'autres approches pour rechercher des informations sur le Web. Ainsi, les trois systèmes cités ci-dessous, font partie des exemples qui se distinguent des moteurs par requêtes grâce à certaines propriétés spécifiques. Cependant, leur fonctionnement global ne diffère que très peu des moteurs par requêtes évoqués ci-dessus.

Les annuaires (ou *directory*). Ils sont basés sur une hiérarchie de catégories définie manuellement. Chaque catégorie est constituée de sous-catégories et/ou d'une liste d'URL. Les annuaires permettent alors d'avoir un classement thématique des documents, sachant qu'un document peut appartenir à plusieurs catégories en même temps. Ils se présentent donc comme un inventaire, disciplinaire (spécialisé dans un domaine) ou généraliste, dans lequel les sites référencés sont classés par catégorie. L'ajout de documents dans cette hiérarchie peut se faire manuellement (vérification du choix utilisateur par l'administrateur) ou automatiquement (ajout du document dans la catégorie choisie par l'utilisateur sans vérification). Deux utilisations de ce type d'annuaire sont possibles. La première consiste à naviguer dans l'annuaire afin de trouver les informations recherchées. Dans ce cas, il n'y a pas de requête de l'utilisateur, ce qui correspond à la tâche de RI dite de navigation. La seconde utilisation est similaire à celle des moteurs par requête, à l'exception près que le système connaît, pour chaque résultat, la catégorie à laquelle il appartient. Dans notre contexte, seule la deuxième utilisation qui se rapproche des moteurs par requête est envisageable. La qualité des annuaires est généralement évaluée par le nombre de documents disponibles ainsi que par la structure des catégories. Les inconvénients majeurs de cette approche sont le faible nombre de documents disponibles, l'absence d'un mécanisme d'expiration et la possibilité d'avoir des catégories ambiguës.

Les méta-moteurs. Ils peuvent être assimilés à des moteurs par requêtes ayant la particularité d'envoyer la requête à plusieurs moteurs simultanément et de faire une synthèse des résultats ainsi obtenus. Les principaux défis des méta-moteurs sont de trouver une méthode efficace pour fusionner les listes de résultats provenant des différents moteurs interrogés, ainsi que de choisir pertinemment ces moteurs.

Les moteurs *peer to peer*. L'idée du concept *peer to peer* vient des systèmes de recherche de fichiers musicaux. Un moteur de recherche *peer to peer* ne requiert pas de serveur central contenant les informations. L'idée est de propager une requête à travers le réseau aux ordinateurs participant à ce système. Les ordinateurs ne possédant aucune information demandée passent la requête. L'ordinateur possédant l'information désirée envoie l'information au demandeur en réalisant le chemin inverse sur le réseau. Le demandeur peut alors accéder aux documents en se connectant directement au fournisseur de l'information. Les avantages de cette technique sont l'absence de serveur central sophistiqué et coûteux, et la présence d'une mise à jour des informations stockées. Par contre cette approche nécessite d'équiper les ordinateurs participant à ce système d'un logiciel spécial. Elle prend également plus de temps et la pertinence des résultats dépend d'algorithmes intégrés et d'une description fournie par le propriétaire de l'information. Bien que cette méthode diffère de par les informations disponibles et leur transfert sur le réseau, du point de vue de l'utilisateur ce système peut être assimilé à un moteur par requête.

1.1.2 Différentes étapes d'un moteur de recherche

La RI sur le Web passe généralement par l'expression d'une requête sur un moteur de recherche. Le fonctionnement de ces moteurs peut être décomposé en différentes étapes, décrites ci-dessous.

La **figure 1.1** est composée de 6 différents modules (utilisateur, interface, indexation, recherche, bases d'index et robots) à travers lesquels sont réalisées différentes actions qui sont numérotées en fonction de leur ordre d'apparition dans le processus de RI. Une distinction peut être faite entre les actions 1 et 2 qui sont réalisées hors ligne (*i.e.* préalablement à l'expression de la requête), et les actions 3 à 7 qui sont réalisées en ligne (*i.e.* lors de l'exécution de la requête). On propose alors de décomposer le fonctionnement des moteurs de recherche en 4 « macro » étapes qui peuvent contenir plusieurs des actions identifiées sur la figure.

1. **Collecte des documents** (partie jaune de la **figure 1.1**). Des robots logiciel (souvent appelés *crawlers* ou *spiders*) parcourent le Web, de liens en liens, afin de recenser automatiquement les adresses des pages visitées (**action 1** de la figure). Cette exploration se fait à des fréquences assez régulières. Plusieurs centaines de robots peuvent ainsi être déployés en parallèle, et ils sont désormais capables de collecter des documents contenus dans des fichiers de différents formats (*e.g.* HTML, PDF, DOC, PPT, XLS).

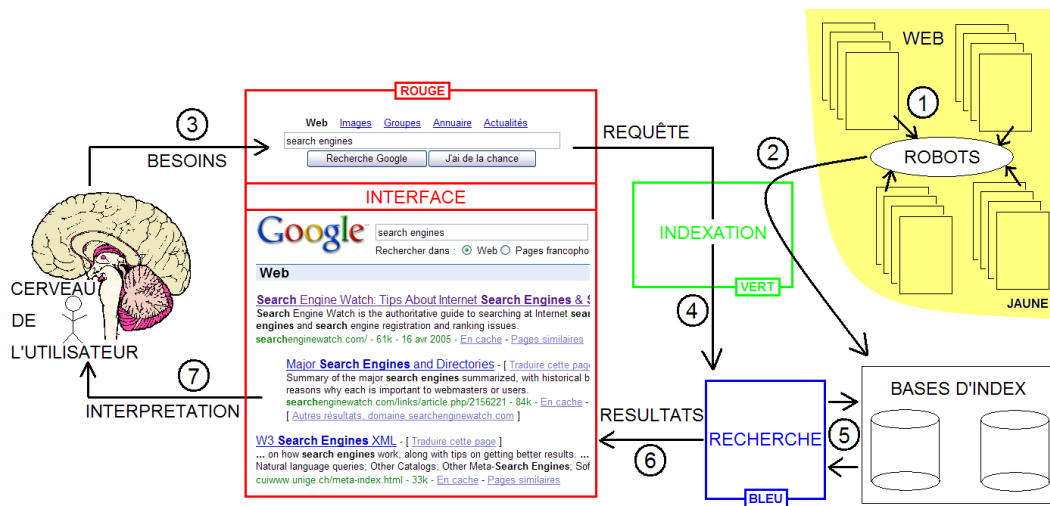


FIG. 1.1 – Description simplifiée du fonctionnement d'un moteur de recherche [Bonnell et Moreau 2005]

2. **Indexation des documents** (partie verte de la **figure 1.1**). L'étape d'indexation a pour objectif d'analyser les pages précédemment collectées afin de construire automatiquement des index des données (**action 2** de la figure). L'index d'un moteur constitue sa base principale de données. Il est composé d'un index principal qui contient les pages Web (documents) capturées par les robots et de fichiers inverses qui contiennent les termes et renvoient aux pages Web. L'indexation des documents peut se faire de façon manuelle ou automatique. Cette indexation peut aussi varier selon certains éléments tels que la taille de l'index, les objets de l'indexation (texte intégral ou champs spécifiques), les méthodes d'indexation (traitement linguistique...), les différences d'indexation selon la nature des ressources (format, texte, image).
3. **Module de recherche** (partie bleue de la **figure 1.1**). L'utilisateur exprime ses besoins d'informations sous la forme d'une **requête par mots-clés** (**action 3** de la figure). Cette requête peut être booléenne ou obéir à une syntaxe d'interrogation plus spécifique. La plupart des moteurs de recherche permettent à l'utilisateur de préciser sa requête en utilisant des opérateurs d'algèbre booléenne ou des opérateurs spécifiques, en spécifiant certaines propriétés sur les documents cherchés (e.g. la date, la langue, la taille ou le type), en utilisant le langage naturel ou bien des dispositifs d'assistance dans la recherche (tels que le tri des résultats, le *clustering*, le raffinement de la requête, l'inclusion des mots de même racine, la vérification de l'orthographe, les requêtes similaires, la recherche de contenus multimédia). Une fois exprimée, la requête est indexée de

façon similaire aux pages Web (**action 4** de la figure). Ensuite le module de recherche effectue un appariement entre la requête et les éléments contenus dans la base d'index (**action 5** de la figure). Cet appariement se fait selon le **modèle de RI** choisi [Baeza-Yates et Ribeiro-Neto 1999]. Il existe différents types de modèles de RI tels que les modèles booléens, les modèles vectoriels ou encore les modèles probabilistes. Le modèle vectoriel, utilisé dans la suite de la thèse, consiste à représenter les requêtes et les documents dans un espace vectoriel engendré par les termes d'indexation.

4. **Restitution (ou visualisation)¹ des résultats** (partie rouge de la **figure 1.1**). Les résultats sont généralement restitués sous la forme d'une liste triée selon un score ou rang (**action 6** de la figure). Il existe différentes méthodes de tri mais les plus courantes sont :
- tri par **pertinence** : calcul du score en fonction de la place des mots dans le document, de la fréquence d'occurrence dans le document par rapport à sa taille, du poids des mots dans la base, de la correspondance d'expression, de la proximité des termes de la requête entre eux...
 - tri par **popularité** : calcul du score basé sur les co-citations (algorithme *Page-Rank* [Brin et Page 1998]) ou sur des mesures d'audience.
 - tri par **calcul dynamique de catégories (clustering)**

Ces listes ordonnées sont généralement affichées sous la forme d'une succession de pages contenant chacune 10 résultats. Il existe cependant d'autres interfaces de représentation des résultats, notamment sous forme cartographique. Enfin l'utilisateur entre à nouveau en jeu afin d'interpréter les résultats fournis par le moteur de recherche (**action 7** de la figure).

La manipulation des systèmes de recherche d'information (SRI), et notamment des moteurs de recherche dans le cas de la RI sur le Web, se fait généralement au travers d'un frontal permettant à l'utilisateur de spécifier sa requête et d'en visualiser les résultats. On constate alors un décalage entre le nom de ces outils et leur fonction du point de vue des utilisateurs. C'est pourquoi nous parlerons désormais dans la suite de la thèse d'**interfaces utilisateur d'information (IUI)**. Ce terme a été proposé par Franck Poisson, ex-Directeur Général et fondateur du bureau France de *Google*.

1.1.3 Interface de restitution

Un des aspects importants pour trouver l'information appropriée parmi les résultats d'une requête est une visualisation pertinente. Trois méthodes de visualisation des résultats peuvent alors être distinguées : la méthode textuelle, la méthode graphique en 2D et la méthode graphique en 3D.

¹Dans cette thèse, on considère la restitution des résultats que sous son aspect visuel. C'est pourquoi les termes **restitution** et **visualisation** peuvent être considérés comme « similaires ». Cependant, bien entendu, le terme de restitution est plus général étant donné qu'il prend aussi en compte d'autres types de restitution (e.g. sonore).

Méthode textuelle. C'est la méthode utilisée par les principaux² moteurs de recherche. Elle consiste à présenter les résultats sous la forme d'une liste. Chaque entrée de la liste est un résultat généralement constitué du titre, d'une description ou du *snippet*³ et de l'URL. Habituellement, le nombre de réponses trouvées est très grand, ce qui explique que la liste soit affichée partiellement (*i.e.* le plus souvent sous la forme d'une succession de pages contenant chacune 10 résultats). Les propriétés de cette liste peuvent être diverses selon le moteur utilisé : titre, description, *snippet*, URL, date de modification, taille, langue, lien vers une traduction, tri des résultats, pages liées sémantiquement à la requête, affichage de plusieurs pages d'un même site, affichage des pages référencées (liens entrants et sortants d'un document). . . Les listes de résultats sont souvent adaptables par l'utilisateur (*e.g.* le nombre de documents par page ou l'affichage de la description détaillée). Cependant, la visualisation textuelle des résultats possède certains inconvénients :

- Le nombre de résultats pouvant être affichés simultanément est limité.
- La disponibilité des résultats est réduite du fait de l'affichage d'un nombre limité de pages par les moteurs de recherche. Le nombre de résultats présenté n'excède souvent pas plusieurs pourcents du nombre total de résultats satisfaisant la requête.
- Un seul critère de classement peut être utilisé dans la visualisation des résultats. En général ce critère représente la pertinence des documents par rapport à la requête, ce qui permet de ranger les résultats par ordre décroissant de pertinence dans la liste des réponses.
- L'impossibilité de comparer facilement différentes recherches. Il n'est pas possible de comparer les résultats provenant de différentes requêtes ou de comparer différentes catégories de pertinence en même temps.
- L'interaction avec l'utilisateur est très limitée. Seul l'accès au document est proposé et aucune sélection d'un groupe de documents n'est possible.
- La perception de l'information textuelle est moins efficace que sa représentation graphique.

Méthode graphique en 2D ou 3D. Ces méthodes sont détaillées dans la section suivante. Mais on peut déjà noter que ces visualisations graphiques des résultats possèdent aussi des inconvénients :

- Bien souvent les méthodes de navigation manquent d'efficacité et sont rarement intuitives. Cela est particulièrement vrai pour les visualisations 3D qui ne sont pas familières pour les utilisateurs.
- Augmentation du temps de calcul pour afficher de telles visualisations.
- Émergence de problèmes liés à la 3D : occultations, choix des points de vue les plus pertinents. . .

²d'après les parts de marché

³Le *snippet* est constitué d'extraits du contenu textuel du résultat, permettant de voir dans quel contexte apparaissent les mots de la requête.

1.1.4 Évaluation de la qualité des résultats

L'évaluation d'un moteur de recherche (ou d'un SRI) consiste généralement à mesurer la qualité des résultats obtenus. La mise en œuvre de cette évaluation s'intéresse alors uniquement à la pertinence des résultats [Mizzaro 1997, 1998]. Il existe pour cela un ensemble de mesures principalement basées sur les notions de rappel et de précision. Soit DP l'ensemble des documents pertinents pour la requête et D l'ensemble des documents présentés à l'utilisateur.

- Mesures classiques [Kent *et al.* 1955], exprimées ici sous forme ensembliste.
- **précision** (P) et **rappel** (R) :

$$P = \frac{|DP \cap D|}{|D|} \quad R = \frac{|DP \cap D|}{|DP|}$$

- **silence** (S) et **bruit** (B) :

$$S = 1 - R = \frac{|DP - D|}{|DP|} \quad B = 1 - P = \frac{|D - DP|}{|D|}$$

- Autres mesures
- **F-mesure** (F) : elle représente la moyenne harmonique pondérée des taux de rappel et de précision. La formule générale de la F-mesure est donnée ci-dessous.

$$F_\alpha = \frac{P \times R}{(1 - \alpha)P + \alpha R} \quad \alpha \in [0, 1]$$

On a alors les relations suivantes : $F_0 = R$ et $F_1 = P$. La valeur la plus courante pour α est 0.5, ce qui permet de donner le même poids au rappel qu'à la précision. Ainsi, la F-mesure s'écrit

$$F_{0.5} = \frac{2 \times P \times R}{P + R}.$$

La F-mesure est issue de la **E-mesure**⁴ (E) qui est beaucoup plus rarement utilisée [van Rijsbergen 1979]. La E-mesure peut s'exprimer de la manière suivante :

$$E = 1 - \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = 1 - F.$$

- **Document cut off values** (DCV) : il s'agit plutôt d'un critère permettant de fixer le nombre de documents à examiner lors du calcul de différentes mesures. Ainsi, on peut, par exemple, calculer une précision avec un DCV à 20 : $P(20)$. Cela signifie que le calcul de la précision ne se base que sur les 20 premiers documents, c'est-à-dire ceux ayant obtenu les plus grands scores (dans le cas d'un modèle vectoriel).

⁴Le « E » vient de *efficiency*.

- **Précision moyenne interpolée (IAP)** : moyenne des 11 précisions globales du système correspondant aux DCV pour lesquels le rappel vaut 0, 10, ..., 90 et 100%. Si certains points ne peuvent pas être atteints en fixant un DCV, les mesures sont interpolées.
- **Précision moyenne non interpolée (NIAP ou MAP)** : moyenne des précisions obtenues pour tous les DCV correspondant au rang d'un bon document dans la liste des réponses.

$$NIAP = \frac{\sum_{d \in |DP \cap D|} P(\text{rang}(d))}{|DP|}$$

Ces différentes mesures permettent d'évaluer la qualité des résultats retournés par les SRI. Leur calcul nécessite toutefois de disposer de jugements de pertinence pour chacun des documents du corpus pour une requête donnée. Cependant l'évaluation de tels systèmes ne doit plus se limiter à la mesure de la qualité des résultats, même si cela reste une préoccupation majeure. Elle doit désormais prendre aussi en compte la restitution des résultats, c'est-à-dire leur organisation et leur visualisation. Pour cela, il est nécessaire de s'interroger sur la manière d'évaluer efficacement une interface de restitution. Ce point est traité dans le **chapitre 5**.

1.2 Visualisation des résultats

La visualisation ou restitution de l'information constitue désormais une phase essentielle dans un processus de RI. Un nombre considérable de méthodes ont déjà été proposées dans le vaste domaine de la visualisation de l'information (**IV**). Cependant, seul le sous-ensemble de ce domaine restreint à la visualisation de résultats de recherche est considéré dans cette étude. Cette section présente les techniques de visualisation les plus courantes en respectant une classification précise.

Il existe de nombreuses méthodes de visualisation de données, et de nombreuses classifications possibles de ces différentes techniques. Un premier exemple de classification consiste à regrouper les différentes techniques selon trois catégories : la visualisation de la structure de l'information, la visualisation des données multivariées et visualisation de l'ensemble des objets d'information [Keim *et al.* 2002]. Si on considère cette classification, alors les techniques de visualisation de l'ensemble des objets d'information semblent les plus adaptées à notre cas. Mais, nous préférons nous intéresser plus particulièrement aux classifications uniquement dédiées aux techniques de visualisation des résultats de recherche [Hascoët et Beaudouin-Lafon 2001] ; et notamment à la classification proposée par Oren Zamir, qui repose simplement sur deux niveaux [Zamir 1998]. Le premier niveau permet de différencier les visualisations des attributs des documents et les visualisations des similarités inter-documents. Cette distinction, bien qu'intéressante, ne nous paraît pas la plus pertinente. En effet, la quasi-totalité des visualisations basées sur les similarités inter-documents permettent

aussi de visualiser des attributs des documents. Par ailleurs, le second niveau proposé s'attache plus aux techniques employées qu'aux éléments visualisés. Ce problème peut être résolu en modifiant ce second niveau afin de s'abstraire des techniques employées [Chevalier 2002]. Cependant, bien que basée sur le type d'objets visualisés, la taxonomie proposée par Max Chevalier reste dépendante du premier niveau de hiérarchie proposé par Oren Zamir. Nous proposons donc d'établir une nouvelle taxonomie des techniques de visualisations des résultats issus d'un SRI (*i.e.* une taxonomie des IUI). Cette classification se doit d'être indépendante des techniques employées, des attributs (ou objets) visualisés, du nombre de dimensions (1D, 2D ou 3D) et de la notion de hiérarchie. Elle est essentiellement basée sur le placement des résultats, c'est-à-dire sur leur organisation visuelle. Une première tentative prenant en compte cette notion de placement a été réalisée [Bonnell *et al.* 2005b], mais elle ne remet pas en question le premier niveau de hiérarchie proposé par Oren Zamir. Cela est désormais le cas dans la nouvelle proposition de taxonomie (*cf.* **figure 1.2**), qui est donc basée sur le placement des résultats, et non sur les différentes implémentations possibles pour une catégorie donnée. Les notions de hiérarchie et de dimensionalité ne sont pas prises en compte dans cette classification bien qu'elles permettent de définir les objectifs de la visualisation. Étant donné qu'elles peuvent être appliquées à toutes les catégories, elles ne sont plus considérées comme des caractéristiques supplémentaires. Dans la suite, des exemples de visualisations sont présentés pour les différentes catégories de la classification proposée.

Afin de clarifier les différentes catégories de la taxonomie, nous utilisons un modèle d'information, général et formel, qui permet de décrire des données de type pages Web [Keim *et al.* 2002]. Les notations initialement proposées par les auteurs du modèle sont conservées. Ainsi, dans notre cas, tout ou une partie du Web constitue l'ensemble discret (noté IM) des objets d'information. Chaque objet d'information (noté IO) représente donc une page ou un site Web (**propriété 1** du **tableau 1.1**). Ces objets d'information sont caractérisés par un ensemble d'attributs (noté AM). Chaque attribut (tel que le nombre d'occurrences des différents termes, la taille, la pertinence, les thèmes...) peut être défini dans un intervalle de valeurs continues ou catégoriques, afin de décrire les propriétés des objets et les caractéristiques de l'information (**propriété 2** du **tableau 1.1**). Les attributs définissent des dimensions et engendrent l'espace d'information (noté IR), et les intervalles de valeurs des attributs définissent l'échelle des axes correspondant de l'espace d'information. La dimensionalité de l'espace d'information correspond donc à la cardinalité de l'ensemble AM (**propriété 3** du **tableau 1.1**). Les attributs et leurs intervalles de valeurs représentent les dimensions de l'espace d'information. Ainsi chaque objet d'information peut être représenté sous la forme d'un point dans l'espace multidimensionnel d'information. Les relations entre les objets d'information, qui peuvent être explicites (hyperliens) ou implicites (basées sur le contenu et donc le sens des documents), sont prises en compte dans la structure de l'information (notée IS). Il s'agit d'une relation sur l'ensemble d'information IM (**propriété 4** du **tableau 1.1**). Ce formalisme peut être associé à la taxonomie. Ainsi, la partie droite de la taxonomie proposée consiste à visualiser tout ou une partie de l'en-

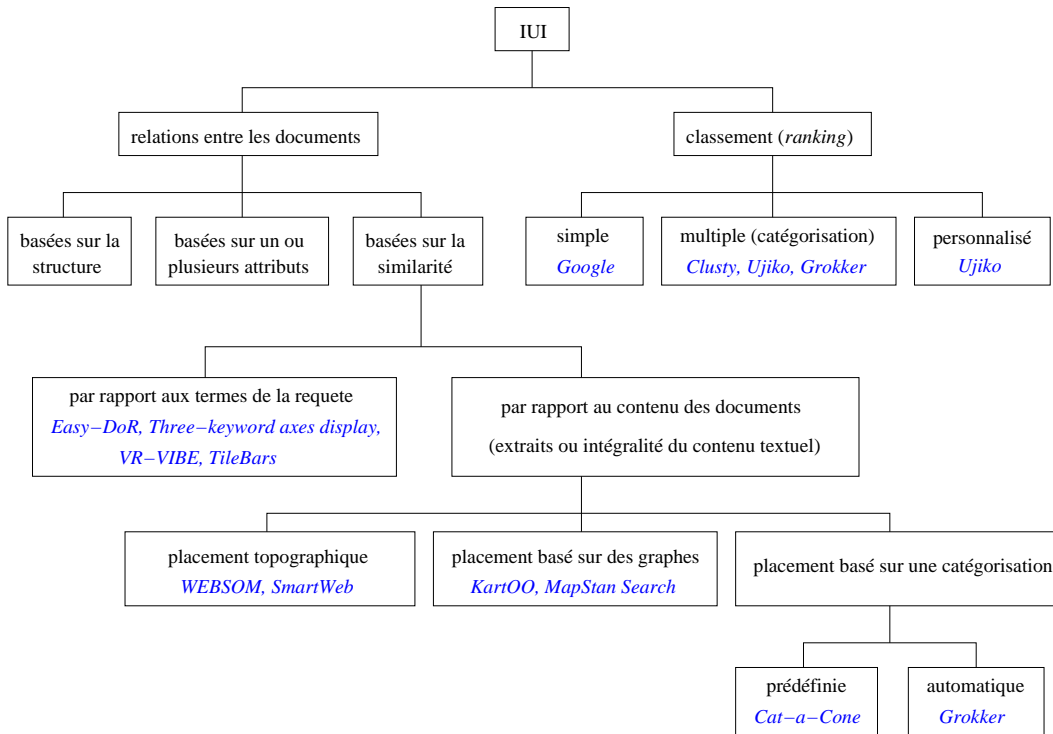


FIG. 1.2 – Taxonomie des différentes approches de visualisation des résultats d'une recherche

semble des attributs AM (et donc d'un sous-espace de IR), alors que la partie gauche se focalise sur la visualisation des relations, c'est-à-dire de tout ou d'une partie de l'ensemble IS . Cette dernière partie propose aussi de visualiser une partie de l'ensemble des attributs AM .

Propriété 1	$IM = IO_1 \dots IO_n$ $\forall i \forall j IO_i = IO_j \Leftrightarrow i = j \quad i, j, n \in \mathbb{N}$
Propriété 2	$AM = \text{attr}(IO_1 \dots IO_n) = A_1 \dots A_k$ $\forall i \forall j A_i = A_j \Leftrightarrow i = j \quad i, j, k, n \in \mathbb{N}$
Propriété 3	$\dim(IR) = AM $
Propriété 4	$IS \subseteq IM \times IM$

TAB. 1.1 – Propriétés du modèle d'information

1.2.1 Visualisation des attributs

Il s'agit de techniques ayant pour objectif de visualiser les valeurs d'un ensemble d'attributs (*i.e.* les propriétés qualitatives et quantitatives des documents). Elles permettent donc de visualiser la valeur de certains attributs mais aussi d'identifier une structure dans ces valeurs. Ainsi il est possible d'explorer la structure de l'information, relativement à certains attributs. Cependant, ces techniques sont plus souvent utilisées dans le domaine de la fouille visuelle de données ou de la visualisation scientifique. Les approches essentielles sont les techniques géométriques et symboliques.

1.2.1.1 Visualisations géométriques

Panel matrices. Cette technique consiste à fournir une configuration de représentations bivariées de l'espace multidimensionnel d'observation dans des matrices. Les matrices de graphes de dispersion (ou *Scatterplot matrices*) sont les exemples les plus connus [Cleveland 1993] (**figure A.1** de l'**annexe A**). Cependant, ce type de représentation n'est absolument pas adapté à un grand nombre de variables. Un autre inconvénient est le manque d'une représentation unifiée des données. D'autres exemples de *Panel Matrices* sont *Prosection Views* [Furnas et Buja 1994] et *HyperSlice* [van Wijk et van Liere 1993].

Coordonnées parallèles et en étoile. Ces deux méthodes permettent de représenter un espace de N-dimensions sur un plan en 2 dimensions. Les coordonnées parallèles [Inselberg 1998] utilisent des axes parallèles et équidistants (**figure A.2** de l'**annexe A**) alors que les coordonnées en étoile [Kandogan 2000] utilisent des axes en forme d'étoile (**figure A.3** de l'**annexe A**). Ainsi, l'espace nécessaire en coordonnées parallèles augmente linéairement avec le nombre de dimension (du fait de l'espace constant inter-axes), contrairement aux coordonnées en étoile ou cet espace est indépendant du nombre de dimensions. Cela permet donc d'avoir, en coordonnées en étoile, une visualisation moins encombrée dans le cas de grandes dimensions.

1.2.1.2 Visualisations symboliques

Il existe deux grandes familles de techniques permettant une visualisation symbolique des attributs : les techniques d'iconification et celles de pixelisation.

Techniques d'iconification. Leur objectif est de représenter les valeurs des attributs d'un document par des traits (ou caractéristiques) d'un icône pouvant être défini arbitrairement. Dans certains cas, ces icônes sont positionnés selon les deux premières composantes principales ou selon deux attributs particuliers. Parmi ces méthodes, quelques exemples sont les *Chernoff Faces* [Chernoff 1973] (**figure A.4** de l'**annexe A**), les *Stick figures* [Pickett et Grinstein 1988] (**figure A.5** de l'**annexe A**) et les *Color Icons* [Levkowitz 1991, Keim et Kriegel 1994] (**figure A.6** de l'**annexe A**).

Techniques de pixelisation. Elles ont pour principe de représenter chaque valeur d'attributs par un pixel coloré et de grouper les pixels appartenant à chaque dimension de l'espace initial dans des régions adjacentes. Cette technique permet donc la visualisation de la plus grande quantité de données possible sur les écrans standards, étant donné qu'elle utilise un pixel par valeur de données [Keim 2000]. Cependant cet affichage des pixels sur l'écran peut être arrangé différemment pour fournir des informations détaillées sur les corrélations, dépendances et points d'intérêts locaux. Le modèle récursif (**figure A.8** de l'**annexe A**) et les segments de cercle (**figure A.9** de l'**annexe A**) font partie des arrangements possibles.

1.2.1.3 Discussion

Ces méthodes permettent de visualiser certains attributs d'un ensemble de documents, mais elles ne semblent pas appropriées pour identifier rapidement les relations entre les documents ainsi que pour accéder aux documents. Une solution consiste alors à proposer un système combinant différentes de ces méthodes [Mann 1999]. Il s'agit d'une approche qui utilise des visualisations simples alternatives groupées autour de la traditionnelle liste de résultats (**figure 1.3**). Cependant l'utilisabilité d'un tel système pour restituer des résultats de recherches sur le Web nous semble discutable. En effet, multiplier les visualisations d'un même ensemble de documents risque d'être un facteur de dispersion de l'efficacité de l'utilisateur dans sa recherche. Ce risque peut être évité en proposant à l'utilisateur une seule interface sur laquelle il peut concentrer ses efforts, quitte à ce qu'elle soit évolutive. C'est pour ces raisons que nous favorisons les approches permettant de visualiser les relations entre documents, ce qui offre une visualisation générale avec des primitives visuelles de plus haut niveau.

1.2.2 Visualisation des résultats sous la forme de classement(s)

1.2.2.1 Classement simple

La visualisation des résultats sous la forme d'une simple liste ordonnée selon un critère de pertinence système reste encore la technique utilisée par la majorité des moteurs de recherche (et surtout par les plus utilisés). La liste de résultats fournie par *Google* en est le meilleur exemple (**figure 1.4**). Par défaut, cette liste présente 10 résultats par page et chaque résultat est décrit par : son titre suivi de son *snippet* et enfin de son URL. Cette technique de visualisation a l'avantage d'être rapide et simple pour l'utilisateur. Cependant elle possède aussi quelques inconvénients majeurs tels que :

- l'absence de relations entre les résultats. Seul un très léger effort de hiérarchie basée sur l'adresse URL a été réalisé. Sinon aucune relation sémantique entre les résultats n'est proposée.
- la désorganisation thématique des résultats. En effet, comme le montre la **figure 1.4**, les résultats appartenant à différentes thématiques sont mélangés dans

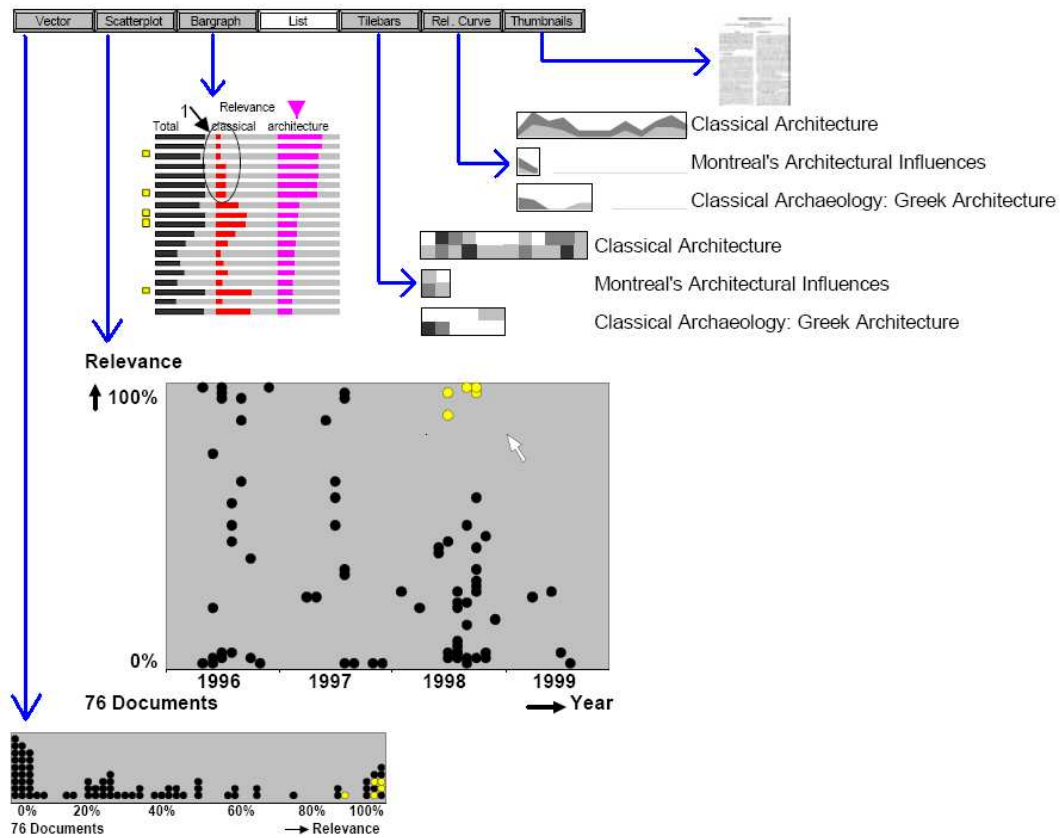


FIG. 1.3 – Approche utilisant plusieurs visualisations simples alternatives [Mann 1999]

la liste des résultats. Cela est lié au classement effectué de manière globale sur l'ensemble des résultats de la requête.

- l'absence de vue d'ensemble des résultats. Seulement une dizaine de résultats peuvent être affichés simultanément. Il n'est donc pas possible, pour l'utilisateur, d'avoir un aperçu de l'ensemble des réponses.

Ces listes de résultats, apparues au tout début de la RI sur le Web, sont donc amenées à évoluer. Ce processus est déjà timidement en cours avec l'apparition de listes évoluées (listes multiples et personnalisées).

1.2.2.2 Classement multiple

*Clusty*⁵ est le méta-moteur de recherche proposé par la société *Vivísimo*⁶. Il organise les 250 premiers résultats de recherche dans des répertoires thématiques basés sur les thèmes principaux dans le texte des résumés des résultats (figure 1.5). Cette

⁵www.clusty.com

⁶www.vivisimo.com

<p>Orange - [Translate this page] Orange (ex-Wanadoo) propose des offres Internet haut débit (ADSL, ADSL2+, READSL) et bas débit. Filiale du groupe France Télécom. www.orange.fr - 69k - 19 Sep 2006 - Cached - Similar pages Météo : Prévisions Météo France... - meteo.orange.fr/ loisirs - r.orange.fr/r/Winfos accueil - www.orange.fr/Une/ Téléphone portable et services... - r.orange.fr/r/Oaccueil_mobile More results from www.orange.fr »</p>		<p>■ opérateur ■ musique ■ ville / tourisme ■ fruit / cuisine</p>
(...)		
<p>Téléphone portable et services mobiles ORANGE - [Translate this page] Achetez votre téléphone portable sur le portail Orange et découvrez les services mobiles sur votre téléphone. mobile.orange.fr - 3k - Cached - Similar pages</p>		
<p>Portail Orange - [Translate this page] Vous éprouvez des difficultés pour vous connecter à Orange World ? ... Orange vous propose de configurer votre téléphone à distance, en suivant nos ... mobile.orange.fr/0/visiteur/PV?PS=CHMOBCONFIG - 13k - Cached - Similar pages</p>		
(...)		
<p>Orange - Ville d'Orange en Provence. Site officiel de la mairie. - [Translate this page] Le site officiel de la Mairie de la ville d'Orange en Provence - France. www.ville-orange.fr - 11k - Cached - Similar pages</p>		
<p>choregies d'orange 2006 - [Translate this page] Festival de musique lyrique dans le théâtre antique d'Orange. www.choregies.asso.fr - 4k - Cached - Similar pages</p>		
<p>Orange Caraïbe - [Translate this page] Tarifs, couvertures et actualité de l'offre de téléphonie portable d'Orange aux Antilles Françaises et en Guyane. www.orangecaraibe.com/ - 43k - Cached - Similar pages</p>		
<p>orange.com - [Translate this page] Wanadoo est désormais Orange. Visitez le site Internet de votre pays pour en ... Orange, Equant et France Télécom se sont regroupés pour former Orange ... www.orange.com/francais/default.asp - 9k - Cached - Similar pages</p>		
<p>Orange Accueil Tourisme - Provence Web - [Translate this page] Ville d'art et d'histoire, Orange brille par son patrimoine historique exceptionnel. Dès l'entrée de la ville, vous vous retrouverez au coeur de la ... www.provenceweb.fr/f/vaucluse/orange/orange.htm - 10k - Cached - Similar pages</p>		
(...)		
<p>ORANGE - office de tourisme - [Translate this page] ORANGE toutes les informations touristiques de l'office de tourisme pour préparer son voyage, son séjour à l'hotel, ses vacances. www.tourisme.fr/office-de-tourisme/orange.htm - 11k - Cached - Similar pages</p>		
<p>Projet Orange - [Translate this page] Projet Orange, groupe, musique. projetorange.com/ - 8k - Cached - Similar pages</p>		
<p>Orange - Wikipédia - [Translate this page] Cette ville était la capitale de la principauté d'Orange, Principauté qui donna son nom à la Maison d'Orange-Nassau, actuelle famille royale néerlandaise ... fr.wikipedia.org/wiki/Orange - 16k - Cached - Similar pages</p>		
<p>Orange - [Translate this page] Bienvenue sur la boutique Internet. internet haut débit. téléphonie. télévision. Connectez votre maison. à vos envies. Vous êtes déjà client ... abonnez-vous.orange.fr/ - 10k - Cached - Similar pages</p>		
(...)		
<p>ORANGE : tout sur l'orange, recettes, saison de l'orange sur L... - [Translate this page] ORANGE : tout sur l'orange, recettes, saison de l'orange sur L'Internaute. www.linternaute.com/femmes/cuisine/encyclopedie/fiche_composant/145/orange.shtml - 64k - 19 Sep 2006 - Cached - Similar pages</p>		

FIG. 1.4 – Un extrait de la liste de résultats proposée par le moteur de recherche *Google* en réponse à la requête « orange » (recherche réalisée sur le Web francophone)

organisation est obtenue grâce à une technique de *clustering* à la volée (développé par *Vivísimo*) qui permet de regrouper les résultats en se basant sur la similarité textuelle et linguistique. Ces répertoires sont générés spontanément avec des titres facilement identifiables et les résultats similaires sont groupés dans ces répertoires. *Clusty* range les répertoires à travers une combinaison des rangs des moteurs de recherche et d'algorithmes propres. Cette approche de clustering permet : d'avoir un aperçu rapide des principaux thèmes associés à la requête, de regrouper ensemble les résultats similaires pour un accès plus rapide, de trouver des résultats qui sont au fond de la liste de résultats (et qui seraient ratés par l'utilisateur sinon), de découvrir des résultats et des relations inattendus.

Bien que cette méthode ait l'avantage de proposer un *clustering* à la volée sur les premiers résultats, elle reste sur un affichage sous forme de listes ordonnées selon un critère de pertinence. On a une liste par répertoire obtenu. Par ailleurs, ces répertoires sont eux aussi classés de façon linéaire selon un critère système calculé par *Clusty*. Ils ne sont donc par organisés selon leur proximité sémantique.

The screenshot shows the Clusty search engine interface. At the top, there is a navigation bar with links for Web+, News, Images, Shopping, Wikipedia, Blogs, Jobs, and Customizet. A search bar contains the word 'orange' and a 'Cluster' button. Below the search bar, there is a 'Cluster by:' dropdown menu set to 'Topics'. The main content area displays a list of search results for 'orange', with a 'Cluster Fruit' containing 6 documents. The results are numbered 1 through 6, each with a title and a brief description. A sidebar on the left lists various topics and their associated document counts, such as 'Orange County' (158), 'Fruit' (6), and 'Network' (5). A 'Search Results' link is visible in the top right corner of the main content area.

FIG. 1.5 – Listes de résultats proposées par le moteur de recherche *Clusty* en réponse à la requête « orange ».

1.2.2.3 Classement personnalisé

*Ujiko*⁷ (développé par la société *KartOO*⁸) s'appuie sur l'index de *Yahoo* afin de proposer une technologie de recherche particulièrement intéressante. D'abord, l'interface *Ujiko* propose au centre de l'écran des thématiques sur lesquelles il suffit de cliquer pour préciser sa recherche (**figure 1.6**). Certains de ces thèmes sont colorés et associés à de petites briques de la même couleur, ce qui indique quels résultats sont associés au thème. En fait, *Ujiko* re trie les sites en tenant compte de leurs thèmes communs : au lieu de vous fournir les résultats en fonction d'un simple critère de pertinence, il ordonne les sites autour du cercle, thématique par thématique. Ensuite, *Ujiko* évolue avec l'expérience de l'utilisateur : de nouvelles fonctions sont proposées au fur et à mesure de l'utilisation de l'interface. Enfin *Ujiko* propose de personnaliser les résultats et prend en compte l'ergonomie. Ces deux points sont traités dans les deux paragraphes ci-dessous.

Personnalisation. Avec *Ujiko*, l'utilisateur peut décider de la pertinence des résultats. Pour cela, il suffit de cliquer sur les pages autour du cercle, ce qui permet entre autres de faire apparaître des boutons de personnalisation. Ainsi, il est possible de filtrer un résultat (il n'apparaîtra plus dans les résultats jusqu'à nouvel ordre), de mémoriser un résultat (il apparaîtra en tête des résultats si vous relancez la requête) ou de modifier un résultat (l'utilisateur peut changer la description, ajouter un commentaire, le ranger dans un dossier. . .). Il existe aussi d'autres fonctions qui apparaissent suivant le degré d'expérience de l'utilisateur, telles que : la création de filtres particuliers, le rangement de vos favoris dans des dossiers, la mise en évidence des pages si elles contiennent certains mots. Enfin, *Ujiko* propose une personnalisation étendue. En effet, lorsque l'utilisateur clique sur un résultat, il remonte automatiquement dans les premiers résultats avec tous les résultats qui ont des thèmes communs.

Ergonomie. Le design d'*Ujiko* a été pensé pour faciliter sa compréhension et sa manipulation. Il s'agit d'un objet en 3 dimensions qui évolue, se transforme, se complexifie lorsque vous avez acquis l'expérience suffisante. En plus, l'utilisateur peut personnaliser l'interface avec ses propres images.

Bien que de nombreuses fonctionnalités aient été proposées dans cette interface, on reste sur un affichage linéaire et limité des résultats. Il y a bien un classement thématique mais il est limité à un ensemble de 10 résultats. De plus, il n'est toujours pas possible d'avoir une vue d'ensemble des résultats. Mais il s'agit cependant d'une évolution très intéressante par rapport à la liste de résultats proposée par *Google*. Et c'est aussi une des rares interfaces à proposer une évolution selon le degré d'expérience de l'utilisateur, ce qui semble être une idée intéressante à exploiter [Shneiderman 2003a].

⁷www.ujiko.fr

⁸www.kartoo.net



FIG. 1.6 – Liste de résultats proposée par le moteur de recherche *Ujiko* en réponse à la requête « orange »

1.2.3 Visualisation des relations entre les résultats

Cette partie s'intéresse aux visualisations qui permettent de représenter les relations entre les différents objets d'information (*IO*), qui peuvent être des documents ou des classes de documents. Ces relations peuvent être explicites (spécifiées par la structure de l'information) ou implicites (obtention de nouvelles relations basées sur la similarité des objets d'information, par application d'algorithmes de calcul automatique tels que le *clustering* hiérarchique). Les techniques de visualisation respectant cette tâche mettent donc en avant les relations entre les objets d'information par rapport à leurs propriétés. Cet ensemble de techniques de visualisation se focalise sur la présentation de tous les aspects des documents, la plupart du temps pour supporter leur identification, leur analyse, ou pour trouver des relations avec les autres éléments. Dans ce contexte, les objets d'information sont souvent caractérisés par un grand nombre d'attributs dans un espace d'information multidimensionnel. La visualisation de tels objets multidimensionnels requiert une représentation sur des dimensions restreintes de l'espace de présentation. Il existe alors de nombreuses techniques pour réduire la dimensionalité de l'espace. Une approche courante de visualisation de l'ensemble des objets est alors l'approche cartographique ou topographique. Ces techniques utilisent généralement deux dimensions pour représenter la similarité thématique des documents ; la troisième dimension étant ensuite généralement utilisée pour représenter l'importance du thème dans une région donnée ou un attribut supplémentaire tel que la pertinence. Dans la suite, nous nous intéressons uniquement

aux visualisations des relations basées sur les similarités entre les documents ou avec les termes de la requête.

1.2.3.1 Visualisations basées sur la distribution des termes de la requête

Les techniques de visualisation présentées dans cette partie sont basées sur l'importance des termes de la requête dans les documents retournés par le système de recherche. L'idée générale consiste à associer un mot-clé de la requête (ou une combinaison de mots-clés) à un axe ou une position de l'espace ; puis de positionner les documents dans cet espace en se basant, le plus généralement, sur la fréquence d'apparition du mot-clé dans le document. Ces techniques permettent donc d'identifier les relations entre les résultats et les mots-clés de la requête. Par contre, elles ne permettent pas réellement de distinguer les relations entre documents. En effet, deux documents peuvent avoir un comportement similaire par rapport aux mots-clés de la requête et pourtant être différents. Nous présentons dans la suite quatre exemples de visualisations basées sur la distribution des termes de la requête.

TileBars. Il s'agit d'une représentation qui permet de visualiser la pertinence des documents par rapport à la requête (**figure 1.7(a)**) [Hearst 1995]. Chaque document est représenté par un ensemble de barres carrelées dont la longueur dépend de celle du document. Chaque barre correspond à une composante de la requête (généralement une catégorie demandée). La correspondance entre un segment du document et une catégorie demandée est représentée par une coloration du carreau correspondant dans la barre de la catégorie par une valeur appropriée sur l'échelle des niveaux de gris. Cette méthode permet d'identifier la pertinence par rapport à la requête des différentes parties d'un document. Ce concept a été repris plus récemment dans la métaphore *Webdocball* qui l'améliore en introduisant l'aspect hiérarchique présent dans la structure des documents [Vegas *et al.* 2003]. Par contre la requête est désormais considérée dans son ensemble et ne peut plus être décomposée en différentes parties.

VR-VIBE. Cette application propose une visualisation, dans un espace virtuel, des résultats issus d'un processus de RI [Benford *et al.* 1995]. L'objectif est de permettre à l'utilisateur de naviguer dans des représentations tridimensionnelles des documents et de communiquer avec d'autres utilisateurs réalisant la même tâche. L'idée générale de la visualisation proposée consiste à placer trois mots-clés de la requête dans un espace 3D ; puis les documents sont placés dans cet espace en fonction de leur pertinence à chacun des mots-clés. L'utilisation de la couleur permet de différencier les différents degrés de pertinence (**figure 1.7(c)**).

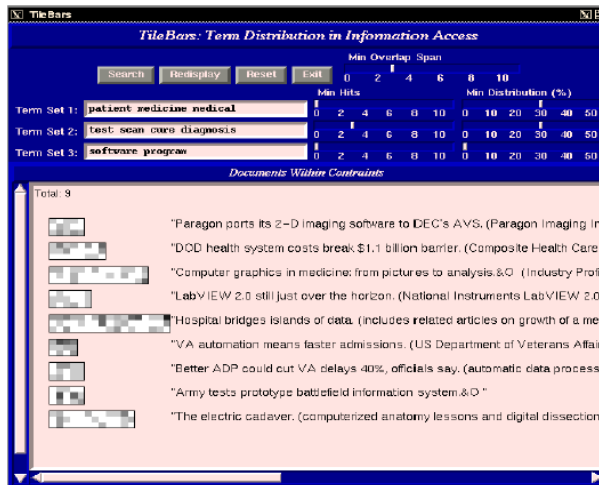
Three-keyword axes display. Cette interface est proposée comme une alternative à l'affichage linéaire du rang des documents [Cugini *et al.* 1996]. Il s'agit de représenter les documents par des icônes dans un espace 3D où chaque axe représente un

ensemble de mots-clés (**figure 1.7(d)**). L'avantage de cette représentation est de permettre à l'utilisateur de voir, de façon interactive, comment les termes de la requête sont liés aux documents.

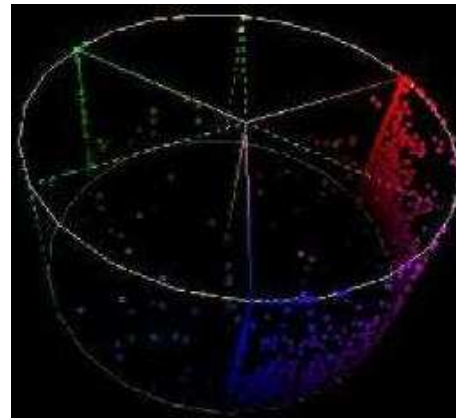
Easy-DoR. Il s'agit d'une interface intelligente et adaptative d'aide à la RI sur le Web [Chevalier 2002]. *Easy-DoR*⁹ repose sur une coopération entre les utilisateurs, à travers notamment leurs navigations sur le Web mais également de leur hiérarchie de signets. L'outil de recommandation pour la connaissance de l'utilisateur permet de faire évoluer la connaissance à long terme de ses centres d'intérêt grâce à de nouveaux documents pertinents. Ces centres d'intérêt sont déduits de sa hiérarchie de signets. Du fait de cette meilleure connaissance, l'utilisateur peut formuler de manière plus précise les requêtes et ainsi obtenir de meilleurs résultats. La recommandation durant la navigation repose sur une représentation des centres d'intérêts des utilisateurs sous la forme d'un multi-arbres afin d'obtenir une représentation unique de tous les centres d'intérêt des utilisateurs. Ce multi-arbres est exploité pour identifier des documents pertinents pour la navigation d'un utilisateur. L'interface de visualisation des résultats de recherche d'information permet d'appréhender les résultats de recherche de façon globale au travers d'une représentation des documents dans un espace 3D en utilisant les couleurs (**figure 1.7(b)**). La combinaison de deux axes d'interprétation (organisation spatiale et couleurs) permet à l'utilisateur une meilleure appréhension de l'importance des différents critères de recherche au sein des documents. L'outil de gestion de la hiérarchie de signets permet à un usager de s'affranchir du travail fastidieux qu'est le suivi des documents mémorisés ainsi que de la réorganisation des signets. Cet outil offre à l'utilisateur la possibilité de connaître les modifications apportées aux documents mémorisés mais également une aide pour organiser ces documents.

Ces approches permettent donc de visualiser les relations entre les différents documents et les termes de la requête. Cependant, aucune information fiable sur les relations entre les documents n'apparaît. En effet, deux documents peuvent réagir de façon similaire aux termes de la requête (et donc être voisins sur ce type de représentation) sans pour autant contenir des informations similaires. Ces techniques permettent uniquement de visualiser la distribution des termes de la requête au sein des différents résultats de la requête. De plus, elles conservent un affichage linéaire restreint (*TileBars*), ou alors elles proposent une vue d'ensemble qui complique la navigation et l'identification d'un document particulier (ainsi que l'accès à ce document). Une approche similaire est *DocCube* qui propose une visualisation 3D ne se basant plus sur les termes de la requête mais sur des hiérarchies de concepts [Mothe *et al.* 2003].

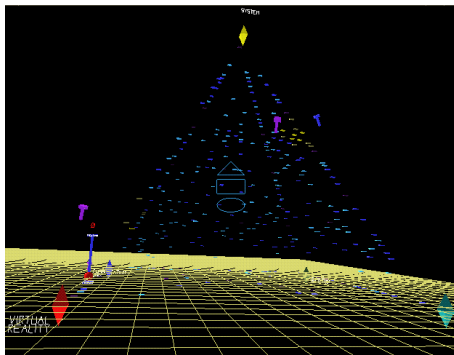
⁹pour *Easy Document Retrieval*



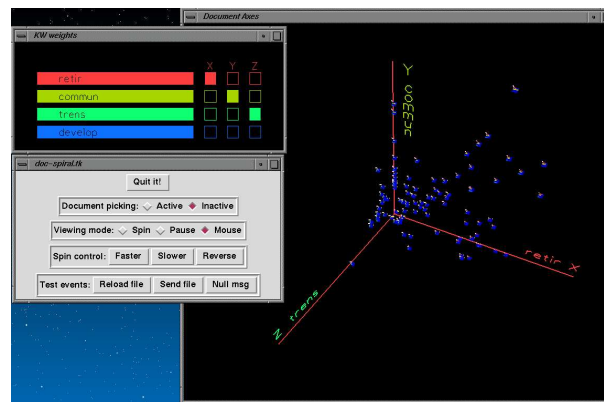
(a) Exemple d'utilisation de *TileBars* pour la visualisation de résultats de recherche dans une base de donnée médicale [Hearst 1995]



(b) *Easy-DoR*



(c) *VR-VIBE*



(d) *Three-keyword axes display*

FIG. 1.7 – Exemples de visualisations basées sur la distribution des termes de la requête

1.2.3.2 Approches cartographiques basées sur la notion de graphes

KartOO. Le méta-moteur de recherche *KartOO*¹⁰ représente les résultats de recherche sous la forme d'une carte. Une fois la requête effectuée, *KartOO* l'analyse, interroge les moteurs de recherche les plus pertinents, sélectionne les sites concernés et les affiche sous forme d'une carte. Dans cette carte, les sites sont représentés par des pages plus ou moins grosses selon leur pertinence. Le survol d'une page (*i.e.* un site) fait alors apparaître une brève description du site et les mots-clés concernés s'illuminent sur la carte (**figure 1.8**). Il est également possible d'affiner la requête en y ajoutant ou soustrayant des thèmes proposés. *KartOO* propose aussi de nombreuses autres options et fonctionnalités telles que : la possibilité d'utiliser le langage naturel, la syntaxe de la requête, le nombre de résultats par carte, l'enregistrement d'une carte, l'affichage du type des pages ou encore du nombre de visites effectuées. Cependant un inconvénient de ce méta-moteur cartographique est l'affichage des résultats sous forme d'une succession de cartes sans réel lien entre elles. Ainsi certains sens d'une requête ne peuvent apparaître qu'après la consultation de quelques cartes.



FIG. 1.8 – *KartOO*, méta-moteur de recherche cartographique en 2D

MapStan Search. Le méta-moteur de recherche *MapStan Search*¹¹ représente, sous forme cartographique, les résultats issus du moteur de recherche *Google*. Ce nouvel outil de recherche cartographe est donc capable de tracer une carte de n'importe quelle

¹⁰<http://www.kartoo.com>

¹¹<http://search.social-computing.com/>

région du Web, à partir de n'importe quel mot-clé. Il propose ainsi une synthèse visuelle des résultats et des proximités entre les pages. Ces résultats sont archivés avec leurs recherches afin d'être suggérés ultérieurement. Grâce à un processus d'analyse unique, *MapStan Search* représente les résultats d'une recherche en utilisant sa connaissance de centaines de recherches similaires. Il utilise le WPS (*Web Positioning System*)¹², première technologie de cartographie d'information personnalisée mise au point, pour analyser et synthétiser les sites issus de la recherche sous la forme d'un plan de quartier composé de places et de rues (**figure 1.9**). L'efficacité de la représentation est aussi améliorée à l'aide de couleurs. Les « meilleurs » sites retournés par le moteur (issus des 100 premières pages de résultats) forment alors les places rouges. L'intensité lumineuse de ces dernières reflète le classement de leur site principal. Ainsi, les places dont le site principal est dans les mieux classés sont jaune pâle. À l'inverse, les places avec un site mal classé sont rouge foncé et elles sont orange pour les résultats intermédiaires. Ces résultats sont complétées avec des places bleues composées de sites suggérés par *MapStan Search*. Ces réponses suggérées ont un rapport étroit avec la requête et sont les plus pertinentes au regard des recherches précédemment capitalisées. Les sites des pages retournées par le moteur de recherche sont placés au centre du plan en fonction de leurs liens. Puis, de manière similaire, les sites suggérés sont placés autour en respectant les relations les unissant entre eux et les sites initiaux. Les requêtes communes à plusieurs sites forment des rues entre les places. Les rues représentent donc les liens entre les places (*i.e.* leur similarité) et leur survol permet de visualiser dans une infobulle les principales requêtes responsables du lien entre les deux places. Un menu contextuel (obtenu en cliquant sur une rue) permet d'effectuer une de ces requêtes. La largeur d'une rue exprime le nombre de requêtes partagées entre les sites des deux places, et la longueur d'une rue est liée à la proximité de ces mêmes places. Cette proximité est importante lorsque les recherches retournent fréquemment les sites des deux places à la fois. Lorsque deux places sont systématiquement ensemble dans les recherches, elles sont si proches qu'elles fusionnent. C'est ainsi que certaines places regroupent un ou plusieurs sites proches. À chaque site correspond une petite boule. Le nom du site principal (le plus représentatif) de la place est affiché partiellement et éventuellement accompagné de son icône. Si le site principal est un résultat de la requête, alors son classement apparaît à la place de sa boule. La taille d'une place traduit, pour une recherche donnée, la pertinence de son site principal. Le survol d'une place permet d'afficher les noms complets de ses sites, et le survol d'un site permet d'afficher dans une infobulle son titre et son résumé. En cliquant sur un site, un menu contextuel apparaît et permet d'ouvrir : les pages du site retournées par le moteur de recherche pour une place rouge (classement suivi du nom) ou les pages suggérées par *MapStan Search* pour une place bleue, la page d'accueil du site et un plan d'exploration du site. Dans un plan d'exploration, *MapStan Search* dispose les sites en fonction de leur similarité avec le site « exploré » (place rouge centrale). Pour

¹²La technologie WPS est l'aboutissement d'un programme de R&D labellisé ANVAR, destiné à développer la première technologie de navigation personnalisée. Elle utilise des algorithmes originaux de filtrage collaboratif, d'analyse de corrélation et de cartographie d'informations, dont les principes ont fait l'objet d'un dépôt de brevet en décembre 2000.

cela, la proximité entre les sites sur l'ensemble des recherches capitalisées est analysée. Il s'agit en quelque sorte d'un catalogue dynamique des sites issus de recherches déjà réalisées. On dispose ainsi d'une classification sans *a priori* qui permet de découvrir rapidement de nouveaux sites et de trouver des pages inédites. *MapStan Search* s'appuie donc sur sa propre base de connaissance pour enrichir le résultat de la recherche avec les pages les plus pertinentes des recherches similaires. On bénéficie alors du résultat de centaines de recherches en une seule requête, ce qui permet d'éviter la multiplication des requêtes. Un des inconvénients de ce moteur est toutefois le temps de calcul de la carte qui va jusqu'à 20 secondes. Sinon, *MapStan Search* est basé sur une approche générique de l'information et peut donc être appliqué à de nombreux domaines.

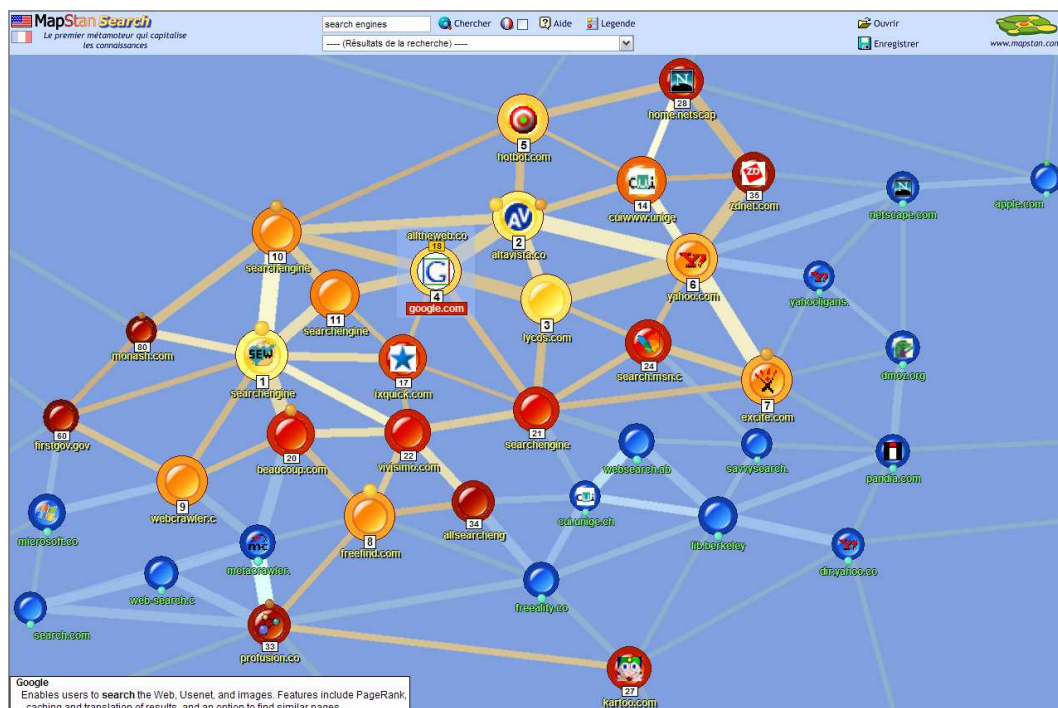


FIG. 1.9 – *MapStan Search*, méta-moteur de recherche cartographique en 2D

1.2.3.3 Visualisations topographiques respectant la notion de proximité sémantique

La majorité de ces visualisations sont basées sur des cartes (pour les visualisations 2D) ou alors sur des univers (pour les visualisations 3D). Quelques exemples représentatifs sont donnés dans les paragraphes suivants.

WEBSOM. Il s'agit d'une méthode permettant d'organiser une collection de documents textuels sur des cartes significatives pour l'exploration et la recherche (**figure 1.10(a)**) [Kohonen *et al.* 2000]. Cette méthode WEBSOM¹³ a été réalisée sur une base de données composée de 6 840 568 résumés de brevets écrits en anglais. La carte utilisée pour la visualisation de ces données est composée de 1 002 240 neurones. Cependant, l'ensemble du processus de calcul de la carte des documents prend environ 6 semaines sur un ordinateur SGI O2000 6 processeurs, et la quantité de mémoire principale requise est environ de 800 MB.

CityOfNews. Il s'agit d'un navigateur Web, 3D immersif, interactif et grandissant dynamiquement [Sparacino *et al.* 2000, 2002]. Ce navigateur est basé sur l'utilisation de la mémoire spatiale de l'utilisateur. Il navigateur cherche et affiche des URL, formant des gratte-ciels et des ruelles de textes et d'images, que les participants peuvent visiter comme s'ils exploraient un paysage urbain d'informations (**figure 1.10(b)**). Le système démarre avec un cadastre d'une ville ou région choisie, et, tandis que le participant suit les chemins, il construit dynamiquement un monde Web 3D virtuel d'informations. Il crée des associations mentales entre l'information et la géographie comme si les gens vivaient dans un Palais de Mémoire personnalisé [Yates 1966].

ThemeView. L'objectif est de donner un aperçu des sujets principaux contenus dans un ensemble de documents. Il s'agit d'une métaphore permettant de visualiser les thèmes d'un ensemble de documents sous la forme d'une carte en relief d'un terrain naturel (**figure 1.10(c)**). Les thèmes similaires sont proches sur la carte, contrairement aux thèmes sans relations qui sont séparés par des distances plus grandes. Les montagnes indiquent les thèmes dominants. Il existe d'autres métaphores de visualisation développées par le *Pacific Northwest National Laboratory*¹⁴.

VxInsight. Cet outil, développé au sein des *Sandia National Laboratories*, permet de découvrir des relations dans de grandes bases de données [Davidson *et al.* 2001, Boyack *et al.* 2002a,b]. Il est alors capable de révéler la structure implicite des données. Il utilise la métaphore du terrain (**figure 1.10(d)**) pour fournir un environnement d'exploration intuitif. Il propose aussi différents niveaux de résolution.

ViOS. L'application ViOS¹⁵ propose une visualisation du Web sous la forme d'un monde 3D en ligne dans lequel l'utilisateur peut naviguer en temps réel (**figure 1.10(e)**). La métaphore de visualisation employée dans ViOS utilise des concepts géographiques (univers de type terrain dans lequel l'utilisateur peut se déplacer) sans pour autant appartenir au monde réel (contrairement notamment à la métaphore de la ville).

¹³*Self-Organizing Maps for Internet Exploration* — <http://websom.hut.fi>

¹⁴<http://infoviz.pnl.gov/technologies.html>

¹⁵*Visual Internet Operating System* — <http://computer.howstuffworks.com/vios.htm>

The Data Mountain. Il s'agit d'une technique pour la gestion de documents, qui permet aux utilisateurs de placer les documents à des positions arbitraires sur un plan incliné dans un environnement virtuel de bureau 3D, en utilisant une simple technique d'interaction 2D (**figure 1.10(f)**) [Robertson *et al.* 1998]. La conception de cette technique a évolué en réponse aux retours utilisateurs et une étude utilisateur montre que *Data Mountain* tire avantage de la mémoire spatiale. Il est également montré que *Data Mountain* a statistiquement des avantages fiables sur le mécanisme de favoris de *Microsoft Internet Explorer* pour gérer des documents d'intérêt dans un espace de travail d'informations.

Cette énumération de systèmes de visualisation cartographique de l'information n'est bien entendu pas exhaustive. Il existe d'autres approches (e.g. les méthodes *Map.net* et *Webmap* dont un aperçu est donné sur le site Web « *An Atlas Of Cyberspaces* » de Martin Dodge¹⁶), ainsi que d'autres études sur l'utilisation des métaphores topographiques dans le domaine de l'IV [Chalmers 1993]. André Skupin a également mené des travaux intéressants sur la cartographie de l'information [Skupin 2000, 2002]. Il est aussi possible d'avoir une interface respectant les proximités sémantiques entre les documents, sans pour autant utiliser une approche cartographique. Un exemple est le modèle « *Spoke and Wheel* » proposé dans le prototype *NIRVE*¹⁷ [Cugini *et al.* 2000]. L'objectif de ce prototype est de présenter à l'utilisateur un aperçu de la structure de l'ensemble des résultats plutôt que de se concentrer sur la recherche d'un document particulier. Plusieurs types de métaphores apparaissent dans ce prototype.

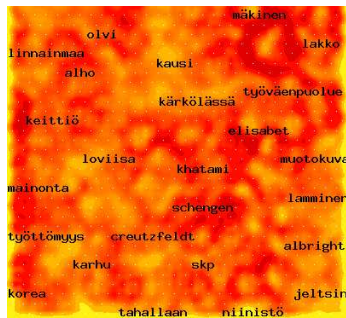
1.2.3.4 Organisation basée sur une catégorisation hiérarchique des résultats

L'organisation des résultats, basée sur une catégorisation hiérarchique, implique de s'intéresser aux méthodes de visualisation hiérarchique. Ces méthodes sont généralement plus largement utilisées pour visualiser la structure de l'information. Quelques approches classiques sont rappelées dans le paragraphe suivant. Ensuite nous revenons aux techniques proposées pour la visualisation hiérarchique de catégories de résultats de recherche.

La visualisation de la structure de l'information (*IS*) peut se faire par des visualisations hiérarchiques ou non hiérarchiques. Concernant les visualisations non hiérarchiques, il s'agit essentiellement de visualisations sous la forme de graphes. Cependant, l'utilisation des graphes pose certains problèmes tels que les difficultés d'affichage (croisement des arcs) ou les difficultés d'interprétation lorsque la taille du graphe augmente. Ces problématiques, traitées par la communauté du *Graph Drawing* ne sont pas abordées ici. Concernant les visualisations hiérarchiques, de nombreuses méthodes ont déjà été proposées, aussi bien en 2D qu'en 3D. Parmi les techniques les

¹⁶<http://www.cybergeography.org/atlas/atlas.html> — traduit en français et adapté par Nicolas Guillard, <http://www.cybergeography-fr.org/atlas/atlas.html>

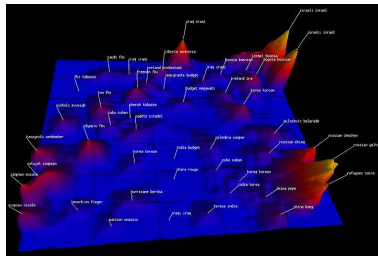
¹⁷*NIST Information Retrieval Visualization Engine* — <http://zing.ncsl.nist.gov/~cugini/uicd/nirve-home.html>



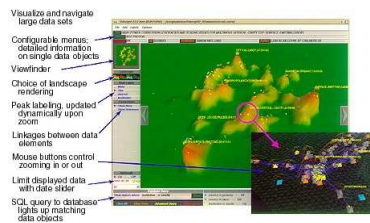
(a) *WEBSOM* : visualisation 2D d'une carte auto-organisatrice



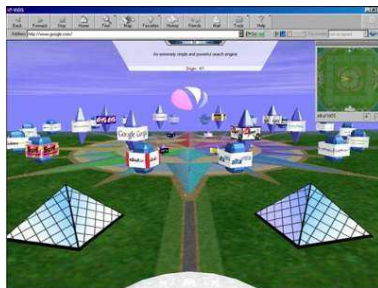
(b) *CityOfNews* : visualisation de pages Web sous la forme d'une ville virtuelle



(c) *ThemeView*



(d) *VxInsight*



(e) *ViOS* : le centre de « l'univers *ViOS* »



(f) *Data Mountain* : affichage de la page sélectionnée en avant plan et de 100 pages placées par l'utilisateur en arrière plan

FIG. 1.10 – Métaphores sous forme de terrain

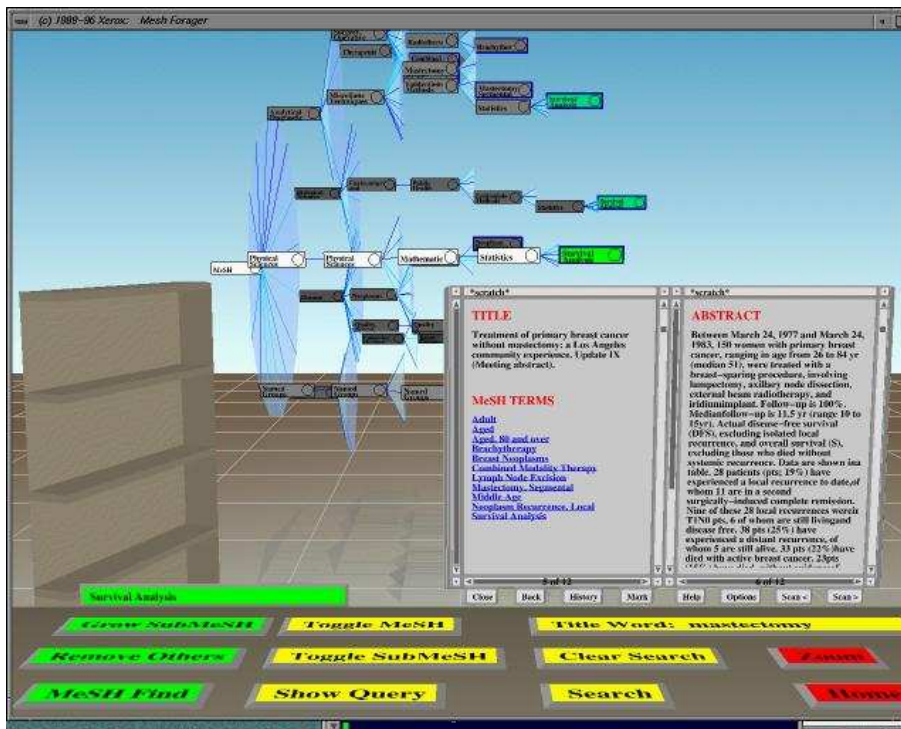
plus courantes, se trouvent les techniques de remplissage de l'espace, telles que les approches *Treemap* [Shneiderman 1992] et *Sunburst* [Stasko et Zhang 2000]. La première est basée sur un tracé rectangulaire (cf. **figure A.10** de l'**annexe A** pour un exemple en 2D et cf. **figure A.15** de l'**annexe A** pour une version 3D de cette technique) alors que la seconde est basée sur un tracé radial (cf. **figure A.12** de l'**annexe A**). Une application classique de ces techniques est la représentation de systèmes de fichiers. Une autre utilisation de la technique *Treemap* est l'application *Newsmap* qui offre une représentation visuelle, sous la forme d'une carte, des actualités (**figure A.11** de l'**annexe A**). Deux autres exemples de visualisation hiérarchique, mais cette fois-ci en 3D, sont les approches *Cone Tree* [Roberston et al. 1991] (**figure A.13** de l'**annexe A**) et *InfoCube* [Rekimoto et Green 1993] (**figure A.14** de l'**annexe A**). *Cone Tree* est l'une des techniques les plus connues dans le domaine de l'IV pour afficher des arbres en 3D. Le second exemple, le cube d'information (*InfoCube*), est une généralisation de l'approche 2D utilisant des boîtes emboîtées. Cependant, cette méthode montre que l'affichage 3D de graphes introduit de nouveaux problèmes comme les occultations ou le choix du point de vue.

Dans le cas de la restitution de résultats de recherche, la visualisation hiérarchique de catégories de documents peut se faire principalement de deux manières. La première consiste à visualiser une hiérarchie de concepts prédéfinie dans laquelle sont classés les documents. Cette solution est cependant orientée vers les annuaires, et ne concerne pas les moteurs par requêtes qui ne disposent pas d'une telle hiérarchie. La deuxième possibilité est de visualiser une hiérarchie basée sur les similarités entre les documents. De telles hiérarchies sont généralement issues de l'application d'algorithmes de *clustering* hiérarchique.

Catégories prédéfinies. *Cat-a-Cone* est un système de restitution de résultats de recherche couplant l'approche *Cone Tree* avec une métaphore de bibliothèque (**figure 1.11**). L'arbre conique permet d'afficher une grande hiérarchie de catégories prédéfinie, et la métaphore de la bibliothèque est basée sur une modification de l'approche *WebBook* du projet *Web Forager* [Card et al. 1996]. Les documents retrouvés sont alors organisés sous la forme de pages d'un livre. Les deux parties de l'interface (*Cone Tree* et *Web-Book*) communiquent entre elles, étant donné que des interactions sur l'une entraînent des modifications de l'autre. L'utilisateur peut aussi stocker un livre (*i.e. des résultats de recherche*) et le réouvrir à n'importe quel moment afin de réutiliser les résultats de recherche.

Catégories obtenues automatiquement. Un exemple de visualisation des résultats basée sur une catégorisation automatique est *Grokker*¹⁸ qui présente les résultats dans des catégories afin d'éliminer les listes désorganisées de résultats qui rendent la recherche frustrante (**figure 1.12**).

¹⁸*Grokker, A New Way to Look at Search* : <http://www.grokker.com>

FIG. 1.11 – *Cat-a-Cone*

1.2.4 Nombre de dimensions

1.2.4.1 Visualisations 3D

Cette section donne un aperçu des différentes métaphores 3D existantes pouvant être utilisées pour visualiser les résultats d'une recherche. Certaines de ces métaphores peuvent initialement avoir été proposées dans un but similaire tel que la visualisation du Web ou de grands ensembles de documents. Nous proposons alors une taxonomie des métaphores 3D de visualisation des résultats de recherche [Bonnell *et al.* 2005c]. Cette classification (voir **figure 1.13**) se veut assez généraliste et elle est uniquement basée sur les caractéristiques visuelles des métaphores.

Par abus de langage, les approches cartographiques représentent ici les méthodes utilisant une organisation des résultats sur un plan 2D. Il existe déjà de nombreuses techniques de visualisation 3D des résultats de recherche, utilisant des métaphores diverses et variées, fondées sur le monde réel ou sur l'imaginaire¹⁹ [Benford *et al.* 1999, Wiza *et al.* 2004]. Quelques exemples illustrant les différentes catégories de cette taxo-

¹⁹Martin Dodge, *Cyber Geography Research* — <http://www.cybergeography.org/>, et le *Pacific Northwest National Laboratory* — <http://infoviz.pnl.gov/technologies.html>

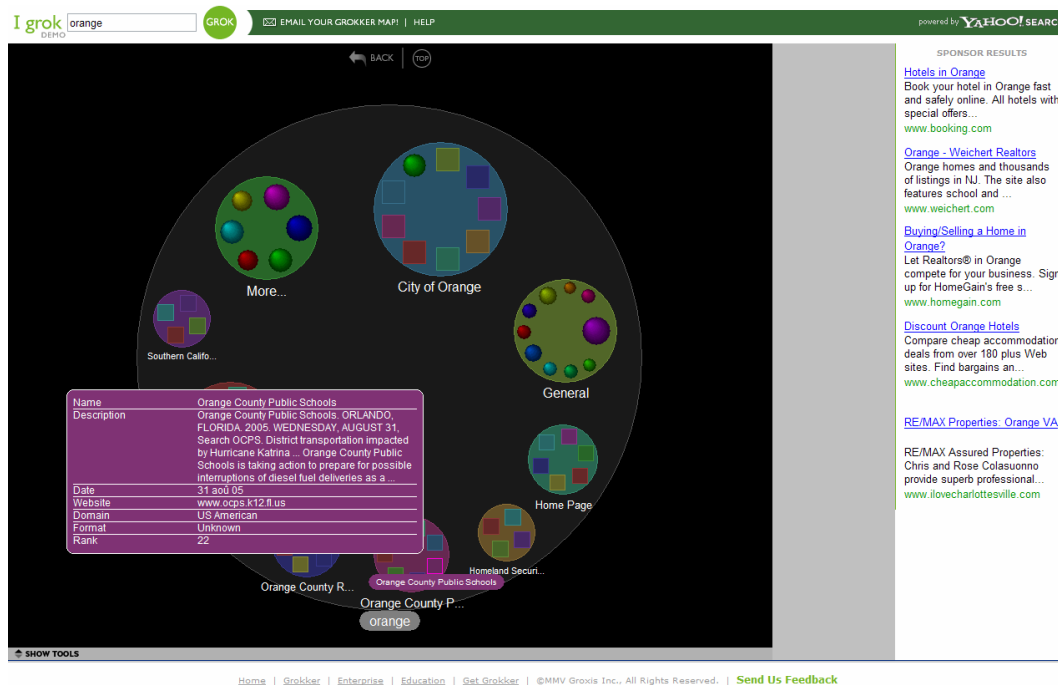


FIG. 1.12 – Grokker

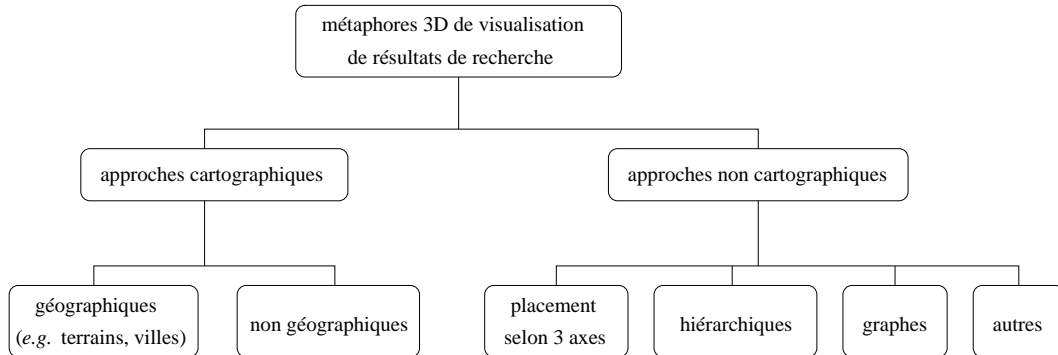


FIG. 1.13 – Taxonomie des métaphores 3D de visualisation pouvant être utilisées par les IUI

nomie ont déjà été présentés précédemment.

Cependant le choix de visualisations 3D n'est plus uniquement guidé par le souhait d'augmenter l'espace de représentation, mais aussi par la volonté de simplifier l'interface Homme-Machine. Ainsi, de nombreuses perspectives s'ouvrent à ce type d'interface : l'utilisation des connaissances de l'utilisateur par des métaphores cognitives, la proposition d'une spatialisation du Web dans lequel l'utilisateur peut se bala-

der (immersion), la proposition d'une mutualisation des recherches par une interface collaborative, ... Cependant, si les performances de la 3D ne sont plus une préoccupation majeure, le problème de l'interaction avec les métaphores 3D reste un défi à relever pour le succès de ces approches.

Les utilisateurs se représentent le Web comme un espace virtuel (en témoigne le vocabulaire employé : *adresse Web, site Web, home page, chat room...*) et pourtant la plupart des moteurs de recherche utilisent des métaphores de livre ou de page (issues du domaine de l'information) pour représenter les résultats (*i.e.* le Web). Il y a donc un décalage entre la façon dont les utilisateurs imaginent le Web et les moyens fournis pour y accéder [Sparacino *et al.* 2002]. Ce constat joue en faveur de nouvelles interfaces géographiques, notamment en 3D. Cependant, pour les utilisateurs, la visualisation 3D reste encore essentiellement liée aux jeux vidéo. En effet, bien que de nombreuses interfaces 3D aient déjà été proposées dans le domaine de la recherche d'information, les applications grand public utilisant ce type d'interface sont très rares ; ce qui explique par ailleurs le manque d'expérience des utilisateurs dans ce domaine. Mais l'émergence de la 3D dans certaines applications incontournables, telles que notre bureau (voir **figure 1.14**), risque de modifier les comportements. En effet, les futurs environnements de bureau annoncés par *Sun Microsystems* (voir **figure 1.14(a)**) ou *Microsoft* (la librairie graphique *Avalon* de son prochain système d'exploitation baptisé *Vista* devrait être dotée de fonctions d'affichage 3D — voir **figure 1.14(d)**), vont probablement modifier les habitudes des utilisateurs face à la 3D. Par ailleurs, des exemples de multifenêtrage 3D avec la version XP de *Windows* sont déjà proposés par des applications telles que *SphereXP*²⁰ (voir **figure 1.14(c)**). D'autres logiciels, tels que *3DNA*, offrent un bureau virtuel 3D dans lequel l'utilisateur peut naviguer librement, et qui sert à la fois d'espace de recherche (local ou sur le Web), de messagerie, de jeu ou encore d'espace multimédia (voir **figure 1.14(b)**). Un autre exemple d'interface utilisateur en 3D ayant pour objectif d'étendre le bureau traditionnel est le projet *The TaskGallery*²¹ (voir **figure 1.14(e)**) mené il y a quelques années par *Microsoft* [Robertson *et al.* 2000]. L'idée est de proposer un nombre illimité de bureaux. Pour cela, l'écran devient une longue galerie avec des tableaux sur les murs qui représentent les différentes tâches. L'utilisateur se déplace rapidement et facilement d'une tâche à l'autre avec une simple série de commandes souris et clavier. Aucun apprentissage particulier n'est nécessaire. En effet, moins les utilisateurs ont besoin de penser à la manière dont leur ordinateur fonctionne, plus ils gardent d'énergie mentale pour leur travail réel. Toutes les applications *Windows* fonctionnent, sans aucune modification, dans cet environnement 3D. Les icônes sont remplacés par des copies d'écran (de fichiers ou d'applications) et l'utilisateur peut personnaliser les affichages des tâches, ce qui permet de tirer profit de la mémoire spatiale de l'utilisateur. De plus, des contrôles de navigation haut niveau évitent à l'utilisateur de se perdre dans l'espace 3D. D'après de premières études utilisateurs, il semble que les utilisateurs adhèrent à ce type d'inter-

²⁰<http://www.spheresite.com/spherexp.html>

²¹<http://research.microsoft.com/ui/TaskGallery/index.htm>

face. Mais des études plus poussées doivent être menées afin d'évaluer les avantages de ce type d'environnements 3D sur l'exécution des tâches. En attendant, le succès de métaphores 3D pour la visualisation de notre bureau pourrait alors signifier le début des interfaces 3D grand public ; à moins que les navigateurs Web 3D ne soient les premiers à s'imposer, à l'image de l'application *SphereXplorer*²² (figure 1.15).



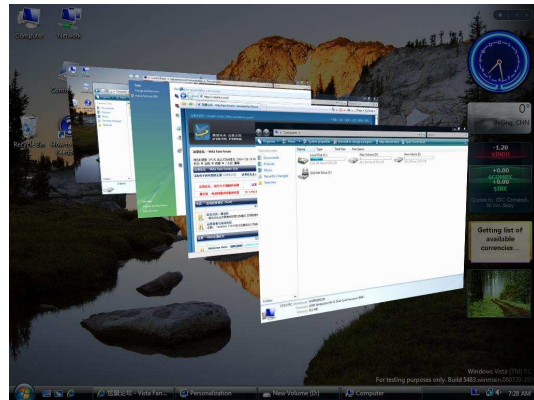
(a) *Looking Glass* : exemple de multifenêtrage du projet de développement *Open Source* d'environnement graphique 3D proposé par *Sun Microsystems*



(b) *3DNA* : logiciel offrant différents mondes 3D qu'il est possible d'explorer et de personnaliser afin d'avoir un bureau immersif



(c) *SphereXP* : application permettant d'étendre de façon naturelle la métaphore classique du bureau et ainsi d'avoir un bureau 3D qui se substitue à celui de *Windows XP*



(d) *Microsoft Vista* : nouvelles possibilités d'affichage proposées dans la nouvelle version, appelée *Vista*, du système d'exploitation *Windows* de *Microsoft*

²²<http://www.spheresite.com/spherexplorer.html>



(e) *The TaskGallery* : prototype de recherche utilisant la 3D et proposant une interface utilisateur qui étend le bureau en une galerie avec un nombre illimité de bureaux

FIG. 1.14 – Bureaux virtuels en 3D



FIG. 1.15 – *SphereXplorer* : navigateur Web 3D basé sur la technologie *SphereXP*

1.2.4.2 2D Vs 3D

Cette section sur la restitution des résultats de recherche ne peut pas se faire sans traiter de « l'éternelle » question : doit-on favoriser les visualisations 2D ou 3D ? Bien entendu cette comparaison peut être étendue aux visualisations textuelles qui sont encore le mode de visualisation privilégié par les moteurs de recherche les plus utilisés. L'objectif n'est pas ici de vous vendre la solution miracle, mais simplement d'apporter

notre point de vue sur la question, tout en prenant en compte les précédentes études réalisées dans le domaine de la recherche d'information. Cependant, on rappelle qu'il est difficile de réaliser des études utilisateur équitables et comparables dans ce domaine, et que le choix du nombre de dimensions est amené à évoluer en fonction des applications, des périphériques d'interaction, des tâches à accomplir et des compétences de l'utilisateur. C'est pourquoi un avis trop figé sur la question nous semble actuellement risqué.

John Cugini a proposé des solutions utilisant la technologie 3D afin d'aider les utilisateurs à comprendre et manipuler les résultats de recherche [Cugini *et al.* 2000]. C'est donc naturellement qu'il a été amené à aborder la question de l'utilité de telles visualisations par rapport aux visualisations textuelles ou 2D. Il tire alors les conclusions suivantes. Si on se base sur l'achèvement de la tâche ou le temps de réponse, les utilisateurs sont plus performants avec une visualisation textuelle qu'avec une visualisation 2D, et avec un modèle 2D par rapport à un modèle 3D. Ce constat est d'autant plus vrai quand il s'agit de tâches très spécifiques telles que la recherche d'un document précis. Cependant, cette comparaison n'est pas vraiment valide étant donné que l'on compare parfois des interfaces 2D (ou 3D) proposant des regroupements de résultats avec des solutions textuelles sans réelle notion de groupes de documents (*e.g.* l'interface actuelle de *Google*). Par ailleurs, après un certain temps d'utilisation, l'écart entre les solutions textuelles et 3D se réduit largement. Si les experts et les novices sont familiers avec les interfaces textuelles, les novices semblent éprouver certaines difficultés à s'adapter aux interfaces graphiques. Un certain temps de pratique semble nécessaire avant que les métaphores spatiales deviennent suffisamment familières au point d'être utilisées sans perte de temps. John Cugini évoque aussi le problème de la lisibilité, particulièrement plus complexe dans les affichages 2D ou 3D comparé aux listes textuelles. Bien entendu, dans le cas d'un faible nombre de résultats, il n'y a probablement pas grand chose à gagner à structurer les réponses dans un espace 2D. Cependant, plus le nombre de réponses prises en compte est élevé, plus le profit des visualisations 2D va apparaître. Enfin l'utilisation de visualisations 3D par rapport aux visualisations 2D pose certains problèmes tels que les occultations ou la désorientation. En effet, les utilisateurs ont des difficultés à retrouver les éléments dans un espace 3D. De plus le modèle 3D est moins familier à manipuler étant donné qu'il est possible d'effectuer des translations et des rotations par rapport aux trois axes. Il faut alors veiller à ne pas utiliser la 3D lorsque cela n'est pas absolument nécessaire, en témoigne le modèle de globe (*Globe Model*) proposé dans le prototype *NIRVE* [Cugini *et al.* 2000].

D'autres travaux se sont aussi intéressés à la comparaison des interfaces 2D et 3D dans le domaine des systèmes de gestion de documents. Par exemple, une étude propose de comparer deux versions — une en 2D et l'autre en 3D — de l'interface *The Data Mountain* présentée précédemment [Cockburn et McKenzie 2001]. Les résultats de cette étude montrent alors que l'utilisateur va plus vite à stocker, organiser et retrouver des pages Web quand il utilise l'interface 2D. Mais cette différence n'est pas

significative. Malgré ce manque de différences significatives entre les interfaces 2D et 3D, des évaluations subjectives montrent une préférence significative pour l'interface 3D. Cette étude a été poursuivie dans le but de contrôler les effets liés à l'utilisation des affichages informatiques [Cockburn et McKenzie 2002]. Des tâches similaires ont alors été réalisées avec des modèles 2D, 2^{1/2}D²³ et 3D — physiques et virtuels — de l'interface *The Data Mountain*. L'objectif est de s'assurer que les résultats sont significatifs de l'utilisation de la mémoire spatiale humaine et non un artéfact de l'interaction humaine avec les affichages informatiques. Ces comparaisons révèlent alors que la capacité de l'utilisateur à localiser rapidement des documents se détériore quand la liberté d'utilisation de la troisième dimension augmente. Le modèle physique 3D est moins efficace que celui en 2D, notamment à cause du manque d'habitude des utilisateurs à un plus grand choix dans le placement des pages Web. Comme lors de l'étude précédente, il n'y a toutefois pas de différence significative entre les trois affichages virtuels.

L'efficacité des interfaces utilisateur graphiques dépend fortement des capacités humaines pour la cognition spatiale. Le passage d'interfaces 2D à des interfaces 3D est jusqu'ici principalement motivé par le fait d'augmenter les performances de l'utilisateur grâce à une utilisation plus naturelle de la mémoire spatiale. Cependant il n'apparaît pas clairement que les affichages 3D fournissent ces avantages. Les résultats d'une étude sur ce sujet suggèrent même fortement que l'efficacité de la mémoire spatiale est indépendante de la présence ou de l'absence d'effets de perspective 3D dans les affichages statiques monoculaires [Cockburn 2004]. Devant la forte corrélation entre les performances des utilisateurs dans diverses tâches et la cognition spatiale, la question suivante reste d'actualité : « Que peut on faire pour mieux exploiter les capacités spatiales humaines dans les interfaces utilisateur ? ».

Il existe de nombreuses autres études sur ce sujet (*i.e.* 2D Vs 3D), avec probablement autant qui concluent en faveur d'interfaces 2D que l'inverse [Sutcliffe et Patel 1996, Ridsen *et al.* 2000]. Cette question mérite toute l'attention qui lui est portée et va probablement rester d'actualité pendant encore de nombreuses années. Nous précisons néanmoins qu'il est difficile de contrôler tous les facteurs lors de ces comparaisons 2D/3D. De notre point de vue, le succès de l'utilisation de la 3D est fortement dépendant du profil des utilisateurs mais aussi du type de tâche. Par exemple, les compétences informatiques de l'utilisateur ont de l'importance pour les visualisations 3D. En effet, des temps de réponse plus rapides avec des interfaces 3D (par rapport à des interfaces 2D) peuvent être constatés chez les utilisateurs ayant de plus grandes compétences informatiques [Sebrechts *et al.* 1999]. Et, d'un point de vue plus général, les coûts liés à l'utilisation de visualisations 3D diminuent avec l'expérience. La performance des visualisations est donc liée à une bonne combinaison de la tâche, de l'utilisateur et de l'interface. Dans ce cas, la performance de visualisations 3D serait compa-

²³La version 2^{1/2}D correspond à un affichage en perspective dans lequel l'utilisateur ne peut pas naviguer dans la profondeur de l'interface.

nable aux outils 2D et textuels. Il apparaît aussi que les caractéristiques de l'interface et la nature des tâches sont plus importantes que la dimensionalité de la visualisation. L'objectif reste donc simplement de fournir des fonctionnalités qui correspondent aux attentes des utilisateurs. Par ailleurs la 3D présente souvent des avantages lorsque les données visualisées sont connues de l'utilisateur. Ce constat doit donc être pris en compte lors des évaluations, notamment en laissant à l'utilisateur un temps d'adaptation à l'interface 3D. Malgré les nombreuses propositions d'interfaces 3D pour la visualisation d'informations de nature non géographique, aucun large succès n'a été obtenu auprès des utilisateurs. En effet, l'usage de la 3D reste pour l'instant confiné aux informations de nature géographique (e.g. *Google Earth*²⁴, les jeux vidéos). La visualisation d'informations électroniques abstraites, sous la forme d'un espace concret en 2D ou en 3D, n'est pas encore vraiment acceptée de l'utilisateur *lambda*, à l'exception des informations liées à des positions géographiques (e.g. l'application *Today's Front Pages* présentée sur le **figure A.7** de l'**annexe A**). Pour conclure, s'il s'avère que les interfaces 3D améliorent la mémoire spatiale des utilisateurs pour localiser des objets dans l'interface, il est vraisemblable que les interfaces 3D amélioreraient alors la performance des utilisateurs du fait de la corrélation établie entre la mémoire spatiale et l'efficacité d'utilisation des interfaces utilisateur graphiques.

1.3 Méthodes d'interaction

En plus des techniques de visualisation, il est indispensable de proposer des techniques d'interaction afin d'avoir une exploration des données efficace. En effet, la visualisation de grandes quantités de données exige des méthodes et techniques adaptées qui prennent en compte la capacité limitée de l'affichage. Des techniques de présentation ont donc été proposées afin d'exploiter efficacement l'espace de l'écran et d'augmenter la quantité des informations visualisées. Les techniques d'interaction permettent quant à elles d'interagir directement avec les données présentées et de changer dynamiquement les visualisations selon les objectifs de l'exploration. Elles permettent aussi de relier et combiner de multiples vues indépendantes. Les techniques d'interaction peuvent être organisées selon les effets qu'elles produisent sur l'affichage [Keim *et al.* 2002] :

- **techniques de navigation** : elles permettent la modification de la projection des données sur l'écran *via* des méthodes automatiques ou manuelles.
- **méthodes de perfectionnement des vues** : elles permettent d'ajuster le niveau de détail de la visualisation ou d'une partie seulement. Cela donne donc la possibilité de mettre en valeur certains sous-ensembles des données.
- **techniques de sélection** : elles permettent à l'utilisateur d'isoler un sous-ensemble des données affichées pour réaliser des opérations telles que : le filtrage, l'analyse quantitative, la focalisation. La sélection peut se faire directement sur la visualisation ou indirectement *via* des mécanismes de boîtes de dialogue ou de requêtes.

²⁴<http://earth.google.fr/>

Par ailleurs, on rappelle une approche d'interaction célèbre dans le domaine de la visualisation d'information : *The Visual Information-Seeking Mantra* [Shneiderman 1996].

Overview first, zoom and filter, then details-on-demand.

*Ben Shneiderman, Professor at the University of Maryland, USA,
The Eyes Have It : A Task by Data Type Taxonomy for Information
Visualizations.*

Cette approche consiste à commencer par une vue d'ensemble des données, puis des possibilités de zoom et de filtrage des données sont proposées et enfin il est possible d'obtenir plus de détails sur certaines données. Différentes techniques d'interaction sont présentées dans les paragraphes suivants.

Zoom. Le zoom est la technique de modification de vue la plus connue. Il est largement utilisé dans de nombreuses applications et convient particulièrement bien aux graphes car l'affichage est fait d'éléments graphiques simples [Herman *et al.* 2000]. Quand on traite de grandes quantités de données, il est essentiel de présenter ces données sous une forme compressée afin d'avoir un aperçu général des données. Cependant, il est tout aussi important de permettre des affichages variables des données sous différentes résolutions. Le zoom permet de réaliser cela en affichant les objets de manière plus grande (zoom géométrique), mais il est aussi possible que le zoom change automatiquement la représentation des données afin de présenter plus de détails aux hautes résolutions (zoom sémantique). Dans les interfaces 3D, le zoom — géométrique et sémantique — sur un objet est souvent représenté par un déplacement soit de la position de l'utilisateur afin de le rapprocher de l'objet, soit de la position de l'objet afin de le faire apparaître en avant-plan de l'environnement 3D. Par ailleurs, l'utilisation du zoom est souvent associé aux techniques de *focus+context* présentées dans la suite de cette section.

Filtrage interactif. Le filtrage interactif est une combinaison d'une sélection et d'un perfectionnement de la vue. Lors de l'exploration de grands ensembles de données, il est important de partitionner l'ensemble de données en segments et de se focaliser sur les sous-ensembles qui intéressent l'utilisateur. Cela peut être réalisé directement par une sélection du sous-ensemble désiré (navigation) ou en spécifiant les propriétés du sous-ensemble désiré (requête). Cependant la navigation dans un très grand ensemble de données est difficile et la requête ne produit pas toujours les résultats attendus. C'est pourquoi des techniques de sélection interactive ont été développées pour améliorer le filtrage interactif dans l'exploration de données. Un exemple d'outil de filtrage interactif est le *Magic Lens* [Fishkin et Stone 1995] qui a un effet similaire à une loupe. Les données sous la loupe sont traitées par le filtre et affichées de façon différente du reste des données. En général, cette région sélectionnée est présentée avec

plus de détails, l'affichage du reste des données restant inchangé. Cependant d'autres représentations de la région sélectionnée fournissant différentes vues sont possibles.

Focus+Context. Un des problèmes classiques du zoom est la perte des informations contextuelles, ce qui peut devenir un obstacle d'utilisation. Ce problème peut être évité avec un ensemble de techniques permettant de focaliser sur la zone d'intérêt en conservant le contexte. Ces techniques sont appelées *focus+context* (ou *fisheye* par abus de langage). Elles permettent de visualiser une partie des données avec un haut niveau de détails (*focus*) alors que le reste est affiché avec un bas niveau de détails (*context*). Cependant, il est indispensable d'avoir des techniques d'interaction pour modifier la région sur laquelle on focalise. Certaines fonctions de manipulation utiles sont alors : le redimensionnement (changement de la taille de la région d'intérêt), la sélection (choix d'un point par l'utilisateur et calcul automatique de la nouvelle région d'intérêt) et le déplacement (de la région d'intérêt). Des combinaisons de ces fonctions peuvent être également réalisées. Un premier exemple est la technique *Fisheye View* qui réalise une réduction du contexte en cachant des noeuds dans la structure [Furnas 1986]. Cette réduction est basée sur la distance avec le *focus* et une mesure d'intérêt. Les *focus* et contexte sont donc spécifiés dans le domaine de l'information plutôt que dans le tracé. Une autre possibilité sont les techniques orientées distorsion qui, contrairement à l'approche précédente, définissent les *focus* et contexte dans le domaine de la présentation. Le contexte est alors déformé de façon à ce que l'espace requis diminue. On a donc une vue détaillée près du point d'intérêt tout en gardant des propriétés à grande échelle de l'ensemble du tracé grâce à l'introduction de distorsions loin du point d'intérêt. De nombreux exemples de cette approche générale, nommée *graphical Fisheye Views* [Sarkar et Brown 1994], existent tels que *Perspective Wall* [Mackinlay et al. 1991]. Ils sont basés sur la géométrie hyperbolique à l'image d'*Hyperbolic Browser* (**figure 1.16**) [Lamping et Rao 1994, Lamping et al. 1995]. Bien que l'approche géométrique *fisheye* soit la plus répandue, il existe d'autres approches de *focus+context* telles que la technique *CardVis* basée sur la métaphore d'un paquet de cartes de jeu [Mukherjea et Hara 1999]. La technique de *focus+context* peut aussi servir à visualiser de grandes tables de données et ainsi leur donner du sens (e.g. la visualisation *Table Lens* [Rao et Card 1994]).

Brushing and Linking. Le *brushing* est un processus de sélection interactive. Il est souvent combiné avec le *linking* qui est un processus de communication des données sélectionnées aux autres vues de l'ensemble des données. Il existe de nombreuses méthodes pour visualiser des données multidimensionnelles, chacune ayant des points forts et des faiblesses. L'idée du *brushing and linking* est de combiner différentes visualisations pour surmonter les imperfections des techniques individuelles de visualisation. On peut par exemple appliquer cette méthode aux visualisations générées par les techniques vues précédemment. Ainsi, les objets sélectionnés sont mis en avant dans toutes les visualisations, rendant possible la détection de dépendances et de corrélations [Mann 1999]. En général, la connexion de plusieurs visualisations à travers un

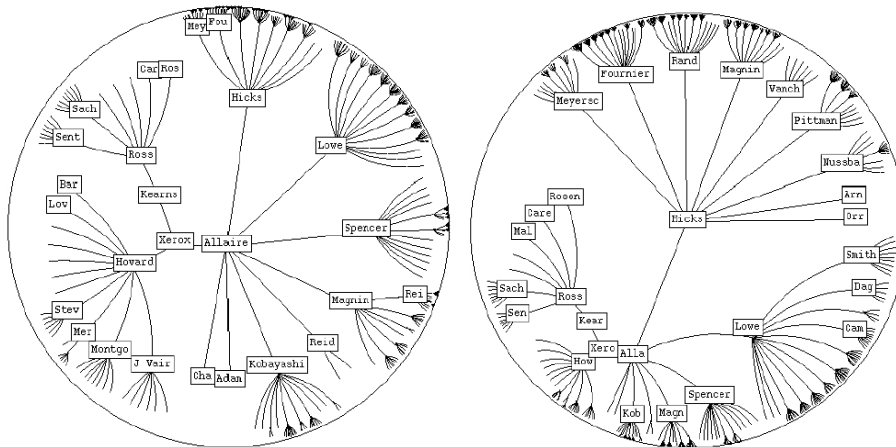


FIG. 1.16 – *Hyperbolic Browser* : illustration d'un changement de *focus* entre l'image de gauche (origine de l'arbre au centre) et celle de droite (où un nouveau *focus* est défini) [Lamping *et al.* 1995]

processus interactif de *brushing and linking* fournit plus d'informations que la considération de ces visualisations de façon indépendante.

Dans les paragraphes précédents, nous avons cité quelques approches d'interaction, généralement utilisées dans les interfaces 2D. Mais, malgré les nombreuses recherches menées dans ce domaine, la majorité des applications grand public restent sur le modèle WIMP (*Window Icon Menu Pointer*) avec des techniques d'interaction génériques telles que le *drag-and-drop* (glisser-déposer). Ce modèle semble toutefois atteindre ses limites face aux nouvelles contraintes (*e.g.* la quantité de données à visualiser, la diversité des supports et des utilisateurs) et même le *drag-and-drop* est amené à évoluer [Hascoët *et al.* 2004]. Le problème est que la pérennité et la fiabilité de ce modèle rend difficile les changements dans ce domaine, bien que certaines nouvelles interactions aient fait leurs preuves en termes d'efficacité. Le coût de changement de paradigme d'interaction est considéré comme trop élevé pour les utilisateurs qui veulent percevoir les bénéfices immédiatement (*i.e.* sans coût d'apprentissage). Pourtant, le seul moyen d'améliorer significativement les interfaces utilisateur est de changer la direction de recherche en passant de la conception d'interfaces à la conception d'interactions [Beaudouin-Lafon 2004]. On constate néanmoins de plus en plus d'applications innovantes proposant de nouvelles interfaces et de nouveaux moyens d'interaction (**figure 1.17**). Parmi ces exemples, on trouve le prototype *BumpTop* qui propose des interactions basées sur une simulation physique réaliste [Agarawala et Balakrishnan 2006] (**figure 1.17(a)**), et l'application *Photosynth*²⁵ qui offre une représentation dans un espace 3D reconstruit d'une grande collection d'images (**figure 1.17(b)**). Ces nouvelles

²⁵<http://labs.live.com/photosynth/>

interactions peuvent aussi être liées à des changements matériels, comme l'utilisation d'une table tactile²⁶ [Han 2005] (**figure 1.17(c)**) ou d'un affichage sous une forme de type holographique tel que le système *Heliodisplay*²⁷. Par ailleurs, les interfaces 3D sont de plus en plus fréquentes et les techniques d'interaction doivent s'adapter aux caractéristiques particulières de ces interfaces (e.g. la nécessité d'avoir différents points de vue).

1.4 Ergonomie

La pertinence d'une visualisation passe évidemment par la prise en considération de certains aspects ergonomiques. Cela va de critères relativement simples et courants à une réflexion plus théorique sur notre système de perception.

Critères de visualisation. Il est nécessaire d'évaluer *a posteriori* la pertinence d'une visualisation. Cette évaluation doit passer par la vérification de critères d'ordre ergonomiques tels que : le temps d'exécution, l'utilisabilité des fonctions, les possibilités et la qualité des interactions, la pertinence des visualisations proposées. On peut alors s'intéresser à quatre facteurs principaux qui influencent l'utilité et l'efficacité d'une visualisation [Mann 1999] :

- **Le groupe d'utilisateur cible.** Il dépend de la manière de penser dans l'espace ou de percevoir l'information.
- **Le type de données.** Il conditionne le choix de représentation graphique. La présence de hiérarchie entre les documents doit être exploitée dans la visualisation. Le nombre de documents influence sur le succès de la visualisation.
- **La tâche à effectuer.** Ce facteur est important pour l'efficacité d'une visualisation. Il y a grand nombre de tentatives pour classer ou évaluer les visualisations pour différentes formes de tâches.
- **Les possibilités techniques.** Elles sont déterminantes pour l'utilisation et le succès de la visualisation.

Il n'existe pas de meilleure visualisation pour tous les cas d'utilisation. Il existe juste au moins quatre facteurs qui influencent l'efficacité des visualisations. De toutes les façons, il semble extrêmement difficile de trouver une visualisation qui satisfasse les attentes de ces quatre facteurs. Il est donc nécessaire de réaliser un compromis entre ces différents facteurs.

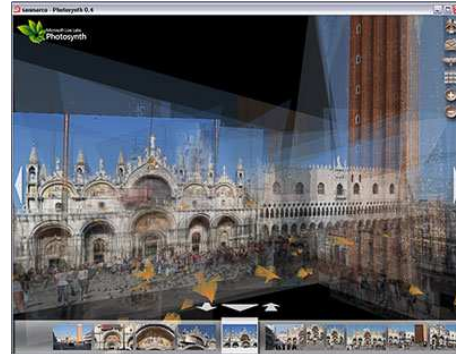
Perception humaine. Le but de ce paragraphe est simplement de rappeler qu'il existe des notions importantes dans ce domaine pour pouvoir traiter efficacement la visualisation de données. En effet, la méthodologie de conception de langages graphiques doit prendre en compte la nature des informations à visualiser mais aussi les capacités de la vision humaine. Ce dernier point fait appel à des théories issues du domaine des

²⁶<http://www.cs.nyu.edu/~jhan/ftirtouch/>

²⁷<http://en.wikipedia.org/wiki/Heliodisplay>



(a) Bureau virtuel classique avec une organisation structurée (1), bureau réel où les objets sont organisés en passant (2), et le prototype *BumpTop* avec des tas comme objet organisationnel fondamental ainsi qu'une simulation physique offrant une interaction fortuite potentiellement plus réaliste (3)



(b) *Photosynth* est une nouvelle technologie de *Microsoft*, qui analyse les similarités présentes dans une grande collection de photos d'un lieu ou d'un objet et les affiche dans un espace 3D reconstruit



(c) Expérimentations de divers scénarios d'application et modalités d'interaction utilisant les interactions multi-points et multi-utilisateurs sur une surface graphique d'interaction



(d) *Heliodisplay*, inventé par Chad Dyner, est un affichage planaire interactif qui projette les images dans l'air et se comporte comme un écran tactile flottant dans l'air

FIG. 1.17 – Nouvelles interfaces, nouvelles applications, nouvelles interactions

sciences cognitives telles que la sémiologie graphique [Bertin 1999] et la théorie de la *Gestalt* [Paul 1979]. L'objectif est de proposer une méthodologie rigoureuse de conception des langages graphiques [Lamy *et al.* 2006]. Nous proposons en guise d'exemple quelques éléments sur la perception visuelle et spatiale.

- **Perception visuelle.** L'étude de notre système de perception visuel peut se révéler utile pour le choix des métaphores de visualisation des données et de leurs propriétés. Certaines erreurs sont à éviter comme le choix de conventions arbitraires dont la signification peut varier selon la zone géographique. Dans la plupart des cas, ces conventions sont adaptées à l'Europe et au nord de l'Amérique. Par exemple, les couleurs rouge (pour signifier « attention ») et vert (pour signifier « OK ») n'ont pas la même signification en Chine. Il faut aussi éviter d'utiliser la couleur pour représenter des données quantitatives continues, étant donné que notre cerveau ne nous permet de détecter qu'un certain nombre de couleurs. Il existe de nombreux autres exemples. Par contre, il peut être utile d'utiliser certaines propriétés de notre système visuel telles que la vision pré-attentive qui permet d'avoir des certitudes à 100% sur certaines valeurs. Toutefois, le nombre de ces valeurs est dépendant de la dimension et des interférences. Par ailleurs, on remarque que bien souvent notre jugement relatif est meilleur (jusqu'à une certaine limite) que notre jugement absolu.
- **Perception spatiale.** Elle n'est pas toujours évidente et dépend des utilisateurs. Cela est d'autant plus vrai pour les affichages 3D où la profondeur est souvent caractérisée par une diminution des textures et des dimensions. Les occultations, bien qu'elles permettent de situer la profondeur de certains objets, sont un problème récurrent de ces interfaces 3D.

L'impact de la perception humaine sur la conception de méthodes de visualisation a été revisité dans de nombreux travaux [Healey et Enns 2002, Kosara *et al.* 2003].

1.5 Discussion

Dans ce chapitre, nous avons présenté les principales approches exploitées dans le cadre de la visualisation de résultats de recherche. L'utilité de ces techniques de visualisation dérive en grande partie de leur capacité à réduire la quantité de travail mental que doit fournir l'utilisateur. Cette thématique est tellement riche qu'il est possible d'écrire un livre entier dessus. L'objectif n'est pas ici d'être exhaustif mais plutôt de donner un aperçu des différentes approches. Pour des informations complémentaires ainsi que d'autres points de vue sur ce sujet, nous dirigeons le lecteur vers d'autres états de l'art similaires [Chevalier 2002, Mann 2002, Käki 2005] ou encore vers d'autres sources documentaires [Geriomenco et Chen 2003].

Dès lors que les résultats de recherche sont visualisés dans un espace multidimensionnel, il nous semble important que les proximités visuelles reflètent les proximités sémantiques des informations affichées. Cette condition a vocation d'aider au maximum l'utilisateur dans sa recherche d'information tout en minimisant sa navigation

dans l'espace des résultats. C'est pourquoi nous nous intéressons plus particulièrement dans la suite de cette thèse aux visualisations sur lesquelles la proximité des documents traduit leur similarité, c'est-à-dire aux approches cartographiques présentées dans la section 1.2.3.3.

Ce chapitre ne traite pas des méthodes de projection (linéaire ou non linéaire) de données qui permettent aussi de visualiser les résultats de recherche. Quelques exemples de ces méthodes sont les analyses factorielles (telles que l'analyse en composantes principales — ACP), la projection de Sammon — *Sammon mapping* [Sammon 1969] — ou les techniques de *Multidimensional Scaling* — MDS. Mais ces méthodes organisent aussi les résultats visuellement du fait de leur projection. Or ce deuxième point n'est abordé que dans le chapitre suivant. Il faut tout de même préciser que ces méthodes ne garantissent pas une utilisation optimale de l'espace de projection (*i.e.* de l'affichage sur l'écran), étant donné que la répartition des données peut être très dense dans certaines régions. Ceci dit, dans le cas de résultats peu nombreux, un post-traitement peut être réalisé afin d'améliorer l'aspect visuel. Un exemple de ce type d'approche est le prototype *JMBrowser* qui utilise la projection de Sammon pour visualiser les résultats d'une recherche de méthodes — dans le langage de programmation *Java* — basée sur leur signature [Bonnell et Le Guernic 2006].

Nous avons donc dressé un panorama des différentes approches en matière de visualisation de résultats de recherche. Toutes ces approches nécessitent une organisation visuelle des résultats. Cette dernière peut être basique ou reposer sur des algorithmes plus complexes. Nous abordons alors le problème de l'organisation des résultats dans le chapitre suivant. On précise toutefois que ces deux parties (la visualisation et l'organisation) peuvent tout à fait être indépendantes jusqu'à un certain point.

Chapitre 2

Organisation des résultats

Devant le grand nombre de résultats que génère une requête sur le Web, l'utilisateur est bien souvent perdu. En effet, il doit faire le tri dans les longues listes de résultats, retournées par la majorité des systèmes de recherche, pour trouver les documents pertinents. La grande taille de ces listes est en partie expliquée par le faible taux de précision des moteurs de recherche. Ce faible taux de précision couplé avec une présentation linéaire des résultats triés selon un critère de « pertinence système » rend difficile la tâche de l'utilisateur. Face à ce constat, deux directions non exclusives sont envisageables pour aider l'utilisateur : la réduction et l'organisation des résultats de recherche. La première possibilité consiste à réduire le nombre de résultats en augmentant le taux de précision des systèmes de recherche. Dans ce but, on peut soit utiliser des méthodes de filtrage, soit raffiner la requête. Cependant cette dernière solution possède certains inconvénients tels que la discontinuité du mécanisme qui ne permet pas de comparer deux raffinements, ou encore l'effort demandé à l'utilisateur pour préciser ces besoins. La seconde possibilité consiste à rendre la navigation dans les résultats de recherche plus facile et plus efficace. Pour cela il faut être capable d'organiser les résultats, ce qui implique de pouvoir rassembler les résultats similaires et donc de s'intéresser aux techniques de classification (et plus précisément aux techniques de *clustering* car on souhaite que le processus de restitution des résultats soit automatique). Cette seconde approche est celle que nous exploitons dans le cadre de cette thèse.

L'organisation des résultats est une étape essentielle pour aider l'utilisateur à exploiter au mieux les réponses issues d'une requête effectuée sur un moteur de recherche. Cet aspect commence à apparaître dans certains moteurs tels que *Clusty* qui propose de regrouper les résultats similaires dans des répertoires grâce à une technique de *clustering* à la volée calculée uniquement sur les premières réponses. Un autre exemple est *Grokker* qui présente les résultats dans des catégories afin d'éliminer les listes désorganisées de résultats qui rendent la recherche frustrante. Dans ce chapitre, nous donnons un aperçu des principales méthodes d'organisation des résultats de recherche. Puis nous proposons notre approche basée sur les cartes auto-organisatrices.

2.1 État de l'art sur l'organisation des résultats de recherche

La RI est une tâche pour laquelle le besoin d'interfaces utilisateur qui organisent les résultats de recherche dans des groupes significatifs, est justifié et se fait amplement ressentir par les utilisateurs. De telles interfaces permettent de donner plus de sens aux résultats et ainsi d'aider les utilisateurs dans leurs recherches. Bien que cette problématique ne soit pas nouvelle, de nombreuses questions restent ouvertes et d'actualité telles que la manière de générer des regroupements exploitables et celle de concevoir des interfaces offrant une exploration utilisant les regroupements. Actuellement, deux grandes familles de méthodes se distinguent : le *clustering* et les catégories hiérarchiques à facettes (HFC¹) [Hearst 2006].

2.1.1 *Clustering*

Le *clustering* est une méthode dont l'objectif est de regrouper des objets selon une mesure de similarité. Étant dans le cas de documents textuels, le calcul de la similarité est généralement basé sur les associations entre les mots et phrases des différents documents. L'utilisation des algorithmes de *clustering* dans le cadre de la RI peut se faire de deux manières. La première repose sur une exécution hors ligne de l'algorithme afin de regrouper les documents similaires d'une collection entière de documents (cf. la méthode *Scatter/Gather* présentée ci-dessous [Cutting *et al.* 1992]). Mais, dans le cas de recherches sur le Web, les documents à considérer sont trop nombreux et trop changeants pour que cette approche puisse être envisagée. En conséquence, la seconde utilisation qui consiste à appliquer l'algorithme de *clustering* uniquement sur les résultats de recherche obtenus en réponse à une requête, semble plus raisonnable et particulièrement adaptée. Un ensemble de conditions préalables à l'utilisation d'algorithmes de *clustering* dans ce cadre applicatif précis a été proposé dans le but de garantir le meilleur résultat possible pour l'utilisateur [Zamir et Etzioni 1998]. Parmi ces exigences, on retrouve les six points suivants.

- **Pertinence.** La méthode doit produire des groupes qui séparent les documents pertinents à la requête de l'utilisateur de ceux qui ne le sont pas.
- **Résumés exploitables.** L'utilisateur a besoin de savoir d'un coup d'œil si le contenu d'un groupe est intéressant. En effet, il ne faut pas remplacer l'examen minutieux de la liste classique de résultats par celui des groupes. La méthode doit donc produire des descriptions concises et précises des groupes.
- **Chevauchement.** Les documents peuvent être multi-thématiques. Il est donc important de ne pas confiner chaque document dans un seul groupe.
- **Tolérance aux *snippets*.** La qualité des groupes doit être assurée même si l'on n'utilise que les *snippets*, et non les documents entiers.
- **Rapidité.** La méthode de *clustering* doit permettre de regrouper un millier de *snippets* en quelques secondes.

¹*Hierarchical Faceted Categories*

- **Incrémentalité.** L'algorithme doit être incrémental afin d'obtenir le résultat le plus rapidement possible.

De nombreux algorithmes de *clustering* de documents apparaissent dans la littérature. Parmi les plus utilisés, on trouve la classification ascendante hiérarchique (CAH). Cet algorithme possède néanmoins certains inconvénients bien connus tels que sa lenteur sur de grandes collections de documents (liée à sa complexité en $O(n^2)$) et sa sensibilité au critère de coupure de l'arbre de classification. Les algorithmes de *clustering* ayant une complexité linéaire sont alors de meilleurs candidats pour satisfaire les exigences d'une exécution en ligne. On retrouve alors l'algorithme *k-means* [MacQueen 1967, Rocchio 1966] qui peut en plus permettre d'avoir des groupes de documents qui se chevauchent. Par contre cette méthode est réputée efficace pour créer des groupes de documents sphériques par rapport à la mesure de similarité utilisée. Or il n'y a aucune raison particulière de croire que les documents (représentés par des vecteurs de mots pondérés et avec une mesure de similarité de type produit scalaire normalisé) forment des groupes approximativement sphériques. La méthode *Single-Pass* est une autre solution de complexité linéaire, qui a l'avantage d'être incrémentale [Hill 1968]. Mais elle souffre aussi du problème lié à la forme sphérique des groupes de documents obtenus. D'autres exemples d'algorithmes de complexité linéaire [Cutting *et al.* 1992, Zamir et Etzioni 1998] sont proposés dans les paragraphes suivants qui traitent de certains travaux majeurs concernant le *clustering* de résultats de recherche.

Cluster Hypothesis. Le regroupement de documents est utilisé dans les SRI depuis plus de 30 ans. À l'origine, l'objectif était d'améliorer l'efficacité — d'un point de vue calculatoire — d'une recherche en réduisant le nombre de documents à comparer avec la requête. Mais une collection de documents ainsi regroupés possède des informations inhérentes qui vont très rapidement être aussi exploitées pour améliorer l'efficacité des résultats des recherches [Jardine et van Rijsbergen 1971]. L'idée, selon laquelle les associations entre les documents fournissent des informations sur la pertinence des documents par rapport aux requêtes, est alors exprimée par la '*Cluster Hypothesis*' : « *closely associated documents tend both to belong to the same clusters and to be relevant to the same request* ». Le regroupement des documents d'une collection peut être utilisé de deux façons. La première consiste à retrouver un ou plusieurs groupes de documents, dans leur intégralité, en réponse à la requête [van Rijsbergen 1979]. La seconde possibilité est d'identifier les groupes qui contiennent vraisemblablement des documents pertinents et ensuite de calculer la similarité entre la requête et chaque document des groupes identifiés. La première solution est plus rapide en temps d'exécution mais la qualité des résultats obtenus dépend de la validité de la '*Cluster Hypothesis*'. Depuis, cette hypothèse a été revisitée de différentes manières [Voorhees 1985, Hearst *et al.* 1995].

Scatter/Gather. Il s'agit d'une méthode qui propose de regrouper hiérarchiquement une collection entière de documents, et qui permet à l'utilisateur de rassembler des groupes à différents niveaux et de les re-séparer [Cutting *et al.* 1992, 1993]. Une éva-

luation de l'efficacité du système pour une navigation dans le contenu d'une grande collection a été réalisée [Pirolli *et al.* 1996]. Dans cette évaluation, *Scatter/Gather* est considéré comme un simple outil de recherche documentaire. Il apparaît alors qu'il induise une image conceptuelle plus cohérente de la collection de documents, un vocabulaire plus riche pour construire les requêtes et qu'il communique des informations sur la structure thématique de la collection de documents. Cette approche de *clustering* peut aussi être appliquée sur les résultats de recherche, c'est-à-dire sur des documents retrouvés comme résultat à une requête [Hearst *et al.* 1995]. Il apparaît que cette méthode permet même de valider la '*Cluster Hypothesis*'. En effet, quand le *clustering* est appliqué aux résultats de recherche, la plupart des documents pertinents tendent à apparaître dans un des *clusters*. Cela implique que la sélection du meilleur *cluster* permet de trouver la plupart des documents pertinents [Hearst et Pedersen 1996].

Grouper. Ce système propose une interface permettant de regrouper dynamiquement, selon leurs similarités, les résultats issus d'un méta-moteur de recherche [Zamir et Etzioni 1999]. *Grouper* utilise l'algorithme de regroupement de documents STC — *Suffix Tree Clustering* — qui est basé sur l'identification des expressions communes aux différents de documents [Zamir et Etzioni 1998]. Dans cette approche, les expressions sont des séquences ordonnées d'un ou plusieurs mots. Les avantages de cet algorithme sont : l'utilisation de groupes de mots ordonnés à la place du modèle classique de « sac de mots », la possibilité d'avoir des groupes de documents qui se chevauchent, son aspect incrémental et sa rapidité (la complexité est linéaire par rapport au nombre de documents). Par ailleurs, cette méthode ne nécessite pas de préciser préalablement le nombre de groupes à réaliser. Dans ce type d'interface, il est aussi essentiel que l'utilisateur puisse identifier rapidement les groupes qui l'intéressent. *Grouper* utilise alors son approche basée sur les expressions (*i.e.* ou groupes de mots ordonnés) extraites des *snippets* pour étiqueter efficacement les groupes de documents. À l'intérieur d'un groupe, les documents sont ensuite triés selon leur similarité par rapport au centroïde du groupe et non par rapport à la requête. Une comparaison a été réalisée entre *Grouper* et un affichage d'une liste triée selon un critère de « pertinence système ». Suite à ce test, trois hypothèses apparaissent. Premièrement, *Grouper* semble retrouver plus facilement un document intéressant. Il paraît aussi être meilleur pour aider l'utilisateur à trouver d'autres documents intéressants supplémentaires. Enfin, cette approche semble préférée des utilisateurs lorsque la tâche à réaliser nécessite de retrouver plusieurs documents. D'un autre côté, il apparaît aussi que *Grouper* soit moins efficace qu'une liste ordonnée si l'utilisateur est intéressé par les premiers résultats de liste. Par contre, *Grouper* est plus efficace pour retrouver d'autres documents supplémentaires apparaissant après les quelques premiers résultats de la liste. Ce constat semble être général à l'ensemble des approches de *clustering* de résultats de recherche. Concernant cette approche, on note toutefois l'absence de clarté dans la manière d'ordonner les groupes. Les auteurs mentionnent le fait que le tri et l'affichage des différents groupes est réalisé d'après une estimation de leur cohérence. On déplore le manque d'information à ce sujet. De plus, une organisation spatiale des groupes selon leur contenu

serait, à notre avis, d'une aide non négligeable pour les utilisateurs qui pourraient alors identifier plus rapidement les groupes intéressants. Un pas dans cette direction peut aussi être réalisé en rendant l'interface hiérarchique et interactive comme les auteurs le mentionnent dans leurs perspectives. Mais le choix de cette solution nécessite alors de faire particulièrement attention à la navigation dans les différents niveaux de hiérarchie afin que l'interface ne perde pas de son utilisabilité.

2.1.2 Catégories hiérarchiques à facettes

Un système de catégories est un ensemble d'étiquettes significatives (*i.e.* mots représentatifs) organisées de manière à refléter les concepts pertinents à un domaine. Il s'agit donc en quelques sortes d'un *thesaurus*. Ces catégories sont créées manuellement, bien qu'il soit possible d'automatiser l'affectation des documents aux catégories avec un certain degré de précision. Un bon système de catégories se doit d'être cohérent et relativement complet. Mais il reste à définir le type de structure de catégories le plus efficace pour l'exploration et la navigation dans les collections d'information. Et il apparaît que l'organisation strictement hiérarchique des catégories est une solution trop « pauvre » pour ce type d'utilisation.

Une représentation alternative est alors apparue ces dernières années, sous le terme de catégories hiérarchiques à facettes [Yee *et al.* 2003]. Cette solution, basée sur une idée très simple, est intermédiaire en termes de complexité mais elle est surtout très riche en termes de flexibilité. Plutôt que de créer une seule grande hiérarchie de catégories, l'idée est de construire un ensemble de hiérarchies de catégories, chacune correspondant à une différente facette (*i.e.* une dimension ou un type de caractéristique) pertinente par rapport à la collection dans laquelle on doit naviguer. *Flamenco* [Yee *et al.* 2003] est un exemple de type d'interface intuitive pour l'exploration et la découverte dans des collections d'information utilisant les HFC.

2.1.3 Discussion

Ces deux méthodes (le *clustering* et les HFC) ont des avantages et des inconvénients [Hearst 2006]. Le *clustering* est une méthode automatique. Cette caractéristique constitue l'avantage majeur de cette méthode qui peut être facilement appliquée à n'importe quelle collection de documents textuels. Par ailleurs, le *clustering* permet :

- de révéler des tendances nouvelles ou inattendues dans un groupe de documents,
- de désambiguïser les requêtes ou de clarifier et préciser une requête vague grâce à l'affichage des thèmes dominants,
- d'éliminer des groupes de documents de la recherche.

Face à ces avantages, le *clustering* présente un certain nombre d'inconvénients parmi lesquels : le manque de prévisibilité, le mélange de plusieurs dimensions simultanément, la difficulté d'étiqueter les groupes, et le manque d'intuitivité des sous-hiérarchies de *clusters*. Certains algorithmes construisent les groupes autour des phrases domi-

nantes. Cette technique permet d'obtenir des étiquettes compréhensibles mais le contenu des groupes ne correspond pas nécessairement à ces étiquettes. Par ailleurs, le *clustering* est une méthode imparfaite dans le sens où elle ne garantit pas que tous les documents semblables soient regroupés ensemble.

Les interfaces basées sur des HFC permettent d'accéder de manière flexible aux différents contenus. Naviguer naturellement dans la hiérarchie se traduit par une requête complexe qui est une conjonction de disjonctions sur les sous-hiérarchies. Les interfaces qui utilisent les HFC donnent des indications à l'utilisateur sur où aller ensuite ainsi que la manière de retourner aux états précédents de l'exploration. Il est aussi possible d'effectuer des recherches textuelles à l'intérieur de la structure de catégories. Cette approche réduit l'effort mental de l'utilisateur en favorisant l'identification au rappel et en lui suggérant des choix — parfois inattendus — à chaque étape. Des études utilisateurs montrent que ce type d'interface est préféré aux listes de résultats, particulièrement pour la tâche dite de navigation [Yee *et al.* 2003]. Par contre, l'inconvénient majeur des HFC concerne la manière d'obtenir les catégories. Ainsi, contrairement au *clustering*, les catégories doivent être connues à l'avance. De plus, dans la majorité des cas, les hiérarchies de catégories sont construites manuellement et l'affectation automatique des catégories aux documents n'est que partiellement réussie. Cependant, des progrès sont réalisés dans la création automatique des HFC [Dakka *et al.* 2005]. Il est notamment possible de créer des HFC presque automatiquement en se basant sur la hiérarchie *WorldNet* [Stoica et Hearst 2004].

Dans le cadre de cette thèse, nous souhaitons pouvoir organiser automatiquement les résultats de recherche. Cela explique que nous nous intéressons plus particulièrement aux techniques de *clustering* d'un point de vue post-traitement des résultats obtenus en réponse à une requête. Par ailleurs, notre objectif est plus de trouver une bonne projection des documents (ou groupes de documents) que la classification en elle-même. L'organisation est donc privilégiée par rapport à la classification. C'est pourquoi nous décidons d'utiliser une carte auto-organisatrice pour organiser les documents. Cet algorithme peut être assimilé à un algorithme de type *k-means* avec une contrainte de voisinage supplémentaire. Par contre, il est réputé pour être plus coûteux en temps de calcul. Cela ne pose toutefois pas de réel problème dans notre cas d'utilisation étant donné que nous avons fait des hypothèses sur le nombre de documents à considérer (de l'ordre de la centaine). De ce fait, les dimensions de la carte restent elles aussi raisonnables. Ce faible nombre de documents n'est pas gênant pour le calcul de la carte auto-organisatrice, du fait que nous sommes plus intéressés par l'organisation spatiale des documents que par la taille des groupes de documents. Les cartes auto-organisatrices ont des propriétés de visualisation indéniables et elles réalisent une projection des données qui préserve la topologie. De plus, la carte obtenue offre une projection lisible et exploitable des résultats, contrairement à la majorité des autres techniques de projection (*e.g.* les méthodes de *multidimensional scaling*). Par ailleurs, les cartes sont un moyen intéressant de représentation des données. Elles permettent de représenter un grand ensemble de documents ainsi que les différents

regroupements qui peuvent être effectués. Leurs avantages sont de fournir une vue globale qui montre les similarités locales entre les documents et d'utiliser une représentation cartographique nécessitant peu d'apprentissage de la part de l'utilisateur.

2.2 Présentation de la méthode des cartes auto-organisatrices

Les cartes auto-organisatrices (SOM²) [Kohonen 1982, 1995] ont suscité beaucoup d'intérêt au cours des 20 dernières années. Elles ont notamment été appliquées à de nombreux domaines. Cette section présente les SOM sans pour autant en détailler toutes les caractéristiques. Des informations complémentaires sont disponibles dans de précédentes thèses [Poinçot 1999, Vesanto 2002].

2.2.1 Principe

Les cartes auto-organisatrices sont une méthode de *clustering* par partitionnement qui permet de projeter non linéairement les données sur une grille de taille et de topologie prédéfinies. Les principes de base de cette méthode sont un apprentissage non supervisé et une organisation spatiale prédéfinie des données. Les cartes auto-organisatrices sont, en fait, des réseaux de neurones à deux couches (pas de couche cachée), où chaque entrée provoque une « excitation particulière » de chaque neurone de sortie.

2.2.2 Algorithme

D'abord quelques notations sont introduites :

- \mathbf{x}_i est le i^e document du corpus et son poids est w_i ,
- \mathbf{m}_k est le vecteur de référence (ou vecteur prototype) associé au k^e neurone de la carte,
- n est le nombre de documents,
- c est le nombre d'unités de la carte,
- h est une fonction de voisinage qui comprend un terme de gain décroissant dans le temps.

Les vecteurs prototypes définissent une tessellation de l'espace d'entrée en ensembles de Voronoi. Ils peuvent être interprétés comme des vecteurs de poids associés aux neurones.

L'algorithme des cartes auto-organisatrices nécessite de fixer préalablement un certain nombre de paramètres. Bien souvent, la connaissance du problème s'avère utile pour effectuer ces choix. Ainsi, il est nécessaire de fixer la dimension de l'espace de départ, la dimension de l'espace de sortie, le nombre de neurones (*i.e.* le nombre de groupes que l'on veut obtenir) ou encore la topologie de l'espace de sortie (*e.g.* les

²Self-Organizing Maps

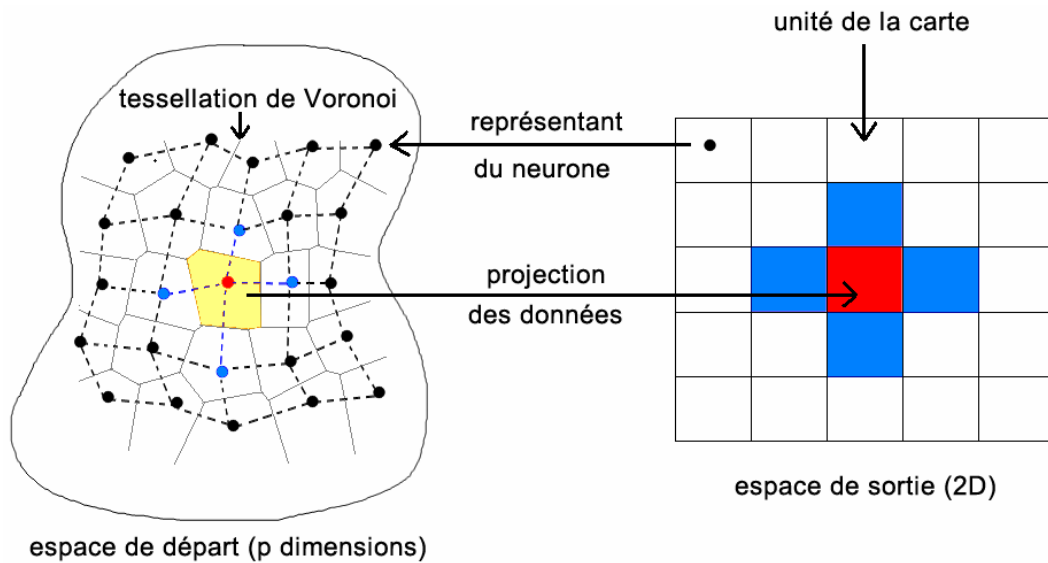


FIG. 2.1 – Description simplifiée du fonctionnement des cartes auto-organisatrices (dans le cas d'une grille 2D carrée et de dimension 5). Les données présentes dans la cellule de Voronoi colorée en jaune sont projetées sur l'unité rouge de la carte. Le représentant de cette unité est le neurone coloré en rouge dans l'espace p-dimensionnel.

grilles carrées ou hexagonales). Concernant la dimension des données, l'algorithme est sensible à l'échelle des données d'entrée, aussi certains pré-calculs sur les données peuvent être nécessaires tels que la réduction du nombre de dimensions.

Il existe deux versions de l'algorithme des cartes auto-organisatrices. La première version, appelée *stochastic SOM*, effectue une mise à jour la carte après chaque nouvelle donnée traitée. Par contre, la seconde version, appelée *batch SOM*, effectue cette mise à jour qu'après avoir traité l'ensemble des données. Ainsi, si I itérations sont nécessaires pour traiter n données, alors la carte est mise à jour $I \times n$ fois dans l'algorithme *stochastic SOM* et I fois dans l'algorithme *batch SOM*.

2.2.2.1 Algorithme *stochastic SOM*

L'étape d'initialisation est traitée dans la partie 2.2.2.5. Ensuite l'étape 2 consiste à comparer le vecteur d'entrée x_i à tous les vecteurs prototypes m_k au sens de la distance euclidienne. Le neurone gagnant (BMU³) est donc celui ayant la distance euclidienne la plus faible avec le vecteur d'entrée. Une fois le neurone gagnant trouvé, l'étape 3 permet d'ajuster le vecteur de poids associé à ce neurone afin de rapprocher son profil à celui de la donnée d'entrée. Il en est de même pour les vecteurs de poids associés aux neurones appartenant à un voisinage donné du neurone BMU(x_i). Ce

³Best Matching Unit

Algorithme 1 *stochastic SOM*

- 1: Initialisation des vecteurs prototypes \mathbf{m}_k
- 2: Sélection du neurone gagnant pour une donnée \mathbf{x}_i

$$\text{BMU}(\mathbf{x}_i) = \arg \min_k \|\mathbf{x}_i - \mathbf{m}_k\|$$

- 3: Mise à jour du profil du neurone gagnant et des neurones voisins

$$\mathbf{m}_k(t+1) = \mathbf{m}_k(t) + h_{\text{BMU}(\mathbf{x}_i),k}(t)[\mathbf{x}_i(t) - \mathbf{m}_k(t)]$$

- 4: Itérer les étapes 2 et 3 jusqu'à convergence

processus est répété jusqu'à convergence dont certains critères sont présentés dans la partie 2.2.2.6.

2.2.2.2 Algorithme batch SOM

Cette version de l'algorithme, parfois appelée version hors ligne, suppose que l'ensemble des observations \mathbf{x}_i soit disponible avant les calculs. Dans cette méthode, V_k peut aussi être appelé ensemble de Voronoi.

Algorithme 2 *batch SOM*

- 1: Initialisation des vecteurs prototypes \mathbf{m}_k
- 2: Quantification vectorielle, où $\bar{\mathbf{m}}_k$ représente la moyenne des observations \mathbf{x}_i de V_k

$$\forall k, \bar{\mathbf{m}}_k = \frac{\sum_{i, \mathbf{x}_i \in V_k} \mathbf{x}_i}{|V_k|}$$

- 3: Étape de lissage

$$\forall k, \mathbf{m}_k = \frac{\sum_{k'=1}^c |V_{k'}| h_{kk'} \bar{\mathbf{m}}_{k'}}{\sum_{k'=1}^c |V_{k'}| h_{kk'}}$$

- 4: Itérer les étapes 2 et 3 jusqu'à convergence

2.2.2.3 Méthode des k -means ou k -moyennes

Si on supprime l'étape de lissage de l'algorithme *batch SOM*, on retrouve la méthode des k -means [MacQueen 1967] qui est elle-même un cas particulier de la méthode des centres mobiles. Cette dernière est une méthode de classification non hié-

rarchique décomposant un ensemble d'individus en un nombre k de classes choisi *a priori*, et ce par un processus itératif convergent de sélection des représentants de chaque classe (un par classe). Une généralisation de la méthode des centres mobiles, dans laquelle chaque classe est représentée par un noyau de plusieurs éléments et non plus par un seul, a aussi été proposée par [Diday 1971] : il s'agit de la technique des nuées dynamiques. Le modèle de regroupement *k-means* est une technique de classification non-supervisée qui permet de partitionner un ensemble fini de données en un nombre fixé de *clusters* ou classes. L'algorithme repose sur une méthode d'affectation/ré-allocation.

Les deux étapes principales de cet algorithme itératif sont donc : l'étape d'affectation des données aux classes et l'étape de mise à jour des centres. La première étape consiste à affecter chaque donnée à une classe selon la règle du plus proche centre. Le calcul de ces distances (entre les données et les centres) se fait en utilisant la distance euclidienne ordinaire. La seconde étape permet de mettre à jour les centres des classes en calculant, pour chaque classe, le barycentre des données qui leur sont respectivement attribuées à l'étape d'affectation.

Algorithme 3 *k-means*

- 1: Initialisation des c centres
- 2: Affectation des données aux classes selon l'équation suivante :

$$\text{BMU}(\mathbf{x}_i) = \arg \min_k \|\mathbf{m}_k - \mathbf{x}_i\| \quad \text{pour } i = 1 \dots n$$

- 3: Mise à jour des centres

$$\mathbf{m}_k = \frac{\sum_{i, \mathbf{x}_i \in V_k} \mathbf{x}_i}{|V_k|} \quad \text{pour } k = 1 \dots c$$

- 4: Itérer les étapes 2 et 3 jusqu'à convergence
-

2.2.2.4 Fonction de voisinage

Cette fonction agit comme un lissage sur la grille, centré sur le neurone gagnant. Il s'agit généralement d'une fonction gaussienne (cf. équation 2.1) ou d'une différence de gaussiennes (aussi appelée « chapeau mexicain »).

$$h_{k, \text{BMU}(\mathbf{x}_i)}(t) = \alpha(t) e^{-\frac{\|\mathbf{r}_k - \mathbf{r}_{\text{BMU}(\mathbf{x}_i)}\|^2}{2\sigma^2(t)}} \quad (2.1)$$

où :

- $\mathbf{r}_{\text{BMU}(\mathbf{x}_i)} \in \mathbb{R}^2$ et $\mathbf{r}_k \in \mathbb{R}^2$ sont les vecteurs position des neurones $\text{BMU}(\mathbf{x}_i)$ et k ,

- $\alpha(t)$ est le taux d'apprentissage qui décroît de façon monotone avec les itérations,
- $\sigma(t)$ est la largeur de la fonction de voisinage qui décroît de façon monotone avec les étapes de régression.

2.2.2.5 Initialisation

Il existe différentes méthodes d'initialisation des vecteurs prototypes (ou neurones). Parmi les plus courantes, on retrouve les méthodes suivantes.

- Initialisation aléatoire : les prototypes sont choisis aléatoirement dans l'espace \mathbb{R}^p . Il faut préciser que cette méthode n'est pas celle qui offre la convergence la plus rapide.
- Initialisation basée sur les données : les prototypes sont choisis parmi les données de l'ensemble d'apprentissage (aléatoirement, en prenant les c premières données ou d'une autre manière).
- Initialisation basée sur une analyse en composantes principales (ACP) : une ACP est réalisée sur l'ensemble des données pour optimiser la répartition initiale des prototypes dans l'espace de description des données. Cette approche permet généralement une convergence plus rapide de l'algorithme.

2.2.2.6 Critère de convergence

Tout comme il existe différentes méthodes d'initialisation, il existe aussi différents critères de convergence dont les plus courants sont les suivants.

- Critère basé sur les itérations : le nombre d'itérations à effectuer est fixé *a priori* ; et une fois ce nombre atteint, l'algorithme s'arrête. Les conditions d'utilisation de ce critère font qu'il ne garantit pas forcément la convergence de l'algorithme.
- Critère basé sur les nouvelles affectations : si le nombre de nouvelles affectations lors de l'étape de quantification vectorielle est inférieur à une certaine valeur prédéfinie, alors l'algorithme s'arrête. Les conditions d'utilisation de ce critère font qu'il ne garantit pas forcément la convergence de l'algorithme.
- Critère basé sur l'erreur quadratique : l'algorithme s'arrête lorsque l'erreur quadratique est inférieure à un certain seuil ϵ prédéfini (stabilisation du coût quadratique). Le calcul de l'erreur quadratique (ou erreur de quantification) est basé sur les équations 2.2 et 2.3, où QE^4 est l'erreur de quantification et MQE^5 l'erreur de quantification moyenne.

$$QE = \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{m}_{\text{BMU}(\mathbf{x}_i)}\|^2 \quad (2.2)$$

$$MQE = \frac{1}{n}QE \quad (2.3)$$

⁴Quantization Error

⁵Mean Quantization Error

Une fois la convergence « atteinte », on dit qu'il y a stabilisation de la carte, Les vecteurs prototypes (\mathbf{m}_k) sont alors considérés comme stationnaires. Cependant, l'obtention d'une carte valide, et au mieux d'une carte optimale, nécessite de générer de nombreuses cartes en faisant varier les paramètres d'apprentissage, et de comparer les cartes obtenues.

2.2.2.7 Complexité

Une implémentation possible de l'algorithme *batch SOM* est proposée, en langage *Java*, dans l'**annexe B**. Cette implémentation ne comporte aucune optimisation particulière. Elle permet de montrer la complexité des différentes étapes de l'algorithme. Ainsi, on remarque que la complexité augmente linéairement avec le nombre de données en entrée. Par ailleurs, la complexité dépend aussi de la topologie utilisée pour la grille de sortie. Si l'on considère une grille carrée (comme dans l'implémentation proposée en annexe), alors la complexité augmente de façon quadratique par rapport à la dimension de la carte. Alors que si l'on considère une grille dans laquelle il est possible d'ajouter un neurone de façon unitaire, la complexité augmente linéairement avec le nombre de neurones. Dans notre cadre applicatif, il s'avère que le nombre de documents est beaucoup plus important que la taille de la carte, ce qui explique que l'étape la plus coûteuse de l'algorithme soit l'étape de quantification vectorielle. Cependant, il est possible d'optimiser cette étape en minimisant le nombre de comparaisons à réaliser entre un document et un prototype de la carte. Dans sa version traditionnelle, cet algorithme compare chaque document à tous les prototypes. Or il est possible d'utiliser, pour chaque document, le neurone gagnant (BMU) de l'itération précédente et ainsi de ne réaliser les comparaisons que dans un voisinage immédiat de ce neurone. Si le nouveau neurone gagnant pour un document est le même qu'à l'itération précédente, alors on arrête les comparaisons. Par contre, s'il s'agit d'un neurone voisin alors on recentre les comparaisons sur ce neurone et sur ses voisins immédiats. On répète ce processus jusqu'à ce que le neurone gagnant soit le neurone central du motif de comparaison. Cela est particulièrement bien adapté aux dernières itérations de l'algorithme où les prototypes sont peu modifiés. Cette optimisation est possible du fait des propriétés des cartes auto-organisatrices.

2.2.2.8 Choix des distances

La distance traditionnellement utilisée dans l'algorithme SOM pour comparer les vecteurs est la distance euclidienne (**équation 2.4**). Il est cependant possible de s'intéresser à d'autres distances telles que la distance du χ^2 (**équation 2.5** où x'_{ij} est la j^e valeur du vecteur \mathbf{x}_i avant normalisation) ou la distance⁶ de Kullback-Leibler (**équation 2.6**). La distance du χ^2 peut être considérée comme une pondération particulière de la distance euclidienne. D'ailleurs un autre type de pondération est celle du *tf.idf* qui consiste à pondérer les variables (avant la normalisation des vecteurs) par le terme

⁶Il s'agit d'une version symétrisée de la dissimilarité de Kullback-Leibler [Rajman et Lebart 1998].

$\log \frac{n+1}{n_j+1}$ où n_j est le nombre de documents du corpus contenant le j^e mot et n le nombre total de documents du corpus (cf. section 2.3.1).

$$d^2(\mathbf{x}_i, \mathbf{m}_k) = \sum_j (x_{ij} - m_{kj})^2 \quad (2.4)$$

$$d_{\chi^2}^2(\mathbf{x}_i, \mathbf{m}_k) = \sum_j \frac{x_{ij}}{\sum_i x'_{ij}} (x_{ij} - m_{kj})^2 \quad (2.5)$$

$$d_{KL}^2(\mathbf{x}_i, \mathbf{m}_k) = \sum_{j, m_{kj} \cdot x_{ij} \neq 0} \log \frac{x_{ij}}{m_{kj}} \quad (2.6)$$

Lors du calcul d'une carte auto-organisatrice de taille 8×8 sur un corpus de 8570 documents textuels représentés par des vecteurs en 164 dimensions, ces quatre distances ont été testées [Bonnell *et al.* 2005d]. Nous avons alors calculé l'erreur de quantification moyenne sur les quatre cartes obtenues. Aucune différence significative n'a été observée, excepté pour la distance de Kullback-Leibler qui présente une erreur plus faible. Mais il ne faut pas tirer de conclusions hâtives étant donné que le choix de l'erreur de quantification n'est pas adapté du fait qu'il favorise la distance euclidienne. Il faut donc réfléchir sur le cadre d'évaluation des différentes distances et effectuer de nouveaux tests sur différents corpus afin de savoir si une distance est plus adaptée pour l'organisation d'informations textuelles.

2.2.3 Étiquetage d'une carte auto-organisatrice

L'interprétation d'une carte nécessite d'avoir préalablement étiqueté les différentes régions de la carte. Ce processus peut être réalisé manuellement en inspectant toutes les données associées à une région ou unité de la carte. Cependant lorsque les ensembles de données sont de grandes dimensions, cela devient très difficile. Il est alors nécessaire d'avoir recours à un processus d'étiquetage automatique des différentes régions d'une carte auto-organisatrice. Plusieurs approches sont possibles, parmi lesquelles les deux suivantes.

La première approche est basée sur le calcul, pour chaque unité de la carte, d'un vecteur d'erreur de quantification δ_k [Merkl et Rauber 1999, Rauber et Merkl 1999, Rauber 1999, Rauber et Merkl 2001]. Ce vecteur est calculé à l'aide de l'équation 2.7 où δ_{kj} représente l'erreur de quantification propre à la j^e dimension, pour la k^e unité de la carte.

$$\delta_{kj} = \sum_{\mathbf{x}_i \in V_k} \sqrt{(m_{kj} - x_{ij})^2} \quad j = 1 \dots p \quad (2.7)$$

Les candidats pour l'étiquetage sont les termes ayant un δ_{kj} en dessous un certain seuil τ_1 . Cependant les candidats doivent être au-dessus d'un second seuil τ_2 afin d'éviter de prendre en considération les termes absents pour représenter les documents d'un neurone particulier.

Une autre approche consiste simplement à projeter les vecteurs unitaires (qui ne contiennent qu'un seul mot) sur la carte afin d'obtenir une carte sémantique [Lin *et al.* 1991]. Elle se décompose en deux étapes :

1. **Division de la carte en régions (caractérisées par un mot).** L'appartenance d'un neurone à une région s'obtient en comparant le neurone à chaque vecteur unitaire et en lui assignant le vecteur unitaire (ou le mot) qui le représente le mieux.
2. **Étiquetage des régions.** Chaque vecteur unitaire est comparé avec tous les neurones et le neurone gagnant est étiqueté avec le mot correspondant au vecteur unitaire. Quand deux mots tombent dans la même région, les deux régions sont fusionnées.

2.2.4 Extensions

Growing Grid (GG). Il s'agit d'un modèle de réseau de neurones similaire au SOM mais avec une architecture adaptative [Fritzke 1995]. En effet, un des inconvénients de l'approche SOM est la taille statique de la carte. Il est parfois difficile de fixer à l'avance la taille de la carte, surtout sans aucune connaissance du type et de l'organisation des données. L'approche *Growing Grid* va alors permettre d'augmenter dynamiquement la taille de la carte en y ajoutant des lignes et des colonnes. Ces ajouts s'effectuent dans les zones où les données sont le moins bien représentées. Cette méthode résout le problème de la taille statique de la carte, mais elle produit de très grandes cartes peu adaptées aux grands ensembles de documents. Par ailleurs, elle suppose d'avoir une carte de structure rectangulaire.

Hierarchical Feature Map (HFM). Il s'agit d'une structure hiérarchique de SOM indépendants [Miikkulainen 1990]. L'idée consiste à ajouter au niveau suivant un SOM pour chaque unité de la carte courante. Cependant, HFM souffre de certains inconvénients du SOM (taille statique des cartes pour chaque niveau) ou encore du problème du choix statique de la profondeur.

Growing Hierarchical SOM (GHSOM). L'objectif de cette méthode [Dittenbach *et al.* 2000a,b, 2002] est de coupler les deux approches précédentes. Le principe est résumé dans l'algorithme suivant.

1. Niveau 0 : une carte avec une seule unité et un vecteur de poids initialisé à la moyenne de toutes les données. On calcule l'erreur de quantification moyenne pour ce neurone (équation 2.8).

$$mqe_0 = \frac{1}{|V_0|} \sum_{\mathbf{x}_i \in V_0} \|\mathbf{m}_0 - \mathbf{x}_i\| \quad (2.8)$$

2. Pour chaque unité i , si l'expression $mqe_i < \tau_2 \times mqe_0$ est fausse, alors on crée une nouvelle carte de taille 2×2 au niveau suivant. Et pour chaque nouvelle carte créée, on effectue les étapes suivantes.
 - On répète l'algorithme SOM pour un nombre fixé d'itérations.
 - On calcule l'unité d'erreur (*Error Unit*) notée e (équation 2.9).

$$e = \arg \max_k mqe_k \quad (2.9)$$

- On ajoute une ligne ou une colonne entre l'unité d'erreur e et son voisin le plus dissimilaire. Les vecteurs de poids de ces nouvelles unités sont initialisés à la moyenne de leurs voisins.
- La carte continue de grandir tant que son erreur de quantification moyenne (MQE) ne satisfait pas l'équation 2.10 où mqe_{parent} est l'erreur de quantification moyenne de l'unité parente correspondante.

$$\text{MQE} < \tau_1 \times mqe_{\text{parent}} \quad (2.10)$$

On trouve des applications de cette méthode dans [Rauber *et al.* 2000]. Une amélioration de cette méthode a également été proposée dans [Dittenbach *et al.* 2001]. Elle concerne l'initialisation des neurones. L'idée est la suivante : plutôt que d'utiliser des données choisies aléatoirement pour initialiser les neurones, on va prendre en compte l'orientation de la carte du niveau précédent. Si un neurone i entraîne la création d'une carte à un niveau inférieur, alors l'initialisation des quatre neurones de la nouvelle carte se fera en fonction des vecteurs de poids du neurone i et de ses voisins. Les expressions de ces initialisations sont données dans [Dittenbach *et al.* 2001]. Cette modification des initialisations permet d'avoir une orientation globale des cartes individuelles dans les différents niveaux de la hiérarchie. En effet, l'orientation d'une carte est désormais conforme à l'orientation de la distribution des données dans sa carte parent dans la hiérarchie. Cet apport à la méthode GHSOM permet d'améliorer l'interprétation des cartes, notamment lorsque des données similaires se trouvent dans différentes branches de la hiérarchie (cela provient du fait que de grands *clusters* peuvent être représentés par plusieurs neurones voisins sur la carte).

Incremental Grid Growing (IGG). [Blackmore et Miikkulainen 1993] Au départ, la grille est composée de quatre neurones connectés et initialisés avec des données choisies aléatoirement. Pour chaque itération de l'algorithme, on effectue les trois étapes suivantes.

1. Adaptation de la grille à la distribution d'entrée à travers le processus des cartes auto-organisatrices.
2. Phase d'expansion. On ajoute de nouveaux neurones dans la région périphérique de la grille qui ne représente pas correctement sa région correspondante dans l'espace d'entrée. Un neurone périphérique est un neurone qui possède moins de quatre voisins. Pour chaque neurone périphérique, on maintient à jour une erreur de quantification cumulative E . Lorsque le neurone gagnant est un

neurone périphérique, alors son erreur de quantification cumulative E est mise à jour selon l'équation 2.11 où \mathbf{x}_i est la donnée et \mathbf{m}_k le neurone gagnant.

$$E(t+1) = E(t) + \sum_j (x_{ij} - m_{kj})^2 \quad (2.11)$$

Le neurone périphérique qui possède alors la plus grande erreur cumulative est le neurone d'erreur. On ajoute donc des nouveaux neurones à chaque place inoccupée de la grille dans le voisinage immédiat du neurone d'erreur. Ces nouveaux neurones sont directement connectés au neurone d'erreur et ils sont initialisés selon les équations 2.12 et 2.13, où \mathcal{N} est l'ensemble des neurones voisins du nouveau neurone et \mathcal{M} est l'ensemble des neurones déjà existants voisins de l'unité d'erreur.

$$\mathbf{m}_{\text{new}} = \frac{1}{|\mathcal{N}|} \sum_{k \in \mathcal{N}} \mathbf{m}_k \quad (2.12)$$

$$\mathbf{m}_{\text{err}} = \frac{1}{|\mathcal{M}| + 1} (\mathbf{m}_{\text{new}} + \sum_{k \in \mathcal{M}} \mathbf{m}_k) \quad (2.13)$$

3. Phase d'adaptation des connexions. On examine les vecteurs des neurones voisins et on détermine si une connexion doit être ajoutée ou supprimée. Si la distance euclidienne entre deux neurones voisins non connectés est inférieure à un seuil (τ_{connect}), une connexion entre ces deux unités est ajoutée à la structure. De même, lorsque la distance euclidienne entre deux neurones connectés est supérieure à un autre seuil ($\tau_{\text{disconnect}}$), la connexion est supprimée de la carte.

Adaptive Hierarchical Incremental Grid Growing (AHIGG). La méthode AHIGG proposée par [Merkl *et al.* 2003] est composée d'un arrangement hiérarchique de réseaux IGG indépendants pour chaque niveau. Elle se résume dans l'algorithme suivant (dont certaines étapes sont similaires au GHSOM).

1. Niveau 0 : une carte avec un seul neurone initialisé à la moyenne de toutes les données. On calcule l'erreur de quantification moyenne pour ce neurone (équation 2.14).

$$mqe_0 = \frac{1}{|V_0|} \sum_{\mathbf{x}_i \in V_0} \|\mathbf{m}_0 - \mathbf{x}_i\| \quad (2.14)$$

2. Pour chaque neurone satisfaisant l'équation 2.15 (où $0 < \tau_2 < 1$), on crée une nouvelle carte de taille 2×2 . Et pour chaque carte créée, on effectue les étapes suivantes.

$$mqe_k > \tau_2 \times mqe_0 \quad (2.15)$$

- Initialisation des neurones selon l'équation 2.16 où \mathbf{v}_{rand} est un vecteur aléatoire de longueur 1.

$$\mathbf{m}_k = \mathbf{m}_{\text{parent}} + mqe_{\text{parent}} \times \mathbf{v}_{\text{rand}} \quad (2.16)$$

- On répète l’algorithme SOM pour un nombre fixé d’itérations
- Phase d’expansion. On cherche l’unité en bordure avec la plus grande erreur de quantification moyenne, appelée par la suite l’unité d’erreur. Puis on génère de nouvelles unités sur les positions inoccupées de la grille qui sont voisines à l’unité d’erreur. Ces nouvelles unités sont directement connectées à l’unité d’erreur et initialisées selon l’équation 2.17 où ϵ est une petite constante telle que $0 < \epsilon \ll 1$.

$$\mathbf{m}_{\text{new}} = \mathbf{m}_{\text{error}} + \epsilon \times \mathbf{v}_{\text{rand}} \quad (2.17)$$

- Phase d’adaptation des connexions : on analyse la distance euclidienne d entre des vecteurs de poids voisins sur la grille. Si $d < \tau_{\text{connect}}$, et les unités ne sont pas connectées, alors une connexion entre les unités respectives est ajoutée. Si $d > \tau_{\text{disconnect}}$, et les unités sont connectées, alors la connexion entre les unités est supprimée.
- La carte continue de grandir jusqu’à ce que l’erreur de quantification moyenne de la carte tombe en dessous d’une certaine fraction τ_1 ($0 < \tau_1 < 1$) de l’erreur de quantification moyenne de son unité parente.

Dans l’algorithme AHIGG, les unités voisines d’une carte ne sont pas nécessairement connectées. La notion de voisinage de l’algorithme SOM est alors adaptée. La mesure de la distance entre deux unités dans l’espace euclidien de la carte est remplacée par une notion de connectivité. Ainsi, la longueur du plus court chemin entre deux unités détermine la force de l’adaptation. Cette adaptation n’est alors plus forcément symétrique par rapport au neurone gagnant (BMU).

MULTISOM. Il s’agit d’une méthode cartographique neuronale représentant l’extension à un contexte multicartes de la méthode SOM [Lamirel *et al.* 2001]. La méthode MULTISOM introduit les principes de points de vue par l’intermédiaire de cartes multiples, et de dynamique dans l’analyse des données par l’intermédiaire de mécanismes de généralisation et de communication intercartes. Grâce aux échanges dynamiques intercartes, on peut réaliser des déductions entre plusieurs analyses sur un même ensemble de données. En fait, cette méthode génère une carte par point de vue. Les améliorations du modèle SOM sont évoquées dans les différents points de la liste suivante.

- **Étiquetage d’une carte.** Chaque carte doit être organisée de façon à être lisible pour l’utilisateur. On va donc chercher à étiqueter les nœuds de la carte en définissant des noms de classes qui représentent au mieux leur contenu. Ce problème dit de « dénomination de classe » n’a pas de solution absolue. On propose ici deux types de stratégies : les stratégies dirigées par les profils des classes et celles dirigées par les profils des données. Les stratégies dirigées par les profils des classes attribuent à chaque classe un nom qui représente la composition des labels des composantes dominantes du vecteur profil (très adapté à la mise en évidence de thématiques⁷ majeures isolées). Les stratégies dirigées par les

⁷Une thématique est la sémantique véhiculée globalement par le profil d’une classe, ou encore par le profil de ses membres.

profils des données attribuent à chaque classe un nom qui représente la combinaison des labels des composantes dominantes du vecteur profil du membre le plus caractéristique de la classe et d'un vecteur profil calculé représentant le profil moyen des membres de la classe. Cette deuxième stratégie fournit des informations complémentaires du fait que les profils de classes représentent une information indirecte (issue de la classification) contrairement aux profils des documents dont l'information est directement issue du contenu des documents.

- **Division d'une carte en aires logiques.** Chaque carte peut être divisée en aires logiques fermées composées par des groupes cohérents de nœuds. Ces aires représentent alors des macro-thématiques dont l'importance est liée à la taille de l'aire.
- **Mécanisme de généralisation.** L'idée est de résumer le contenu d'une carte en un ensemble de cartes plus génériques *via* un processus de généralisation. À partir de la carte de départ, on va introduire de nouveaux niveaux de classification plus synthétiques en réduisant progressivement le nombre de nœuds de façon cohérente. Cette généralisation garde la structure du voisinage et la cohérence des aires logiques. La cohérence d'une thématique est alors liée à la stabilité, au cours du processus de généralisation, du nombre de nœuds la représentant et des frontières des aires auxquelles elle est associée.

Une autre caractéristique de cette méthode est l'utilisation de cartes multiples. Chaque point de vue est représenté par une carte. Le mécanisme de communication intercartes permet alors à l'utilisateur de mettre en évidence des relations sémantiques entre des thématiques de points de vue différents.

- **Notion de point de vue.** Les points de vue permettent de séparer l'espace de description des documents en sous-espaces correspondant à des sous-ensembles de mots-clés. Un point de vue appartient à l'ensemble des parties de l'espace de description des documents et l'union des points de vue recompose l'espace de description des documents. On peut ainsi avoir de nombreux sous-ensembles de points de vue différents. Les points de vue peuvent être recouvrants ou encore respecter une description explicite (champs de description de documents).
- **Mécanisme de communication intercartes.** La communication est basée sur l'utilisation de neurones intermédiaires de transmission intercartes. Ces neurones intermédiaires sont en fait les documents qui ont été re-projetés sur les cartes. Ce processus de communication est divisé en deux étapes : la génération de l'activité thématique sur la carte source (directement par l'utilisateur ou *via* la projection d'une requête utilisateur) et la transmission de l'activité vers les cartes destination. La transmission s'opère également en deux étapes, à savoir la transmission descendante de la carte source vers les neurones-documents puis la transmission ascendante des neurones-documents vers les cartes destination. Ce processus de transmission est une sorte de déduction non supervisée, étant donné qu'il estime les corrélations entre des thématiques rattachées à différents points de vue. Le mode de calcul de cette corrélation sémantique peut être possibiliste (chaque classe d'une carte destination hérite de l'activité retransmise par ses données les plus actives) ou probabiliste (chaque classe d'une carte destination

hérite de l'activité moyenne retransmise par l'ensemble de ses données) suivant les désirs de l'utilisateur. La transmission de l'activité (ascendante et descendante) peut être modulée en considérant le degré d'appartenance d'un document à une classe comme un facteur d'atténuation. On note cependant que ce processus de communication ne fonctionne correctement qu'à condition qu'une partie significative des données joue le rôle d'intermédiaire entre les différentes cartes.

Ce modèle MULTISOM a été appliqué à un ensemble de 1000 brevets du domaine de l'ingénierie des huiles de moteur. Les résultats et l'interprétation des cartes se trouve dans [Lamirel *et al.* 2001]. Un des axes de recherche reste la validation des déductions thématiques intercartes en couplant le modèle MULTISOM avec un modèle symbolique utilisant une classification conceptuelle basée sur les treillis de Galois.

2.2.5 Interprétation

Une fois la carte auto-organisatrice calculée, une étape cruciale reste son interprétation. Pour cela, il est notamment nécessaire de visualiser efficacement les différents *clusters* qui apparaissent dans une carte auto-organisatrice. Parmi les nombreux travaux sur ce sujet, nous citons deux approches qui permettent d'améliorer la visualisation des *clusters* sur une carte auto-organisatrice [Merkl et Rauber 1997]. Ces deux méthodes ne modifient pas l'algorithme, ce qui permet de garder la robustesse de l'approche SOM. Elles se contentent d'utiliser l'information obtenue pendant l'apprentissage ou présente dans toute carte résultant de la phase d'apprentissage. La première méthode appelée *Adaptive Coordinates* permet de capturer les mouvements des différents neurones dans un espace de sortie « virtuel » bi-dimensionnel. Les coordonnées adaptatives sont donc utilisées pour suivre le mouvement des neurones pendant la phase d'apprentissage. La seconde méthode appelée *Cluster Connections* utilise les distances entre les neurones voisins pour déterminer : s'ils appartiennent au même *cluster* et doivent être connectés, ou s'ils se situent sur une frontière de *clusters* et ne doivent pas être connectés.

Il existe toutefois de nombreux autres travaux dans la littérature sur l'analyse et l'interprétation de cartes auto-organisatrices. Par exemple, une première étape devenue classique consiste à regrouper les neurones similaires *via* un algorithme de *clustering* de type CAH ou *k-means* [Vesanto 2002]. La complexité des algorithmes utilisés dans cette étape n'est pas une préoccupation majeure étant donné le faible nombre de prototypes à considérer comparé au nombre de données de départ. L'interprétation des prototypes ou groupes de prototypes nécessite par ailleurs de s'intéresser à la projection des variables sur la carte. Afin d'éviter une analyse manuelle des différentes projections (représentées par une coloration des unités de la carte selon la valeur, pour la variable projetée, du prototype correspondant) une seconde étape est proposée. Elle consiste à calculer une nouvelle carte auto-organisatrice sur les variables, sachant que ces dernières sont définies par leur projection sur la carte des données [Lemaire et Clérot 2005]. Un regroupement des neurones de cette seconde carte est aussi réalisé, ce qui

permet d'identifier rapidement les corrélations et les relations entre les variables. Cette méthode peut aussi être utilisée comme une méthode de réduction des dimensions et de sélection des variables.

Comme la plupart des outils exploratoires, les cartes auto-organisatrices aident à découvrir rapidement des motifs dans les données. Elles fournissent des cartes claires et lisibles mais elles devraient être accompagnées de procédures statistiques lorsqu'une interprétation plus profonde est nécessaire. Les cartes auto-organisatrices, obtenues par un algorithme souvent considéré comme une boîte noire, peuvent être considérées comme un compromis entre un ensemble de *clusters* définis dans un espace de grande dimension et les plans générés par une paire d'axes principaux. Une immersion des cartes auto-organisatrices dans un cadre analytique (l'algèbre linéaire de l'analyse de contiguïté) a alors été proposé [Lebart 2005].

2.3 Organisation des résultats

Cette section présente la méthode employée pour organiser les résultats. Dans le contexte de recherches sur le Web, les documents sont les pages Web retournées par le moteur de recherche, en réponse à une requête de l'utilisateur. Ici, seule l'information textuelle des documents est utilisée. Cette information permet d'avoir une représentation vectorielle (vecteurs de mots) des pages, ce qui est fréquemment utilisé dans le domaine de la RI [Salton *et al.* 1975]. Le nombre de résultats à traiter doit aussi être spécifié parce qu'il est crucial pour les choix d'organisation et de visualisation. Une étude montre que 81.7% des utilisateurs essayent une nouvelle recherche s'ils ne sont pas satisfaits avec les réponses qu'ils trouvent à l'intérieur des trois premières pages de résultats [iProspect 2004]. Une mise à jour récente de cette étude confirme ce constat, étant donné que le pourcentage d'utilisateurs ne dépassant pas les trois premières pages de résultats est désormais de 88% [iProspect 2006]. Cependant il serait trop restrictif de seulement considérer les 30 premiers résultats (10 résultats par page). En effet cette étude a été réalisée sur des moteurs de recherche proposant une visualisation linéaire des résultats (listes ordonnées) et les utilisateurs veulent peut-être voir plus de résultats sur des interfaces de visualisation 2D ou 3D. Face au manque d'études pour les visualisations 2D et 3D, ce nombre est actuellement limité à un maximum de 200 résultats. En pratique, les visualisations proposées dans les chapitres suivants ne prennent en compte que les 50 à 100 premiers résultats, ce qui est plus que les 30 résultats avec lesquels les utilisateurs sont satisfaits. Ce faible nombre de résultats n'est pas réellement un problème pour calculer la carte auto-organisatrice, parce que l'organisation des résultats est plus importante que le *clustering* en lui-même.

L'organisation basée sur les SOM peut être divisée en trois principales étapes : une étape de pré-traitement pour définir la représentation des documents, une étape de calcul avec l'exécution de l'algorithme, et une étape de post-traitement pour regrouper les groupes similaires.

2.3.1 Pré-traitement des résultats

Les résultats sont les réponses d'une requête effectuée sur un SRI. Ils peuvent être multimédia, mais seul le contenu textuel est exploité pour les organiser. Ainsi, chaque résultat peut se résumer à une liste de mots.

Cette première étape s'intéresse à la représentation des documents et plus précisément à la manière dont les vecteurs de mots sont définis à partir du contenu des documents. Les termes contenus dans les pages indexées⁸ sont extraits afin de représenter les documents. Avec cette extraction de termes nous sommes confrontés à un grand nombre de mots. Ainsi notre premier traitement consiste à supprimer les mots vides grâce à un dictionnaire. Pour chaque mot restant, on calcule le produit de sa fréquence d'apparition dans le corpus par sa fréquence documentaire inverse⁹ (*idf* pour *inverse document frequency* [Spärck Jones 1972]). Ensuite les mots sont classés par ordre décroissant des valeurs ainsi calculées. Les mots sélectionnés pour l'organisation des résultats sont ceux qui correspondent aux 200 premières valeurs. Ce choix, fixé de manière empirique, est suffisant pour avoir une bonne représentation des documents et est compatible avec l'efficacité du temps d'exécution. Il peut néanmoins être modifié dans l'interface du prototype présenté dans le chapitre 4. Chaque mot sélectionné est alors associé à une variable de la carte auto-organisatrice. Ainsi le nombre maximum de dimensions pour l'espace de départ est 200. Une solution alternative à cette sélection de variables est d'utiliser une méthode de réduction des dimensions (*e.g.* l'ACP ou le *Random Mapping* [Kaski 1998]). Dans ce cas de méthodes linéaires, les dimensions réduites sont des combinaisons linéaires des dimensions de départ. Elles perdent alors de leur interprétabilité (notamment lors de l'étiquetage des unités de carte auto-organisatrice) dans un processus de traitement automatique comme proposé ici. Cela explique notre choix en faveur de la sélection des variables présentée précédemment.

Toutes les variables de l'espace de départ n'ont pas la même importance pour le *clustering* des résultats. Cela est pris en compte par la pondération *tf.idf* (*cf.* **équation 2.18** où n_j est le nombre de documents qui contiennent le j^e mot [Salton et Buckley 1988]) qui permet d'augmenter ou de diminuer l'importance de certains mots. Cette pondération des termes peut aussi être interprétée comme une pondération particulière de la distance euclidienne. On rappelle que le poids du j^e mot dans le document x_i est donné par l'**équation 2.18**. Après avoir appliqué cette pondération, les vecteurs représentant les documents sont normalisés.

$$tf.idf(\text{mot}_j, x_i) = tf(\text{mot}_j, x_i) \times idf(\text{mot}_j) \quad (2.18)$$

$$idf(\text{mot}_j) = \log \frac{n + 1}{n_j + 1} \quad (2.19)$$

La sélection des variables rend possible la présence de documents juste représentés

⁸Une page indexée est seulement la page Web associée à l'URL, et non le site entier.

⁹c'est-à-dire le nombre de documents du corpus qui contiennent ce mot

par un faible nombre de variables, ce qui signifie que certains documents peuvent être mal représentés. Dans le domaine de la fouille de données, ces documents mal représentés sont généralement supprimés avant l'analyse. Cependant, dans le contexte de recherches sur le Web, ces documents ne peuvent pas être supprimés. En effet, la suppression de documents peut être assimilée à cacher de l'information à l'utilisateur. Ainsi, un poids est donné à chaque document. Ce poids est élevé pour les documents bien représentés et bas pour les documents mal représentés. Dans un premier temps, la pondération des documents utilisée est définie de la façon suivante : chaque document a un poids proportionnel à son nombre de variables non nulles.

2.3.2 Auto-organisation des résultats

Cette seconde étape correspond à l'exécution de l'algorithme 4 qui est une version modifiée de l'algorithme *batch SOM*. En effet, une pondération des documents est prise en compte dans l'algorithme proposé. On précise que w_i est le poids associé au i^e document et que W_k est la somme des poids des documents qui appartiennent à l'ensemble de Voronoi associé au k^e neurone.

Algorithme 4 Algorithme *batch SOM* avec pondération des données.

1: Initialisation

2: Quantification vectorielle :

Calcul de la BMU : $\forall i \quad \text{BMU}(\mathbf{x}_i) = \arg \min_k \|\mathbf{x}_i - \mathbf{m}_k\|$

Calcul de la nouvelle valeur : $\forall k \quad \bar{\mathbf{m}}_k = \frac{1}{W_k} \sum_{i=1, \text{BMU}(\mathbf{x}_i)=k}^n w_i \mathbf{x}_i$

3: Étape de lissage :

$$\forall k \quad \mathbf{m}_k = \frac{\sum_{k'=1}^c W_{k'} h_{kk'} \bar{\mathbf{m}}_{k'}}{\sum_{k'=1}^c W_{k'} h_{kk'}}$$

4: Retour à l'étape 2 tant que la convergence n'est pas atteinte

Dans cette approche basée sur les SOM et utilisant une pondération des documents, des valeurs particulières des poids peuvent être interprétées de la façon suivante. Si tous les poids des documents sont fixés à 1, cet algorithme est l'algorithme *batch SOM* classique. Fixer le poids d'un document à 0 signifie que ce document est seulement projeté sur la carte calculée (*i.e.* ce document n'est pas utilisé pendant le calcul). Comme le but est de classifier les résultats de recherches sur le Web, une contrainte doit être ajoutée sur l'algorithme : il doit être déterministe. La même requête sur le même corpus doit toujours fournir le même résultat. Pour cela, la version *batch* de l'algorithme est calculée sur une carte de taille prédéfinie (5×5) et les pre-

mières données sont toujours utilisées pour initialiser les neurones.

Concernant l'étiquetage des différentes unités de la carte (*i.e.* neurones), la méthode *LabelSOM* est utilisée [Rauber et Merkl 2001]. Chaque étiquette correspond à un mot et les étiquettes sont ordonnées par ordre croissant de l'erreur de quantification de la variable correspondant à ce mot.

2.3.3 Post-traitement des résultats organisés

Cette dernière étape consiste à regrouper les neurones similaires afin d'obtenir les différents thèmes de la recherche. De nombreuses méthodes traitent de ce problème comme les *k-means* ou la CAH. Étant dans le contexte de recherches sur le Web, on rappelle la nécessité d'avoir une solution déterministe. Lors de ce post-traitement, on ne connaît pas à l'avance le nombre de regroupements optimal pour faciliter la recherche de l'utilisateur. Par ailleurs, ce dernier peut vouloir des groupes de thématiques plus ou moins générales (*i.e.* différents niveaux d'abstraction). Dans ce but, il nous semble plus approprié d'utiliser une CAH. La complexité de cet algorithme ne pose ici aucun problème étant donné le faible nombre de données à traiter. Ce choix permet à l'utilisateur de naviguer, si besoin, dans les différents niveaux hiérarchiques.

Pour cette étape de *clustering*, les données d'entrée sont les neurones ayant un nombre de documents non nul. En effet, les neurones ne contenant aucune donnée n'apportent pas d'information utile pour le calcul des regroupements de neurones. Il est préférable de ne pas les considérer, ce qui permet d'améliorer le temps de calcul de la CAH et de garder ces neurones comme séparateurs visuels des différents regroupements de neurones. La CAH est alors calculée sur ces neurones « non vides ». L'idée est d'extraire les principaux thèmes de la recherche afin de donner à l'utilisateur un aperçu des principaux thèmes de la recherche. Cependant il n'est pas raisonnable de couper arbitrairement dans la hiérarchie de *clusters* obtenue, étant donné que le nombre de groupes naturels dans les données est inconnu *a priori* et dépendant de la requête effectuée. Nous proposons alors de couper le dendrogramme au niveau du saut maximum, ce qui n'empêche toutefois pas l'utilisateur de se déplacer dans l'ensemble de la hiérarchie s'il le souhaite. Chaque groupe est étiqueté avec les étiquettes les plus fréquentes des neurones qu'il représente. Bien que cette méthode soit basique, elle est actuellement suffisante pour donner à l'utilisateur un étiquetage des différents thèmes basé sur les mots.

2.4 Évaluation de l'organisation

L'évaluation de l'organisation des documents passe, dans un premier temps, par des mesures de la qualité d'un algorithme de *clustering*. En effet, l'objectif est d'évaluer si les documents similaires sont bien regroupés ensemble et si des documents de thèmes différents sont bien séparés. Pour cela, on retrouve les critères classiques

de compacité et de séparabilité. Ces critères permettent de valider les objectifs des techniques de *clustering* qui sont de minimiser la variance intra-classe (critère de compacité) et de maximiser la variance inter-classes (critère de séparabilité). Cependant, les techniques de *clustering* topologique nécessitent la prise en compte d'un autre critère afin d'évaluer l'organisation des classes les unes par rapport aux autres. Dans ce cas, ce critère remplace celui de la séparabilité qui n'a plus vraiment de sens. Ce type de critère d'évaluation existe dans le cadre des cartes auto-organisatrices [Lesot *et al.* 2003].

La validité de l'organisation des résultats proposée dans ce chapitre repose sur celle des cartes auto-organisatrices, qui n'est plus à prouver. Bien qu'il soit possible de calculer l'erreur de quantification, les critères liés à l'organisation des résultats ne sont pas vraiment utilisables du fait de l'absence d'étiquetage préalable et manuel du corpus. Pourtant cette étape est bien plus importante ici que l'évaluation de la compacité des groupes et elle est nécessaire, notamment pour valider les différents paramètres liés à cette méthode. Actuellement ces paramètres sont fixés de manière empirique. Par ailleurs, si les résultats observés sur des requêtes multi-thématiques (*e.g.* « orange ») sont globalement bons, nous n'avons pas approfondi l'impact de notre méthode sur des requêtes possédant une seule thématique.

2.5 Discussion

L'organisation des documents présentée ici est basée sur les cartes auto-organisatrices de Kohonen. Cette méthode, qui est uniquement basée sur la distribution des mots, a l'avantage de respecter la proximité « sémantique » des données. Cela nous permet ainsi d'avoir un premier niveau d'abstraction si on se déplace au niveau des neurones. Alors une classification ascendante hiérarchique est appliquée sur les neurones en utilisant le critère d'agrégation de Ward. Ainsi, cela fournit un second niveau d'abstraction qui donne à l'utilisateur les principaux thèmes de sa recherche. Ces thèmes sont définis par une combinaison des étiquettes des neurones. Ces niveaux abstraits ne sont pas basés sur des ontologies ou un étiquetage des documents. Cependant, croiser les étiquettes des neurones avec une ontologie ou utiliser des données sémantiques sur les documents pendant l'algorithme peuvent être des perspectives intéressantes. Cette approche est aussi ouverte et proche d'autres approches plus sémantiques telles que les *Topic Maps* [Le Grand et Soto 2002]. En effet, les cartes auto-organisatrices permettent de naviguer dans plusieurs niveaux d'abstraction comme les *Topic Maps* (les *topics* sont représentés ici par les neurones ou les groupes de neurones). Cependant une méthode qui fournit la meilleure organisation possible dans tous les cas n'existe probablement pas. C'est pourquoi plusieurs méthodes d'organisation doivent être définies afin de sélectionner la plus adaptée à chaque cas. Par exemple, il ne semble pas pertinent de calculer une carte auto-organisatrice quand le nombre de résultats à la requête est extrêmement faible. Par ailleurs, on rappelle que si deux documents voisins sur une carte auto-organisatrice sont aussi semblables que possible (il peut y avoir

un « fossé » entre deux unités voisines), il se peut que deux documents semblables ne soient pas proches sur la carte auto-organisatrice. Il s'agit là d'un artéfact connu des cartes auto-organisatrices.

L'utilisation des cartes auto-organisatrices proposée dans ce chapitre n'exploite cependant pas les nombreuses possibilités de cette approche. En effet, le succès de cette méthode a encouragé de nombreux travaux sur cet algorithme. Ainsi nous identifions plusieurs perspectives intéressantes telles que l'utilisation de cartes hiérarchiques, mais aussi de cartes de taille dynamique (méthode *GHSOM* présentée dans la section 2.2.4), ou encore de cartes multiples (méthode *MULTISOM* présentée dans la section 2.2.4). D'autres perspectives particulièrement intéressantes, concernant l'algorithme *batch SOM*, sont l'incrémentalité [Baez-Monroy et O'Keefe 2005] et l'interactivité. L'incrémentalité (ou la progressivité) de l'algorithme n'a de sens ici que si le temps de calcul est trop élevé pour l'utilisateur, donc dans l'hypothèse où l'on souhaite augmenter soit le nombre de résultats, soit le nombre de dimensions, soit la taille de la carte. L'idée consiste alors à fournir à l'utilisateur des résultats intermédiaires à différentes étapes du calcul, ce qui peut se traduire sur l'interface par un affichage dynamique des résultats. Cela nécessite cependant que l'ordre général de la carte ne soit plus modifié pour proposer à l'utilisateur une première version des résultats. En effet, cette contrainte permet de s'assurer que l'affichage des résultats suivants ne modifie pas radicalement l'ordre de la carte. Elle permet donc aussi d'éviter à l'utilisateur de produire de nouveaux efforts pour s'approprier une « nouvelle » carte. On précise aussi qu'une réduction du temps de calcul de la carte peut être envisagée en utilisant une technique d'échantillonnage [Bonnell *et al.* 2005d]. Il reste toutefois à vérifier l'impact d'une telle méthode sur la qualité de l'organisation des données. Concernant l'interactivité, un objectif est par exemple de donner à l'utilisateur la possibilité d'identifier des documents pertinents ou non pertinents (soit au cours de l'algorithme, soit *a posteriori*) et d'essayer de prendre en compte les choix de l'utilisateur sans calculer une nouvelle carte Hatano *et al.* [1999]. Une idée possible est alors d'augmenter l'importance (et donc la pondération) des éléments pertinents et, à l'inverse, de diminuer l'importance des éléments non pertinents (ou de les supprimer). Cela peut être pris en compte dans la carte auto-organisatrice en effectuant des mises à jour locales, c'est-à-dire des itérations de l'algorithme avec une fonction de voisinage restreinte aux voisins immédiats (*i.e.* comme cela est le cas pour les dernières itérations de l'algorithme).

D'un autre côté, l'approche que nous proposons exploite l'ensemble du contenu textuel des documents. Il serait alors intéressant d'évaluer la qualité de notre méthode dans le cas où elle est uniquement appliquée sur les *snippets*. De précédents travaux montrent que la tâche de classification (dans le cas de plusieurs algorithmes dont celui des *k-means*) n'est que très faiblement dégradée lorsque l'on utilise les *snippets* [Zamir et Etzioni 1998]. Cet impact reste à vérifier sur l'organisation des résultats.

Par ailleurs, l'application d'un post-traitement sur le positionnement des neurones est une solution pour optimiser leur placement qui est une information essentielle du

fait de son impact direct sur la visualisation. Ainsi un neurone serait plus proche de ses voisins les plus similaires que des autres. Une telle approche permet alors d'effacer la notion de « fossé invisible » pouvant exister entre deux unités voisines de la carte pourtant très dissimilaires. Par ailleurs, nous ne nous sommes pas intéressés à l'organisation des documents au sein d'une unité de la carte. Pour finir, il faut noter que d'autres approches cartographiques peuvent être envisagées telles que l'exemple original des automates cellulaires [Azzag *et al.* 2006].

Dans ce chapitre, un bref rappel des différentes approches possibles pour organiser les résultats a été réalisé. Puis nous avons présenté notre méthode d'organisation basée sur les SOM. Il nous reste maintenant à définir une représentation visuelle de ces résultats désormais organisés, ainsi que les interactions associées. Cela est l'objectif du chapitre suivant.

Chapitre 3

Visualisation

Dans le chapitre précédent, nous avons proposé une méthode d'organisation des résultats basée sur leur contenu. Il faut désormais proposer une métaphore de visualisation exploitant au mieux l'organisation obtenue. Dans le domaine de la visualisation de l'information, le choix d'une métaphore est toujours très difficile car il a des implications directes sur l'utilisabilité des interfaces.

3.1 Introduction

3.1.1 Définition des éléments à visualiser

Les éléments que l'on souhaite visualiser sont les résultats issus d'une recherche textuelle sur un moteur de recherche sur le Web. Ces résultats sont accompagnés de certaines informations telles que leur titre, leur URL, leur résumé, leur score de pertinence ou encore leur copie d'écran. Ces informations sont disponibles dans la base de données mais il n'est pas impératif de toutes les exploiter. Ce sont les choix en matière de visualisation qui vont déterminer les informations à utiliser. Ces résultats possèdent d'autres caractéristiques intéressantes, il s'agit des relations entre eux. En effet, les résultats ont été préalablement organisés (*cf.* chapitre 2) afin d'obtenir une représentation spatiale basée sur les proximités (d'un point de vue statistique) qui peuvent exister entre les différents documents. Cette organisation est une information qui a un impact important sur les choix en termes de visualisation.

3.1.2 Approche proposée

Lors de recherches sur le Web, la navigation dans l'ensemble des résultats souffre de l'absence d'une structure apparente. De ce fait, il est notamment difficile pour l'utilisateur de retrouver des informations qu'il a déjà vues. L'objectif est alors de proposer une interface utilisateur basée sur une métaphore qui permette de l'aider dans la compréhension des résultats, en utilisant des concepts déjà connus de l'utilisateur. Il semble cohérent de vouloir représenter l'espace d'information par une métaphore

spatiale. Ce type de métaphore permet entre autres d'exploiter certaines capacités humaines. En effet, l'utilisateur a l'habitude d'organiser les objets de manière spatiale et il a donc développé des capacités cognitives spatiales que ce type de métaphore permet d'exploiter (*e.g.* l'organisation des objets dans l'espace et le raisonnement par rapport à leur location). Ces métaphores spatiales peuvent exprimer les relations entre les objets. Cependant, elles posent généralement le problème du passage à l'échelle. Nous nous intéressons alors plus particulièrement à la métaphore de la ville car sa structure spatiale est adaptée à notre problématique et elle est réputée pour ne pas souffrir du problème du passage à l'échelle (notamment grâce à l'utilisation de cartes pour se repérer et à l'infrastructure de la ville) [Dieberger 1997]. Un autre avantage à ce choix est que les utilisateurs connaissent déjà les stratégies pour naviguer efficacement dans une ville inconnue. Cette métaphore de la ville a déjà été abordée dans quelques travaux [Dieberger 1994, Dieberger et Frank 1998, Rossi et Varga 1999].

Le choix de la métaphore de la ville nous entraîne vers la proposition d'une interface 3D. Cela respecte les contraintes initiales du sujet de cette thèse dans lequel il est proposé d'explorer l'utilisation de métaphores 3D de visualisation de résultats de recherche. On peut néanmoins se poser la question de la légitimité de l'usage de la 3D par rapport aux interfaces 2D. Cette question a déjà été abordée dans la section 1.2.4 et aucune conclusion fiable à ce sujet n'émerge actuellement. Cependant, il semble possible (bien que non prouvé) que la 3D permette de maximiser l'efficacité de l'utilisation de l'espace sur l'écran.

3.2 Métaphore 3D de la ville

3.2.1 Principe

Nous exploitons ici la métaphore de la ville dans une version relativement simplifiée. Les bâtiments sont utilisés pour représenter les résultats de recherche ainsi qu'un certain nombre de leurs caractéristiques. Les quartiers de la ville correspondent alors à des groupes de résultats. Ils sont séparés par des rues. L'organisation des quartiers dans la ville est réalisée de façon à donner du sens aux résultats afin d'aider l'utilisateur dans sa tâche. La navigation dans la ville peut se faire selon plusieurs modes : à pied (utilisation de la distance de Manhattan), en survol (utilisation de la distance euclidienne) ou par téléportation. Par ailleurs, une seule ville statique est utilisée pour représenter les résultats d'une recherche.

3.2.2 Caractéristiques

La métaphore de la ville proposée a évolué au cours de la thèse. Nous présentons d'abord les caractéristiques liées à la première version. Puis nous expliquons les différentes modifications apportées pour aboutir à la version actuelle de la métaphore.

3.2.2.1 Première version

La métaphore de la ville est utilisée ici pour représenter les résultats issus d'une recherche (*i.e.* les sites Web). Les propriétés majeures de cette métaphore sont récapitulées dans le **tableau 3.1**. Chaque bâtiment représente une page Web. Les bâtiments ont une hauteur proportionnelle à la taille des pages. Ils possèdent un premier toit dont la couleur indique si la page a déjà été visitée. Les bâtiments associés à des pages contenant des images ont aussi un deuxième toit sur lequel l'utilisateur peut cliquer pour accéder à une nouvelle métaphore représentant uniquement les images de cette page Web. D'un point de vue organisation, les bâtiments se trouvent sur un quadrillage 2D où les axes représentent les mots-clés les plus fréquents¹ et les noms des hôtes. Chaque bâtiment se trouve donc à l'intersection de deux rues : l'une représente un mot-clé et l'autre un nom d'hôte. On peut sélectionner une rue « mot-clé » et une rue « hôte ». Cette sélection est alors visuellement représentée par une coloration au sol de la rue (**figures 3.1** et **3.2**). Le survol d'un bâtiment par le curseur de la souris entraîne l'apparition d'informations contextuelles sur le résultat associé (*e.g.* l'URL, la description et des mots-clés). La position de cet affichage est calculée en fonction de celle du curseur de la souris sur l'écran. Enfin, l'ouverture de la page Web dans une nouvelle fenêtre du navigateur Web se fait en « double cliquant » sur le bâtiment correspondant.

primitive visuelle	information associée
bâtiment	site Web
hauteur du bâtiment	taille du site Web
couleur du premier toit d'un bâtiment	site déjà visité par l'utilisateur (toit rouge) ou non (toit gris)
deuxième toit	présence d'images dans le site Web correspondant
action graphique	interaction associée
sélection d'un bâtiment	ouverture du site Web associé
pointage d'un bâtiment	affichage d'informations contextuelles sur le site Web associé (URL, description, mots-clés...)
sélection du deuxième toit d'un bâtiment	passage à la métaphore de la galerie qui affiche les images du site Web sélectionné

TAB. 3.1 – Propriétés de la métaphore

Problèmes liés à la métaphore de la ville. Cette première version de la métaphore de la ville souffre d'un certain nombre de problèmes listés ci-dessous.

¹À chaque page Web, on associe une liste de mots-clés les plus pertinents. Les mots-clés les plus fréquents sont donc ceux qui apparaissent le plus souvent dans l'ensemble des résultats.

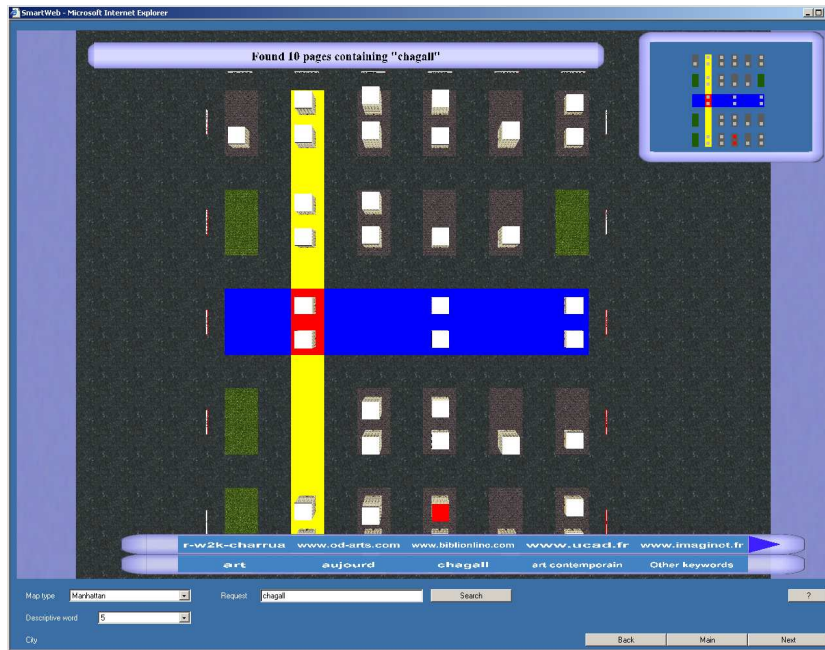


FIG. 3.1 – Vue de dessus de la métaphore de la ville

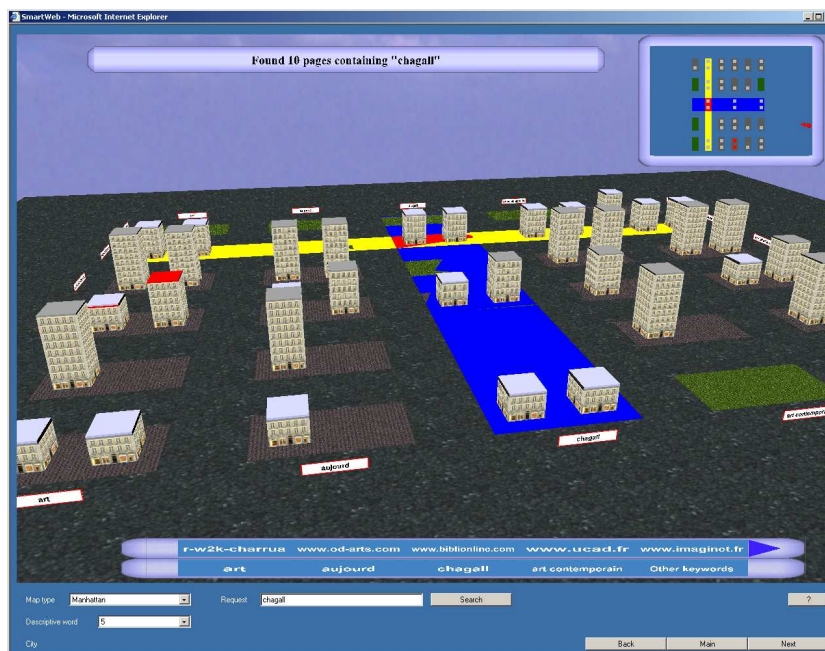


FIG. 3.2 – Vue panoramique de la métaphore de la ville

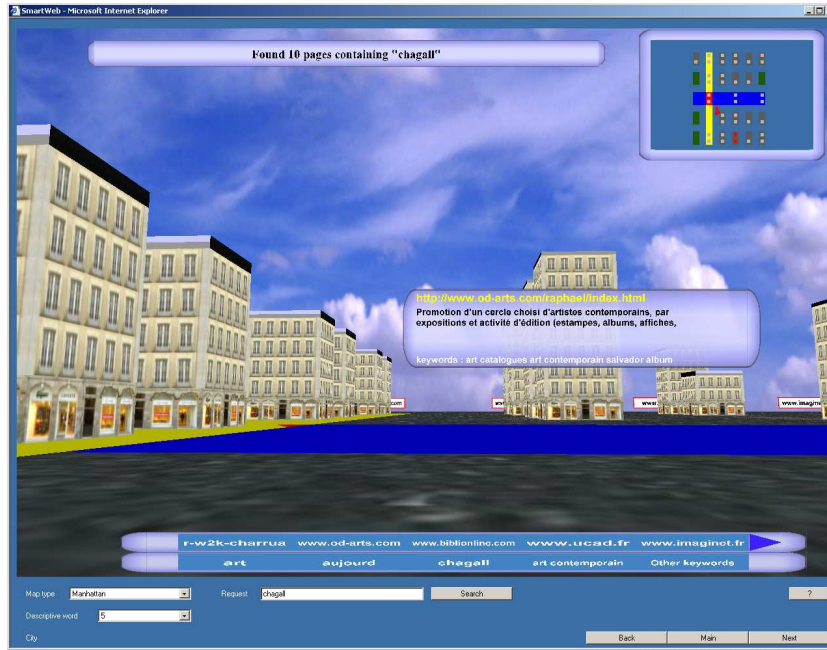


FIG. 3.3 – Métaphore de la ville

- **Taille.** La construction d'un tel environnement 3D n'est envisageable que pour des villes de petite taille, c'est-à-dire de l'ordre d'un millier de bâtiments. Cela limite donc le nombre de résultats de recherche sachant qu'un site apparaît une fois par mot-clé qu'il contient.
- **Temps.** Le nombre de bâtiments (*i.e.* le nombre de résultats et le nombre de mots-clés sélectionnés pour représenter l'axe associé) influe sur le temps de calcul de la ville. Quelques exemples de temps de calcul sont donnés dans le **tableau 3.2**.

Requêtes	Nombre de résultats	Temps d'attente (s)	Taille (Ko)
matisse	50	6	277
picasso	173	12	658
toile	423	34	1551
artistique	565	54	2043
artistique OR toile	982	88	3415

TAB. 3.2 – Temps de calcul de la ville 3D pour différentes requêtes et pour un nombre de mots-clés égal à six. On rappelle que dans le pire des cas le nombre de bâtiments peut être égal à six fois le nombre de résultats.

- **Navigation.** La redondance des sites ne rend pas forcément intuitive la navigation dans l'espace de résultat.
- **Mots-clés.** Afin de rendre les résultats plus interprétables, il serait intéressant

d'utiliser une analyse factorielle des correspondances (AFC) dans le but de regrouper certains mots-clés (cette analyse peut également se faire en amont, lors de la constitution de la liste des mots-clés relative à chaque document).

- **Scalabilité.** La visualisation manque de scalabilité du point de vue de la représentation de l'information. Il faudrait augmenter le niveau d'abstraction dans le cas où les résultats sont relativement nombreux.

Cependant, il reste un problème récurrent, à savoir qu'une visualisation dépend du nombre de résultats. Il est donc difficile de trouver une métaphore qui donne des résultats satisfaisants dans tous les cas de figure.

Aspect multimédia. Il est aussi envisagé d'avoir des informations sur le contenu multimédia des documents (images, vidéos, sons...). Dans un premier temps, on s'intéresse aux images. Une autre métaphore permet alors d'avoir un aperçu, sous la forme d'une galerie, des images contenues dans un site Web. Elle ne propose pas d'organisation particulière de ces images et elle offre deux modes de visualisation à l'utilisateur : un mode 2D et un mode 3D. Dans le premier, les images sont affichées sous forme réduite (vignettes) sur un plan 2D (**figures 3.4 et 3.5**). Les dimensions de ces vignettes restent proportionnelles à celles des images d'origine. De plus, les images considérées comme des icônes peuvent être filtrées. Les propriétés de cette visualisation 2D de la métaphore sont récapitulées dans le **tableau 3.3**. Le survol d'une vignette par le pointeur de la souris entraîne l'affichage d'informations contextuelles sur l'image associée (*e.g.* l'URL, la description, la taille et les dimensions). La position de cet affichage est calculée en fonction de celle du curseur de la souris sur l'écran. Par ailleurs, le support de la vignette survolée change de forme et de couleur. Un « clic gauche » de la souris sur une vignette de la visualisation 2D a pour conséquence de la faire apparaître en avant-plan (zoom). Il est alors possible de la supprimer de la représentation. Il est également possible de créer des signets d'images à l'aide du bouton central de la souris, et d'ouvrir une image dans une nouvelle fenêtre du navigateur Web en « double cliquant » sur la vignette correspondante. Concernant la visualisation 3D, les images sont désormais affichées dans une galerie 3D dans laquelle l'utilisateur peut se promener (**figure 3.6**). Les propriétés de cette visualisation 3D sont les mêmes que pour celle en 2D à l'exception de la suppression et de l'affichage en avant-plan.

3.2.2.2 Évolution

Suite à cette première version de la métaphore de la ville ainsi qu'aux problèmes identifiés, des modifications sont apportées à cette métaphore afin de la faire évoluer. La modification majeure concerne l'organisation des résultats. En effet, l'organisation selon les deux axes précédemment proposés est peu efficace et engendre trop de redondance d'information. Nous proposons désormais d'organiser les résultats selon leurs proximités sémantiques en utilisant l'approche des cartes auto-organisatrices présentée dans le chapitre précédent (*cf.* section 2.3). La carte auto-organisatrice est en fait une grille 2D que nous utilisons comme cadastre de notre ville (**figure 3.7**). En

action graphique	interaction associée
sélection d'une image	affichage de l'image en avant-plan (s'il n'y en a pas déjà une)
sélection de l'image en avant-plan	retour de cette image avec les autres sur le plan 2D
pointage d'une image	affichage d'informations contextuelles sur l'image (URL, description, dimensions...) et le support de l'image devient circulaire et de couleur rose (de façon temporaire)
« double-clic » sur une image	ouverture de l'image dans une nouvelle fenêtre du navigateur Web
« clic » sur le bouton de suppression qui apparaît en haut à droite de l'image lorsqu'elle est en avant-plan	suppression de l'image de la visualisation
« clic » avec le bouton central de la souris sur une image qui n'est pas déjà un signet	signet de l'image et le support de l'image devient circulaire et de couleur verte (de façon permanente)
« clic » avec le bouton central de la souris sur une image qui est déjà un signet	enlève le signet et le support de l'image redevient rectangulaire et de couleur bleue

TAB. 3.3 – Propriétés de la visualisation 2D

d'autres termes, le sol de la ville correspond à la carte auto-organisatrice. Cette nouvelle organisation des données supprime le problème de redondance d'informations constaté dans la version précédente de la métaphore. Les quartiers correspondent alors aux différentes unités de la carte auto-organisatrice (*i.e.* aux différents neurones).

Dans chaque quartier, les documents restent représentés par des bâtiments. Par contre, ces derniers sont modifiés. La texture des bâtiments n'est plus uniforme mais elle est désormais représentative du contenu de la page Web associée. En effet, les textures sont en fait les copies d'écran des différentes pages Web. Concernant la hauteur des bâtiments, elle ne représente plus la taille des pages Web mais leur pertinence à la requête. Il s'agit d'une « pertinence système » qui correspond au rang calculé par l'algorithme *PageRank* de *Google*, et dont la représentation utilise une approche sous la forme d'intervalles. Cela s'explique par le fait qu'une définition de la hauteur proportionnelle au rang ne permette pas de différencier visuellement des résultats avec des rangs proches et entraîne des confusions liées à la perspective de l'affichage. Trois valeurs sont alors proposées pour la hauteur des bâtiments. Les 10 premiers résultats correspondent aux bâtiments les plus hauts, les 20 suivants aux bâtiments de hauteur intermédiaire et les autres résultats aux bâtiments de petite taille. Par ailleurs le



FIG. 3.4 – Visualisation 2D des images contenues dans un site Web



FIG. 3.5 – Visualisation 2D des images contenues dans un site Web : sélection d'une image ayant pour conséquence l'affichage de sa vignette en avant-plan



FIG. 3.6 – Visualisation 3D des images contenues dans un site Web

deuxième toit, présent sur les bâtiments qui possèdent des images, a été supprimé. Il est désormais possible d'accéder à la représentation des images d'une page Web en « double cliquant » sur le bâtiment correspondant. Un dernier point sur les bâtiments concerne les favoris. L'utilisateur peut maintenant identifier les résultats qui font partie de ses favoris. Il s'agit des bâtiments possédant un drapeau rouge sur le toit. Les favoris sont des documents qui intéressent *a priori* l'utilisateur. Grâce à cette interface, il est alors possible d'identifier les résultats proches des favoris, ce qui risque potentiellement d'intéresser l'utilisateur.

Une autre modification concerne l'affichage contextuel des informations liées aux différents résultats. Dans la précédente version, les informations liées à plusieurs pages Web ne peuvent pas être vues simultanément. Il est alors difficile de comparer deux réponses et cela nuit à l'exploitation rapide des résultats. Désormais, l'affichage contextuel apparaît à une position fixe dans l'interface et il est accompagné de l'affichage d'informations sur quatre résultats proches sémantiquement (*i.e.* voisins sur l'interface). Il est aussi possible de stocker un affichage en mémoire dans un but de comparaison ultérieure. L'affichage contextuel lié à un résultat est également enrichi par des mots-clés qui correspondent aux premières étiquettes de l'unité de la carte auto-organisatrice contenant ce résultat.

L'organisation des données étant basée sur une carte auto-organisatrice, les résul-

tats d'un même quartier sont normalement similaires d'un point de vue statistique. Ils ont donc de fortes chances de traiter d'informations similaires. Un post-traitement de la carte auto-organisatrice (basé sur une CAH) est réalisé, ce qui permet de regrouper visuellement les quartiers contenant des informations similaires et de naviguer à un niveau plus abstrait dans les données. Chaque regroupement de quartiers (*i.e.* de neurones) est associé à une couleur qui est affichée sur le sol des quartiers. Des informations sur ces regroupements sont données dans les barres situées en bas de l'interface. Chaque barre est associée à la couleur du regroupement qu'elle représente et elle contient une liste de mots-clés représentatifs de ce regroupement. Différents niveaux d'abstraction sont possibles pour ces regroupements (*i.e.* différentes coupures dans le dendrogramme issu de la CAH). Il serait alors intéressant d'ajouter à l'interface une fonctionnalité permettant à l'utilisateur de naviguer librement dans ces différents niveaux d'abstraction sur les résultats. L'objectif est d'aider l'utilisateur à identifier les différents thèmes liés à sa requête et de lui permettre de naviguer plus rapidement dans l'ensemble des résultats. Une autre fonctionnalité serait bien entendu de proposer à l'utilisateur des possibilités de filtrage des résultats (par exemple selon leur thématique).

Ne faisant pas réellement partie des objectifs de nos travaux, la métaphore de la galerie n'est pas modifiée. Par contre il est désormais possible de l'utiliser pour visualiser les résultats de recherche (**figures 3.8(a)** et **3.8(b)**). Cette utilisation ne diffère toutefois pas de celle d'un affichage linéaire d'une liste de plusieurs centaines de résultats, à l'exception de la présence d'une vue d'ensemble des résultats.

Les évolutions proposées dans cette partie correspondent à la version de la métaphore actuellement implémentée dans le prototype présenté dans le chapitre suivant.

3.2.3 Navigation

3.2.3.1 Première version

La navigation est un point essentiel qui doit permettre d'utiliser efficacement les métaphores développées. Les possibilités offertes pour la navigation évoluent selon la métaphore utilisée. Dans la métaphore de la ville, la navigation est peu contrainte et offre une liberté totale de déplacement dans la scène 3D. Cependant cela n'est pas intuitif et aisé pour un utilisateur novice des environnements 3D (*e.g.* l'utilisation du clavier et de la souris, la sensibilité et la vitesse des déplacements). Des points de vue sont offerts à l'utilisateur pour faciliter sa navigation ainsi que des barres de navigation qui permettent de se positionner au départ de la rue désirée. On a également un plan 2D de la ville qui permet à l'utilisateur de se situer dans cet univers qui peut parfois être assez vaste. À l'inverse, dans la métaphore de la galerie, on dispose d'une navigation plus contrainte qui semble plus intuitive et facile pour un utilisateur novice. Cela est d'autant plus vrai pour la visualisation 3D. En effet, afin que l'utilisateur ne se perde pas dans la galerie 3D, ses déplacements se font à travers un chemin central

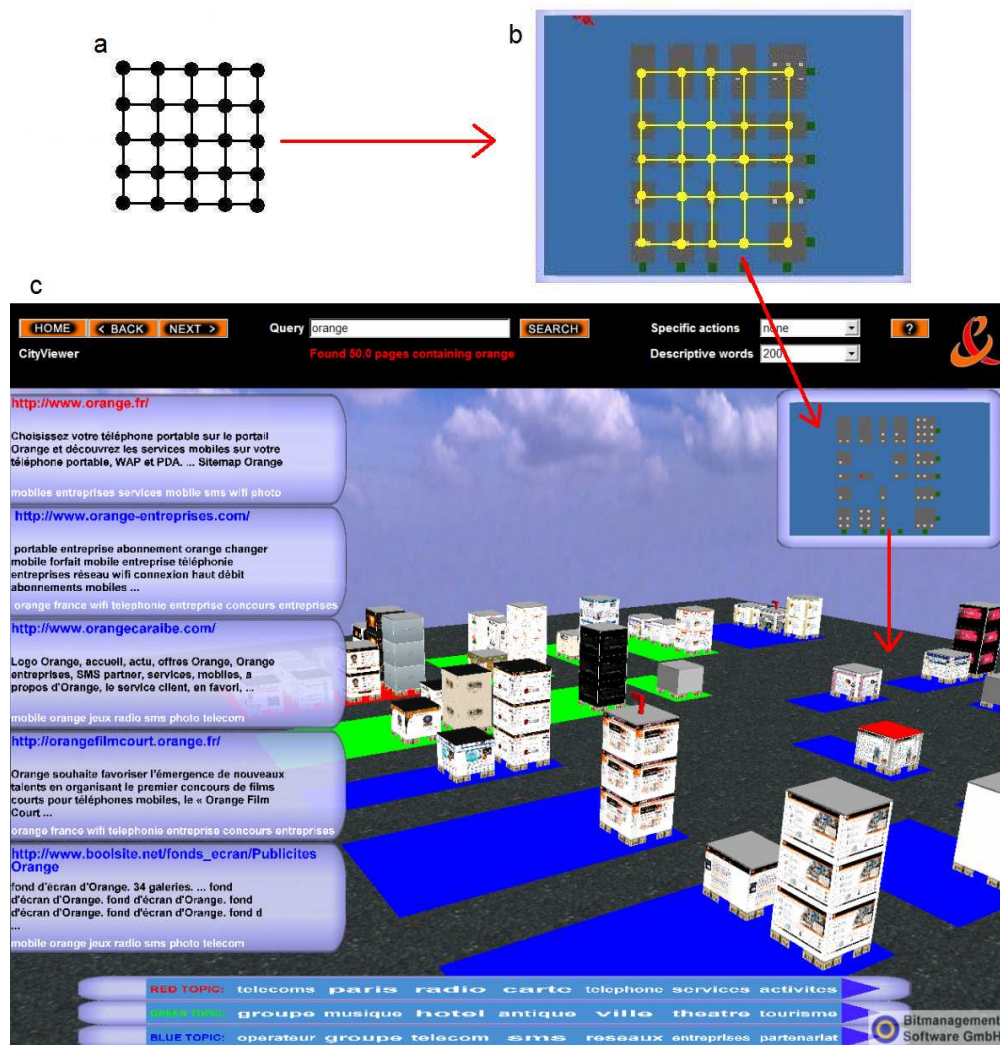
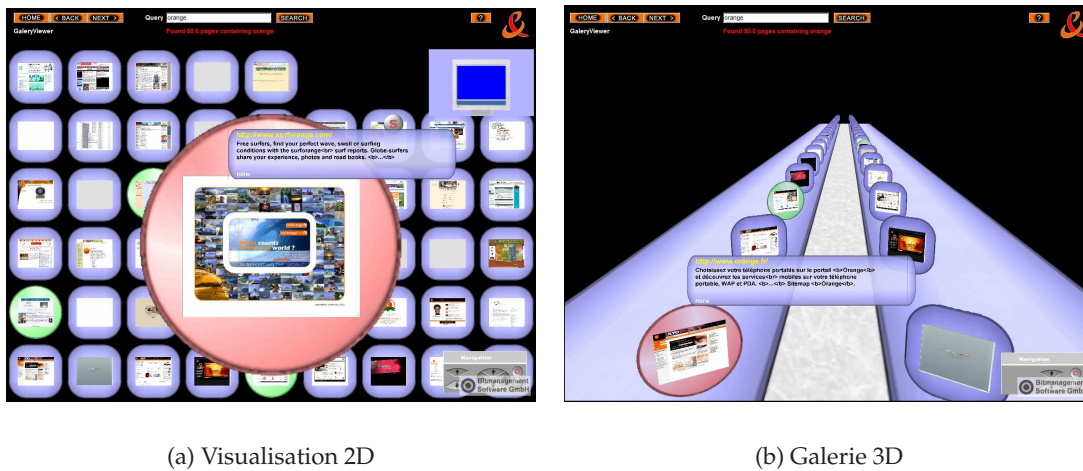


FIG. 3.7 – Différents niveaux de la métaphore de la ville : la carte auto-organisatrice précédemment calculée (a) est affichée sur le sol de la ville, comme cela apparaît sur la vue de dessus (b) ou directement sur la scène 3D (c).

qu'il ne peut quitter que pour prendre de l'altitude (pour avoir une vue plus globale). De même, les angles de rotation sont limités à certaines valeurs suffisantes pour une visualisation correcte des images.

3.2.3.2 Évolution

Dans la seconde version de la métaphore de la ville, le plan 2D est désormais interactif. Il est donc possible de se déplacer rapidement à n'importe quel quartier de la ville en cliquant dessus dans le plan 2D. Cela se traduit par le déplacement, dans la



(a) Visualisation 2D

(b) Galerie 3D

FIG. 3.8 – Visualisation des résultats de recherche avec la métaphore de la galerie

scène 3D, de l'utilisateur qui se retrouve alors devant le quartier sélectionné. L'organisation des résultats proposée dans cette seconde version de la métaphore de la ville est aussi un avantage en termes de navigation. En effet, la proximité visuelle (dans l'environnement 3D) des documents similaires permet de minimiser les déplacements de l'utilisateur dans certains cas. Dans cette seconde version, et en partie à la demande des utilisateurs, il est possible d'entrer dans les bâtiments qui possèdent une ouverture sur chaque côté. Pour l'instant aucune information supplémentaire sur le résultat n'est fournie à l'intérieur du bâtiment (les textures à l'intérieur sont identiques de celles de l'extérieur). Mais il est envisagé d'y ajouter de nouvelles informations sur les résultats. Par ailleurs, nous ne cherchons pas à garder le réalisme du monde physique mais à aider l'utilisateur dans sa tâche. Aussi, une fois à l'intérieur d'un bâtiment, le toit est transparent et permet ainsi de garder dans certains cas des informations sur le contexte (*i.e.* une vue sur le reste de la ville). Et, dans le but de simplifier la navigation, il est possible de sortir du bâtiment par le toit. Il s'agit juste là d'un exemple pour montrer que l'objectif est de faciliter la tâche de l'utilisateur et non de représenter fidèlement la réalité. Dans un environnement 3D virtuel, nous avons en effet la possibilité de s'affranchir des contraintes du monde réel et ainsi de proposer des métaphores plus riches [Shneiderman 2003a].

3.2.3.3 Améliorations

Il faut garder comme principal objectif la simplification des déplacements dans la scène 3D afin d'aider l'utilisateur à accomplir sa tâche. Une idée consiste à contraindre davantage la navigation dans la ville pour que l'utilisateur novice dans la navigation en 3D ne puisse pas se perdre dans l'environnement (*cf.* la navigation dans la métaphore de la galerie). Une mise en œuvre possible de cette idée est de limiter les déplacements de l'utilisateur aux différentes rues de la ville. L'affichage d'informations

d'aide à la navigation sur le sol de la ville est aussi un autre moyen d'aider l'utilisateur à se repérer. Par ailleurs il est utile, dans un espace 3D, de proposer différents points de vue à l'utilisateur. Plusieurs points de vue prédéfinis sont déjà disponibles dans notre métaphore (*e.g.* une vue de dessus et une vue de face de la ville). Il serait alors intéressant de donner à l'utilisateur la possibilité de définir des points de vue dynamiques (*i.e.* des signets de positions dans l'environnement 3D en quelques sortes).

3.3 Discussion

Dans cette section, nous identifions un ensemble de critères à prendre en compte lors de la conception de métaphores 3D dans notre cadre applicatif. Puis nous dressons un bilan de notre métaphore de ville virtuelle.

3.3.1 Critères spécifiques aux métaphores 3D

L'affichage des résultats de recherche dans un espace 3D implique la prise en compte d'un ensemble de critères spécifiques. En général, il n'est pas évident de prendre en compte tous ces critères de façon optimale. Un compromis doit donc être trouvé. L'objectif est alors d'identifier les principaux critères spécifiques à l'utilisation d'interfaces 3D pour la visualisation de résultats de recherche, afin de proposer une aide à la construction de telles interfaces. Il faut également préciser que le problème des occultations n'est pas mentionné explicitement ici, étant donné qu'il s'agit d'un problème récurrent, présent dans plusieurs critères tels que le placement des résultats, la navigation ou encore le choix de la métaphore. L'attitude à adopter consiste généralement à minimiser ou prioriser ces occultations, ainsi qu'à proposer différents points de vue. Les critères que nous avons identifiés sont présentés dans les paragraphes suivants. Il existe néanmoins d'autres formulations de recommandations relatives à la construction d'interfaces 3D d'un point de vue plus général [Shneiderman 2003b].

Choix de la métaphore. Il existe une grande liberté dans le choix de la métaphore. Cependant certaines métaphores sont préférées pour leur aspect cognitif, telles que les métaphores de livre, de bibliothèque, de ville ou plus généralement les métaphores géographiques. Ces dernières permettent notamment de fournir une cartographie de l'information en proposant à l'utilisateur une métaphore qu'il connaît déjà. Le choix de métaphores appartenant au monde réel pose cependant un problème : celui du réalisme du rendu. Ce piège doit être évité en proposant justement une visualisation respectant les principes de la métaphore sans pour autant chercher à ressembler graphiquement à la réalité. En effet, plus la visualisation tente d'être réaliste, plus l'utilisateur identifie des incohérences qui le détournent de son but initial qu'est la recherche d'information. La métaphore doit donc être au service des fonctionnalités et non à la recherche du réalisme.

Placement des résultats. Deux types de placement se distinguent principalement. Le premier consiste à positionner les résultats dans l'espace en utilisant les 3 dimensions. Les distances entre les résultats doivent donc être évaluées par l'utilisateur dans un espace tridimensionnel. Or il s'avère que cette évaluation n'est pas triviale et nécessite souvent d'avoir plusieurs points de vue. C'est pourquoi le second type de placement est plus apprécié. Il consiste à placer les résultats sur un plan 2D. La troisième dimension n'est alors pas nécessaire pour évaluer les distances entre les résultats, ce qui rapproche l'utilisateur du monde réel, notamment pour certaines métaphores comme celle de la ville. En effet, se déplacer dans une ville revient à se déplacer au sol (information 2D) et ensuite la troisième dimension est éventuellement utilisée. L'idée est donc de respecter cette habitude en se servant de la troisième dimension uniquement pour représenter une information supplémentaire telle que la pertinence, mais pas une information relative au placement.

Navigation. Cet aspect pose généralement moins de problèmes pour les interfaces 2D où la navigation est restreinte aux déplacements 2D et au zoom. Le passage à une interface 3D donne la possibilité à l'utilisateur de se déplacer selon le troisième axe, ce qui permet alors une immersion de l'utilisateur dans l'espace des résultats (et augmente le phénomène d'occultations). Les interfaces 3D offrent aussi généralement plusieurs modes de navigation (tels que le survol ou la marche au sol), ainsi que la possibilité d'effectuer des rotations. L'absence d'interfaces 3D grand public, excepté les jeux, rend l'utilisateur peu familier avec les environnements 3D et avec leur manipulation (navigation et interaction). De plus les moyens d'interaction ne sont pas vraiment adaptés à la navigation 3D (souris, écran), comme en témoigne l'utilisation de la souris proposée par les navigateurs VRML. Des travaux proposant d'autres types d'interaction pour la navigation ont été réalisés, tels que [Sparacino *et al.* 2002] pour la navigation dans une ville, mais sans réel succès pour l'instant. De plus, il est très probable que la mise en œuvre de nouveaux moyens d'interaction nécessitera de nouveaux équipements pour les ordinateurs actuels. Il faut aussi préciser que ces moyens d'interaction ne doivent pas se traduire par des contraintes trop fortes sur les utilisateurs pour être acceptés.

Position initiale. Il faut placer l'utilisateur dans l'espace des résultats. Ce placement peut se faire à l'intérieur ou à l'extérieur de la scène ; cependant le deuxième choix permet d'avoir une vue d'ensemble des résultats. Le choix peut aussi être guidé par la thématique la plus importante, les résultats les plus pertinents, la meilleure position pour un aperçu de l'ensemble des résultats, la présence de signets dans les résultats. . .

Immersion. Une interface 3D peut proposer une immersion de l'utilisateur dans la scène ou alors se contenter de lui proposer une manipulation 3D de l'extérieur de la scène. L'immersion de l'utilisateur dans l'environnement 3D entraîne rapidement une perte de repères. Il faut alors l'aider à se repérer en lui proposant, par exemple, sa position sur un plan 2D ou encore des points de repères dans la scène.

Performances. Les performances de la 3D ne représentent plus aujourd’hui un handicap au choix de ce type d’interface. Par exemple, dans notre cas, l’organisation des résultats est une tâche plus coûteuse que la génération et l’affichage de l’interface 3D. Par contre, il faut faire attention à la taille du fichier contenant l’environnement 3D, plus particulièrement lorsque sa création se fait sur le serveur. Le cadre applicatif n’impose pas d’avoir des scènes 3D réalistes, ce qui permet de sous-exploiter les capacités de la 3D et donc d’avoir des performances acceptables.

Utilisateur. La prise en compte de l’utilisateur n’est pas propre à la visualisation 3D de résultats mais il s’agit plutôt d’un critère général à toute IHM. Dans notre cas, il faut que tous les choix en matière d’affichage répondent à une attente de l’utilisateur. Par ailleurs, il faut également tenir compte de la catégorie de l’utilisateur, c’est-à-dire de l’ensemble de ses capacités, afin d’adapter la visualisation. Le respect de ce critère s’évalue généralement par des tests utilisateurs (*cf.* chapitre 5).

3.3.2 Bilan

Nous proposons dans ce chapitre une métaphore de visualisation basée sur la ville pour représenter des résultats de recherche préalablement organisés par le calcul d’une carte auto-organisatrice. Ce choix de métaphore cognitive est adaptée pour visualiser des données organisées et elle permet d’exploiter la mémoire spatiale de l’utilisateur. Elle améliore l’appréhension des résultats, grâce notamment à une vue d’ensemble des relations de proximité entre les différents documents. Par ailleurs, la taille de la ville reste suffisamment petite pour ne pas perdre l’utilisateur dans l’espace 3D. En effet, il ne faut pas transformer le problème de la perte de l’utilisateur dans l’hyper-espace par celui de la perte de l’utilisateur dans l’environnement 3D. Ainsi l’expression « *getting lost in hyperspace problem* » ne doit pas devenir « *getting lost in 3D environment problem* » et encore moins « *getting lost in hyperspace and in 3D environment problem* ».

Il reste néanmoins certains points à améliorer dans la version actuelle de la métaphore afin de la rendre encore plus efficace. Par exemple, les différents niveaux d’abstraction ne sont pas exploités à leur maximum. Plus de détails sur les résultats peuvent être fournis, notamment en utilisant l’intérieur des bâtiments. D’un autre côté, les différents regroupements de quartiers n’apparaissent pas suffisamment clairement à l’utilisateur. Il faudrait alors représenter visuellement cette abstraction de manière plus efficace en réfléchissant, entre autres, à de nouvelles primitives visuelles. Par ailleurs, ces regroupements sont caractérisés par des mots-clés. On peut rendre cet étiquetage plus riche en y ajoutant des informations sur quelques résultats représentatifs de ces regroupements. Cette approche est déjà utilisée dans le système *Grouper*. Dans notre cas, cela permettrait d’aider l’utilisateur à choisir les zones qui l’intéressent dans la ville. Cette caractérisation de groupes de résultats à l’aide de documents représentatifs peut aussi être proposée au niveau des différents quartiers.

Nous venons de définir dans ce chapitre la métaphore de la ville qui est utilisée pour représenter des résultats de recherche. Il est donc désormais possible de mettre en œuvre nos propositions en matière d'organisation et de visualisation de résultats de recherche. Cela est fait dans le prototype *SmartWeb* présenté dans le chapitre suivant.

Chapitre 4

Prototype SmartWeb

Dans ce chapitre, nous présentons le prototype *SmartWeb* qui met en œuvre nos propositions faites dans les deux chapitres précédents. Différents points sont traités, allant de l'architecture à l'utilisation pratique en passant par les caractéristiques du prototype.

4.1 Présentation générale

SmartWeb est un prototype développé par la Division R&D de France Télécom. L'objectif de ce prototype est de proposer de nouvelles méthodes de visualisation de résultats de recherche, ainsi que de nouveaux types d'interactions. SmartWeb intègre donc l'ensemble des solutions proposées concernant l'organisation (cf. **chapitre 2**) et la visualisation (cf. **chapitre 3**) des résultats de recherches sur le Web. Bien que le prototype permette de proposer de nouvelles méthodes de visualisation indépendamment du nombre de dimensions utilisées, nous nous intéressons plus particulièrement sur les conditions à réunir pour offrir une navigation efficace en 3D dans les résultats de recherches effectuées sur le Web.

On rappelle le contexte actuel de la restitution des résultats de recherche. Une recherche sur le Web retourne généralement un grand nombre de réponses qui sont affichées partiellement et linéairement sous la forme d'une liste. Cette dernière est composée d'informations essentiellement textuelles (e.g. URL, titre, *snippet*). Ces informations peuvent être plus ou moins riches selon le moteur de recherche utilisé. De plus, cette liste de résultats est triée selon un critère de « pertinence système », généralement basé sur la popularité des résultats (e.g. algorithme *PageRank* de Google). On a donc, en résultat, une liste de grande taille affichée sur plusieurs pages. Ce type d'organisation des résultats ne permet pas à l'utilisateur d'exploiter correctement les résultats de sa recherche. Par exemple, ce type de restitution des résultats souffre de l'absence de vue d'ensemble, ainsi que de l'impossibilité de « filtrer » efficacement les résultats d'un point de vue visuel. Cela explique sûrement l'émergence de nouveaux concepts organisant l'information sous la forme cartographique, en deux dimensions

(e.g. *KartOO* ou *MapStan Search*) ou, plus rarement, en trois dimensions (e.g. *Antartica*). Cependant, les solutions proposées jusqu'ici ne traitent pas vraiment l'organisation des résultats de façon efficace, et les interfaces 3D sont plus rarement exploitées.

Une première maquette, nommée « 3Dsearch », a été réalisée par l'université d'économie de Poznan (PUE¹). Elle met en avant la possibilité de construire dynamiquement des contenus 3D en se basant sur un nouveau méta-langage, le X-VRML. Cette première maquette est le point de départ du prototype SmartWeb.

4.2 Architecture

4.2.1 Architecture d'un point de vue conceptuel

L'architecture conceptuelle du prototype est présentée sur la **figure 4.1**. Il s'agit d'une architecture relativement classique, que nous allons expliquer en respectant l'ordre chronologique de l'utilisation du prototype par l'utilisateur.

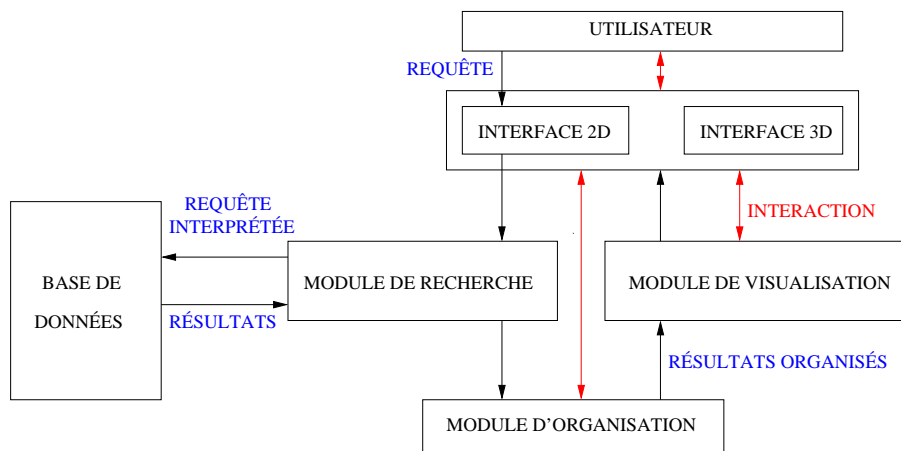


FIG. 4.1 – Schéma conceptuel du prototype

Tout d'abord, l'utilisateur entre sa requête (booléenne ou non) *via* l'interface 2D. Cette requête est alors transmise au **module de recherche** qui crée la requête SQL² associée à requête de l'utilisateur. Puis le module de recherche interroge la **base de données**. On précise que la base de données utilisée dans le cadre de ce prototype est statique, c'est-à-dire qu'on ne prend pas en compte l'évolution des documents dans le temps (*i.e.* l'aspect temporel ou dynamique). La gestion et les caractéristiques de la base de données sont expliquées dans la suite. Ensuite, le module de recherche récupère l'ensemble des résultats et les stocke dans une structure de données interne

¹Poznan University of Economics

²Structured Query Language

adaptée pour une réutilisation efficace de ces résultats par les autres modules. Ces résultats passent alors par le **module d'organisation** qui permet d'organiser les résultats indépendamment des méthodes de visualisation utilisées par la suite. Puis, une fois organisés, les résultats sont envoyés au **module de visualisation** qui va définir la métaphore de visualisation à utiliser. C'est donc dans ce module qu'est réalisée l'association entre les paramètres graphiques de la visualisation et les informations sur les résultats. Enfin, les informations sont transmises à l'**interface** (2D pour les informations de contrôle et 3D pour les résultats de recherche). Dans cette architecture conceptuelle, ce sont les modules d'organisation et de visualisation qui nous intéressent particulièrement, ainsi que les interactions possibles avec l'interface utilisateur.

4.2.2 Architecture d'un point de vue implémentation

Cette partie s'intéresse à l'architecture relative à l'implémentation du prototype. Ainsi la **figure 4.2** présente l'architecture simplifiée de l'application et la **figure 4.3** propose, de façon plus complète, le schéma d'implémentation du prototype. Dans la suite de cette sous-section, nous donnons des explications à propos de la **figure 4.3**, sur laquelle on distingue les modules stockés sur le serveur de ceux exécutés sur le poste client.

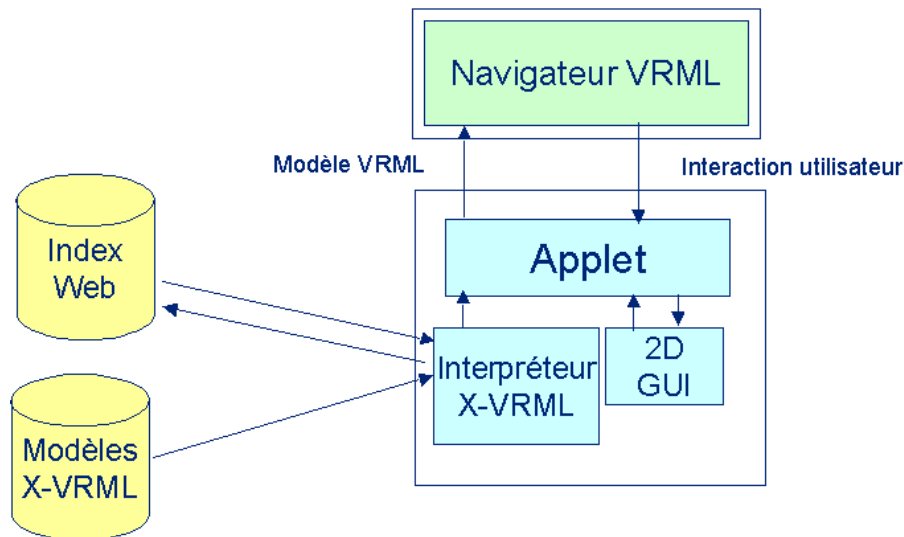


FIG. 4.2 – Architecture de l'application

4.2.2.1 Partie serveur

Côté serveur, la mise en œuvre de ce prototype nécessite la présence de deux serveurs : un serveur de base de données et un serveur Web. Ce dernier contient les différents modèles d'interfaces ainsi que le code (fichiers binaires correspondant à

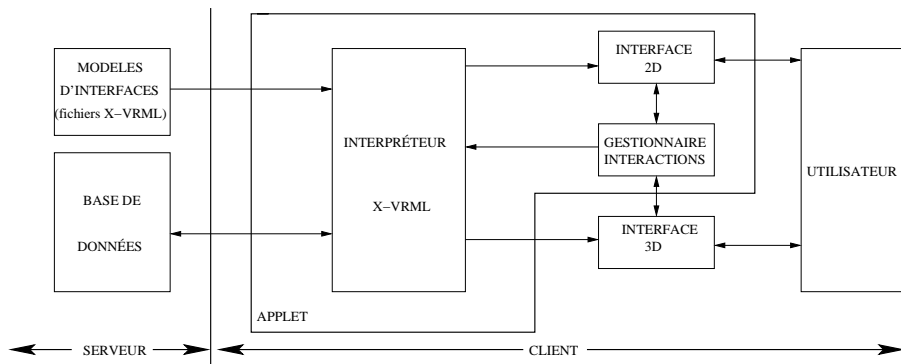


FIG. 4.3 – Schéma d'implémentation du prototype

l'interpréteur X-VRML, à l'interface 2D ainsi qu'au gestionnaire des interactions) à télécharger et à exécuter sur le poste client.

La base de données. Initialement la base de données était gérée par un serveur *Oracle*. Dans la version actuelle du prototype, les bases de données de test sont gérées par *MySQL*. La création de la requête SQL par le module de recherche diffère légèrement suivant le type de base de données (*Oracle* ou *MySQL*). Cependant, ces deux types de bases de données sont gérés par le prototype. Il suffit de préciser le type de base utilisé dans le fichier de paramètres du prototype. La base de données contient les descripteurs classiques³ des pages Web (*e.g.* le titre, l'URL, le *snippet*). Ces descripteurs sont alors tous théoriquement accessibles *via* le prototype. Mais, en pratique, certains descripteurs peuvent ne pas être utilisés par le module de visualisation. La description du type de base de données utilisée est proposée en **annexe C**. La base de données se trouvant côté serveur, cela implique d'avoir un ordinateur jouant le rôle de serveur de base de données. Dans le cadre de ce prototype, les différentes bases de données de test sont remplies en utilisant l'API⁴ fournie par *Google*.

Les modèles d'interfaces. Côté serveur, différents modèles d'interfaces, correspondant à différentes métaphores de visualisation, sont disponibles. Chacun de ces modèles est décrit par un ou plusieurs fichiers au format X-VRML, le plus souvent accompagnés de fichiers au format WRL⁵ (dans lesquels sont définis des prototypes externes utilisés dans le code X-VRML). Ces fichiers, définissant les modèles d'interfaces, sont stockés dans un répertoire du serveur Web. Dans le cas de notre prototype SmartWeb, nous utilisons un serveur Web *Apache*. Actuellement, la sélection du modèle d'interface se fait manuellement parmi toutes les métaphores de visualisation disponibles. En d'autres termes, c'est l'utilisateur qui choisit la métaphore au moment où il exprime

³c'est-à-dire ceux utilisés par la majorité des moteurs de recherche

⁴*Application Programming Interface*

⁵L'extension WRL correspond à des fichiers contenant du code VRML.

sa requête. Cette sélection pourrait cependant aussi se faire de manière automatique ou semi-automatique. Dans le cas d'une sélection automatique, c'est le prototype qui impose la métaphore de visualisation en fonction des caractéristiques de la requête et des résultats. Dans le cas d'une sélection semi-automatique, le prototype fait une présélection dans les modèles d'interfaces disponibles et le choix final, dans cette présélection, se fait par l'utilisateur. Les fichiers décrivant les modèles d'interfaces sont stockés sur le serveur, mais les fichiers correspondant à la métaphore de visualisation sélectionnée seront transférés sur le poste client.

4.2.2.2 Partie client

D'un point de vue pratique, la partie client doit juste supporter l'interface utilisateur qui se présente sous la forme d'une *applet Java* et d'un navigateur VRML.

Applet Java (interpréteur X-VRML, interface 2D et gestionnaire d'interactions). L'utilisateur accède au prototype *via* une URL qui pointe sur le serveur. C'est alors que le code *Java* (applet et interpréteur X-VRML), stocké sur le serveur Web avec les modèles d'interfaces, est transmis au poste client afin d'y être exécuté. Le poste client doit donc être muni d'un environnement d'exécution *Java* (JRE⁶) compatible avec la version *Java* utilisée pour développer le prototype. Mais cela est le cas de la majorité des ordinateurs. Pour exemple, nous utilisons actuellement la *Microsoft Java Virtual Machine* (MSJVM) incluse dans *Internet Explorer* (pour les systèmes d'exploitation *Windows XP* antérieurs à l'installation du *Service Pack 2*). Une fois l'*applet* chargée et le modèle d'interface sélectionné (dans l'interface 2D de l'*applet*), les fichiers correspondants sont téléchargés sur le serveur et passés à l'*applet*. Ainsi l'interpréteur X-VRML (écrit en *Java*) peut interpréter le code X-VRML et générer dynamiquement un fichier VRML qui est stocké dans un espace temporaire du poste client (C:\Temp\vv_res.vrml). Ce fichier VRML peut être de taille plus ou moins grande suivant le nombre de résultats pris en compte. Il est écrasé à chaque nouvelle requête de l'utilisateur. Aucune gestion de cache n'est gérée dans la version actuelle du prototype. Les informations de l'interface 2D sont affichées par l'*applet Java*.

Interface 3D. Elle correspond au navigateur VRML. Le fichier VRML (généré par l'interpréteur X-VRML) est interprété par le navigateur VRML afin d'afficher les informations contenues dans l'interface 3D. Initialement, le navigateur utilisé est *Blaxxun Contact*. Cependant, il est désormais possible d'utiliser son « successeur » *BS Contact VRML* (proposé par l'entreprise *Bitmanagement Software GmbH*). Le prototype est dépendant de l'utilisation d'un de ces deux navigateurs, car nous utilisons leur API pour réaliser la communication entre le navigateur VRML et le code *Java*. Le navigateur permet d'afficher la scène VRML générée par l'*applet Java*. L'installation de l'un ou l'autre

⁶Java Runtime Environment

de ces navigateurs VRML se fait très simplement, sous la forme de *plugs-in* du navigateur Web. Il s'agit là d'une contrainte sur la configuration du poste client, afin de pouvoir utiliser le prototype.

Remarque. Dans l'architecture proposée, l'interprétation du fichier X-VRML se fait sur le poste client. Cependant, il est envisageable de réaliser cette opération sur le serveur (*via* une *servlet*). D'un côté, cette modification de l'architecture permet d'alléger la charge de travail sur le poste client et de diminuer la taille du code *Java* à télécharger sur le serveur (il n'y a plus besoin de télécharger la partie concernant l'interpréteur X-VRML). D'un autre côté, cette solution ajoute le transfert, à chaque nouvelle requête, du fichier WRL (issu de l'interprétation du code X-VRML) du serveur vers le client. Cela peut poser problème si le nombre de résultats pris en compte pour la requête est important (*i.e.* on a un fichier WRL de grande taille, pouvant aller jusqu'à quelques méga-octets).

4.3 Caractéristiques du prototype

4.3.1 Langage X-VRML

Le X-VRML est un langage co-développé et co-breveté par France Télécom R&D et l'université d'économie de Poznan. Il s'agit d'un langage à balises, basé sur XML, permettant d'enrichir les possibilités du VRML. En quelques sortes, le X-VRML peut être assimilé à un méta-langage de plus haut niveau que le VRML. Dans le cas du prototype SmartWeb, c'est le fichier (ou modèle) X-VRML qui définit tous les aspects de la restitution des résultats, tels que les requêtes à la base de données, les traitements des requêtes, l'organisation et la présentation des résultats, la navigation, les interactions. Un extrait de fichier X-VRML est proposé en **annexe C**.

4.3.1.1 Apports du X-VRML

Cette partie présente les apports majeurs du langage X-VRML par rapport au VRML. En effet, ce nouveau méta-langage offre des fonctionnalités plus riches, qui n'existent pas dans le VRML. Ainsi, avec le X-VRML, il est possible :

- de réaliser une connexion et des requêtes à une base de données. Ces deux actions sont rendues possibles grâce à la définition de deux balises (ou *tags*) : `SW_CONNECT` et `SW_QUERY`.
- de faire des itérations sur une structure de données grâce à la balise `FTRD_UTIL`.
- de réaliser des actions spécifiques telles que l'organisation des données avec la balise `SW_CITY_MAP` et le chargement de nouvelles métaphores grâce à la balise `<listener_action>` (avec une persistance des informations).
- de définir une interface 2D grâce aux balises `<model_interface>` et `<component>`.

4.3.1.2 Spécifications du X-VRML

Le langage X-VRML est basé sur XML, c'est-à-dire qu'il s'agit d'un langage de balises et que chaque fonction ouverte doit être explicitement fermée. Ainsi, nous distinguons deux types de balises dans le X-VRML :

- les commandes englobant d'autres actions. Elles ont alors une balise d'ouverture et une balise de fermeture. Par exemple, on peut citer la commande permettant de se connecter à une base de données : `<SW_CONNECT>` (balise d'ouverture) et `</SW_CONNECT>` (balise de fermeture). La formulation générique est donc la suivante : `<name>` pour une balise d'ouverture et `</name>` pour une balise de fermeture.
- les commandes n'englobant pas d'autres actions. Elles ont alors une seule balise qui sert d'ouverture et de fermeture. Par exemple, on peut citer la commande permettant d'envoyer une requête à une base de données : `<SW_QUERY . . . />`. La formulation générique est donc la suivante : `<name . . . />`.

Chaque balise du fichier X-VRML correspond à une méthode *Java*. Un extrait du fichier X-VRML définissant la métaphore de la ville est donné en **annexe C**.

4.3.1.3 Comparaison avec X3D et Java 3D

Qu'est ce que le X3D ? X3D⁷ est un standard libre pour la distribution de contenus 3D. Ce n'est pas une API de programmation, ni un simple format de fichier pour l'échange de la géométrie. Il combine à la fois les descriptions comportementales géométriques et de temps d'exécution dans un seul fichier qui a un nombre de différents formats de fichier disponible pour cela, incluant le XML. C'est la prochaine révision de la spécification ISO VRML97, incorporant les dernières avancées dans les caractéristiques du matériel graphique commercial aussi bien que les améliorations architecturales basées sur des années de retours venant de la communauté de développement VRML97. X3D est donc un format de fichier 3D, compatible XML et de standard libre, qui permet la communication temps réel de données 3D à travers toutes les applications (y compris les applications réseau). Il a un ensemble riche de caractéristiques (propriétés) qui permettent son utilisation dans la visualisation technologique et scientifique, la CAO et l'architecture, la visualisation médicale, l'apprentissage et la simulation, le multimédia, le divertissement, l'éducatif. . . X3D est un standard considérablement plus mature et raffiné que son prédécesseur VRML, ainsi les auteurs peuvent réaliser les comportements qu'ils attendent. Plus d'informations sur les raisons d'utiliser X3D (basé XML et présenté comme un meilleur choix de développement que VRML) sont disponibles à l'adresse http://www.web3d.org/x3d/x3d_vs_vrml.html (les 10 meilleures raisons de migrer).

Le langage X-VRML a été créé parce que le VRML est trop pauvre pour satisfaire nos besoins particuliers tels que l'envoi de requêtes à une base de données, l'organisa-

⁷*eXtensible 3D* — <http://www.web3d.org> (*Web 3D Consortium*)

tion des données, les interactions, les itérations... Ainsi, le X-VRML permet d'ajouter de nombreuses fonctionnalités. De plus le X-VRML est extensible et il est possible d'ajouter facilement des nouvelles balises (ou *tags*) qui doivent être définies en *Java*. Cependant le X-VRML a aussi quelques « inconvénients ». Par exemple, l'interpréteur X-VRML est écrit en *Java*. Ce choix permet d'être indépendant de la plate-forme utilisée, mais il n'optimise pas l'efficacité en termes de temps d'exécution. Et la version actuelle semble être dépendante du navigateur VRML, en raison de l'API qui permet la communication entre le code *Java* et le navigateur VRML. Ainsi, comme le X3D est plus riche en caractéristiques (propriétés) que le VRML, est-il possible d'utiliser le X3D pour résoudre nos problèmes ? Malheureusement, la spécification actuelle du langage X3D n'est pas suffisante pour répondre à tous nos problèmes. Cependant, une nouvelle réflexion basée sur les versions courantes de X3D et de X-VRML serait intéressante pour identifier les points clés de ces deux langages dans notre cadre particulier d'application. Et alors, il serait possible d'envisager deux solutions :

- inclure certaines nouvelles fonctionnalités dans le langage X3D,
- présenter X-VRML comme un langage complémentaire à X3D (évidemment, cela ne pose aucun problème de modifier X-VRML pour remplacer la génération de fichiers au format VRML par la génération de fichiers au format X3D).

Dans notre cas particulier, les données visualisées sont non géographiques et abstraites. Le langage 3D utilisé (X-VRML, X3D ou d'autres) doit donc prendre ce point en compte. Cela peut être fait avec l'intégration de fonctionnalités spéciales orientées interface (comme une bonne gestion de l'affichage textuel dans une scène 3D). L'objectif est d'être capable de proposer des interfaces 3D intuitives pour visualiser des données abstraites.

Une autre possibilité pour construire des interfaces 3D est l'utilisation de l'API *Java 3D*⁸. Cette API est en fait une hiérarchie de classes *Java* permettant de concevoir des interfaces 3D. D'un premier abord⁹, cette solution semble envisageable pour développer nos métaphores de visualisation de résultats de recherche. Elle possède néanmoins certains inconvénients. D'abord le code doit être compilé contrairement au X-VRML qui est juste interprété. Ensuite, le *Java 3D* ne permet pas d'exprimer une métaphore 3D aussi simplement qu'en X-VRML. Ce dernier offre une grande liberté dans la création des métaphores de part la manipulation de ses balises et d'un total accès à l'ensemble des primitives visuelles de la métaphore (*i.e.* au code VRML). L'équivalent en *Java 3D* nécessiterait que l'utilisateur écrive des lignes de code *Java*. Toutefois ce besoin de production de code *Java* existe aussi en X-VRML, mais uniquement pour créer des nouvelles balises dans le langage. Cela ne concerne donc pas directement l'utilisateur en tant que concepteur de métaphores.

⁸<https://java3d.dev.java.net/>

⁹Les auteurs n'ont pas d'expérience sur l'utilisation de cette API.

4.3.2 Personnalisation

L'utilisation du langage X-VRML rend possible la personnalisation de l'interface de restitution des résultats de recherche. Il faut toutefois préciser que la personnalisation d'une interface requiert actuellement des connaissances, de la part de l'utilisateur, sur le langage VRML et dans une moindre mesure sur le langage XML. L'expression des métaphores en langage X-VRML est suffisamment simple et compréhensible pour permettre à l'utilisateur de la modifier, voire même de créer sa propre métaphore. Par exemple, une simple modification des paramètres de la balise `SW_CITY_MAP` a pour conséquence de changer le type d'organisation des résultats. En effet, l'algorithme utilisé pour organiser les données est choisi en fonction de la chaîne de caractère associée au paramètre `TYPE` de cette balise (e.g. `'kohonen'` pour une carte auto-organisatrice ou `'axis'` pour une organisation basée sur deux axes représentant deux variables). De plus, les paramètres associés à chaque type d'organisation sont accessibles (e.g. la taille de la carte ou celle des vecteurs de mots représentant les documents). Il s'agit là que d'un exemple parmi les nombreuses balises du langage. Par ailleurs, toutes les primitives visuelles sont exprimées en VRML. Ce code est donc aussi directement modifiable par l'utilisateur qui peut changer la représentation des documents (i.e. les bâtiments de la métaphore de la ville) en quelques lignes de code VRML. Des personnalisations automatiques sont aussi possibles. Ainsi la métaphore peut s'adapter en fonction d'informations extérieures. Par exemple, l'horloge du système est utilisée pour modifier l'aspect visuel du ciel (nuit/jour). D'autres informations pourraient être exploitées telles que la météo et la position géographique de l'utilisateur. Il s'agit là de personnaliser au maximum l'interface pour que l'utilisateur se retrouve dans son « chez lui » virtuel. Pour finir sur ce point, une interface graphique d'aide à la personnalisation mériterait d'être implémentée afin d'aider davantage les utilisateurs novices. Pour les autres, il est fort probable qu'ils préfèrent directement accéder au code X-VRML et VRML, ce qui leur offre beaucoup de liberté dans la définition de l'interface de visualisation.

4.4 Utilisation du prototype

Dans cette section, nous présentons une utilisation standard du prototype Smart-Web. Dans ce but, le fonctionnement global du prototype est d'abord expliqué et un exemple basé sur la métaphore de la ville est ensuite proposé.

4.4.1 Fonctionnement

Dans un premier temps, le prototype fonctionne de façon très similaire à un moteur de recherche. En effet, l'utilisateur spécifie une requête textuelle qu'il juge représentative de ses besoins. Ensuite, le prototype va rechercher dans sa base de données les réponses à la requête. La partie système de recherche utilise des solutions existantes étant donné que la thèse n'a pas vocation à contribuer sur cet aspect. Ici nous exploitons les réponses fournies par le moteur de recherche *Google*. Par contre, les différences

par rapport à un SRI classique concernant la restitution des résultats. L'utilisateur doit spécifier, en même temps que sa requête, l'interface de visualisation qu'il souhaite utiliser. Ce choix peut se faire parmi plusieurs interfaces 2D, 2^{1/2}D ou 3D. Dans le cadre de cette thèse, nous nous sommes concentrés sur la visualisation 3D basée sur la métaphore de la ville. Une fois la requête effectuée, les résultats sont alors affichés selon la métaphore de visualisation choisie. Dans cette interface, on y retrouve les informations classiques sur les résultats (URL, titre, *snippet*, rang, copie d'écran) ainsi que des informations plus riches (e.g. les relations entre les documents). Il est aussi possible de passer d'une métaphore de visualisation à une autre lors de l'exploration des résultats de recherche. Un exemple d'utilisation du prototype est présenté dans la section suivante.

4.4.2 Exemple

L'exemple d'utilisation proposé utilise la métaphore de la ville présentée dans la section 3.2. Nous distinguons dans la suite la partie spécification des besoins — qui est classique — de celle concernant l'utilisation proprement dite de la métaphore de visualisation — qui constitue l'originalité du prototype.

4.4.2.1 Spécifier sa requête

- Étape 1.** Entrez votre requête dans le champ intitulé « Query ». Votre requête peut être simple ou booléenne. Ainsi la requête « orange » retournera toutes les pages Web contenant le mot « orange » et la requête « orange fruit » retournera toutes les pages Web contenant à la fois les mots « orange » et « fruit ». Dans ce dernier cas, la requête est booléenne : l'espace est interprété comme l'opérateur booléen « ET » (c'est le cas de la plupart des moteurs de recherche e.g. *Google*).
- Étape 2.** Sélectionnez le type d'interface que vous souhaitez grâce au menu déroulant intitulé « Choose your interface ».
- Étape 3.** Cliquez sur le bouton « SEARCH » pour lancer votre recherche. Vous pouvez aussi lancer votre recherche en tapant sur la touche « entrée » de votre clavier (uniquement lorsque vous êtes dans le champ « query »).

4.4.2.2 Utilisation de la métaphore de ville

Représentation graphique d'un résultat. Chaque résultat de la requête est représenté par un bâtiment. Toutes les empreintes au sol des bâtiments sont identiques. Par contre la hauteur des bâtiments représente la « pertinence système » des résultats (selon l'algorithme *PageRank* de *Google*). Cependant nous avons choisi de représenter cette pertinence sous la forme de trois intervalles. Ainsi, les 10 premiers résultats sont

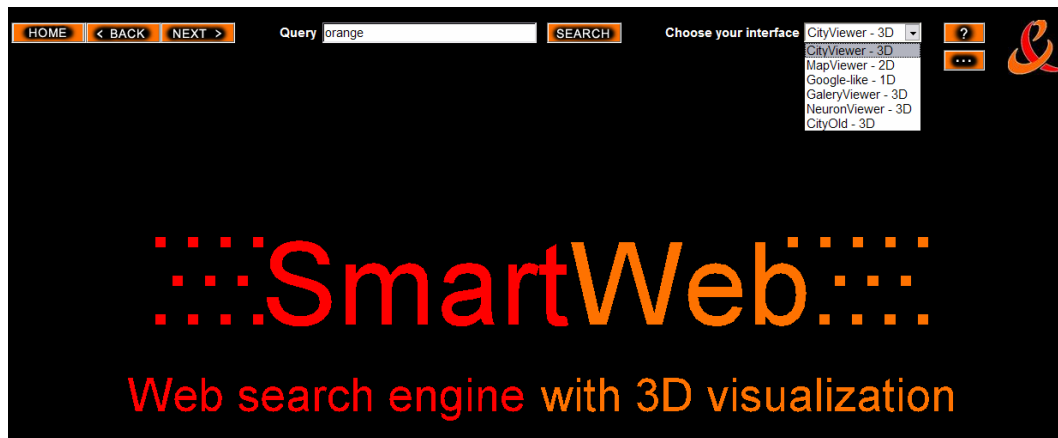


FIG. 4.4 – Écran d'accueil du prototype



FIG. 4.5 – Métaphore de la ville

associés à des bâtiments de même taille (les plus hauts). Ensuite les 20 suivants sont associés à des bâtiments de taille moyenne et enfin les autres résultats sont représentés par les bâtiments de petite taille. En effet, si nous avons choisi de représenter la hauteur des bâtiments proportionnellement à la pertinence (*i.e.* au rang calculé par l'algorithme *PageRank*), alors cela aurait posé plusieurs problèmes tels que : la différenciation des rangs proches ou les confusions dues à la perspective liée à l'affichage 3D. La texture des bâtiments correspond à une copie d'écran (image au format JPG) de la page HTML associée à l'URL du résultat. La texture est la même sur toutes les faces du bâtiment. Un simple « clic gauche » de la souris sur un bâtiment permet d'ouvrir la page Web associée dans une nouvelle fenêtre de votre navigateur. De plus,

chaque bâtiment possède un toit, de couleur grise par défaut. Si la page Web associée au bâtiment a déjà été visitée par l'utilisateur, alors le toit devient rouge. Cela permet d'identifier rapidement les pages déjà visitées. De même, un drapeau rouge apparaît sur les bâtiments correspondant à des pages Web présentes dans les favoris (*bookmarks*) de l'utilisateur. Cela permet de donner un point de repère à l'utilisateur. En effet, d'après l'organisation des résultats, les bâtiments proches d'un bâtiment avec un drapeau rouge correspondent à des résultats proches d'un favori de l'utilisateur, donc à des résultats susceptibles de l'intéresser. Enfin, il est possible de « double cliquer » sur un bâtiment afin de charger une nouvelle métaphore (*cf.* partie 4.4.2.3) qui permet de visualiser uniquement les images contenues dans la page Web associée.

Organisation des résultats. Les résultats sont placés sur le sol de la ville d'après la carte auto-organisatrice calculée. Un neurone (ou une unité) de la carte auto-organisatrice est représenté par un quartier. Ainsi deux bâtiments se trouvant dans le même quartier de la ville correspondent à deux résultats similaires. De façon analogue, les propriétés des cartes auto-organisatrices garantissent que deux quartiers voisins sont aussi similaires que possible (*i.e.* associés à deux thématiques proches). Les différents quartiers sont ensuite regroupés grâce à la CAH réalisée sur les neurones de la carte auto-organisatrice. Les différents regroupements obtenus sont différenciés par la couleur affichée sur le sol des quartiers. Ainsi, tous les quartiers ayant la même couleur au sol sont associés à une même « méta-thématique ». L'utilisateur a la possibilité de réaliser des « méta-thématiques » plus ou moins nombreuses selon le degré d'abstraction qu'il souhaite avoir sur les résultats.

Affichages 2D dans le navigateur VRML. Il est difficile d'afficher correctement du texte dans des interfaces 3D. Pour résoudre ce problème, nous proposons d'utiliser des affichages 2D (appelés *2D Layers*), c'est-à-dire que ces affichages ne bougent pas lors de la navigation dans l'espace 3D. Cela s'obtient tout simplement en fixant ces affichages en premier plan. Ces affichages sont au nombre de trois. Le premier, situé en haut à droite de l'interface 3D, est le plan 2D de la ville (*i.e.* la vue de dessus). Le point rouge représente la position courante de l'utilisateur et la flèche rouge correspond à sa direction de visualisation. Le second affichage 2D se trouve sur la gauche de l'interface 3D. Il contient les informations classiques (titre, *snippet* et URL) sur le résultat pointé par la souris (en haut) ainsi que sur 4 résultats voisins. Le troisième affichage 2D se trouve en bas de l'interface 3D. Il donne une description, sous la forme d'un ensemble de mots-clés, des différentes « méta-thématiques ».

Navigation. L'espace 3D proposé à l'utilisateur est immersif, c'est-à-dire qu'il est possible de naviguer librement à l'intérieur suivant tous les axes (rotation et translation). La navigation est classique et correspond aux fonctions proposées par la plupart des navigateurs VRML. On retrouve ainsi les fonctions suivantes :

- **Mouvements.** On maintient le bouton gauche de la souris enfoncé tout en bougeant la souris dans le sens du déplacement que l'on souhaite réaliser. Ces 4

actions de base peuvent aussi être effectuées à l'aide des flèches directionnelles du clavier.

- **Menu.** Un « clic » droit de la souris permet d'accéder à un menu dans lequel il est possible de paramétrer un ensemble de valeurs telles que la vitesse des mouvements ou l'affichage de l'avatar (représentant l'utilisateur). Ce menu permet également de sélectionner le mode de navigation (*e.g.* marche au sol, survol) ou encore un des différents points de vue prédéfinis. Les différentes actions du menu ont des équivalents en termes de raccourcis clavier.

Cependant, ce mode de navigation dans un environnement 3D est loin d'être trivial pour l'utilisateur *lambda*, même si certaines catégories d'utilisateurs s'en sortent mieux que d'autres. Afin de simplifier les déplacements dans la ville, il est alors possible d'utiliser le plan 2D qui est interactif. Ainsi, un simple « clic » sur un quartier du plan 2D a pour conséquence de déplacer l'utilisateur devant ce quartier dans la scène 3D.

Interface 2D proposée par l'applet. Elle est composée d'éléments basiques. On y trouve ainsi différents boutons associés à différentes actions :

- bouton « HOME » pour retourner à l'écran d'accueil,
- bouton « BACK » pour aller à la représentation précédente (recul dans l'historique),
- bouton « NEXT » pour aller à la représentation suivante (avance dans l'historique),
- bouton « SEARCH » pour démarrer la recherche,
- bouton « ? » pour accéder à l'aide en ligne dont le contenu dépend du contexte, c'est-à-dire de la métaphore sélectionnée.

Cette interface dispose aussi d'un champ textuel (intitulé « QUERY ») pour entrer la requête. Une fois la requête effectuée, le nombre de résultats présentés dans l'interface de restitution (ici il s'agit du nombre de bâtiments dans la ville virtuelle) est affiché sous le champ de la requête. Enfin, selon la métaphore de visualisation sélectionnée, certains éléments additionnels peuvent apparaître.

4.4.2.3 Utilisation de la métaphore de la galerie

Lorsque l'utilisateur « double clique » sur un bâtiment de la ville (*cf.* **figure 4.6**), l'applet charge alors une nouvelle métaphore, appelée galerie, qui permet de visualiser uniquement les images contenues dans la page Web sélectionnée. Les images de dimensions trop petites ou ne contenant que très peu de couleurs sont considérées comme des icônes et elles sont donc exclues de la visualisation.

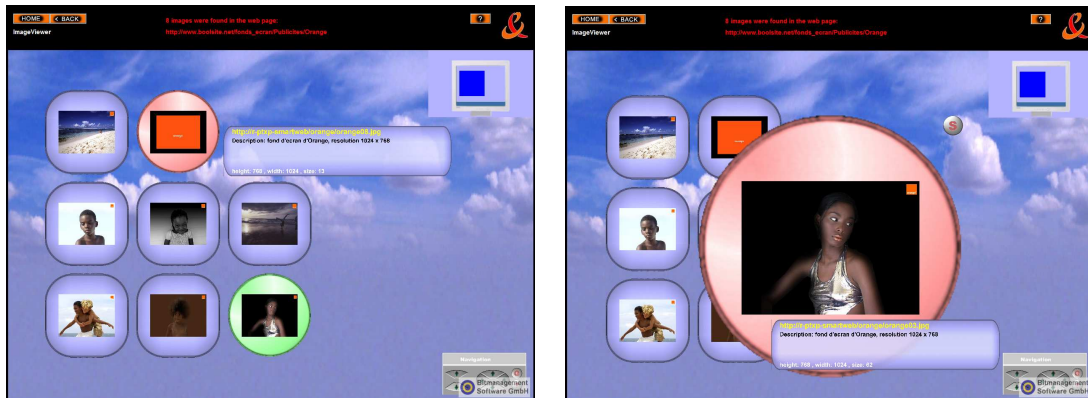
Dans un premier temps, cette métaphore propose un affichage 2D des images (**figure 4.7(a)**). Il est possible de déplacer cette palette d'images et de zoomer en utilisant les boutons de navigation proposés en bas à droite de l'interface, ou à l'aide du cadre en haut à droite de l'interface permettant de déplacer rapidement la palette d'images. L'utilisateur peut aussi sélectionner une image (par simple « clic gauche »



FIG. 4.6 – Représentation dans la ville 3D de la page Web sélectionnée (par « double clic ») afin de visualiser les images qu’elle contient

de la souris) et la faire ainsi apparaître en premier plan (**figure 4.7(b)**). Il est alors possible de supprimer l’image de la représentation en cliquant sur le bouton apparaissant en haut à droite de la vignette sélectionnée. Un simple « clic gauche » de la souris sur l’image en avant-plan a pour conséquence de replacer cette image avec les autres de la palette. Il est également possible, à tout moment, d’ajouter l’image à ses favoris en cliquant sur la vignette associée à l’aide du bouton central de la souris. Cette action répétée une seconde fois sur une image apparaissant déjà dans les favoris a pour conséquence de la supprimer des favoris. Lorsque le curseur de la souris est sur une vignette, un affichage 2D contextuel contenant des informations sur l’image (URL, description, dimensions et taille) apparaît. Cette action reste valide même lorsqu’une vignette est en avant-plan. Il est aussi possible d’ouvrir l’image dans une nouvelle fenêtre du navigateur Web en « double cliquant » sur la vignette associée à cette image.

Ensuite, dans un second temps, il est possible de passer à une version 3D de cette métaphore de la galerie (en cliquant sur le bouton rouge situé dans le cadre des boutons de navigation en bas à droite de l’interface). Les vignettes sont alors organisées linéairement sous la forme d’une galerie 3D composée d’un unique chemin central (**figure 4.8**). Afin d’aider l’utilisateur, la navigation est restreinte à ce chemin central sur lequel il est possible d’avancer, de reculer et d’effectuer des rotations avec un angle maximum limité à un affichage central des vignettes. Le même système d’affichage



(a) Représentation 2D des images d'un site Web

(b) Sélection d'une image

FIG. 4.7 – Représentations 2D des images d'un site Web

d'informations contextuelles sur les images est disponible dans cette version 3D de la galerie. Par ailleurs les actions réalisées dans la version 2D (favoris et suppression d'images) restent persistantes lors du passage à la version 3D, et inversement. Cela est possible grâce au stockage de ces informations par le prototype.

4.5 Applications

Le prototype n'est actuellement utilisé qu'à des fins expérimentales. Il est pourtant tout à fait envisageable de déployer une version en ligne. Les seules conditions imposées sur la configuration logicielle de l'ordinateur de l'utilisateur sont la présence d'un interpréteur *Java*, d'un navigateur Web et d'un navigateur VRML¹⁰. Cela est tout à fait raisonnable et ces applications sont déjà présentes sur la plupart des ordinateurs. Concernant la génération dynamique de l'environnement 3D, elle peut être réalisée sur le poste client ou sur le serveur suivant l'architecture choisie (cf. fin de la partie 4.2.2.2). Enfin, au sujet de la base de données, l'objectif n'est pas de réaliser nos propres index du Web. L'API de *Google*, utilisée à des fins de recherche, doit alors être remplacée par un « partenariat » avec un moteur de recherche possédant une base d'index acceptable. Il est alors possible qu'une légère adaptation soit nécessaire pour adapter le prototype au mécanisme permettant d'interroger ces nouvelles bases d'index (sauf dans le cas d'un accord avec *Google*). À ces conditions, le prototype *SmartWeb* pourrait être disponible en ligne, et ainsi permettre la réalisation d'un test utilisateur

¹⁰Pour l'instant, le prototype n'a été testé qu'avec le navigateur Web *Internet Explorer* de *Microsoft* et le navigateur VRML *BS Contact* de *Bitmanagement Software GmbH*. Mais une plus grande portabilité est possible moyennant des ajustements d'ordre technique.



FIG. 4.8 – Représentation, sous la forme d’une galerie, des images d’un site Web

à grande échelle.

Le prototype a déjà fait l’objet de nombreuses démonstrations durant la période 2003-2005 :

- Démonstration pour la visite de Pascal Viginier, directeur de France Télécom R&D, public interne à France Télécom R&D, durée : ponctuelle, lieu : Cesson-Sévigné, date : 14 mars 2003.
- Démonstration pour la visite de Gérard Eude, directeur de la Recherche de France Télécom R&D, public interne à France Télécom R&D, durée : ponctuelle, lieu : Cesson-Sévigné, date : 24 mars 2003.
- Démonstration au VisionL@b (espace démonstration) de France Télécom R&D, public interne à France Télécom R&D, durée : demi-journée, lieu : Cesson-Sévigné, date : 8 avril 2003.
- Démonstration au VisionL@b (espace démonstration) de France Télécom R&D, public interne à France Télécom R&D, durée : demi-journée, lieu : Cesson-Sévigné, date : 15 avril 2003.
- Démonstration / test à l’université de Paris 13, public composé d’étudiants en informatique, durée : 2 jours, lieu : Villetaneuse, date : lundi 28 et mardi 29 avril 2003.

- Démonstration / test au centre des arts d’Enghiens-les-Bains, grand public, durée : journée, lieu : Enghiens-les-Bains, date : mercredi 30 avril 2003.
- Démonstration dans le cadre d’un projet avec des étudiants de l’école de design de Nantes (EDN), public composé d’étudiants de l’école de design de Nantes, durée : ponctuelle, lieu : Nantes, date : 24 février 2004. Participation au suivi du projet ayant abouti (en juin 2004) à la livraison, de la part des étudiants, de six maquettes de visualisation (2D ou 3D) de résultats de recherche.
- Démonstration pour la visite de Pierre-Antoine Badoz, responsable du laboratoire de rattachement de cette thèse (IRIS), public interne à France Télécom R&D, durée : ponctuelle, lieu : Cesson-Sévigné, date : 10 juin 2004.
- Démonstration lors d’une visite au pôle de recherche de rattachement de cette thèse, public interne à France Télécom R&D, durée : ponctuelle, lieu : Lannion, date : jeudi 17 février 2005.
- Démonstration pour la visite de Paul Friedel, directeur de la Recherche de France Télécom R&D, public interne à France Télécom R&D, durée : ponctuelle, lieu : Cesson-Sévigné, date : lundi 20 juin 2005.
- Démonstration aux Journées Portes Ouvertes de l’IRISA (pour les 30 ans), grand public, durée : 3 jours, lieu : Rennes, date : du vendredi 14 au dimanche 16 octobre 2005.
- Démonstration lors d’une visite à *Voilà (Wanadoo)* qui gère le portail de recherche d’information de *Wanadoo*, public interne à *Voilà*, durée : ponctuelle, lieu : Issy-Les-Moulineaux, date : le mardi 29 novembre 2005.
- Démonstration au Salon des Résultats de la Recherche de France Télécom R&D, public interne à France Télécom, durée : journée, lieu : Issy-Les-Moulineaux, date : jeudi 1^{er} décembre 2005.

À l’issue de ces nombreuses démonstrations, nous avons eu beaucoup de retours informels de la part d’utilisateurs de profils variés. Ces informations nous permettent aujourd’hui d’avoir un certain recul sur notre prototype. Ainsi, il semble par exemple que le cadre applicatif que nous proposons dans cette thèse ne soit pas le seul adapté. C’est pourquoi nous citons, ci-dessous, d’autres contextes applicatifs possibles pour nos travaux :

- **Visualisation des actualités** (ou *news*). Le contenu des actualités peut en effet être traité de façon similaire aux pages Web issues d’une requête. Cependant, il faudrait pouvoir prendre en compte le fait que les actualités appartiennent à des catégories particulières pouvant être prédéfinies (*e.g.* monde, économie, *people*, sport). Dans ce cadre applicatif, on rappelle l’exemple *Newsmap* qui est une approche 2D basée sur une méthode *Treemap* et proposée par Marcos Weskamp (*cf.* **figure A.11** de l’**annexe A**). Dans la cas de la visualisation des actualités, l’aspect temporel des données doit aussi être pris en compte du fait qu’il s’agisse d’une information cruciale pour ce type d’application.
- **Visualisation de flux ou fils RSS**¹¹ (ou *RSS feeds*). La visualisation de flux RSS est

¹¹Really Simple Syndication ou Rich Site Summary

aussi un cadre applicatif intéressant et relativement nouveau. Il rejoint en partie le cas de la visualisation des actualités, notamment de part la prise en compte de l'aspect temporel des documents. Actuellement, donner de l'importance à la « fraîcheur » des informations nécessite d'apporter des modifications à notre prototype.

- **Visualisation de favoris** (ou *bookmarks*). Un exemple de ce type d'application est *Data Mountain* [Robertson *et al.* 1998]. Techniquement ce contexte applicatif est similaire à celui que nous proposons, étant donné que les favoris sont des pages Web. Cependant ce type d'application offre deux avantages supplémentaires. D'abord, il est possible d'utiliser le classement hiérarchique défini par l'utilisateur. Ensuite, l'utilisateur peut se repérer plus facilement dans l'espace de visualisation de ces favoris étant donné qu'il connaît déjà toutes les pages Web représentées.
- **Visualisation de systèmes de fichiers**. Une autre possibilité est en effet de visualiser tout ou une partie d'un système de fichiers. Ce type de visualisation de fichiers basée sur une organisation selon leur contenu pourrait alors être à l'origine de découvertes non triviales dans les relations entre fichiers. Il serait aussi intéressant d'exploiter l'organisation hiérarchique des fichiers, ce qui n'est pas possible avec la version actuelle du prototype. Par ailleurs, un autre problème est l'aspect multimédia des données (texte, image, son, vidéo). Or notre prototype ne gère que les données textuelles. Pour les autres médias, il faut réfléchir à leur représentation ainsi qu'à d'autres types d'organisation. La visualisation sémantique de ces différents médias dans une interface commune reste encore un défi à relever. Actuellement le prototype ne peut donc tirer profit que des fichiers contenant des informations textuelles. De plus, dans le cas d'un nombre important de fichiers, un pré-calcul de la carte auto-organisatrice est nécessaire si l'on souhaite garder une utilisation temps-réel.
- **Visualisation du contenu d'une base de données**, voire d'une partie du Web. Ce type d'application est très proche de notre cadre applicatif, à condition que les données soient textuelles. Comme dans le cas précédent, la seule contrainte est de calculer, hors ligne, la carte auto-organisatrice si le nombre de données est trop important. Sinon l'utilisabilité de l'application peut être grandement remise en cause. Une autre solution consiste à réaliser le calcul de la carte en ligne, mais sur un sous-ensemble des données (les autres données sont alors juste projetées sur la carte). Notre prototype est d'ailleurs peut-être plus adapté à ces tâches de navigation. Vincent Lemaire l'a même rebaptisé « *Ballad on the Web* » au cours d'un *brainstorming*.

Dans ce chapitre, nous avons présenté le prototype *SmartWeb* qui intègre nos propositions en termes d'organisation et de visualisation des résultats de recherche. Il faut rappeler qu'il s'agit d'un prototype ayant pour objectif de tester nos travaux de recherche, ce qui explique parfois l'absence d'optimisation en matière d'implémentation. Maintenant, il reste encore à évaluer le prototype auprès des utilisateurs. Cette

étape est d'autant plus importante lorsqu'il s'agit d'interfaces utilisateur. Nous proposons alors de consacrer le prochain chapitre à l'évaluation du prototype *SmartWeb*.

Chapitre 5

Évaluation

De nombreux systèmes de visualisation des résultats de recherche ont déjà été proposés au cours des 10 dernières années. Or il s'avère difficile de définir le réel apport de ces systèmes ou interfaces sans passer par une évaluation. Dans ce chapitre, nous proposons donc une évaluation de notre IUI mais également une réflexion sur la mise en œuvre d'une telle évaluation. Ainsi ce chapitre se décompose en trois parties. D'abord, nous proposons une évaluation utilisateur basée sur un questionnaire. Il s'agit de la méthode la plus courante mais dont l'interprétation est bien souvent difficile. À partir de ce constat, nous portons une réflexion sur les méthodes d'évaluation dans le cas des IUI, ce qui implique de s'intéresser au déroulement de la phase d'évaluation (préalables, panel utilisateur, contraintes) mais aussi aux critères à utiliser.

5.1 Première évaluation

Cette première évaluation de notre IUI intervient volontairement à un stade précoce par rapport à l'avancement de notre prototype. En effet, l'objectif est d'avoir un retour utilisateur au cours du développement de l'IUI afin de pouvoir prendre en considération leurs attentes. Cette évaluation repose principalement sur un questionnaire utilisateur. Nous allons dans la suite respecter une certaine chronologie en expliquant tout d'abord le contexte de cette étude utilisateur, puis le protocole mis en place et enfin une interprétation des résultats.

5.1.1 Contexte

Cette première évaluation s'est déroulée sur 3 jours consécutifs en avril 2003. Elle a été réalisée sur un panel de 60 utilisateurs qui se décompose en deux parties : 48 étudiants de l'université de Paris 13 lors des deux premiers jours (20 étudiants le premier jour et 28 le second jour) et 12 personnes lors du troisième jour. Les utilisateurs des deux premiers jours ont utilisé le prototype en binôme mais ils ont chacun répondu individuellement au questionnaire proposé. Les 12 utilisateurs du troisième jour étaient

des visiteurs du centre des arts d'Enghien-les-Bains. Aucun utilisateur n'avait déjà utilisé notre prototype. Cette étude repose uniquement sur un questionnaire utilisateur, réalisé d'après les propositions de Ben Shneiderman en matière d'évaluation d'IHM [Shneiderman 1998, page 136].

La version du prototype évaluée lors de ce premier test est celle présentée dans la partie 3.3.2.1 du **chapitre 3**. Les différences majeures entre cette première version et la version actuelle du prototype sont les suivantes.

- L'organisation des résultats n'est pas basée sur le calcul d'une carte auto-organisatrice, mais simplement sur un affichage selon deux axes (noms de domaine et mots-clés les plus fréquents).
- La texture des bâtiments est uniforme et ne montre pas un aperçu du résultat (copie d'écran de la page Web).
- L'affichage contextuel des informations sur un résultat n'est pas accompagné de l'affichage contextuel des informations sur quatre résultats voisins.
- Le plan 2D n'est pas interactif, *i.e.* il ne peut pas être utilisé pour se déplacer rapidement à un quartier précis de la ville.

5.1.2 Protocole

Tout d'abord, nous avons réalisé un questionnaire utilisateur. Pour cela nous sommes essentiellement basés sur les célèbres propositions de [Shneiderman 1998] en matière d'évaluation d'interfaces utilisateur. Ensuite ce test a été soumis à une ergonome afin d'améliorer son efficacité. Ainsi, certaines questions ont été modifiées afin de ne pas laisser de questions trop ouvertes. Cela nous a également permis, pour chaque question, de nous demander ce que l'on souhaitait vraiment évaluer et quel type de réponse nous attendons. Enfin ce travail avec une ergonome nous a aussi amené à modifier l'échelle de notation. Initialement, Ben Shneiderman propose une évaluation en 9 points. Nous avons ramené cette échelle à 5 points seulement, ce qui nous semble amplement suffisant. Enfin, les questions ont été choisies de telle façon à garder un test assez court et peu contraignant pour l'utilisateur. En effet, nous souhaitons qu'il puisse répondre à ce test en 10 minutes environ. Le questionnaire ainsi réalisé et proposé aux utilisateurs est présenté dans son intégralité sur les **figures 5.1** et **5.2**.

5.1.3 Résultats

L'interprétation des résultats de ce test utilisateur se décompose en deux parties. La première partie permet de définir le profil des utilisateurs, puis la seconde partie est l'évaluation proprement dite du prototype SmartWeb.

Questionnaire pour la satisfaction de l'interface utilisateur de l'application SMARTWEB

Évaluez votre satisfaction du système.

- Essayez de répondre à toutes les questions.
- NA : Pas Applicable
- Pour envoyer vos réponses, cliquez sur : **Soumettre**

Soumettre

IDENTIFICATION

Age

Sexe masculin féminin

1. EXPERIENCE DU SYSTEME SMARTWEB

1.1 Combien de temps avez vous travaillé sur SmartWeb?

1.2 En moyenne, combien de temps passez vous par semaine sur SmartWeb?

2. EXPERIENCE PASSEE

2.1 Avec combien de systèmes d'exploitation avez vous déjà travaillé?

2.2 Parmi les appareils suivants, sélectionnez ceux que vous avez personnellement utilisé et avec lesquels vous êtes familier :

ordinateur personnel non oui ordinateur portable non oui écran couleur non oui

écran tactile non oui lecteur de CD-ROM non oui souris non oui

track ball non oui joystick non oui stylet non oui

tablette graphique non oui affichage via des lunettes non oui scanners non oui

2.3 Parmi les logiciels suivants, sélectionnez ceux que vous avez personnellement utilisé et avec lesquels vous êtes familier :

outils de traitement de texte non oui logiciels graphiques non oui

tableurs non oui logiciel de base de données non oui

2.4 Parmi les systèmes suivants, sélectionnez ceux que vous avez personnellement utilisé et avec lesquels vous êtes familier :

reconnaissance vocale non oui internet non oui e-mail non oui

CAO conception assistée par ordinateur non oui chat non oui jeux en 3D (sur ordinateur) non oui

jeux en réseau (sur ordinateur) non oui autres jeux (sur ordinateur) non oui

2.5 Quel moteur de recherche utilisez-vous principalement?

Google YAHOO Voila Lycos Altavista KartOO autres

Quel avantage majeur lui trouvez vous?

nombre de résultats pertinence des résultats fonctionnalités présentation des résultats rapidité autres

Quel inconvénient majeur lui trouvez vous?

nombre de résultats pertinence des résultats fonctionnalités présentation des résultats lenteur autres

2.6 Quel est votre catégorie de sites Web préférés?

actualités commerce sciences informatique divertissements art - culture autres

2.7 Avez vous des bookmarks?

non oui, mais moins de 50 oui, entre 50 et 100 oui, entre 100 et 150 oui, plus de 150

Entrez la profondeur de votre hiérarchie de bookmarks (entrez 1 pour aucune hiérarchie)

2.8 En moyenne, combien de temps passez vous par semaine sur un navigateur Web?

FIG. 5.1 – Questionnaire utilisateur : profil

3. REACTION GENERALE A SMARTWEB		1	2	3	4	5	NA
3.1 Comment trouvez vous SmartWeb?	affreux	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	merveilleux <input type="radio"/>
3.2 Comment trouvez vous SmartWeb?	frustrant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	satisfaisant <input type="radio"/>
3.3 Comment trouvez vous SmartWeb?	ennuyeux	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	stimulant <input type="radio"/>
3.4 Comment trouvez vous SmartWeb?	confus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	clair <input type="radio"/>
3.5 Utilisation de SmartWeb	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
3.6 Vitesse du système	trop lent	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	assez rapide <input type="radio"/>
3.7 Fiabilité du système	peu fiable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	fiable <input type="radio"/>
4. AFFICHAGE		1	2	3	4	5	NA
4.1 Lecture des caractères sur l'écran	dur	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
4.2 L'affichage contextuel simplifie la tâche	pas du tout	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	beaucoup <input type="radio"/>
4.3 Organisation de l'information sur l'écran	confus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	très clair <input type="radio"/>
4.4 Séquence des affichages (requête puis métaphore ville et enfin accès à la page Web ou à la représentation des images)	confus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	très clair <input type="radio"/>
5. TERMINOLOGIE ET INFORMATIONS DU SYSTEME		1	2	3	4	5	NA
5.1 Usage des termes à travers le système	incohérent	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cohérent <input type="radio"/>
5.2 La terminologie est liée à la tâche	jamais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	toujours <input type="radio"/>
5.3 Position des messages sur l'écran	incohérent	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cohérent <input type="radio"/>
5.4 Entrées de texte	confus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	clair <input type="radio"/>
5.5 L'ordinateur informe au sujet de son avancement	jamais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	toujours <input type="radio"/>
6. APPRENTISSAGE		1	2	3	4	5	NA
6.1 Apprendre à utiliser le système	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
6.2 Exploration de nouvelles fonctionnalités par l'essai et l'erreur	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
6.3 Se rappeler le nom et l'usage des commandes	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
6.4 L'exécution des tâches est directe	jamais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	toujours <input type="radio"/>
6.5 Destiné à tous les niveaux d'utilisateurs	jamais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	toujours <input type="radio"/>
7. AIDE EN LIGNE		1	2	3	4	5	NA
7.1	inutile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	utile <input type="radio"/>
7.2	confus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	clair <input type="radio"/>
8. METAPHORE VILLE		1	2	3	4	5	NA
8.1 Trouvez vous la métaphore	inadaptée	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	adaptée <input type="radio"/>
8.2 Compréhension de la métaphore	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
8.3 Pertinence de la métaphore	pas pertinente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pertinente <input type="radio"/>
8.4 Que pensez vous de l'usage de la 3D pour ce type d'application?	inadapté	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	adapté <input type="radio"/>
8.5 Navigation	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
8.6 Déplacement dans la ville	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
8.7 Avez vous trouvé les deux barres de navigation?	difficilement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facilement <input type="radio"/>
8.8 Trouvez vous le plan en haut à droite utile pour se repérer?	pas du tout	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	énormément <input type="radio"/>
9. REPRESENTATION DES IMAGES		1	2	3	4	5	NA
9.1 Avez vous trouvé la représentation 2D des images d'une page Web?	difficilement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facilement <input type="radio"/>
9.2 Trouvez vous la représentation 2D	inadaptée	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	adaptée <input type="radio"/>
9.3 Compréhension de la représentation	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
9.4 Navigation	difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facile <input type="radio"/>
9.5 Avez vous trouvé la galerie 3D des images d'une page Web?	difficilement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	facilement <input type="radio"/>
9.6 Métaphore 3D de la galerie	inutile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	utile <input type="radio"/>
9.7 Fonction de bookmark	inutile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	utile <input type="radio"/>
9.8 Fonction de suppression	inutile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	utile <input type="radio"/>
10. IDEES		1	2	3	4	5	NA
10.1 Ajout d'un historique	inutile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	utile <input type="radio"/>
10.2 Visualisation plus abstraite sous forme de groupes de pages Web	inutile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	utile <input type="radio"/>

Quelles améliorations suggérez vous?

COMMENTAIRES

[Haut](#)
Contact : nicolas.bonnel@francetelecom.com

FIG. 5.2 – Questionnaire utilisateur : évaluation du prototype

5.1.3.1 Profil des utilisateurs

Un test utilisateur n'a de sens que si l'on est capable de définir le profil des utilisateurs ayant testé le système à évaluer. Dans ce but, un ensemble de questions a été posé aux utilisateurs afin d'obtenir des informations sur leur expérience. Les réponses à ces différentes questions sont synthétisées dans le **tableau 5.1** et représentées graphiquement sur les **figures 5.3** à **5.16**.

	IDENTIFICATION		
	Âge		
			Nombre d'utilisateurs
	RÉPONSES :	0	1
		12	1
		15	1
		16	2
		20	1
		21	13
		22	17
		23	9
		24	7
		25	1
		26	2
		31	1
		33	1
		35	1
		46	2
	Sexe		
			Nombre d'utilisateurs
	CHOIX :	masculin	47
		féminin	13
1	EXPÉRIENCE DU SYSTÈME SMARTWEB		
1.1	Combien de temps avez-vous travaillé sur SmartWeb ?		
			Nombre d'utilisateurs
	CHOIX :	NA	0
		moins d'une heure	56
		entre 1 heure et 1 jour	4
		entre 1 jour et 1 semaine	0
		entre 1 semaine et 1 mois	0
		entre 1 mois et 6 mois	0
		entre 6 mois et 1 an	0
		entre 1 an et 2 ans	0
		plus de 2 ans	0
1.2	En moyenne, combien de temps passez-vous par semaine sur SmartWeb ?		
			Nombre d'utilisateurs
	CHOIX :	NA	23
		moins d'une heure	33
		entre 1 heure et 4 heures	4
		entre 4 heures et 10 heures	0
		plus de 10 heures	0

2	EXPÉRIENCE PASSÉE		
2.1	Avec combien de systèmes d'exploitation avez-vous déjà travaillé ?		
		Nombre d'utilisateurs	
	CHOIX : NA	1	
	aucun	1	
	1	4	
	2	20	
	3-4	27	
	5-6	3	
	plus de 6	4	
2.2	Parmi les appareils suivants, sélectionnez ceux que vous avez personnellement utilisés et avec lesquels vous êtes familier :		
		NON	OUI
	ordinateur personnel	2	58
	ordinateur portable	15	45
	écran couleur	4	56
	écran tactile	36	24
	lecteur de CD-ROM	2	58
	souris	1	59
	<i>track ball</i>	50	10
	<i>joystick</i>	27	33
	stylet	41	19
	tablette graphique	50	10
	affichage <i>via</i> des lunettes	58	2
	scanners	13	47
2.3	Parmi les logiciels suivants, sélectionnez ceux que vous avez personnellement utilisés et avec lesquels vous êtes familier :		
		NON	OUI
	outils de traitement de texte	3	57
	logiciels graphiques	21	39
	tableurs	23	37
	logiciels de base de données	38	22
2.4	Parmi les systèmes suivants, sélectionnez ceux que vous avez personnellement utilisés et avec lesquels vous êtes familier :		
		NON	OUI
	reconnaissance vocale	54	6
	Internet	3	57
	e-mail	3	57
	CAO : Conception Assistée par Ordinateur	47	13
	<i>chat</i>	17	43
	jeux en 3D (sur ordinateur)	19	41
	jeux en réseau (sur ordinateur)	30	30
	autres jeux (sur ordinateur)	12	48

2.5	Quel moteur de recherche utilisez-vous principalement ?	
		Nombre d'utilisateurs
CHOIX :	<i>Google</i>	53
	<i>YAHOO</i>	4
	<i>Voila</i>	0
	<i>Lycos</i>	0
	<i>Altavista</i>	0
	<i>KartOO</i>	2
	autres	1
	Quel avantage majeur lui trouvez-vous ?	
CHOIX :	nombre de résultats	17
	pertinence des résultats	21
	fonctionnalités	2
	présentation des résultats	8
	rapidité	11
	autres	1
	Quel inconvénient majeur lui trouvez-vous ?	
CHOIX :	nombre de résultats	15
	pertinence des résultats	17
	fonctionnalités	4
	présentation des résultats	7
	lenteur	3
	autres	14
2.6	Quelle est votre catégorie de sites Web préférée ?	
		Nombre d'utilisateurs
CHOIX :	actualités	11
	commerce	1
	sciences	4
	informatique	11
	divertissements	18
	art - culture	6
	autres	9
2.7	Avez-vous des bookmarks ?	
		Nombre d'utilisateurs
CHOIX :	non	36
	oui, mais moins de 50	16
	oui, entre 50 et 100	7
	oui, entre 100 et 150	0
	oui, plus de 150	1
	Entrez la profondeur de votre hiérarchie de bookmarks	
		Nombre d'utilisateurs
RÉPONSES :	0	30
	1 (aucune hiérarchie)	16
	2	6
	3	3
	4	1
	6	1
	10	2
	20	1

2.8	En moyenne, combien de temps passez-vous par semaine sur un navigateur Web ?	
		Nombre d'utilisateurs
	CHOIX : NA	3
	moins d'une heure	3
	entre 1 heure et 4 heures	25
	entre 4 heures et 10 heures	16
	plus de 10 heures	13

TAB. 5.1 – Résultats de la première partie (définition du profil des utilisateurs) du test utilisateur réalisé sur 60 personnes d'âges et de professions différents

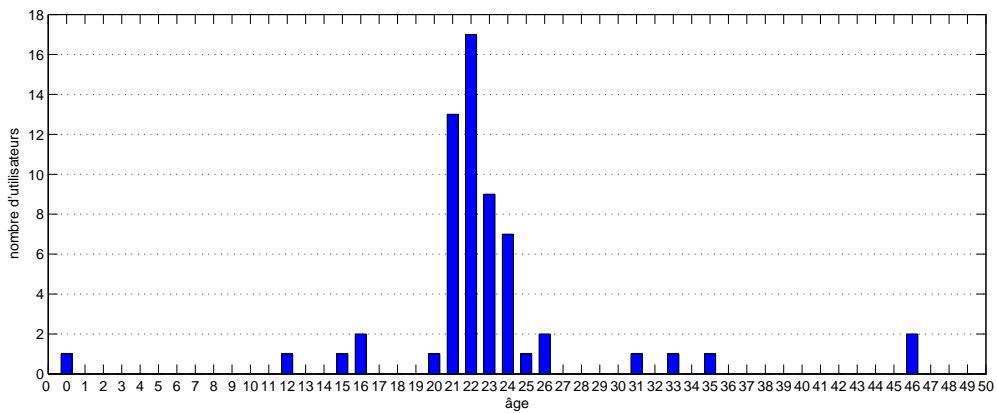


FIG. 5.3 – Âge

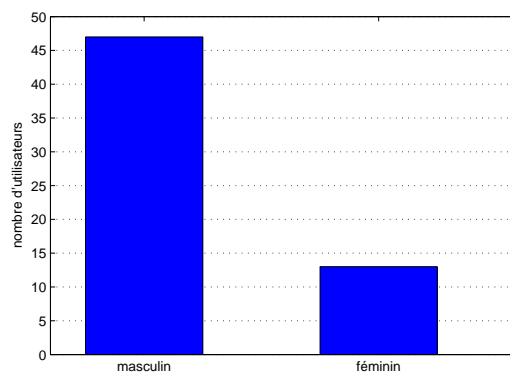


FIG. 5.4 – Sexe

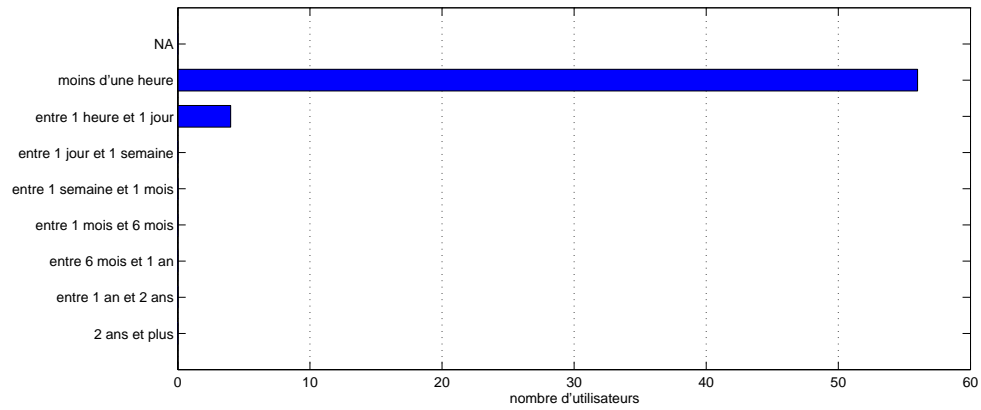


FIG. 5.5 – Temps passé à travailler sur SmartWeb

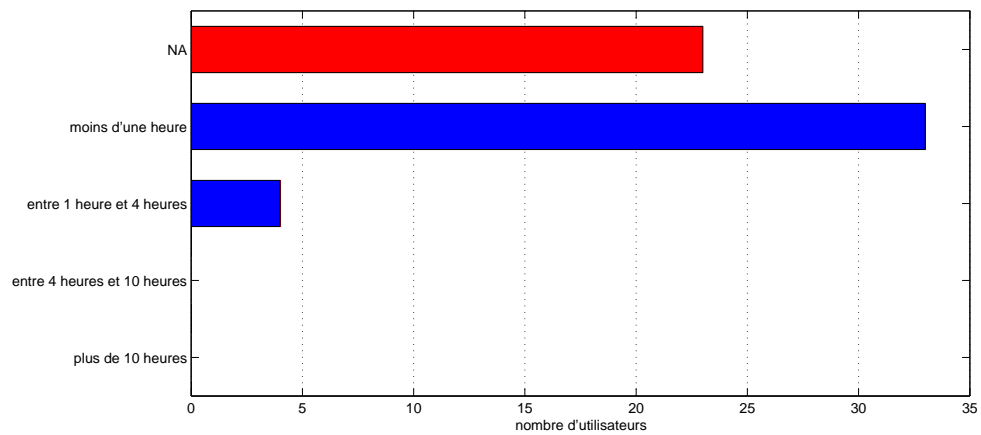


FIG. 5.6 – Temps moyen passé par semaine sur SmartWeb

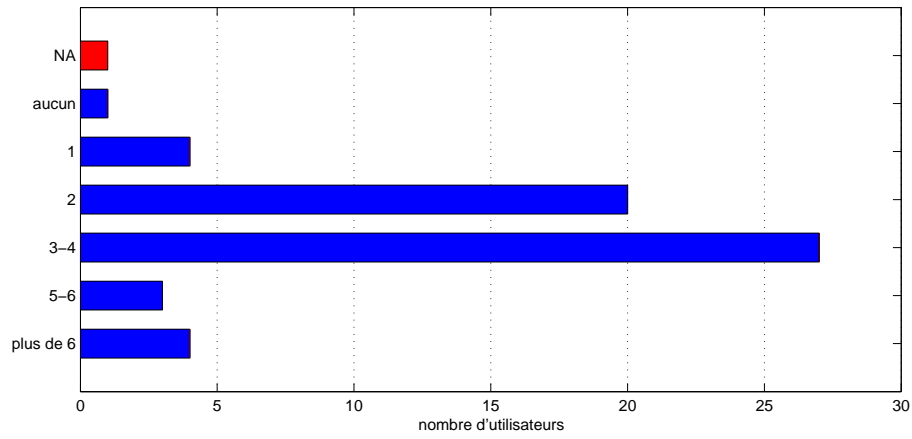


FIG. 5.7 – Nombre de systèmes d'exploitation déjà utilisés

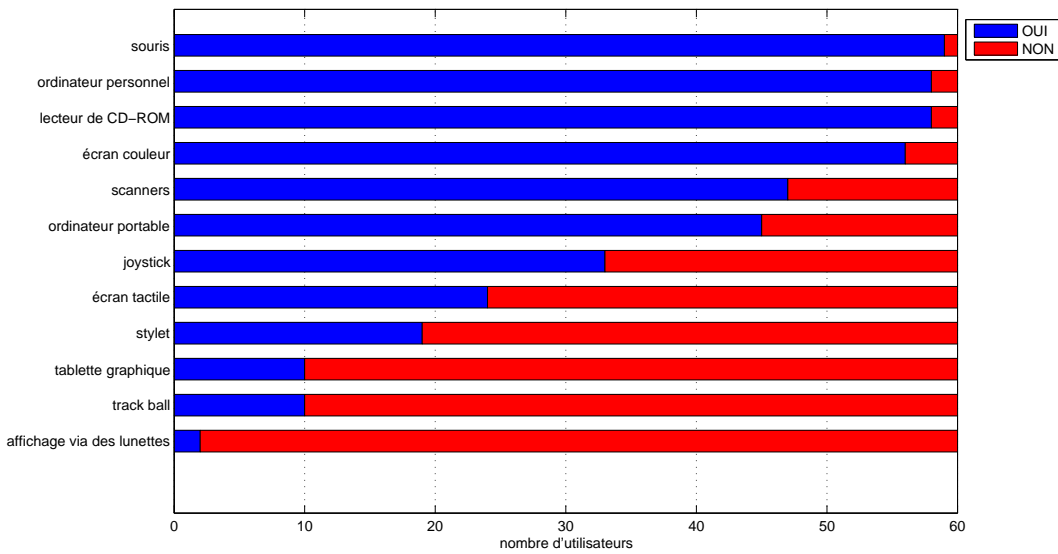


FIG. 5.8 – Appareils déjà utilisés et avec lesquels l'utilisateur est familier

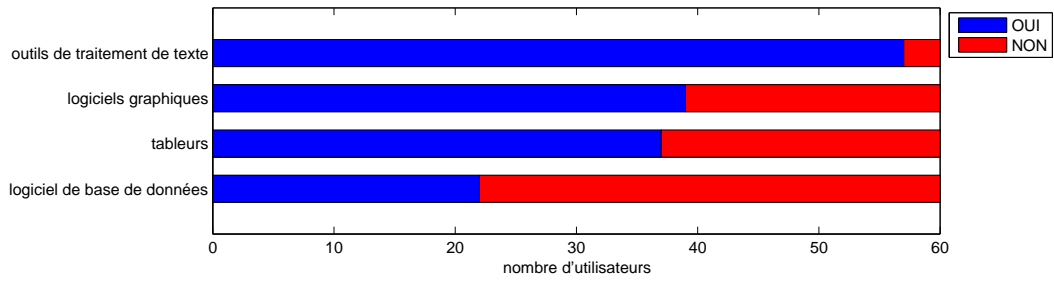


FIG. 5.9 – Logiciels déjà utilisés et avec lesquels l'utilisateur est familier

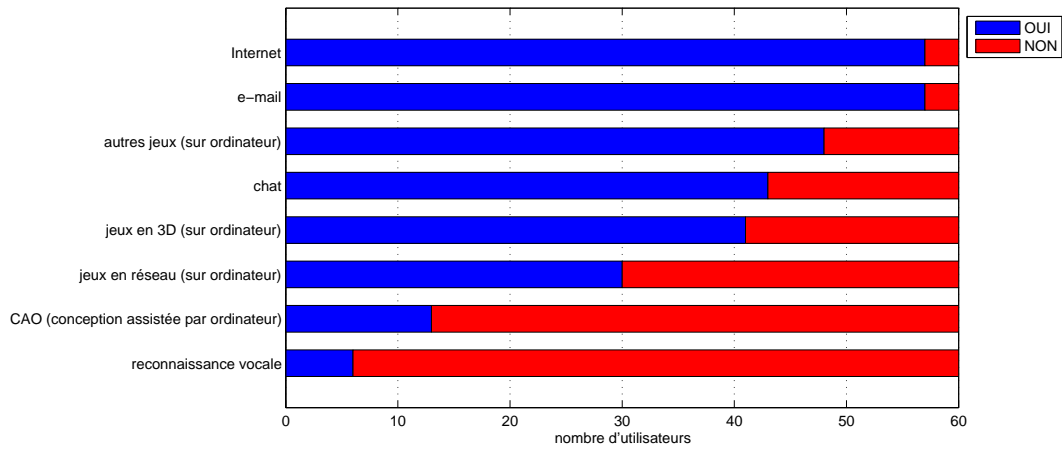


FIG. 5.10 – Systèmes déjà utilisés et avec lesquels l'utilisateur est familier

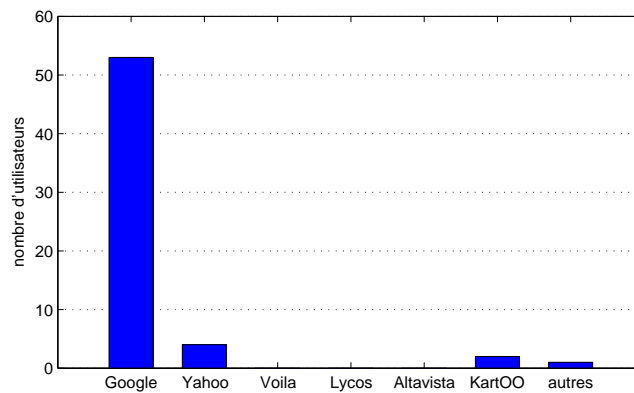


FIG. 5.11 – Moteur de recherche principalement utilisé par les utilisateurs

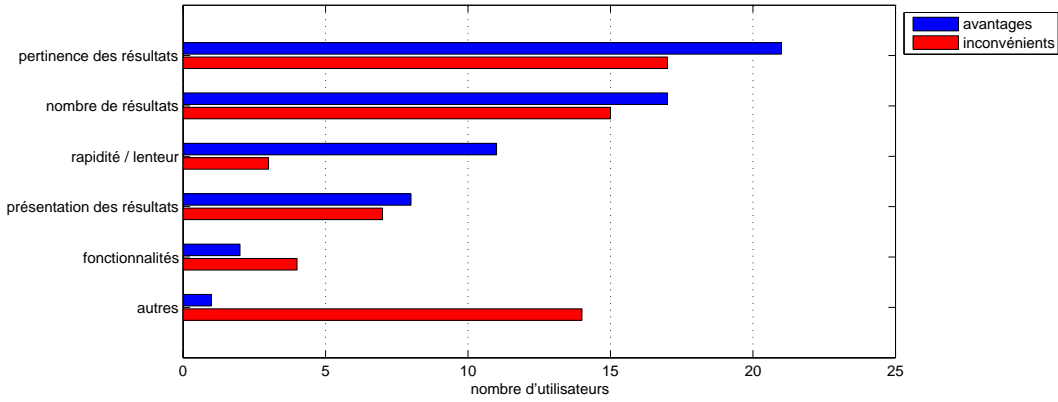


FIG. 5.12 – Avantage et inconvénient majeurs du moteur de recherche utilisé

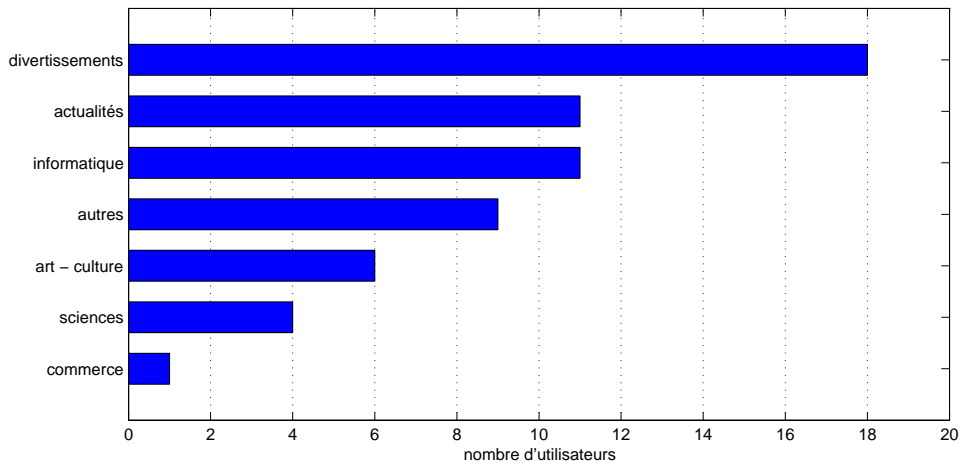
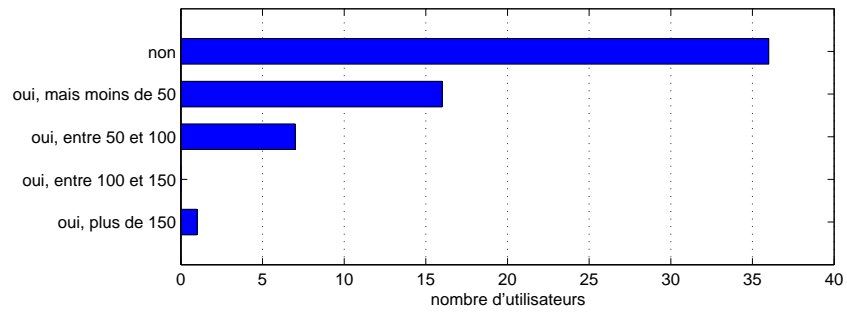
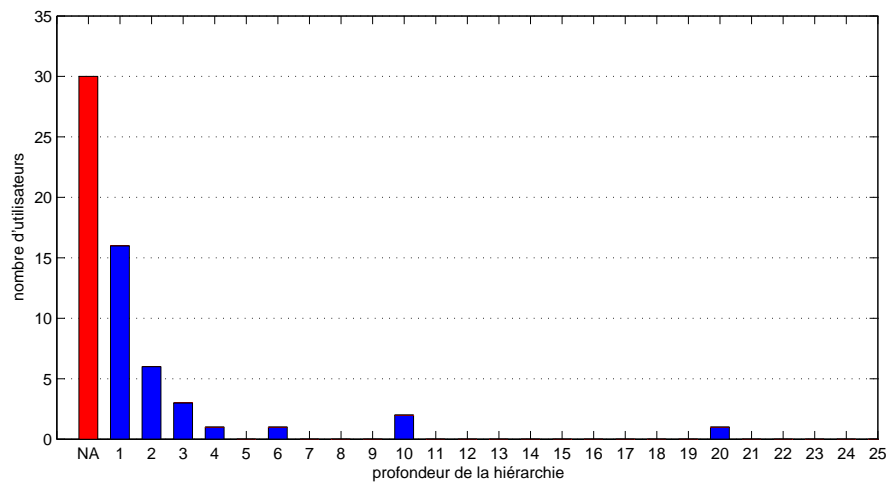


FIG. 5.13 – Catégorie de sites Web préférée par les utilisateurs

FIG. 5.14 – Nombre de *bookmarks* utilisés par les utilisateursFIG. 5.15 – Profondeur de la hiérarchie de *bookmarks*

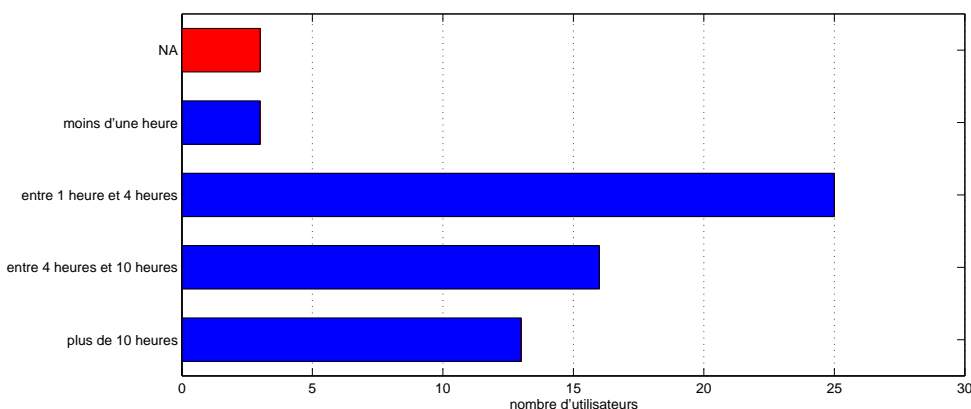


FIG. 5.16 – Temps passé en moyenne par semaine sur un navigateur Web

Nous proposons maintenant d'exploiter les réponses à ces différentes questions afin de caractériser le panel d'utilisateurs. Tout d'abord, ce panel n'est pas représentatif de l'ensemble de la population, comme le montre la distribution des âges (**figure 5.3**) ou la répartition selon le sexe (**figure 5.4**). En effet, comme précisé dans le contexte de cette évaluation (*cf.* 5.1.1), le panel est composé de 40 étudiants, ce qui explique la forte distribution des utilisateurs sur la tranche 20-26 ans. Les autres âges (notamment les personnes ayant moins de 20 ans ou plus de 30 ans) correspondent à certains utilisateurs « atypiques¹ » du troisième jour de notre évaluation. Cette forte concentration d'étudiants, qui plus est en informatique (d'où la répartition des âges), dans le panel explique les résultats obtenus sur les **figures 5.7 à 5.10**. En effet, on constate que les utilisateurs sont, en majorité, très familiers avec l'outil informatique. Cependant on reste dans une utilisation « standard » de l'informatique, ce qui explique le faible nombre d'utilisateurs familiers avec certains « périphériques » spécifiques (**figure 5.8** : *track ball*, stylet, tablette graphique, affichage *via* des lunettes ; **figure 5.10** : reconnaissance vocale). Toutefois un nombre élevé d'utilisateurs sont familiers avec les jeux 3D, ce qui doit être pris en compte lors de l'exploitation des réponses aux questions relatives aux métaphores 3D de visualisation. Il est aussi intéressant de noter que la quasi-totalité des utilisateurs utilisent le moteur de recherche *Google*, c'est-à-dire un affichage linéaire des résultats. Cela est sans surprise et reste représentatif des parts de marché de ce moteur pour la population francophone (avec plus de 80% de parts de marché). Par ailleurs, on constate des contradictions dans les réponses présentées sur la **figure 5.12**. Les utilisateurs considèrent la pertinence des résultats et leur nombre comme les principaux avantages du moteur de recherche qu'ils utilisent, mais aussi comme les principaux inconvénients. Concernant la rapidité du moteur de recherche utilisé, ils semblent satisfaits. Cela n'est pas vraiment le cas pour la présentation des résultats. Enfin ils utilisent tous régulièrement un navigateur Web (**figure 5.16**). Par

¹par rapport à la majorité des utilisateurs constituant notre panel

contre peu d'entre eux utilisent des favoris (**figure 5.14**), et ils sont encore moins nombreux à les organiser (**figure 5.15**). On précise aussi que les **figures 5.5** et **5.6** n'ont pas vraiment d'intérêt ici car, comme mentionné dans le contexte de cette évaluation, aucun utilisateur n'a déjà utilisé le prototype. On précise cependant que le temps de manipulation de notre prototype varie, suivant les utilisateurs, de 30 minutes (pour la majorité des utilisateurs des 2 premiers jours) à une heure (pour certains utilisateurs du troisième jour).

Dans ce type d'évaluation, il faut faire attention à l'interprétation de certaines valeurs. En effet, un certain nombre d'erreurs, de la part des utilisateurs, sont généralement constatées. Il peut s'agir d'erreurs de saisies — réponse « 0 » pour l'âge (**figure 5.3**), réponse « aucun » pour le nombre de systèmes d'exploitation utilisés (**figure 5.7**) — ou d'erreurs de compréhension des questions posées.

En conclusion, la majorité des utilisateurs composant ce panel répondent au profil suivant : « **personne âgée de 20 à 26 ans, familière de l'informatique et notamment des jeux 3D, utilisant principalement le moteur de recherche Google et n'ayant jamais utilisé notre prototype** » .

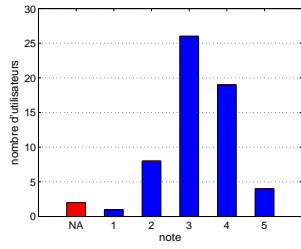
5.1.3.2 Évaluation du prototype SmartWeb

Tout d'abord, les réponses des utilisateurs aux questions relatives au prototype sont synthétisées dans le **tableau 5.2**. Ensuite, nous proposons une représentation graphique (sous la forme d'histogrammes) de ces réponses. Les questions sont regroupées par figures (numérotées de **5.17** à **5.24**) afin de respecter les 8 catégories proposées dans le test : réaction générale au prototype, affichage, terminologie et informations du système, apprentissage, aide en ligne, métaphore de la ville, représentation des images et idées. Puis nous en proposerons une interprétation. On rappelle que les notes vont de 1 (plus mauvaise note) à 5 (meilleure note).

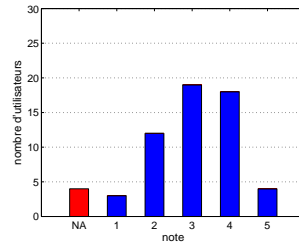
QUESTION		NOTE (n)						\bar{n}
		1	2	3	4	5	NA	
3	RÉACTION GÉNÉRALE À SMARTWEB							
3.1	Comment trouvez-vous SmartWeb ?	1	8	26	19	4	2	3.3
3.2	Comment trouvez-vous SmartWeb ?	3	12	19	18	4	4	3.1
3.3	Comment trouvez-vous SmartWeb ?	2	3	20	22	11	2	3.6
3.4	Clarté du prototype	5	17	22	8	7	1	2.9
3.5	Facilité d'utilisation du prototype	3	12	21	13	9	2	3.2
3.6	Vitesse du système	5	13	14	14	11	3	3.2
3.7	Fiabilité du système	3	14	15	19	3	6	3.1
4	AFFICHAGE							
4.1	Facilité de la lecture des caractères sur l'écran	4	9	15	15	14	3	3.5
4.2	L'affichage contextuel simplifie la tâche ?	3	7	10	25	12	3	3.6
4.3	Clarté de l'organisation de l'information sur l'écran	1	11	22	17	8	1	3.3

QUESTION		NOTE (n)						\bar{n}
		1	2	3	4	5	NA	
4.4	Clarté de la séquence des affichages (requête puis métaphore ville et enfin accès à la page Web ou à la représentation des images)	4	5	20	19	10	2	3.4
5 TERMINOLOGIE ET INFORMATIONS DU SYSTÈME								
5.1	Cohérence dans l'usage des termes à travers le système	0	6	23	14	9	8	3.5
5.2	La terminologie est liée à la tâche	0	6	24	17	1	12	3.3
5.3	Cohérence dans la position des messages sur l'écran	0	2	22	18	9	9	3.7
5.4	Clarté des entrées de texte	1	6	10	12	22	9	3.9
5.5	L'ordinateur informe au sujet de son avancement	3	17	13	10	8	9	3.1
6 APPRENTISSAGE								
6.1	Apprendre à utiliser le système	2	11	12	17	17	1	3.6
6.2	Exploration de nouvelles fonctionnalités par l'essai et l'erreur	1	9	13	21	12	4	3.6
6.3	Se rappeler le nom et l'usage des commandes	0	8	12	23	14	3	3.8
6.4	L'exécution des tâches est directe	0	5	19	24	7	5	3.6
6.5	Destiné à tous les niveaux d'utilisateurs	5	16	18	4	15	2	3.1
7 AIDE EN LIGNE								
7.1	Utilité de l'aide en ligne	0	1	11	8	16	24	4.1
7.2	Clarté de l'aide en ligne	1	4	15	10	3	27	3.3
8 MÉTAPHORE DE LA VILLE								
8.1	Pertinence de la métaphore	3	6	13	24	12	2	3.6
8.2	Facilité dans la compréhension de la métaphore	1	10	11	21	15	2	3.7
8.3	Facilité de la navigation	4	11	16	17	10	2	3.3
8.4	Facilité du déplacement dans la ville	7	16	11	9	14	3	3.1
8.5	Facilité d'accès aux deux barres de navigation	14	12	7	14	12	1	3
8.6	Utilité du plan en haut à droite pour se repérer	12	12	10	13	12	1	3
9 REPRÉSENTATION DES IMAGES								
9.1	Facilité d'accès à la représentation 2D des images d'une page Web	6	12	18	8	13	3	3.2
9.2	Pertinence de la représentation 2D	1	4	12	24	15	4	3.9
9.3	Facilité de la compréhension de la représentation	2	3	18	16	17	4	3.8
9.4	Facilité de la navigation	1	4	9	25	17	4	3.9
9.5	Facilité d'accès à la galerie 3D des images d'une page Web	14	12	12	9	8	5	2.7
9.6	Utilité de la métaphore 3D de la galerie	3	9	7	13	23	5	3.8
9.7	Utilité de la fonction de <i>bookmark</i>	2	5	18	12	13	10	3.6
9.8	Utilité de la fonction de suppression	4	5	15	13	13	10	3.5
10 IDÉES								
10.1	Utilité de l'ajout d'un historique	4	6	9	12	24	5	3.8
10.2	Utilité d'une visualisation plus abstraite sous la forme de groupes de pages Web	1	1	17	11	23	7	4

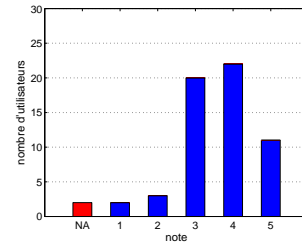
TAB. 5.2 – Résultats de la seconde partie (évaluation du prototype) du test utilisateur réalisé sur 60 personnes d'âges et de professions différents. La valeur 1 correspond à la plus mauvaise note et la valeur 5 à la meilleure note. La colonne NA — pour *Not Applicable* — représente les réponses non exprimées et la colonne \bar{n} correspond à la note moyenne (arrondie au dixième).



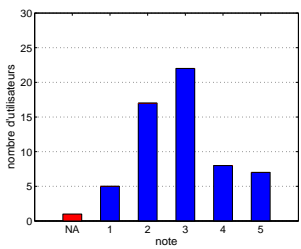
(a) 3.1 : SmartWeb est-il merveilleux (5) ou affreux (1)?



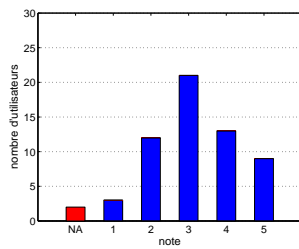
(b) 3.2 : SmartWeb est-il satisfaisant (5) ou frustrant (1)?



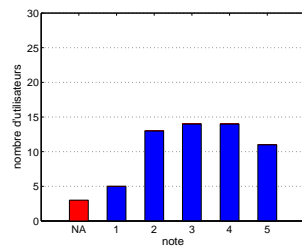
(c) 3.3 : SmartWeb est-il stimulant (5) ou ennuyeux (1)?



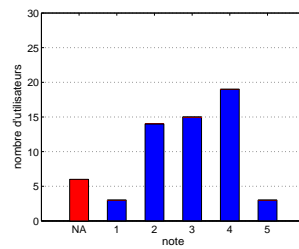
(d) 3.4 : Clarté du prototype SmartWeb



(e) 3.5 : Facilité d'utilisation de SmartWeb

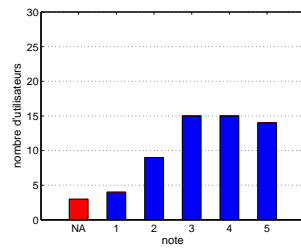


(f) 3.6 : Rapidité du système

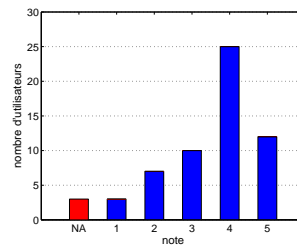


(g) 3.7 : Fiabilité du système

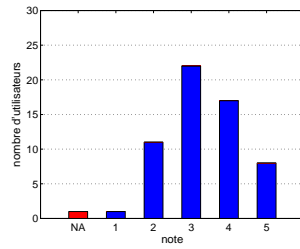
FIG. 5.17 – Réaction générale à SmartWeb



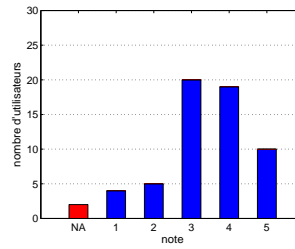
(a) 4.1 : Facilité de lecture des caractères sur l'écran



(b) 4.2 : Simplification de la tâche grâce à l'affichage contextuel

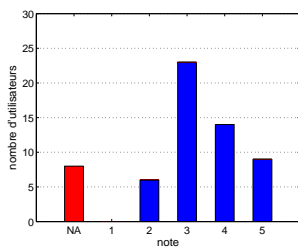


(c) 4.3 : Clarté de l'organisation de l'information sur l'écran

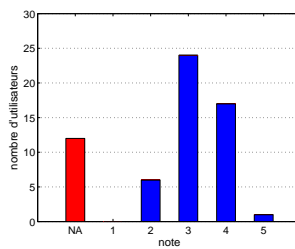


(d) 4.4 : Clarté de la séquence des affichages

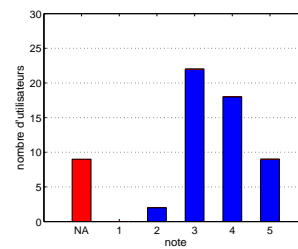
FIG. 5.18 – Affichage



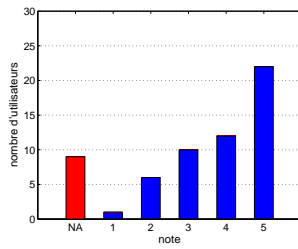
(a) 5.1 : Cohérence dans l'usage des termes à travers le système



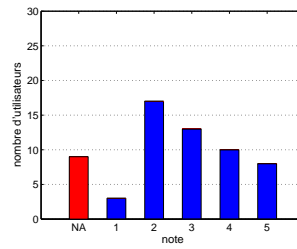
(b) 5.2 : La terminologie est liée à la tâche



(c) 5.3 : Cohérence de la position des messages sur l'écran

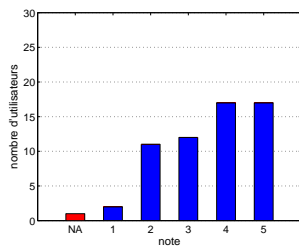


(d) 5.4 : Clarté des entrées de texte

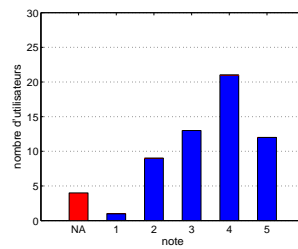


(e) 5.5 : L'ordinateur informe au sujet de son avancement

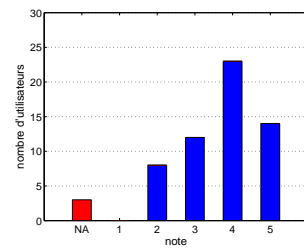
FIG. 5.19 – Terminologie et informations du système



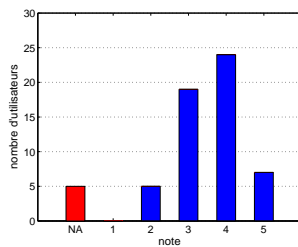
(a) 6.1 : Facilité à apprendre à utiliser le système



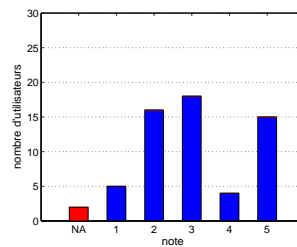
(b) 6.2 : Facilité d'exploration de nouvelles fonctionnalités par l'essai et l'erreur



(c) 6.3 : Facilité de mémorisation du nom et de l'usage des commandes

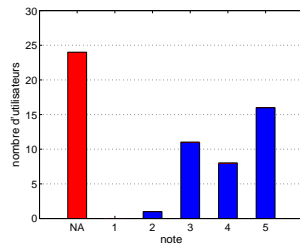


(d) 6.4 : L'exécution des tâches est directe

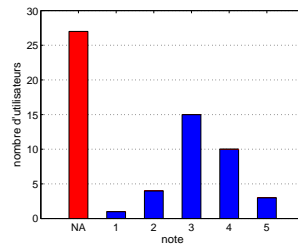


(e) 6.5 : Destiné à tous les niveaux d'utilisateurs

FIG. 5.20 – Apprentissage

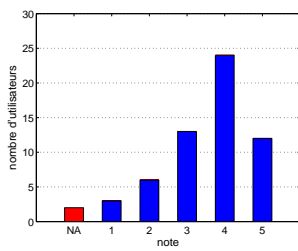


(a) 7.1 : Utilité de l'aide en ligne

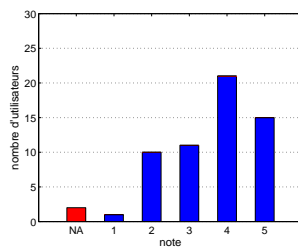


(b) 7.2 : Clarté de l'aide en ligne

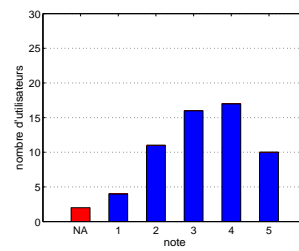
FIG. 5.21 – Aide en ligne



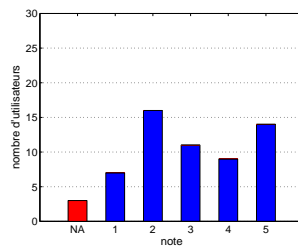
(a) 8.1 : Pertinence de la métaphore



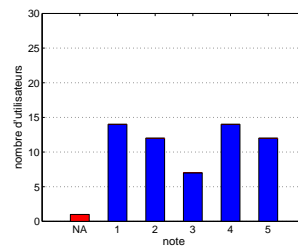
(b) 8.2 : Facilité de compréhension de la métaphore



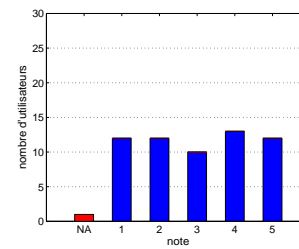
(c) 8.3 : Facilité de la navigation



(d) 8.4 : Facilité du déplacement dans la ville

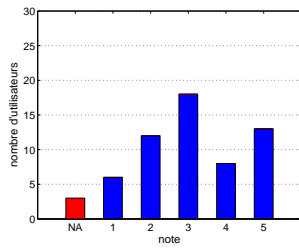


(e) 8.5 : Facilité pour trouver les deux barres de navigation

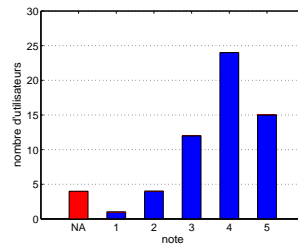


(f) 8.6 : Utilité du plan en haut à droite pour se repérer

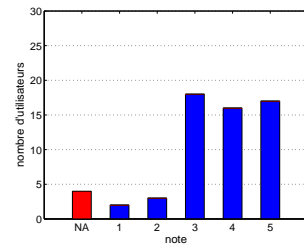
FIG. 5.22 – Métaphore de la ville



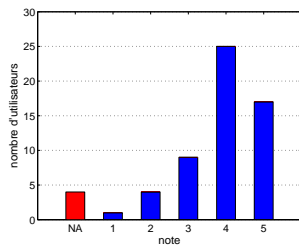
(a) 9.1 : Facilité pour trouver la représentation 2D des images d'une page Web



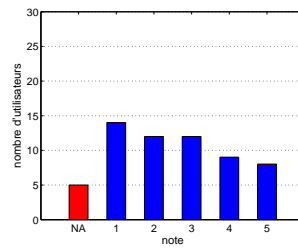
(b) 9.2 : Pertinence de la représentation 2D



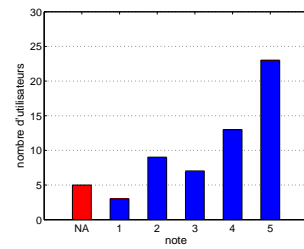
(c) 9.3 : Facilité de compréhension de la représentation



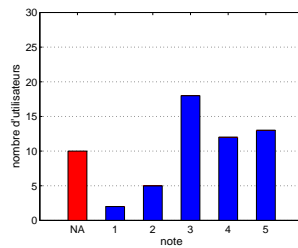
(d) 9.4 : Facilité de la navigation



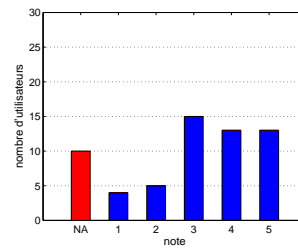
(e) 9.5 : Facilité pour trouver la galerie 3D des images d'une page Web



(f) 9.6 : Utilité de la métaphore 3D de la galerie



(g) 9.7 : Utilité de la fonction de bookmark



(h) 9.8 : Utilité de la fonction de suppression

FIG. 5.23 – Représentation des images

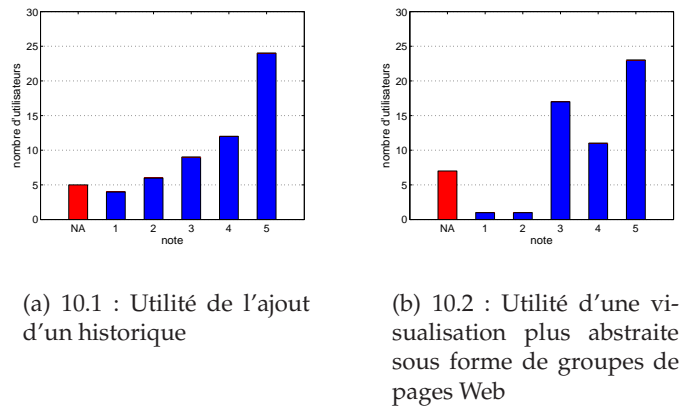


FIG. 5.24 – Idées

D'un point de vue général, il semble que les utilisateurs ont tendance à favoriser les notes centrales au détriment des notes extrêmes. D'ailleurs une idée intéressante serait de proposer une échelle de notes impaire afin de « forcer » l'utilisateur à prendre partie. Ce test révèle cependant que les utilisateurs ne trouvent pas d'inconvénient majeur à utiliser notre interface 3D. Le choix de la métaphore de la ville est d'ailleurs plutôt bien accepté (**figure 5.22(a)**) malgré la nécessité d'améliorer certains points tels que les déplacements dans la métaphore (**figure 5.22(d)**) ou la clarté du prototype (**figure 5.17(d)**). D'après la **figure 5.20(e)**, les capacités de l'utilisateur doivent mieux être prises en compte ou alors il faut mieux cibler le public visé par cette métaphore. Enfin il semble que les utilisateurs soient prêts à utiliser des degrés d'abstraction sur les résultats tels que le *clustering* (**figure 5.24(b)**), ce qui est pris en compte dans la seconde version de la métaphore de la ville (cf. partie 3.3.2.2 du **chapitre 3**). L'interprétation que nous venons de fournir n'est basée que sur un sous-ensemble de questions traitant de points-clés. Il est fastidieux d'essayer d'analyser manuellement toutes les questions sans aucune garantie sur leur possible interprétabilité. C'est pourquoi nous proposons une analyse globale dans la section suivante.

5.1.3.3 Analyse globale des résultats

Il existe d'autres façons d'exploiter les résultats issus de ce type de questionnaire. En particulier, il est intéressant d'analyser les réponses d'un point de vue global afin d'identifier les différentes tendances qui peuvent apparaître. Étant donné le type de données dont on dispose, il semble approprié de réaliser une analyse des correspondances multiples (ACM) [Lebart *et al.* 2000]. Pour cette analyse, nous disposons de 60 individus et de 39 variables. Chaque individu, représenté par une ligne du tableau, correspond à un utilisateur ayant réalisé le test. Et les variables correspondent aux différentes questions posées. Ainsi, chaque variable possède les 6 mêmes modalités dont

5 représentent les différentes valeurs de l'échelle de notation. La dernière modalité est réservée aux utilisateurs qui ne souhaitent pas se prononcer pour une question donnée (modalité NA pour « non exprimé »). Afin d'obtenir des résultats (*i.e.* des plans factoriels) aussi interprétables que possible, nous réalisons une sélection dans les 39 variables. Ainsi, un ensemble de questions représentatif de notre évaluation est défini. Il constitue les variables nominales actives utilisées dans l'ACM. Ces variables correspondent aux 16 questions récapitulées dans le **tableau 5.3**. Les autres questions composent l'ensemble des variables nominales illustratives qui pourront être projetées sur les plans factoriels. Par ailleurs, toutes les modalités ne possèdent pas le même nombre de votes. Par exemple, si l'on considère les 16 questions du tableau 5.3, la distribution des réponses selon les 6 modalités est présentée dans le **tableau 5.4**. On constate alors que les modalités correspondant aux notes 1 et NA sont plus faiblement représentées. Ce constat reste valide sur l'ensemble des questions. Or, lors du calcul de l'ACM, les modalités trop faiblement représentées sont ventilées aléatoirement dans les autres modalités afin de ne pas influencer démesurément sur les résultats. Il nous semble alors plus pertinent de fusionner, dès le départ, les modalités correspondant aux notes 1 et 2. Il faut alors faire attention à l'interprétation des modalités lors de l'analyse des résultats de l'ACM.

		NOTE (<i>n</i>)						\bar{n}
		1	2	3	4	5	NA	
QUESTION								
3.2	Comment trouvez-vous SmartWeb ?	3	12	19	18	4	4	3.1
3.3	Comment trouvez-vous SmartWeb ?	2	3	20	22	11	2	3.6
3.5	Facilité d'utilisation du prototype	3	12	21	13	9	2	3.2
3.6	Vitesse du système	5	13	14	14	11	3	3.2
4.1	Facilité de la lecture des caractères sur l'écran	4	9	15	15	14	3	3.5
4.2	L'affichage contextuel simplifie la tâche ?	3	7	10	25	12	3	3.6
4.4	Clarté de la séquence des affichages (requête puis métaphore ville et enfin accès à la page Web ou à la représentation des images)	4	5	20	19	10	2	3.4
6.1	Apprendre à utiliser le système	2	11	12	17	17	1	3.6
6.5	Destiné à tous les niveaux d'utilisateurs	5	16	18	4	15	2	3.1
8.1	Pertinence de la métaphore de la ville	3	6	13	24	12	2	3.6
8.2	Facilité dans la compréhension de la métaphore de la ville	1	10	11	21	15	2	3.7
8.4	Facilité du déplacement dans la ville	7	16	11	9	14	3	3.1
9.2	Pertinence de la représentation 2D des images	1	4	12	24	15	4	3.9
9.4	Facilité de la navigation dans les images	1	4	9	25	17	4	3.9
9.6	Utilité de la métaphore 3D de la galerie	3	9	7	13	23	5	3.8
10.2	Utilité d'une visualisation plus abstraite sous la forme de groupes de pages Web	1	1	17	11	23	7	4

TAB. 5.3 – Variables actives pour l'ACM

L'ACM est alors réalisée avec le logiciel *Spad* (version 6.0) en considérant les 60 individus comme actifs et seulement les 16 variables sélectionnées précédemment

Modalités	1	2	3	4	5	NA
Nombre total de votes	48	138	229	274	222	49
Moyenne sur le nombre total de votes	3	8.6	14.3	17.1	13.9	3.1

TAB. 5.4 – Distribution des votes sur les 16 questions sélectionnées

comme actives. Suite à la fusion préalable de certaines modalités, les variables nominales ne possèdent plus que 5 modalités dont la modalité NA. Cette dernière est extrêmement peu fréquente par rapport aux 4 autres modalités. Elle risque donc inévitablement d'être explicative du premier axe factoriel. Nous fixons alors le seuil de ventilation des modalités actives à 7%, c'est-à-dire que les modalités dont l'effectif est inférieur à 7% du nombre total d'individus (*i.e.* inférieur à 4, 2) sont ventilées aléatoirement dans les autres modalités. Notre analyse porte sur 16 variables et 5 modalités. Le choix du nombre de valeurs propres à conserver va donc se baser sur la valeur 0,25 (pour la valeur propre). Nous ne conservons alors que les deux premiers axes factoriels étant donné que la première valeur propre vaut 0,283 et la seconde 0,213. Ces deux axes représentent un pourcentage cumulé (de conservation de l'inertie) de 16,51%. L'ensemble des sorties textuelles (*i.e.* résultant de l'ACM) proposées par le logiciel *Spad* sont données dans l'**annexe D**.

Interprétation. Sur le plan factoriel 1×2 , nous avons relié par des segments les différentes modalités de nos 16 variables, à l'exception de la modalité NA (**figures 5.25 à 5.29**). Un premier aperçu de ces graphiques montre que leur interprétation n'est pas vraiment évidente et que les résultats du questionnaire semblent difficilement interprétables (comme cela a déjà été constaté précédemment). Dans un premier temps, nous identifions les individus et les modalités explicatives des deux premiers axes factoriels. Une attention particulière est aussi portée sur leur qualité de représentation sur le plan factoriel 1×2 . Ainsi, comme cela peut être constaté visuellement, l'axe 1 sépare les modalités 1 et 4 qui correspondent aux notes extrêmes (*cf.* **figures D.6 et D.7** de l'**annexe D** où figurent notamment les contributions et les qualités de représentation des modalités 32-3, 35-1, 35-4, 36-4, 44-4, 61-4, 82-4 et 84-4). D'après la **figure D.10** de l'**annexe D**, les individus bien représentés sur cet axe sont ceux associés aux libellés Ind9, Ind10, Ind14 et Ind48. Il s'agit alors d'utilisateurs distribuant des bonnes notes aux questions sélectionnées. La même analyse sur l'axe 2 met en avant les modalités 61-3, 94-3, 96-3, 102-3 et les individus Ind18, Ind33, Ind38, Ind60. Ces utilisateurs sont caractérisés soit par un système de notation favorisant des notes légèrement supérieures à la moyenne, soit par l'attribution de mauvaises notes (*e.g.* l'individu 60). Les variables actives ne sont pas vraiment corrélées. Cependant, l'allure générale de certains segments nous laisse penser une distribution proche des notes pour certaines questions. C'est le cas de la **figure 5.26** qui correspond d'ailleurs à des questions appartenant à la même rubrique (*i.e.* traitant de sujets similaires). Certaines variables illustratives sur le profil des utilisateurs sont également projetées (**5.30**). Elles concernent le sexe (deux modalités : féminin et masculin), l'âge (quatre modalités : NA, 12-16

ans, 20-26 ans et 31-48 ans), la pratique de jeux 3D (deux modalités : oui et non) et le temps moyen passé par semaine sur un navigateur Web (cinq modalités : NA, moins d'une heure, entre une et quatre heures, entre quatre et dix heures, plus de dix heures). La quasi-totalité de ces modalités sont projetées près de l'origine et ne sont donc pas interprétables. On note toutefois que la modalité « 12-16 ans » semble être caractéristique des utilisateurs distribuant des notes relativement faibles. Cela est confirmé par les données brutes. Mais ce constat doit être relativisé étant donné la forte inégalité de la répartition des utilisateurs dans les différentes catégories d'âge. La catégorie « 20-26 ans » est en effet largement sur-représentée par rapport aux autres. Une classification des individus en trois groupes est proposée sur la **figure D.11** de l'**annexe D**. On y retrouve aussi la séparation entre les modalités « excellent » et « mauvais » des variables actives (sur le premier axe factoriel). La classe contenant le plus d'individus semble représentative des modalités centrales, ce que nous avons déjà constaté sur les histogrammes précédents. Enfin, une caractérisation plus fine de ces différentes classes peut être réalisée par les modalités des variables nominales (**figure D.12** de l'**annexe D**).

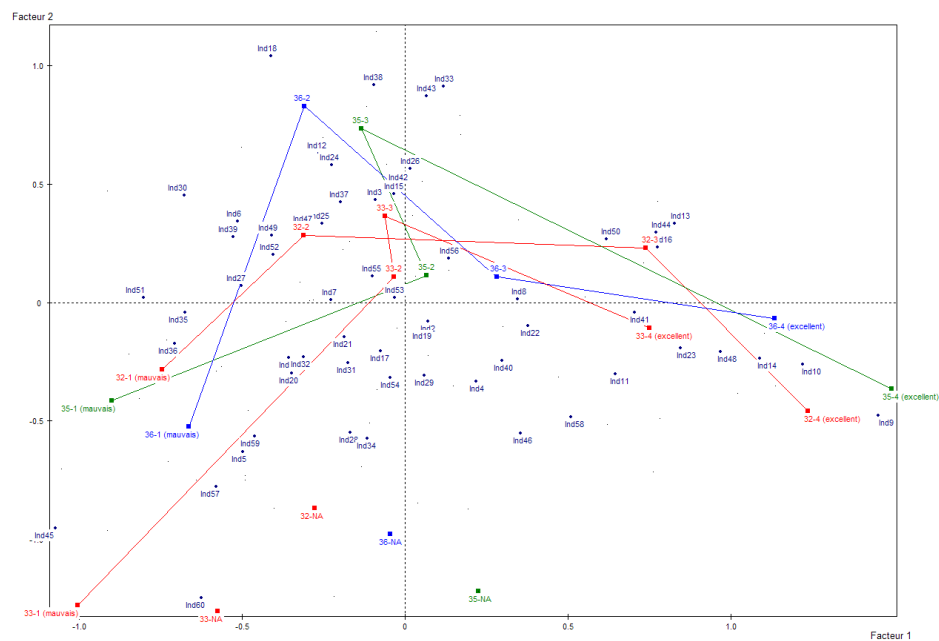


FIG. 5.25 – Plan factoriel 1 × 2 (variables 3.2, 3.3, 3.5 et 3.6)

5.1.3.4 Discussion

Ce test utilisateur initialement basé sur les propositions de [Shneiderman 1998], peut probablement être amélioré en y intégrant des questions plus spécifiques aux interfaces 3D de restitution de résultats de recherche et des scénarios d'utilisation. Ce-

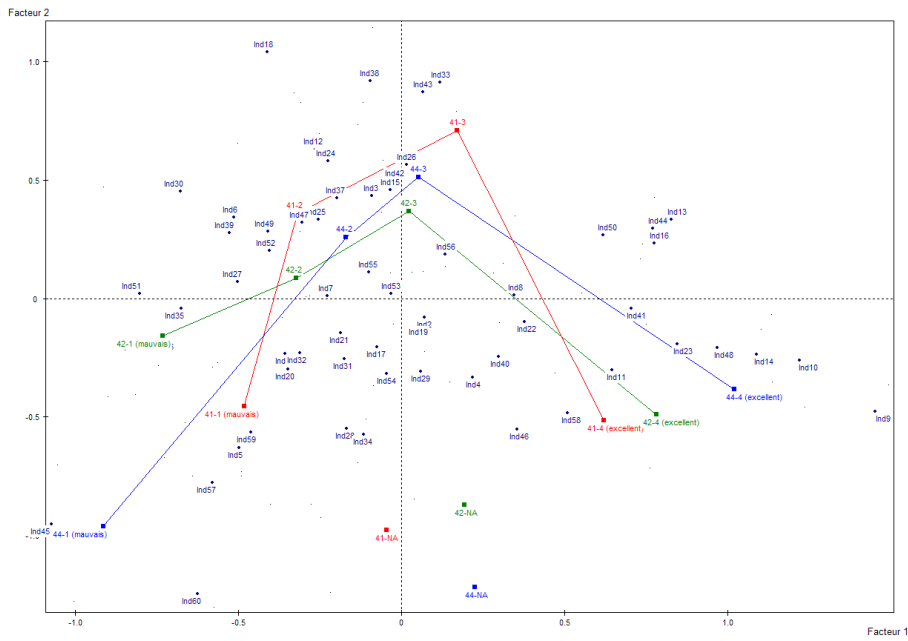


FIG. 5.26 – Plan factoriel 1 × 2 (variables 4.1, 4.2 et 4.4)

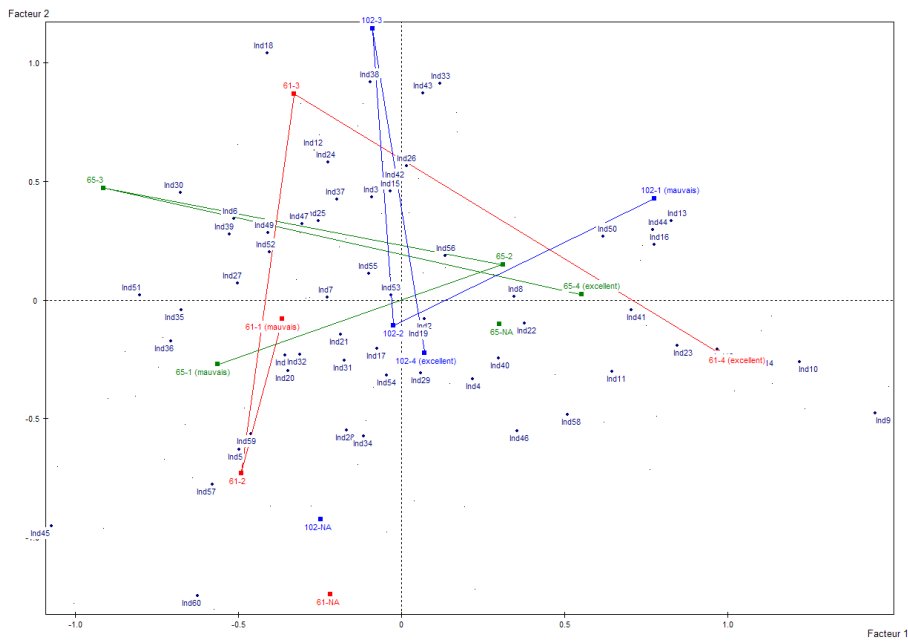


FIG. 5.27 – Plan factoriel 1 × 2 (variables 6.1, 6.5 et 10.2)

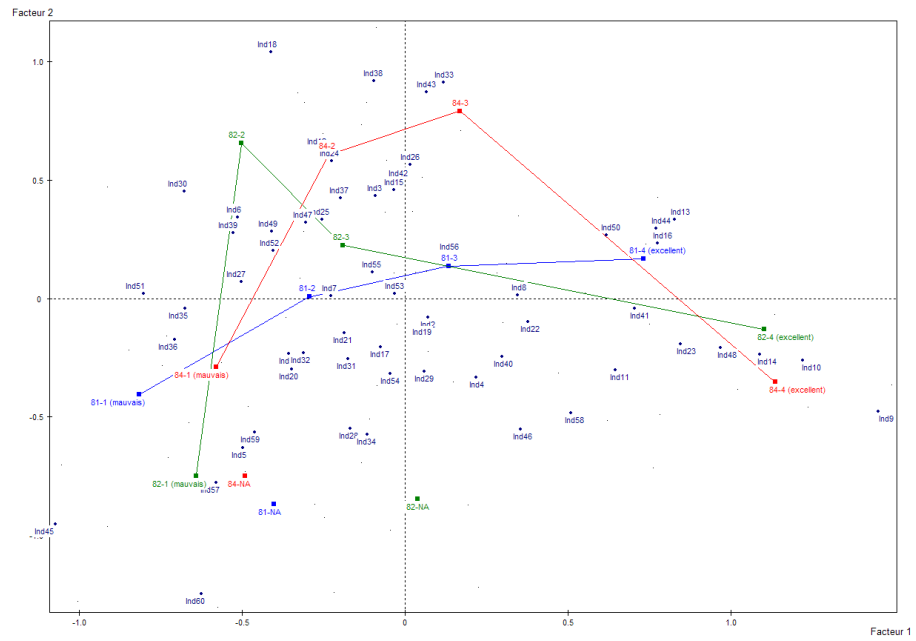


FIG. 5.28 – Plan factoriel 1 × 2 (variables 8.1, 8.2 et 8.4)

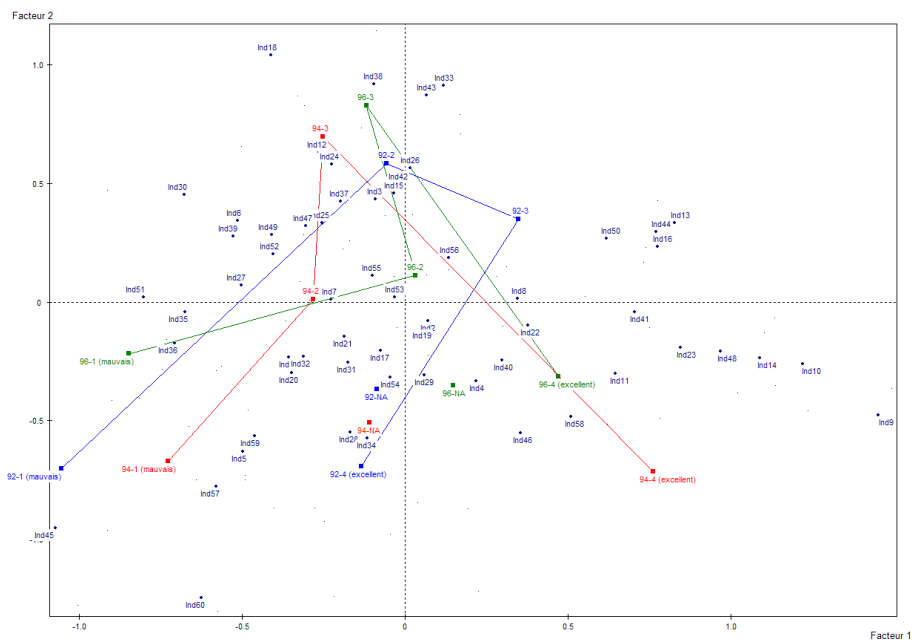
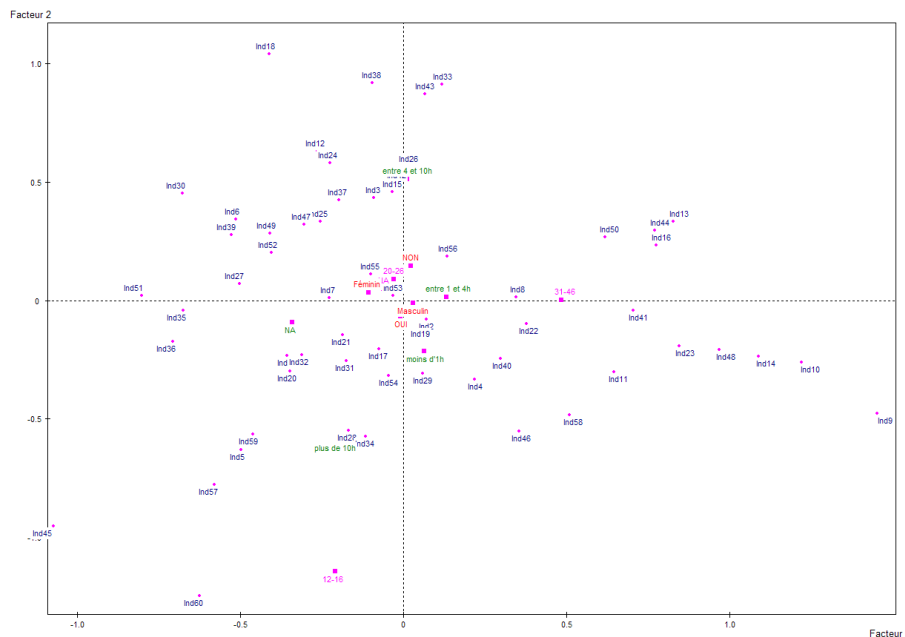


FIG. 5.29 – Plan factoriel 1 × 2 (variables 9.2, 9.4 et 9.6)



pendant, cette méthode est essentiellement orientée utilisabilité et ne nous semble pas suffisante. Elle ne permet notamment pas d'évaluer l'impact de l'IUI sur la tâche accomplie par l'utilisateur. C'est pourquoi nous proposons, par la définition d'un cadre d'évaluation, d'essayer d'évaluer plus pertinemment ces IUI. Cela passe inévitablement par la définition d'un ensemble de critères nous permettant d'évaluer, de comparer et de classer les différentes IUI en fonction des tâches, des utilisateurs... Les propositions faites dans la suite de cette section 5.2 correspondent aux travaux présentés dans les articles intitulés « Évaluation des Interfaces Utilisateur d'Information » et « Critères d'évaluation pour les interfaces des systèmes de recherche d'information » [Bonnell et Chevalier 2006b,a].

De nombreuses IUI ont déjà été proposées (cf. **chapitre 1**) ; certaines sont utilisées alors que d'autres n'ont pas dépassé le stade du prototype. Le problème majeur réside toutefois dans les trop grandes disparités entre ces différentes interfaces, ce qui implique des évaluations très hétérogènes et ce qui rend toute comparaison extrêmement difficile. Afin d'illustrer notre propos, nous rappelons quelques unes de ces IUI. La plus classique et la plus utilisée est l'interface qui propose un affichage linéaire des résultats sous la forme d'une liste triée selon un critère de « pertinence (système) » (e.g. *Google*). Une évolution de cet affichage linéaire est la représentation de classes (ou catégories) de documents sous la forme de listes multiples (e.g. *Grouper* [Zamir et Etzioni 1999] ou *Clusty*). La classification des résultats peut aussi être hiérarchique (e.g. *Grokker*). Cette approche de *clustering* peut également être utilisée dans les interfaces 2D cartographiques, notamment *via* l'utilisation des cartes auto-organisatrices [Lagus *et al.* 1996, Poinçot 1999]. D'autres interfaces proposent des visualisations 2D sous forme cartographique, généralement basées sur les relations qui existent entre les différents résultats (e.g. *KartOO* ou *MapStan*). Dans ce type d'interfaces l'utilisateur peut visualiser les liens entre les résultats et par exemple certains « mots-clés » qu'ils partagent. Enfin certaines IUI vont plus loin dans la restitution des résultats en utilisant pour cela une certaine « spatialisation 3D » des documents restitués. Il existe globalement deux approches : la première est orientée 3D « simple » où l'intérêt de la 3D est de permettre d'obtenir un espace plus important pour l'affichage [Hearst et Karadi 1997, Cugini *et al.* 2000, Chevalier 2002], et la seconde peut être qualifiée d'environnement virtuel 3D au travers duquel l'utilisateur peut « se promener » dans les résultats. Cette dernière approche repose sur l'idée de métaphores 3D cartographiques (e.g. *Smart-Web* [Bonnell *et al.* 2005a, 2006] ou *ViOS*).

Cependant le problème majeur pour l'évaluation de ces IUI est qu'elles ne reposent pas toutes sur le même système de recherche et ne proposent pas toutes les mêmes traitements (tels que le *clustering* ou le filtrage des résultats). De ce fait, identifier la « meilleure » interface est assez difficile surtout que cette appréciation dépend de différents éléments. Nous avons identifié ces éléments permettant ainsi de construire une plate-forme d'évaluation et de comparaison des IUI. Mais avant de présenter ces différents critères, nous présentons dans la section suivante les approches classiques d'évaluation des IUI afin d'en identifier les différentes limites. Ensuite la section 5.2.3

est consacrée à la proposition de critères d'évaluation. Enfin la section 5.2.4 donne des précisions sur le déroulement d'une telle phase d'évaluation, et notamment sur les contraintes à respecter.

5.2.2 Évaluation d'une IUI

La méthode la plus courante pour l'évaluation d'une IUI est la réalisation d'une étude utilisateur. Cette technique consiste à proposer un questionnaire aux utilisateurs afin qu'ils évaluent et notent l'IUI, par le biais d'une série de questions. Ce questionnaire est très souvent inspiré des propositions de Ben Shneiderman en matière d'évaluation d'interfaces utilisateur et est très orienté utilisabilité [Shneiderman 1998]. Ce principe a notamment été adopté lors de l'évaluation du prototype *Easy-DoR* [Chevalier 2002] ou encore de notre prototype *SmartWeb* [Bonnell *et al.* 2006]. Cependant, il s'avère que les résultats de ces tests, dans le cas d'une évaluation d'une IUI, sont difficilement interprétables. En effet, les questions posées admettent souvent une réponse subjective, ne sont pas toujours adaptées au contexte des IUI et ne permettent pas tout simplement d'apprécier l'impact de l'IUI sur le processus de recherche d'information [Chevalier et Hubert 2005]. Or, l'interface utilisateur est essentielle dans un tel processus pour permettre à l'utilisateur d'arriver à ses fins dans de bonnes conditions (si tenté que ce qu'il recherche soit disponible). Ainsi, l'IUI devient un élément aussi crucial que le modèle sous-jacent au SRI. Il est donc important de mesurer l'impact d'une IUI sur le processus mais également de valider les hypothèses émises lors de la conception de l'IUI (par exemple, l'IUI est-elle réellement destinée à des utilisateurs *lambda* ou permet-elle réellement de remplir telle ou telle tâche ?). Il est donc nécessaire de proposer une évaluation, complémentaire des évaluations plus orientées utilisabilité [Fekete et Plaisant 2004], permettant de caractériser plus finement ces IUI dans leur utilisation dans le contexte de la RI. En effet, dans un premier temps, ce n'est pas tant la comparaison intrinsèque des IUI qui nous motive mais plus l'évaluation des IUI en prévision de les classer afin d'identifier, selon le contexte d'utilisation, quelles sont les IUI les plus adéquates.

Deux possibilités concernant cette évaluation peuvent alors être distinguées : l'évaluation d'une IUI utilisant son propre système de recherche (*i.e.* ses propres résultats de recherche) et l'évaluation d'une IUI indépendante de tout système de recherche. Dans le premier cas, nous pouvons retrouver les principaux *benchmarks* qui ont été mis en place dans le contexte de la recherche d'information, et notamment ceux rattachés aux campagnes d'évaluation TREC, CLEF ou encore INEX. Or ces campagnes d'évaluation ne sont pas initialement prévues pour évaluer l'interface utilisateur, ce qui explique l'absence de certains critères spécifiques aux IUI. De plus, dans ces campagnes d'évaluation, les IUI sont évaluées en même temps que le système de recherche propre à chaque participant. Ainsi, même si l'interaction est mise en évidence, l'impact de l'IUI est noyé dans les performances intrinsèques du système. Cette façon d'évaluer ne permet pas, en soi, de définir si une IUI est performante ou non. Ces types d'évaluations supposent que l'intérêt de l'IUI est corrélé à la performance du système.

La question qu'il faudrait alors se poser est : comment se comporterait une bonne interface avec un système peu performant ? Cette question peut faire sourire et nous pensons tous avoir la réponse. Or, nous pouvons penser que l'interface peut « relever » les performances du système. Par exemple, une IUI basée sur une classification des résultats peut proposer des classes tout à fait pertinentes même si le système ne fait pas de la haute précision.

Dans le cadre de la thèse, nous nous intéressons donc essentiellement au deuxième cas d'évaluation, c'est-à-dire à celui où l'évaluation des IUI est réalisée indépendamment du système de recherche. Si jusqu'à présent les systèmes « les plus chanceux » avaient le privilège d'être évalués, les comparaisons de l'impact de ces IUI sur la RI étaient quasi-inexistantes du fait de la trop grande hétérogénéité des systèmes. Pour réaliser l'évaluation des interfaces de manière transparente, il est donc nécessaire de développer un SRI (virtuel) gérant la phase d'évaluation et sur lequel les IUI seront connectées afin d'être évaluées. En plus de ce SRI, différents indicateurs doivent permettre de mesurer la qualité d'une IUI en fonction d'un contexte spécifique (type d'utilisateur, type de résultat. . .). Nous pouvons supposer que ces indicateurs soient principalement liés à l'utilisateur qui peut être caractérisé par trois questions : « Qui est-il ? Que veut-il ? Pour quoi faire ? » [Lainé-Cruzel 1999]. Cette vision est confortée par le fait que certaines caractéristiques de l'utilisateur sont primordiales dans un processus de RI (connaissance du domaine et connaissance pratique) [Hölscher et Strube 2000]. Ainsi d'après ces approches, nous pouvons faire émerger deux catégories de critères qui sont liés soit à l'utilisateur (« Qui est-il ? ») soit à sa tâche (« Que veut-il ? Pour quoi faire ? »). Si nous ne considérons que ces critères nous biaisons l'évaluation ainsi réalisée. En effet, aucun élément du système ne rentre en compte. Cependant comment mesurer l'impact d'une IUI sur un SRI peu performant si l'on ne se concentre que sur l'utilisateur. Pour remédier à cela, nous proposons une autre catégorie de critères liés au système et à l'ensemble résultat des documents. Grâce à ces différents critères nous pouvons ainsi caractériser au mieux l'impact d'une IUI sur le processus de RI en fonction de différents points de vue.

5.2.3 Critères d'évaluation

Dans cette sous-section, nous proposons les critères d'évaluation que nous avons identifiés selon les différents points de vue, permettant de comparer des IUI. La liste de critères proposée n'est pas exhaustive et certains critères peuvent ne pas être adaptés à toutes les interfaces. L'idée étant de faire émerger des relations entre des critères et la réussite ou non d'une recherche d'information. La sélection des critères à prendre en compte doit être effectuée en amont de la phase d'évaluation et doit correspondre à une caractérisation des différentes tâches évaluées. En effet, l'évaluation d'une IUI, pour une tâche donnée, se traduit par la prise en compte d'un certain nombre de critères et éventuellement par une pondération de ces critères. Pour chacun des critères, nous présentons comment, en pratique, il peut être mesuré.

5.2.3.1 Critères liés à l'utilisateur

Il est important de préciser que les critères présentés dans cette partie peuvent être évalués *a priori* lors de la préparation de la phase d'évaluation et notamment lors du choix du panel d'utilisateurs (cf. 5.2.4 et 5.2.4.4).

Critère 1 Niveau de connaissance du domaine lié à l'information recherchée

La réussite d'une recherche d'information dépend entre autres du niveau de connaissance du domaine lié à l'information recherchée [Hölscher et Strube 2000]. En effet, cette connaissance intervient à différents niveaux du processus : lors de la requête, lors de l'analyse des résultats obtenus afin d'identifier ceux qui correspondent à ses besoins. . . Ce critère est donc très important car il traduit en quelque sorte l'expertise de l'utilisateur dans le thème lié aux informations qu'il recherche.

En pratique : – utilisation d'une échelle ouverte comme néophyte, intermédiaire, expert.

Critère 2 Niveau de connaissance pratique de l'outil informatique

Ce critère est lié au critère précédent. Il permet de savoir si un utilisateur sait manipuler l'environnement informatique de la RI. Cela peut aller de la connaissance de l'utilisation de la souris et du clavier, du fonctionnement des SRI ou alors des interfaces 3D. . .

En pratique : – utilisation d'une échelle ouverte comme (néophyte, intermédiaire, experte),
– utilisation d'un questionnaire permettant de classer les utilisateurs dans des classes prédéfinies.

5.2.3.2 Critères liés à la tâche de l'utilisateur

Critère 3 Réussite de la tâche proposée

L'utilisateur doit juger s'il a réussi ou non à réaliser la tâche qui lui était proposée. Cet achèvement peut être binaire ou être associé à une échelle ouverte (note de 1 à 5 par exemple). L'intérêt de ce critère est qu'il permettra de calculer l'adéquation entre l'IUI, la tâche et le type d'utilisateur par exemple.

En pratique : – jugement de satisfaction explicite de la part de l'utilisateur par rapport à la tâche à réaliser. Ce jugement peut être évalué par l'attribution d'une note sur 5 de la part de l'utilisateur.

Critère 4 Temps nécessaire pour terminer la tâche

Ce critère consiste à mesurer le temps mis par l'utilisateur pour terminer la tâche qu'il doit réaliser. Contrairement au critère précédent, il ne prend pas en compte le fait que la tâche ait été réussite ou non. Il apporte simplement une information supplémentaire qui est le temps passé pour réaliser la tâche.

En pratique : – mesure du temps mis par l'utilisateur pour réaliser la tâche (qu'il l'ait réussie ou non).

Critère 5 Ressources nécessaires pour remplir une tâche

Le temps pour achever une tâche n'est pas forcément gage de qualité surtout si l'on considère des usagers peu habitués aux applications informatiques. Ainsi, le nombre d'actions à réaliser ainsi qu'un malus dans le cas d'un retour en arrière pourrait être couplé au temps mis pour terminer la tâche.

En pratique : – comptage du nombre d'actions réalisées par l'utilisateur durant la réalisation de sa tâche (par exemple, clic souris = 1 action et retour en arrière = 2 actions).

5.2.3.3 Critères liés au système

Dans cette catégorie de critères, nous trouvons deux grandes tendances. Les critères liés au système lui-même et ceux qui sont liés à l'ensemble de documents retrouvés. Nous avons choisi de ne pas dissocier ces deux groupes de critères car nous considérons qu'ils dépendent tous deux du SRI utilisé.

Critère 6 Temps de réponse

Il s'agit là d'un critère simple mais inévitable. En effet, la rapidité est une notion fondamentale pour l'utilisateur lors d'une tâche de recherche d'information. Le calcul du temps de réponse du système consiste simplement à mesurer le temps écoulé entre la validation de la requête et l'affichage des résultats (*i.e.* le temps de traitement des résultats par l'IUI). On considère ici que l'affichage des résultats correspond à l'affichage du dernier composant graphique de l'interface. Cependant, il n'est plus suffisant de se contenter d'une seule valeur. En effet, les traitements des IUI sont de plus en plus variés. Les deux traitements les plus fréquents sont généralement l'affichage et le calcul d'un *clustering* à la volée. Dans ce cas, il semble important d'être en mesure de différencier le temps de calcul pour ces deux tâches. Cette différence permet d'effectuer une meilleure comparaison des systèmes et d'identifier les raisons de leurs potentielles lacunes. Cependant, étant donné que toutes les IUI seront évaluées sur le même SRI, seul le temps entre la soumission de la requête et l'affichage effectif des résultats peut être mesuré.

En pratique : – mesure du temps de réponse du système par des compteurs de temps (dans le code).

Critère 7 Compréhension et utilisabilité de l'interface

Il s'agit d'un critère subjectif pouvant être évalué par des questions posées à l'utilisateur sur le fonctionnement de l'interface [Shneiderman 1998, page 136]. Cependant, afin de garantir « l'égalité » des évaluations par rapport à ce critère, les mêmes conditions doivent être respectées : temps d'explication de l'interface, temps de manipulation de l'interface, présence d'une aide en ligne. Dans le cas des IUI, le succès d'une interface passe bien souvent par une compréhension intuitive de l'interface (*i.e.* sans explications pour les fonctionnalités fondamentales). On peut aussi ne pas chercher à évaluer ce critère, considérant qu'il influe plus ou moins dans l'évaluation d'autres critères tels que le temps de recherche de résultats pertinents.

En pratique : – fixer les temps d'explication et de manipulation de l'interface,
– autoriser ou non l'utilisation d'une aide en ligne,
– effectuer un test utilisateur prédéfini en termes de questions et d'échelles de notation. On propose dans un premier temps de se baser sur les propositions de Ben Shneiderman avec une échelle de notation en 5 points. Cependant, une échelle de notation comportant un nombre pair de notes permettrait d'éviter les tendances centrales.

Critère 8 Temps pour retourner à un document précédemment identifié

Ce critère mesure le temps nécessaire à l'utilisateur pour qu'il retrouve un document qu'il a précédemment identifié (ou qu'il connaît) dans l'interface proposée. Il peut s'agir d'un résultat que l'on vient de voir dans l'espace des réponses d'une requête (*i.e.* on n'a pas quitté l'espace des réponses) ou d'un document dont on sait qu'il appartient aux résultats d'une requête (*i.e.* on a quitté l'espace des réponses pour y revenir plus tard). Ce critère prend donc aussi en compte la capacité de mémorisation de l'utilisateur, qui peut notamment être améliorée dans le cas de métaphores de visualisation spatiales (utilisation de notre mémoire spatiale).

En pratique : – identification d'un document que l'utilisateur connaît (soit parce qu'il s'agit d'un résultat déjà vu dans les réponses d'une requête, soit parce qu'il s'agit d'un document que connaît l'utilisateur),
– mesure du temps mis par l'utilisateur pour le retrouver (requête + navigation dans l'espace des résultats),
– mesure du nombre d'actions réalisées par atteindre ce but.

Critère 9 Capacités de personnalisation et d'adaptation

Ce critère indique si l'interface est personnalisable et/ou adaptative ? La personnalisation concerne la possibilité offerte à l'utilisateur de modifier certains paramètres de l'interface alors que l'aspect adaptatif est relatif à la prise en compte par l'IUI de certaines connaissances qu'elle possède de l'usager. Dans le cas où l'IUI est personnalisable, les points suivants pourraient alors être considérés : quelles parties (ou fonctions) sont personnalisables, quelles compétences sont nécessaires du point de vue de l'utilisateur, quel est le temps nécessaire pour l'utilisateur pour réaliser ces personnalisations, quel est l'apport de la personnalisation. . . Concernant le critère d'adaptation, on pourrait entre autres vérifier si l'interface exploite les précédentes requêtes formulées par l'utilisateur pour adapter, réorganiser les résultats [Lainé-Cruzel 1999]. Un autre point consiste à vérifier si l'IUI est en mesure de présenter la représentation de l'usager qu'elle possède. Ce dernier point est prédominant pour une bonne acceptation de la personnalisation proposée. Et là aussi il faudrait pouvoir identifier l'apport de l'adaptation (le résultat est-il plus pertinent ? . . .). Ce critère de personnalisation et d'adaptation de l'IUI reste cependant trop dépendant de nombreux aspects (tels que du type d'interface ou de personnalisation proposés) pour tenter une généralisation de l'évaluation de ce critère.

En pratique :

- *l'utilisation d'un questionnaire pour les réalisateurs des IUI est envisageable,*
- *évaluation de l'ensemble des critères proposés avec et sans l'aspect adaptatif (respectivement personnalisable) de l'interface. Ainsi, cela permet de mettre en avant le réel apport de ces fonctionnalités pour l'utilisateur.*

Les critères qui suivent sont des indicateurs liés à l'ensemble de documents retrouvés.

Critère 10 **Nombre de documents retrouvés**

Ce critère indique le nombre de documents retournés par le SRI. En effet, une IUI peut être efficace pour un nombre limité de documents retrouvés et ne plus l'être pour un nombre plus important.

En pratique : – *simple comptage des documents retournés par le SRI.*

Critère 11 **Longueur des documents retrouvés**

Ce critère indique la longueur (valeur moyenne et écart-type) des documents retrouvés. Cette longueur peut être mesurée, par exemple, par un comptage du nombre de mots présents dans les documents.

En pratique : – *analyse à la volée des documents retrouvés.*

Critère 12 Répartition des documents

Ce critère indique si les documents retrouvés sont homogènes par exemple en ce qui concerne le thème lié à l'information recherché. En effet, si l'on considère des documents textuels, il est important de vérifier s'ils traitent tous du même thème ou pas. Cela permet de mettre en évidence si les IUI permettent à l'utilisateur de s'orienter rapidement vers les documents du thème lié aux informations qu'il recherche.

En pratique : – analyse du centre des différents clusters obtenus à la volée sur les documents retrouvés (s'ils sont proches on peut considérer que la répartition est homogène sinon elle est hétérogène).

Critère 13 Type de documents

Ce critère indique quels types d'informations sont restituées par le SRI. Par exemple, il peut s'agir d'informations textuelles seules ou d'informations semi-structurées comme XML (avec leur structure). En effet, le but est d'identifier si les IUI sont adaptables selon le type d'information. Une IUI peut tout à fait être adaptée pour présenter des informations textuelles mais peut ne pas l'être pour des informations ayant une structure type XML et ce du fait de la métaphore proposée.

En pratique : – réalisation d'une étude a priori du fonctionnement du SRI.

Critère 14 Performances intrinsèque du système

Ce critère indique si le résultat est un résultat très pertinent ou pas. Cela traduit l'efficacité entre autre du SRI.

En pratique : – mesure fixée a priori et associée à chaque requête de la phase d'évaluation (rappel/précision, F-mesure...).

Pour obtenir des résultats interprétables, ces différents critères devront être choisis et combinés pour mesurer l'adéquation entre l'IUI, la tâche de l'utilisateur et le type d'individu. Le choix des critères à prendre en compte dépendra de la tâche à réaliser ainsi que du contexte. Ainsi, chacune des tâches identifiées dans la phase d'évaluation doit être associée à ces critères organisés selon leur importance dans la tâche. Par exemple, si l'utilisateur recherche un document précis, le critère de temps de réponse peut être prédominant sur le reste. Ce critère est encore plus important pour la tâche si on la considère dans un contexte autre que les SRI classiques et notamment sur le Web.

5.2.4 La phase d'évaluation

Afin de pouvoir évaluer et comparer efficacement différentes IUI, un certain nombre de contraintes doivent être respectées. Ces contraintes doivent être prises en compte dès la conception de l'interface afin de pouvoir garantir leur respect. Ainsi l'IUI doit être :

- **indépendante** du système de recherche (*i.e.* des données). Il est en effet nécessaire que l'IUI accède aux données *via* des interfaces proposées par les systèmes de recherche et libres d'utilisation.
- **modulable** en termes de fonctionnalités, c'est-à-dire que les fonctionnalités doivent être adaptables au système de recherche utilisé. En effet, l'interface peut être destinée à plusieurs outils ou applications.
- **capable** de gérer des jeux de requêtes prédéfinis.

Ces différents aspects sont présentés dans les sous-sections suivantes.

5.2.4.1 Données

Le premier point à vérifier est de s'assurer que les interfaces traitent le même ensemble de résultats, c'est-à-dire que les données visualisées sont identiques. Dans notre cas, les résultats peuvent être déclarés identiques à condition de respecter trois égalités : celle du contenu, celle du nombre et celle des descripteurs disponibles (*i.e.* les mêmes informations de base sur les documents sont disponibles mais chaque IUI peut les traiter différemment). La solution la plus simple pour répondre à cette contrainte fondamentale à l'évaluation et à la comparaison de deux IUI, est d'utiliser les API fournies par certains moteurs de recherche. Un système de recherche de référence pourrait être proposé si un moteur de recherche proposait une API suffisamment riche en termes de fonctionnalités. Malheureusement, les API existantes ne donnent pas accès à l'ensemble des descripteurs disponibles dans les bases d'index. Une solution pourrait alors être la proposition d'un système de recherche destiné à l'évaluation des IUI et fournissant une API riche. Il faudrait que l'API propose des fonctions d'accès suffisamment riches pour que chaque IUI y trouve son compte. Cependant, des besoins supplémentaires de la part d'une IUI peuvent exister, ce qui se traduirait par une différence en ressources nécessaires. Dans ce cas, il faudrait aussi que l'API soit suffisamment ouverte pour la faire évoluer et y intégrer ces nouveaux besoins afin d'améliorer le cadre d'évaluation et de comparaison de ces interfaces.

5.2.4.2 Fonctionnalités des IUI

Il est nécessaire que les différentes IUI évaluées puissent s'adapter en fonction de l'API proposée ; les fonctionnalités de ces IUI étant très souvent dépendantes des informations fournies par l'API. Ainsi, en supposant que les moteurs de recherche adoptent une API partagée, l'utilisation d'une IUI avec n'importe quel moteur sera désormais possible. Ce n'est pas le cas aujourd'hui est cette vision reste à l'heure actuelle utopique.

5.2.4.3 Jeux de requêtes

Il est important que chaque IUI exploite les mêmes jeux de requêtes permettant de couvrir les différentes tâches que l'on souhaite évaluer. Par exemple, en recherche d'information, nous pourrions identifier les tâches suivantes :

- Recherche d'un élément précis (exemple : existe-t-il une page Web professionnelle de Fabienne Moreau ?). En général le résultat peut être assimilé à une information booléenne dans le sens où l'utilisateur cherche une information très précise.
- Recherche de l'existence d'une information *a priori* connue (exemple : je recherche les sites Web parlant des lois de Murphy).
- Recherche exploratoire, c'est-à-dire panorama d'un thème donné (exemple : je recherche tout ce qu'il existe dans le domaine de la gestion des connaissances).

Chacune de ces tâches n'implique pas, par leur nature, la même restitution en termes d'IUI. Ces tâches doivent être définies et caractérisées avant la phase d'évaluation car le jeu de requête devra répondre aux différentes tâches identifiées. Ensuite, pour chacune de ces tâches, différents ensembles de documents résultats devront être proposés. Le choix de ces ensembles de documents résultats est primordiale car il doit couvrir les différentes possibilités en termes de caractéristiques du système (ensemble de documents restreint ou très important, rappel très faible ou très élevé...).

5.2.4.4 Les évaluateurs

Le choix des « évaluateurs » devra également être réalisé avec soin pour obtenir un panel d'utilisateurs représentatif des catégories d'utilisateurs pour lesquelles les IUI vont être testées. Ainsi, les utilisateurs pourront par exemple être répartis en neuf classes selon les caractéristiques des utilisateurs en RI et ce en respectant les échelles choisies pour représenter les critères énoncés ci-dessus (cf. 5.2.3.1 et 5.2.3.2) : néophyte, intermédiaire ou expert.

5.2.5 Conclusion

Il existe un grand nombre d'outils de recherche d'information qu'il s'agisse de SRI spécialisés ou de moteurs de recherche sur le Web. Ces différents outils tentent de répondre au mieux aux attentes des utilisateurs aussi variés que les besoins en information qu'ils tentent d'assouvir. La stratégie actuelle est de faire progresser les outils en termes de fonctionnalités intrinsèques (amélioration du processus d'indexation, adaptation...). Une fonctionnalité liée à l'utilisateur reste la conception de l'interface. Cette IUI est la passerelle directe entre l'outil de recherche et l'utilisateur au travers de laquelle les documents retrouvés sont présentés. Ces IUI doivent ainsi être bien réfléchies pour permettre à une catégorie d'utilisateurs de remplir des tâches spécifiques. Cependant, à l'heure actuelle nous ne sommes pas en mesure de vérifier si au moins une IUI répond bien aux attentes. Le besoin en évaluation est donc important pour vérifier ces hypothèses mais également pour vérifier si éventuellement une IUI peut remplir d'autres tâches non prévues initialement ou si elle est adaptée à d'autres catégories d'utilisateurs.

Par ailleurs, cette évaluation permet en outre de réaliser une comparaison et une classification des IUI en fonction des tâches qu'elles permettent de remplir par exemple. Nous proposons dans ce contexte une approche de l'évaluation des IUI au regard de différents critères. Ces critères permettent la caractérisation des tâches que l'on souhaite évaluer. Grâce aux évaluations et aux mesures ainsi calculées au travers des différentes évaluations, une synthèse pour chaque IUI pourra être fournie. Cependant, la phase d'évaluation n'est pas si simple à réaliser et surtout à organiser [Fekete et Plaisant 2004]. Nous devons désormais poursuivre notre réflexion sur les critères d'évaluation et la caractérisation des différentes tâches à partir de ceux-ci. Nous devons également approfondir le mode opératoire de cette phase d'évaluation afin d'obtenir et de valider les résultats que nous espérons aussi objectifs que possible.

5.3 Discussion

L'évaluation d'interfaces utilisateur de SRI est loin d'être une problématique nouvelle. Des études réalisées dans ce domaine montrent les difficultés pour évaluer l'efficacité des SRI du fait, entre autres, du concept évasif de pertinence et de l'hétérogénéité des systèmes [Brajnik *et al.* 1996]. Les aspects quantitatifs classiques (*e.g.* le rappel et la précision) ne sont plus suffisants et doivent désormais être accompagnés d'aspects qualitatifs — plus difficiles à définir — tels que la perception de l'utilisateur et les aspects cognitifs. Généralement, ces études concluent en proposant des recommandations intéressantes pour l'évaluation et la conception des IUI. Mais l'absence de synthèse constructive de toutes ces recommandations se fait ressentir.

Cette problématique se retrouve dans le domaine plus général de la visualisation de l'information, où de nombreuses méthodes d'évaluation des IHM ont déjà été proposées et sont régulièrement utilisées⁶. L'intérêt et l'importance de cette thématique sont désormais pleinement reconnus, comme en témoigne les travaux de Jean-Daniel Fekete d'après lesquels nous tirons le constat suivant [Fekete 2005]. Les méthodes d'évaluation actuelles, qui trouvent leurs origines dans les domaines de la psychologie, de la sociologie et de l'informatique plus théorique, sont insuffisantes pour évaluer certains aspects des systèmes de visualisation de l'information tels que l'apprentissage ainsi que les analyses et les découvertes non triviales. Bien entendu, des *benchmarks* ont été mis à disposition au cours de ces dernières années afin d'augmenter la portée scientifique des comparaisons qui s'en trouvent moins biaisées. Mais cela ne constitue qu'une étape et ne remplace pas le manque de critères efficaces. Le Graal serait alors des « modèles liés à la perception, à la cognition et à la boucle perception/action permettant de prévoir les performances et caractéristiques techniques de visualisations et d'interactions avant même de les avoir implémentées » (Jean-Daniel Fekete). En attendant la fin de cette quête, l'évaluation empirique a de beaux jours

⁶CHARM — *Choosing Human-Computer Interaction Appropriate Research Methods* — <http://www.otal.umd.edu/charm/>

devant elle.

Nos propositions (faites à la section précédente) ne sont pas pourvues de toute cette ambition. Il s'agit simplement d'une première étape de travaux exploratoires ayant pour but de proposer un cadre d'évaluation des IUI le plus objectif possible. Ce dernier repose sur un ensemble de critères, non exhaustifs, prenant en compte l'utilisateur, la tâche à réaliser, l'interface et le système sous-jacent. Il doit alors permettre de classer et de comparer les différentes IUI proposées, à condition de les évaluer indépendamment du système de traitement de l'information utilisé. Ces premières propositions sont orientées sur la visualisation de résultats de recherche, mais elles peuvent être adaptées à d'autres domaines tels que la fouille visuelle de données. Pour cela, il faut toutefois introduire de nouveaux critères afin de prendre en compte de nouvelles caractéristiques telles que l'interaction dans le processus de visualisation. Un outil d'aide à l'évaluation d'IUI, incluant ces différents critères, pourrait alors s'avérer utile. Mais, dans un premier temps, il nous faut appliquer ces propositions à notre interface afin d'en fournir une seconde évaluation que nous espérons plus interprétable, et afin d'ajuster nos critères.

Conclusion

Bilan

Les systèmes de recherche d'information ne doivent plus seulement se préoccuper de la qualité des résultats retournés par le système, bien que cela reste un problème majeur. Les SRI doivent désormais prendre en compte la restitution des résultats à l'utilisateur aussi bien du point de vue de l'organisation que de ceux de la visualisation et de l'interaction. Dans cette thèse, nous présentons une méthode d'organisation et de visualisation tridimensionnelle des résultats issus d'une requête sur un moteur de recherche. L'organisation repose sur une carte auto-organisatrice particulière de par les pondérations utilisées. Une étape de pré-traitement permet d'ajuster la représentation des données (sélection et pondérations des mots et des documents). Puis une étape de post-traitement basée sur une CAH appliquée sur les différentes unités de la carte, permet d'avoir des groupes plus représentatifs. Ainsi, la méthode proposée se décompose en trois niveaux : un premier niveau avec la visualisation des documents, un second niveau avec la visualisation des neurones (regroupements de documents similaires) et un troisième niveau avec la visualisation des « thématiques » majeures (regroupements de neurones). L'idée consiste à dégager les différents sens de la requête en montant dans la hiérarchie et à permettre à l'utilisateur de « filtrer visuellement » les résultats. Concernant les temps d'exécution, ces algorithmes (SOM et CAH) sont réputés pour être assez coûteux. Bien que cet aspect ne soit pas une priorité de nos travaux, les hypothèses faites sur le nombre de documents ainsi que les nombreuses optimisations techniques ou algorithmiques existantes permettent de rester dans des temps raisonnables. Ensuite, nous proposons une visualisation 3D basée sur la métaphore de visualisation de la ville. Cette métaphore, empruntée au monde réel, se révèle être très efficace pour représenter des données préalablement organisées. Ainsi, le plan du cadastre de la ville correspond à la carte auto-organisatrice calculée à partir des résultats de recherche. Ces derniers sont représentés par des bâtiments dont les textures sont des copies d'écran des pages Web respectives. Cela fait déjà plusieurs années que cette information (permettant la pré-visualisation des résultats) est intégrée dans notre métaphore de ville virtuelle ; alors qu'à l'époque les moteurs de recherche ignoraient cette information. Aujourd'hui elle est présente dans de nombreux moteurs de recherche ! La métaphore de la ville permet donc d'identifier rapidement les différents documents grâce à la présence d'une vue d'ensemble et de textures représentatives du contenu des documents. Ces tex-

tures rendent même possible l'identification de documents pertinents sans avoir à utiliser le mécanisme d'essai-erreur (chargement d'une page Web, identification de son contenu puis retour à la liste des résultats). Cette métaphore a également l'avantage de tirer profit de la mémoire spatiale des utilisateurs. L'ensemble de nos travaux sont intégrés dans un prototype nommé *SmartWeb*. Ce prototype est doté d'un ensemble de caractéristiques particulières. Il dispose de différentes métaphores de visualisation des résultats de recherche, dont la principale est celle de la ville. Son fonctionnement est similaire à un moteur de recherche classique ; seule la partie restitution des résultats est modifiée. *SmartWeb* propose des interfaces hybrides, c'est-à-dire composées d'une partie exclusivement en 2D et d'une autre en 3D. Cela est possible grâce à un couplage entre la scène 3D et l'interface 2D afin de permettre l'échange d'information entre les deux parties. Les interfaces graphiques (ainsi que les interactions qui y sont associées) sont générées dynamiquement grâce à l'interpréteur X-VRML. En effet, les métaphores sont exprimées au moyen de ce nouveau langage : le X-VRML. Ainsi, la métaphore sélectionnée pour représenter les résultats est interprétée en ligne afin de générer l'interface 2D et la scène 3D (génération d'un fichier VRML). Une étude utilisateur du prototype avec la métaphore de la ville a été réalisée en se basant sur les propositions de Ben Shneiderman en matière d'évaluation d'IHM. Les résultats sont intéressants mais ils nous paraissent difficilement interprétables. Cette simple approche est insuffisante pour être représentative de la réelle utilisabilité de l'IHM. Par ailleurs, dans le cadre d'interfaces 3D, il pourrait être utile de laisser un temps d'apprentissage à l'utilisateur (étant donné le manque d'expérience de la majorité des utilisateurs dans la manipulation d'interfaces 3D).

L'aspect multi-thématique de nos travaux nous a amené à étudier différents domaines de recherche : l'organisation de documents textuels, les interfaces de visualisation 3D et leur évaluation, la recherche d'information textuelle. C'est pourquoi il nous semble essentiel de préciser les points clés de nos travaux qui sont :

- une taxonomie des IUI⁷ basée sur l'organisation visuelle des résultats et qui s'affranchie donc des différentes techniques ou caractéristiques,
- une adaptation de l'algorithme des cartes auto-organisatrices afin de répondre au mieux aux besoins de notre cadre applicatif (pré-traitement des données, pondération des documents et exploitation des différents groupes obtenus *via* une CAH),
- une métaphore de visualisation 3D basée sur le concept d'une ville virtuelle, ce qui offre une organisation efficace des documents aussi bien que des facilités dans l'exécution de certaines tâches (grâce à l'identification rapide des documents, à une utilisation accrue de la mémoire spatiale, ...),
- un prototype basé sur l'utilisation d'un nouveau langage — le X-VRML,
- un cadre d'évaluation pour améliorer les tests d'utilisabilité des IUI.

⁷Les IUI sont considérées dans le cadre de la restitution des résultats de recherche.

Nos travaux sont encourageants mais l'absence de réelle expérience dans ce domaine⁸ est un handicap. On se doit alors de rester prudent et critique par rapport à nos travaux, au risque de n'avoir apporté qu'un prototype supplémentaire. Ainsi, les remarques des utilisateurs, suite au test réalisé ainsi qu'aux nombreuses démonstrations du prototype, nous ont permis de confirmer nos doutes sur certains aspects du prototype. Bien que la métaphore de la ville soit globalement bien appréciée du public, la navigation dans un tel espace doit être améliorée si l'on souhaite réduire le temps d'exécution de certaines tâches. En effet, pour la recherche d'une information n'apparaissant pas dans les premières pages des traditionnelles listes de résultats, notre interface peut s'avérer être plus efficace (grâce notamment à la présence d'une vue d'ensemble des différents documents et « thématiques » ainsi qu'à l'organisation spatiale des résultats). Par contre, pour la recherche d'une information présente dans les premières réponses d'une requête, notre interface est moins performante que les listes de résultats. En effet, le prototype *SmartWeb* ne biaise pas l'affichage des réponses en favorisant une thématique particulière (*i.e.* en faisant un *a priori* sur les besoins de l'utilisateur). Il faut donc chercher à améliorer notre approche ou remettre en question notre cadre applicatif. Ces différents points sont abordés dans les perspectives.

Perspectives

D'abord, nous proposons des perspectives immédiatement issues de nos travaux dans le but de les améliorer. On distingue alors les perspectives liées à l'organisation des résultats de celles liées à la métaphore de visualisation. Concernant l'organisation, il serait intéressant de rendre l'algorithme interactif. Actuellement la seule solution proposée dans le cadre de la RI est de recalculer une nouvelle carte auto-organisatrice en prenant en compte les choix réalisés par l'utilisateur sur la carte initialement proposée [Hatano *et al.* 1999]. Ce processus d'interactivité pourrait être amélioré, soit en permettant à l'utilisateur d'interagir au cours du calcul de la carte, soit en optimisant le calcul d'une nouvelle carte basée sur les choix effectués par l'usager. Par ailleurs, il est possible d'avoir des hiérarchies de SOM et de ne pas être obligé de prédéfinir préalablement la taille des différentes cartes (*cf.* section 2.2.4). Cette approche mériterait d'être adaptée au cadre applicatif de cette thèse, à savoir la visualisation des résultats de recherche. Cependant, cette proposition implique de faire très attention à la complexité algorithmique et aux temps de calcul. Un échec sur ces points est suffisant pour rendre caduc cette solution dans ce cadre applicatif qui nécessite des calculs en ligne et une interaction temps réel. Une autre amélioration possible concerne nos règles de pré-traitement. Pour cela, il faut notamment prendre en compte des corpus liés à différents types de recherche. Bien entendu, l'étiquetage d'une carte auto-organisatrice n'est pas toujours significatif pour les utilisateurs. Il s'agit d'un problème connu et identifié. Son amélioration est alors un sujet de recherche à part entière. Actuellement, l'approche statistique semble

⁸Tous les précédents prototypes de visualisation 3D de données abstraites sont restés au stade de prototypes ou d'applications « petit public ».

atteindre ses limites. Il faudrait alors peut-être se tourner vers la prise en compte d'informations supplémentaires (e.g. des métadonnées ou des informations linguistiques). Ensuite, des modifications peuvent aussi être apportées à la métaphore de visualisation (la ville) afin d'améliorer l'utilisabilité de notre prototype. Les points essentiels concernent la simplification de la navigation dans l'espace 3D et l'amélioration de l'interaction entre les différents niveaux hiérarchiques. Bien que l'utilisateur ait déjà la possibilité de créer simplement⁹ son propre modèle (i.e. sa propre métaphore), une idée intéressante serait de proposer une boîte à outils — ou *ToolBox* — permettant de concevoir graphiquement de telles métaphores. Sur ce principe, il est envisageable d'adapter le choix de la métaphore en fonction des objectifs de la recherche et du profil de l'utilisateur. Mais automatiser ce processus reste une étape difficile et dangereuse (en termes de satisfaction utilisateur) à franchir. Un dernier point sur l'interface de visualisation concerne l'évaluation du point de vue de l'utilisabilité. Cette partie est fondamentale et notre proposition de cadre d'évaluation des IUI doit encore être validée par une mise en œuvre pratique. Alors, suite à ce test pratique de nos propositions, il sera encore probablement nécessaire d'effectuer un certain nombre de mises au point des critères d'évaluation. D'autres améliorations, d'ordre plus technique, peuvent être apportées au prototype afin de diminuer son temps de réponse (lié à l'algorithme d'organisation et à la génération dynamique du code VRML). Il faut préciser que cet aspect, dont les solutions techniques existent déjà en grande partie, n'était pas une priorité dans nos travaux. De même, l'architecture du prototype doit être modifiée si l'on souhaite imaginer un portage efficace et réalisable de cette application sur d'autres plates-formes, et plus particulièrement sur des systèmes embarqués.

Ensuite, nous évoquons un certain nombre de perspectives dites applicatives. En effet, l'approche générale présentée dans cette thèse peut être adaptée à d'autres problèmes (cf. section 4.5). Par exemple, une idée est de fournir une représentation personnalisée du Web à travers les résultats de recherche. Les recherches sur le Web de l'utilisateur permettent alors d'enrichir la représentation selon un critère défini (pages visitées, premiers résultats). Cette représentation du Web serait alors un espace de visualisation qui grandit dynamiquement au fur et à mesure des recherches de l'utilisateur (à l'image de *CityOfNews* [Sparacino *et al.* 2000]). Mais cette approche nécessite de gérer efficacement un espace de grandes dimensions, ce qui introduit de nouveaux problèmes tels que la navigation, le temps de chargement de l'interface, ou encore la gestion de la mise à jour des données. On peut aussi décider d'utiliser une carte auto-organisatrice calculée à partir des connaissances de l'utilisateur (telles que ses favoris) pour afficher les résultats de recherche. D'autres domaines applicatifs sont aussi adaptés à l'approche proposée dans cette thèse, tels que la visualisation de flux RSS, d'actualités, de favoris ou encore d'une partie d'un système de fichiers. Pour certaines applications, la caractéristique temporelle des documents a une importance particulière. Si cela semble évident pour les flux RSS ou les actualités, cet aspect est actuelle-

⁹Cette action nécessite quand même d'être capable d'écrire du code en langage à balises (HTML, XML) ainsi que du code VRML.

ment délaissé pour le cas des pages Web. Pourtant, l'information temporelle peut se révéler utile pour certaines tâches de RI. Mais sa prise en compte pose quelques problèmes tels que la définition de la date des informations contenues dans un document ou la manière de gérer les mises à jour partielles des documents.

En ce qui concerne les perspectives plus générales, ou plutôt de pistes de recherche qui nous semblent intéressantes, nous nous concentrons sur de récentes propositions en matière d'interface utilisateur. Les interfaces utilisateur définissent à la fois les métaphores utilisées pour visualiser l'information et les interactions offertes à l'utilisateur. Leur rôle est fondamental dans le succès des applications informatiques. Il est donc très intéressant de poursuivre notre réflexion sur les interfaces et plus particulièrement selon trois axes correspondant à des propositions originales de Ben Shneiderman au cours de ces dernières années. L'ordre de traitement de ces différents axes ne doit pas être associé à un quelconque degré de priorité.

Le premier axe consiste à introduire une nouvelle composante dans ces interfaces : l'amusement (ou *fun*) [Shneiderman 2004]. Cette notion, assez vaste, est restreinte ici au *fun-in-doing*¹⁰. Pour y contribuer, il faut chercher à atteindre les objectifs suivants : fournir les fonctions adéquates aux utilisateurs pour accomplir leurs tâches, offrir de l'utilisabilité et de la fiabilité pour éviter la sensation de frustration souvent rencontrée par les utilisateurs, et accrocher les utilisateurs avec des caractéristiques amusantes. C'est plus particulièrement ce dernier point qui est encore absent lors de la conception de nombreuses interfaces. Or l'utilisateur accepte volontiers ces caractéristiques amusantes à condition qu'elles ne le détournent pas de sa tâche. Cependant, il n'existe pour l'instant aucun modèle pour l'intégration de l'amusement dans les interfaces. Cela reste donc essentiellement lié aux compétences des concepteurs. De même, aucun critère d'évaluation n'est proposé à ce sujet, et on reste malheureusement sans solution face au constat de Ben Shneiderman : « *quality has no metrics, but you know it when you see it* ». C'est d'autant plus vrai pour les caractéristiques liées à l'amusement. On ne peut pas vraiment prédire leur impact, mais un test d'utilisation de l'interface révèle rapidement l'apport de ces fonctionnalités. Pour finir sur cet aspect intéressant pour l'avenir des interfaces mais aussi étroitement lié aux attentes actuelles des utilisateurs, voici une courte citation [Shneiderman 2004] :

User interfaces are taking their place in the world of fashion and style, which is great news. Just as dining out is more than getting a balanced diet and wearing clothes is more than staying warm, interfaces are becoming a personal statement. It's great that designers and researchers are turning attention to fun, as a separate design space, distinct from functionality, usability, and reliability. Did anyone no-

¹⁰Le *fun-in-doing* correspond à l'amusement lié à des actions ou à des objectifs, par opposition au *fun-in-not-doing* qui est lié à l'absence d'actions ou d'objectifs. Le *fun-in-doing* peut alors être décrit par des sensations joyeuses intervenant pendant des activités physiques — telles que danser ou faire du snowboard — ou associées à des défis mentaux — tels que résoudre des problèmes, jouer de la musique ou découvrir quelque chose de nouveau. Ces descriptions du *fun-in-doing* servent de base pour une réflexion sur la manière dont la technologie peut être conçue afin de produire plus d'amusement pour les utilisateurs.

... tice that fun is part of functionality ? (...) However, guidelines, models, and principles alone will never guarantee success. Designers have to develop their own style and then test, test, test, and test again. Excellence in design is a great facilitator of fun. Are you ready to have fun designing playful and liberating user interfaces ?

*Ben Shneiderman, Professor at the University of Maryland, USA,
Designing for Fun : How Can We Design User Interfaces to Be More Fun ?*

Ainsi ce constat suit les tendances actuelles des applications Web et notamment de la personnalisation des portails qui deviennent des interfaces propres à chaque utilisateur (cf. la notion de *Portal 2.0* qui est traitée dans la suite).

Le second axe s'intéresse aux interfaces 3D de visualisation et se réfère au point de vue proposé dans l'article intitulé « Why Not Make Interfaces Better than 3D Reality ? » [Shneiderman 2003b]. Nombreux sont ceux qui, ces dernières années, ont essayé de construire des interfaces imitant au mieux la réalité 3D, c'est-à-dire le monde réel. Une idée sous-jacente est de supposer que plus l'interface ressemble au monde réel (e.g. *CityOfNews* [Sparacino et al. 2002], *ViOS*, *3DNA*), plus son utilisation est facile. Et pourtant les études utilisateur révèlent que les interfaces 3D peuvent être moins performantes du fait, entre autres, des difficultés de navigation, de la désorientation de l'utilisateur, des occultations ou encore de la complexité des actions [Sutcliffe et Patel 1996, Ridsen et al. 2000, Cockburn et McKenzie 2002]. Cependant, ces problèmes ne sont pas liés à la qualité de la retranscription de la réalité étant donné qu'ils apparaissent aussi dans le monde réel. Face à ce constat, nombreux sont ceux qui prônent le retour à des visualisations 2D dans lesquelles la navigation et les interactions sont mieux maîtrisées, du fait de la plus grande expérience en matière d'interfaces dans le domaine de la 2D. L'objectif n'est pas ici de discuter de la légitimité de ce point de vue mais de s'intéresser aux moyens d'améliorer les interfaces 3D. Ces dernières doivent cependant cohabiter avec des périphériques pas toujours adaptés (e.g. des écrans 2D, l'usage de la souris ou du clavier). Dans le domaine de l'IV, ces interfaces 3D doivent faciliter la tâche de l'utilisateur et non chercher à reproduire la réalité le plus fidèlement possible. La première approche consiste à contraindre ces interfaces 3D (mouvements, actions, placement des objets sur un plan 2D) afin de les rendre plus simples. Initialement, nos travaux présentés dans les chapitres 3 et 4 se situent dans cette catégorie d'interfaces 3D. Il faut ensuite aller plus loin en enrichissant ces interfaces 3D par rapport au monde réel. Par exemple, on peut y intégrer des caractéristiques telles que la téléportation (changement rapide de contexte par simple « clic » de la souris sur une vue d'ensemble), la vision rayons X pour voir au-delà des objets, le zoom sémantique (apparition de l'objet en premier plan) ou encore des actions utilisateur plus riches. L'objectif est bien sûr de diminuer les occultations, de simplifier la navigation et d'offrir des fonctionnalités plus riches. Il est aussi important de rappeler l'objectif de toute interface dans le domaine de l'IV : minimiser le nombre d'actions nécessaires pour accomplir les différentes tâches. Ainsi l'utilisabilité (en termes de contrôle) de l'interface est aussi important que l'affichage graphique.

Since navigation complexity is a key determinant of user success, it might be more important to fight for two versus three clicks than to debate 2D versus 3D.

*Ben Shneiderman, Professor at the University of Maryland, USA,
Why Not Make Interfaces Better than 3D Reality?*

Un bémol doit cependant être apporté aux interfaces 3D qui ne sont pas forcément une source d'amélioration dans le cadre de l'IV. Contrairement à la visualisation scientifique, ici les variables sont plus souvent nominales que continues et l'objectif reste de découvrir des motifs, des regroupements, des tendances, des comportements atypiques ou des séparations dans les données. On remarque alors que certaines métaphores 3D, pourtant appréciées du public d'un point de vue immersion, s'avèrent moins efficaces pour achever les différentes tâches (telles que la recherche). L'utilisateur adule les interfaces 3D immersives parce qu'elles améliorent l'utilisation de la mémoire spatiale. Il faut alors vérifier avec attention le réel apport de la 3D en termes d'utilisabilité et s'assurer qu'une interface en 2D n'apporte pas de meilleurs résultats.

Le troisième et dernier axe sur les interfaces utilisateur, dans le cadre d'un accès à l'information, concerne l'aspect multi-couches des interfaces [Shneiderman 2003a]. En effet, satisfaire tous les utilisateurs avec une seule et même interface est extrêmement dur. Cela est notamment dû à la diversité des profils des utilisateurs et des tâches à accomplir. Il est alors plus raisonnable d'essayer de répondre à une certaine catégorie d'utilisateurs et/ou de tâches. La conception d'interfaces multi-couches s'avère être une solution adaptée à ce problème. Ce type d'interfaces permet aux utilisateurs novices de démarrer avec un ensemble limité de fonctionnalités (couches basses). Il est aussi adapté aux utilisateurs expérimentés qui évoluent rapidement vers les couches plus élevées de l'interface qui disposent de fonctionnalités plus riches. Un exemple de cette approche multi-couches est le moteur de recherche *Ujiko*. L'implémentation de ce type d'interfaces pose toutefois certains problèmes parmi lesquels le choix du nombre de couches, la compatibilité entre les différentes couches, la gestion de l'apprentissage de l'utilisateur ou encore la définition de couches satisfaisant le maximum d'utilisateurs. L'absence de règles de conception de ces interfaces (excepté l'utilisation des théories cognitives) rend primordial les tests d'utilisabilité.

Enfin, nous profitons de ces derniers paragraphes pour resituer nos travaux dans le contexte actuel et très changeant du Web, et plus particulièrement par rapport aux évolutions étiquetées Web 2.0 et *Search 2.0*. En effet, il nous paraît difficile de ne pas évoquer ce phénomène devenu à la mode au point d'entendre dire au cours d'une discussion : « Moi, je n'ai pas encore été sur le Web 2.0 ». Le terme Web 2.0 a été proposé par Dale Dougherty de la société *O'Reilly Media* pour incarner l'évolution du Web [O'Reilly 2005]. De nombreuses définitions¹¹ — parfois contradictoires — de ce concept existent. Bien entendu, certains aspects du Web 2.0 reviennent souvent comme des leitmotiv tels que la dimension sociologique, le basculement des techniques vers des services, le recentrage sur l'utilisateur, syndication des contenus, les nouvelles

¹¹http://en.wikipedia.org/wiki/Web_2

approches (API, RSS, folksonomies¹², réseaux sociaux, Web sémantique¹³, AJAX¹⁴, CSS¹⁵, XHTML¹⁶...), la réappropriation humaine du Web, **l'amélioration des interfaces utilisateur**, l'interopérabilité. Avec le Web 2.0, on passe alors d'une collection de sites Web à une vraie plate-forme fournissant des applications Web aux utilisateurs. Cette plate-forme remplacera peut-être la majorité des applications locales ainsi que celles liées au système d'exploitation, à l'image de ce qui se passe déjà actuellement avec certaines applications telles que la messagerie, le stockage des photos, le traitement de texte, la messagerie instantanée. On peut aussi désormais parler de *Web Operating System* dont un exemple est *YouOS*¹⁷. Seulement, le problème est qu'on accède à ces systèmes d'exploitation Web par l'intermédiaire d'un navigateur Web, ce qui suppose donc d'avoir installé un système d'exploitation en local sur l'ordinateur ! Mais recentrons nous sur le Web 2.0. Ce qui est gênant dans cette histoire, c'est que de nombreuses idées du Web 2.0 ont déjà été employées avant la création de ce terme. Il semble alors tout à fait légitime de se demander si le Web 2.0 n'est pas plus lié à une évolution des utilisateurs (et donc de son utilisation) qu'à une nouvelle génération du *Word Wide Web* d'un point de vue technologique. En ce sens, le Web 2.0 n'est alors qu'une récupération de technologies existantes réarrangées à la sauce 2.0.

Cette appellation 2.0 pour désigner un Web seconde génération est devenu un phénomène de mode au point d'estampiller 2.0 de nombreuses applications étroitement liées au Web, sous prétexte qu'elles doivent accompagner son évolution. Ainsi, on parle d'*Information Access 2.0*¹⁸ qui se décompose en deux parties : *Portal 2.0*¹⁹ et *Search 2.0*²⁰. On trouve aussi le terme de *Moteur 2.0*. À qui le tour ? De plus, il s'agit de termes difficilement définissables tellement leurs frontières sont floues. Bien qu'ils fassent déjà partie de l'héritage commun de la terminologie du Web, ils sont bien souvent « redéfinis » à la convenance de leurs utilisateurs. Loin d'en proposer une définition, nous donnons quelques éléments associés à ces termes afin de mieux en définir les enjeux. Ainsi le *Search 2.0* est associé aux nouvelles technologies de recherche qui sont conçues pour combiner la scalabilité des moteurs de recherche existants avec des modèles de pertinence nouveaux et améliorés. Ces nouvelles technologies prennent en compte les préférences des utilisateurs, la collaboration et de nombreuses autres capacités spécialisées qui rendent l'information plus productive. Elles proposent aussi de nouvelles stratégies de recherche telles que la possibilité de chercher des informations de fil en aiguille ou par association d'idées (ce qui permet d'obtenir des résultats qu'une recherche plein texte ne peut pas proposer). Cela dit, le mécanisme d'essai-erreur (reformulation de la requête si la réponse n'apparaît pas dans les premiers résultats), bien que contraignant, ne disparaît pas pour autant étant donné qu'il s'avère

¹²<http://fr.wikipedia.org/wiki/Folksonomie>

¹³http://fr.wikipedia.org/wiki/Web_sémantique

¹⁴<http://fr.wikipedia.org/wiki/AJAX>

¹⁵http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

¹⁶<http://fr.wikipedia.org/wiki/Xhtml>

¹⁷<https://www.youos.com/>

¹⁸Ce terme peut être traduit en français par l'expression « Accès à l'information 2.0 »

¹⁹Ce terme peut être traduit en français par l'expression « Portail 2.0 »

²⁰Ce terme peut être traduit en français par l'expression « Recherche 2.0 »

efficace dans certains cas. Actuellement la majorité des acteurs dans le domaine des moteurs de recherche veulent devenir « le prochain *Google* ». Pour cela, ils cherchent à être plus pertinent en mettant en œuvre de nouveaux modèles et de nouvelles fonctionnalités. D'ailleurs, la personnalisation et les interfaces utilisateur sont annoncés comme deux points clés dans l'évolution des moteurs de recherche au cours des prochaines années. Ces innovations sont cohérentes avec l'évolution du Web vers le 2.0; et si c'était simplement ça le *Search 2.0*? Concernant la partie *Portal 2.0*, il s'agit de portails nouvelle génération²¹ qui permettent à l'utilisateur de personnaliser sa page d'accueil en agrégeant divers contenus ou sources d'information (e.g. des flux RSS, des services Web tels que la météo locale ou la bourse, ses favoris, ses e-mails, des moteurs de recherche). Cette personnalisation donne l'impression à l'utilisateur de retrouver du pouvoir, ce qui fait encore défaut dans les moteurs de recherche où il reste encore trop souvent spectateur des résultats proposés par les algorithmes de recherche. Étant donné le contexte actuel, il apparaît clairement que les IUI vont être un point capital du *Search 2.0*. Bien qu'on constate actuellement les gros efforts réalisés sur la catégorisation des résultats ou sur la personnalisation, tous les défis ne sont pas au même degré de maturité, aussi bien du point de vue technique que de celui de l'utilisateur. La démocratisation des interfaces 3D pour des applications grand public d'accès à l'information devra probablement attendre une évolution technologique (au niveau des périphériques d'interaction grand public tels que la souris ou le clavier) et comportementale (au niveau des utilisateurs). Ainsi, en jouant aux néologismes, il faudra peut-être attendre le *Search 3.D* pour voir des IUI 3D performantes, conviviales et simples d'utilisation.

Pour conclure, on remarque qu'il reste encore de nombreux défis à relever concernant les IUI, même si certains sont dans un état d'avancement significatif. Ainsi, il est nécessaire d'organiser visuellement les documents selon leur contenu (*i.e.* selon leur sens). Les méthodes actuelles, essentiellement statistiques, donnent des résultats plus qu'encourageants si l'on considère qu'elles traitent les mots comme des objets dépourvus de sens. Cependant, il serait aussi intéressant de voir l'impact sur ces résultats si on introduisait des connaissances linguistiques. Par ailleurs, certains aspects restent problématiques comme l'étiquetage d'un groupe de documents qui reste encore très souvent peu interprétable par l'utilisateur. Ensuite, il faut proposer des métaphores de visualisation adaptées aux données, à la tâche et à l'utilisateur. En ce sens, les interfaces multi-couches semblent être une alternative intéressante à condition que leur manipulation ne soit pas trop contraignante pour l'utilisateur. De plus, dans le domaine de l'IV, l'utilisation d'interfaces 3D ne doit se faire que sous une forme améliorée et enrichie par rapport à la réalité 3D. Il ne faut également pas hésiter à rendre ces interfaces plus amusantes pour l'utilisateur sans pour autant le détourner de sa tâche. Pour cela, il est nécessaire de repenser les métaphores de visualisation, sans négliger la navigation et les interactions qui sont deux points extrêmement importants. Enfin,

²¹Deux exemples sont *Netvibes* (<http://www.netvibes.com/>) et *WebWag* (<http://www.webwag.com/>).

afin de valider ces IUI, des critères d'évaluation adaptés et interprétables doivent être mis en place dans une démarche commune d'évaluation.

Une interface de visualisation adaptée à l'utilisateur et à ses différentes tâches peut lui être d'une aide non négligeable dans sa gestion de l'information. Dotée de méthodes intelligentes, elle permet d'organiser et de filtrer visuellement les données. Ces deux tâches sont un bon point de départ pour faire face à l'infobésité. Actuellement cette pathologie ne peut être raisonnablement traitée qu'en aval par des applications ou interfaces intelligentes accompagnées d'une modification des comportements des utilisateurs. En effet, un contrôle des données à leur diffusion reste aujourd'hui une utopie.

Glossaire

ACM	Analyse des Correspondances Multiples
ACP	Analyse en Composantes Principales
AFC	Analyse Factorielle des Correspondances
AHIGG	Adaptive Hierarchical Incremental Grid Growing
AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
BMU	Best Matching Unit
CAH	Classification Ascendante Hiérarchique
CAO	Conception Assistée par Ordinateur
CLEF	Cross-Language Evaluation Forum
CSS	Cascading Style Sheets
DCV	Document Cut off Values
GG	Growing Grid
GHSOM	Growing Hierarchical Self-Organizing Map
HFC	Hierarchical Faceted Categories
HFM	Hierarchical Feature Map
HTML	HyperText Markup Language
IA	Intelligence Artificielle
IAP	Interpolated Average Precision
IGG	Incremental Grid Growing
IHM	Interface Homme-Machine
INEX	Initiative for the Evaluation of XML Retrieval
IUI	Interface Utilisateur d'Information
IV	Information Visualization
JRE	Java Runtime Environment
MAP	Mean Average Precision
MDS	Multidimensional Scaling
MQE	Mean Quantization Error
NIAP	Non-Interpolated Average Precision
QE	Quantization Error
RI	Recherche d'Information
RSS	Really Simple Syndication ou Rich Site Summary
SOM	Self-Organizing Map

SQL	Structured Query Language
SRI	Système de Recherche d'Information
TAL	Traitement Automatique des Langues
TREC	Text REtrieval Conference
URL	Uniform Ressource Locator
VRML	Virtual Reality Modeling Language
X3D	eXtensible 3D
XHTML	eXtensible HyperText Markup Language
XML	eXtended Markup Language

Annexe A

Compléments sur les méthodes de visualisation

Cette annexe catalogue certaines méthodes de visualisation citées mais non détaillées dans le chapitre 1. En effet, bien que les concepts de ces méthodes soient intéressants, elles semblent moins adaptées à la visualisation de résultats de recherche (*i.e.* de documents textuels de type page Web).

A.1 Techniques basées sur la visualisation des attributs

A.1.1 Visualisations géométriques

Matrice de graphes de dispersion [Cleveland 1993]
— *scatterplot matrix* —

Il s'agit d'une configuration de graphes de dispersion dans une matrice pour montrer les différentes combinaisons de variables (**figure A.1**). Les graphes de dispersion individuels peuvent révéler des corrélations entre des variables (*e.g.* la linéarité) et la matrice complète peut être utile pour une exploration initiale de l'ensemble des données. Cette collection de graphes de dispersion est organisée de façon analogue à une matrice de covariance, avec une représentation de la variable a par rapport à la variable b dans la a^e ligne et b^e colonne de la matrice. Les graphiques de la diagonale peuvent montrer la distribution des variables individuelles, ou simplement être remplacés par les noms des variables.

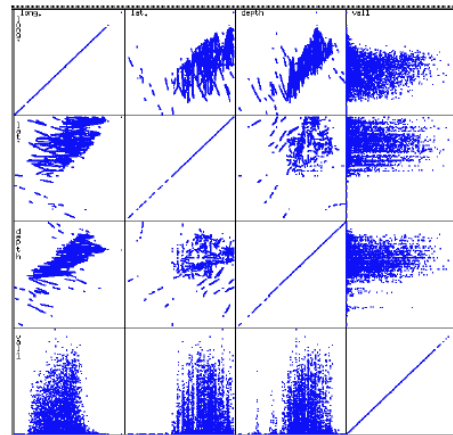


FIG. A.1 – *Scatterplot matrix*

Coordonnées parallèles [Inselberg 1998]
 — *parallel coordinates* —

Cette méthode permet de représenter des données multidimensionnelles dans un espace 2D. Son principe consiste à représenter les dimensions de l'espace de départ par des axes qui doivent être parallèles et équidistants. Une donnée est alors représentée en reliant ses coordonnées sur chaque axe par des droites, ce qui correspond à une représentation de la valeur de ses attributs. Il est aussi possible d'utiliser les couleurs pour différencier différentes valeurs des attributs ou différentes données (**figure A.2**).

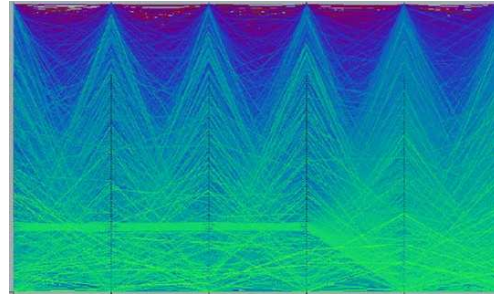


FIG. A.2 – Coordonnées parallèles

Coordonnées en étoile [Kandogan 2000]
 — *star coordinates* —

Cette méthode permet de représenter des données multidimensionnelles dans un espace 2D. Son principe consiste à représenter les dimensions de l'espace de départ par des axes radiaux. Une donnée est alors représentée en reliant ses coordonnées sur chaque axe par des droites, ce qui correspond à une représentation de la valeur de ses attributs. Les coordonnées en étoiles permettent alors de représenter une donnée par une ligne polygonale ou encore par un point (**figure A.3**).

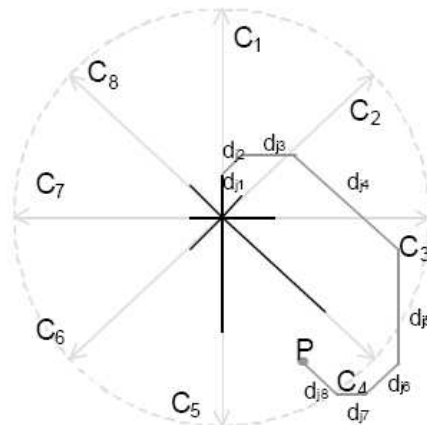


FIG. A.3 – Coordonnées en étoile

A.1.2 Visualisations symboliques

A.1.2.1 Techniques d'icônification

Les visages de Chernoff [Chernoff 1973]
— *Chernoff Faces* —

Cette technique de visualisation utilise des caractéristiques faciales pour représenter les tendances dans des valeurs des données multidimensionnelles, et non les valeurs spécifiques elles-mêmes (**figure A.4**). Bien qu'il s'agisse clairement d'une limitation, la connaissance des tendances dans les données peut aider à déterminer quelles parties des données sont d'un intérêt particulier. Les visages (ou figures) de *Chernoff* sont particulièrement efficaces parce qu'ils relient les données à des caractéristiques faciales, c'est-à-dire à quelque chose que nous sommes habitués à différencier. Différentes dimensions des données sont associées à différentes caractéristiques faciales telles que la largeur du visage, le niveau ou le rayon des oreilles, la longueur ou la courbure de la bouche, la longueur du nez, espacement des yeux.

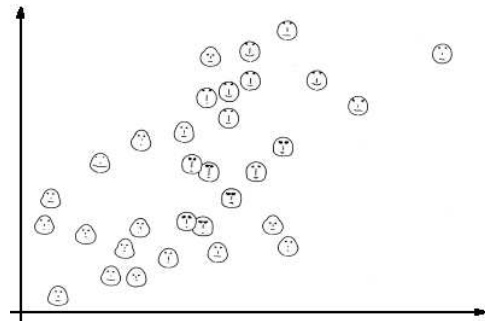


FIG. A.4 – *Chernoff Faces*

Au total, 15 attributs peuvent être représentés, et des variables additionnelles pourraient être encodées en rendant les visages asymétriques [Flury et Riedwyl 1981]. Par ailleurs, deux attributs (généralement il s'agit plutôt des deux premières composantes principales) peuvent être utilisés pour placer les visages dans l'espace 2D. Le pouvoir des visages de *Chernoff* vient de leur grande condensation des données et leur moyen intéressant de présentation. Bien que le visionnage répétitif de grandes tables de données soit fastidieux, les visages de *Chernoff* peuvent améliorer significativement la « digestion » des données [Scott 1992]. Cependant, un inconvénient majeur de cette technique est que l'attribution subjective des expressions faciales aux variables affecte la forme du visage. Ainsi, des permutations dans l'attribution des caractéristiques peuvent causer un taux d'erreur de l'ordre de 25 pour la tâche de classification des visages dans des groupes [Chernoff et Rizvi 1975]. Cela signifie que classer deux visages comme « assez similaires » est considérablement influencé par l'attribution des variables à des caractéristiques faciales spécifiques. En outre, une critique faite à cette technique est la redondance liée à la caractéristique symétrique des visages de *Chernoff* [Flury et Riedwyl 1981, Turner et Tidmore 1980]. Comme les approches *star graph* ou *radar plot*, le pouvoir de montrer des relations multiples dans les visages de *Chernoff* est limité dans un mode statique. Mais, les visages de *Chernoff* animés peuvent répondre partiellement à ce problème.

Stick Figures
[Pickett et Grinstein 1988]

La technique des *Stick figures* propose une représentation des attributs par le biais des angles et des longueurs des membres d'un icône (figure A.5). Si les icônes sont suffisamment denses par rapport aux axes choisis, il est alors possible de voir des modèles de textures qui varient suivant les caractéristiques des documents. Deux attributs (généralement il s'agit plutôt des deux premières composantes principales) peuvent être utilisés pour placer les icônes dans l'espace 2D.

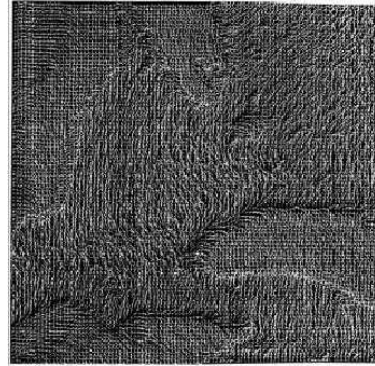
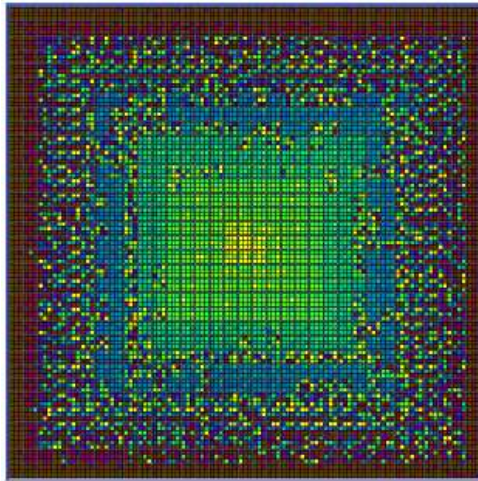


FIG. A.5 – *Stick Figures*

Color Icons [Levkowitz 1991]



Color Icons est une technique qui permet de représenter des documents sous la forme d'un tableau dont les différents champs correspondent aux valeurs des différents attributs qui sont représentées par des couleurs. Ici, l'arrangement est en spirale (figure A.6).

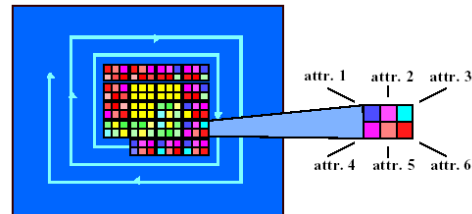


FIG. A.6 – *Color Icons*

Today's Front Pages

<http://www.newseum.org/todaysfrontpages/>

Today's Front Pages est une présentation en ligne d'une des expositions les plus populaires du *Newseum* (**figure A.7**). Chaque matin, plus de 400 journaux de partout dans le monde soumettent leurs premières pages au *Newseum* via Internet. Quand le *Newseum* s'ouvrira en 2007 au centre de la ville de Washington, la galerie des premières pages (*i.e.* des unes) donnera aux visiteurs un aperçu des nouvelles de la journée venant des unes de 80 journaux de chaque état, du district de Columbia et de pays à travers le monde. En attendant, la sélection entière des premières pages de chaque jour est disponible sur [newseum.org](http://www.newseum.org), chaque jour à partir de 9h30.

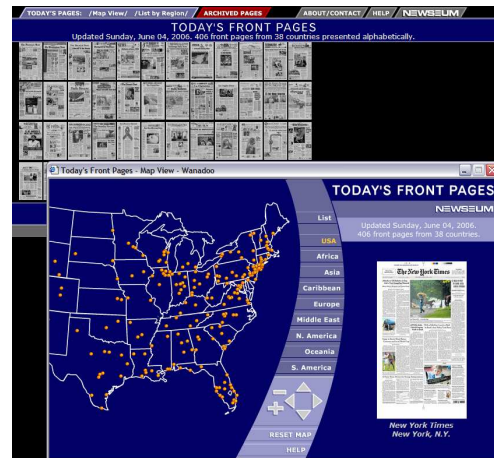


FIG. A.7 – *Today's front pages*

A.1.2.2 Techniques de pixelisation

Motif récursif [Keim *et al.* 1995]
— *recursive pattern* —

Le motif récursif est une technique de visualisation qui a été développée pour visualiser de grandes quantités de données multidimensionnelles. La technique est basée sur un modèle récursif générique qui généralise un vaste éventail d'arrangements basés sur les pixels pour afficher de grands ensembles de données. En instanciant la technique avec les paramètres adéquats (dépendant des données et de l'application), l'utilisateur peut largement influencer la structure des visualisations obtenues en résultat. Comme la technique utilise un pixel pour présenter chaque valeur des données, la quantité des données

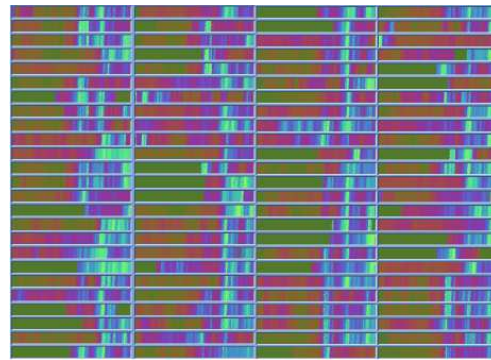


FIG. A.8 – Exemple de motif récursif

qui peuvent être affichées est seulement limitée par la résolution de la technologie d'affichage et par les limites de la perceptibilité humaine. Par ailleurs, cette technique est particulièrement avantageuse pour les ensembles de données qui ont un ordre naturel selon une dimension, telles que les données de séries temporelles [Keim *et al.* 1995]. Le motif récursif propose un arrangement récursif de long en large des pixels. L'utilisateur peut spécifier des paramètres pour chaque niveau de récursivité et ainsi contrôler l'arrangement des pixels pour former des sous-structures sémantiquement significatives. L'élément basique de chaque niveau de récursivité est un motif de taille h_i et de largeur w_i spécifié par l'utilisateur. Les pixels sont alors arrangés dans un rectangle de taille h_i et de largeur w_i , de gauche à droite, puis en dessous, en arrière, de droite à gauche et ainsi de suite (**figure A.8**). Cet arrangement basique est réalisé dans tous les niveaux de récursivité, la seule différence étant que les éléments basiques arrangés au niveau i sont les motifs résultants du niveau $i - 1$. Il existe d'autres types d'arrangements possibles.

Segments de cercle [Ankerst *et al.* 1996]

— *circle segments* —

L'idée basique de la technique de visualisation des segments de cercle est d'afficher les dimensions des données comme des segments d'un cercle. Ainsi, si les données possèdent k dimensions, alors le cercle est partitionné en k segments, chaque segment représentant une dimension des données. Les données sont donc représentées dans un cercle divisé en segments. À l'intérieur des segments, les valeurs des données appartenant à une dimension sont arrangées du centre du cercle vers l'extérieur, sous une forme de *va et vient*, orthogonalement à la ligne divisant le segment en deux (*i.e.* sur une ligne orthogonale à la médiatrice de la corde définie par le segment). Ainsi, dans un segment, chaque valeur de la dimension représentée est visualisée par un simple pixel coloré (**figure A.9**). La technique des segments

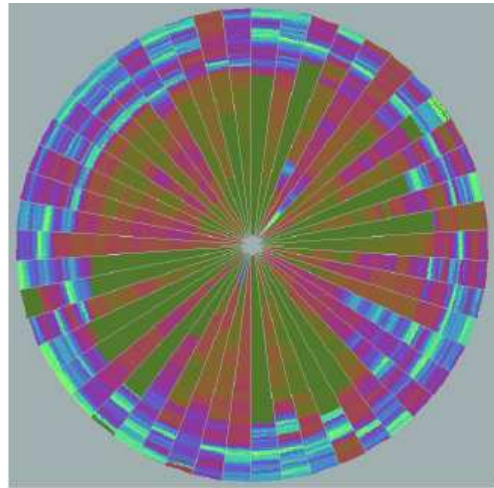


FIG. A.9 – Exemple d'arrangement selon la technique des segments de cercle

de grandes quantités de données, et fournit des visualisations plus expressives que les autres techniques célèbres telles que la technique du motif récursif et les traditionnelles courbes (*line graphs*) [Ankerst *et al.* 1996]. Le principal avantage de cette technique est que la représentation générale de l'ensemble de données en entier est mieux perceptible, incluant les potentielles dépendances, les analogies, et les corrélations entre les dimensions. De plus, près du centre, tous les attributs sont proches les uns des autres, augmentant ainsi la comparaison visuelle de leurs valeurs.

A.2 Visualisations hiérarchiques

A.2.1 Visualisations 2D

Treemap [Shneiderman 1992]

<http://www.cs.umd.edu/hcil/treemap-history/>

La technique *Treemap*, proposée par Ben Shneiderman au début des années 90, est une méthode très célèbre de visualisation dite de « remplissage de l'espace ». Elle permet d'afficher des hiérarchies (ou arbres) dans un espace 2D. Pour cela, la technique *Treemap* utilise un tracé rectangulaire avec une sous-division récursive selon le nombre et la taille des noeuds fils. La racine de la hiérarchie est alors représentée par le rectangle entier. Les couleurs sont généralement utilisées pour différencier le type des différents nœuds. Il existe de nombreuses extensions de ce principe tel que les *Cushion Treemaps* (figure A.10) [van Wijk et van de Wetering 1999]. Une application classique de cette méthode est la visualisation de systèmes de fichiers.

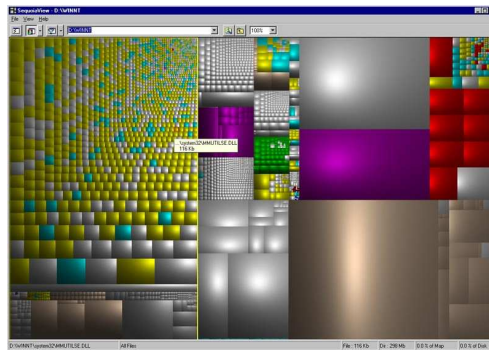


FIG. A.10 – *Treemap*

Newsmap [Marcos Weskamp]

<http://www.marumushi.com/apps/newsmap/newsmap.cfm>

Newsmap est une application qui reflète visuellement le paysage changeant constamment de l'agrégateur d'actualités *Google News*. Un algorithme de visualisation de type *treemap* aide à afficher l'énorme quantité d'informations récoltées par l'agrégateur. Les *Treemaps* sont traditionnellement des visualisations de l'information contraintes dans l'espace. L'objectif de *Newsmap* pousse ce but un cran plus loin et fournit un outil pour diviser l'information dans des groupes rapidement reconnaissables qui, quand ils sont présentés ensembles, révèlent des motifs sous-jacents

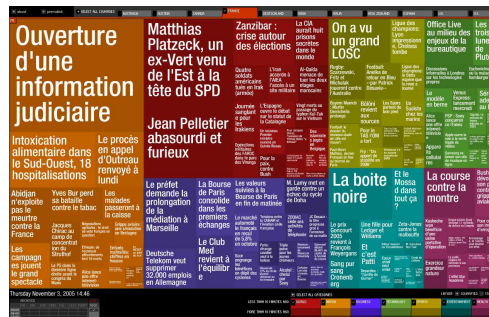


FIG. A.11 – *Newsmap*

dans les actualités rapportées à travers les cultures et à l'intérieur des sujets d'actualités en changement constant autour du monde (**figure A.11**). Mais *Newsmap* ne prétend pas remplacer l'agrégateur *googlenews*. Son objectif est simplement de montrer visuellement les relations entre les données et les motifs invisibles dans les médias d'actualités. On ne pense pas afficher une vue non biaisée des actualités, au contraire on pense accentuer ironiquement le biais dans les actualités.

SunBurst [Stasko et Zhang 2000]

La technique *SunBurst* est une visualisation alternative de remplissage de l'espace, qui utilise un tracé radial plutôt que rectangulaire (cf. *Treemap*). Dans *SunBurst*, les objets d'une hiérarchie sont affichés radialement, avec le haut de la hiérarchie au centre (le petit cercle au centre représente la racine) et les feuilles les plus profondes très loin du centre. Chaque niveau est caractérisé par l'ajout d'un nouvel anneau au graphe. Les anneaux sont bien entendu divisés en secteurs selon le nombre et la taille des nœuds dans les niveaux respectifs. Ainsi, l'angle est balayé par un objet et la couleur de l'objet correspond à certains attributs des données. Par exemple, dans une visualisation d'un système de fichiers, l'angle peut correspondre à la taille du fichier (ou du répertoire) et la couleur peut correspondre au type de fichier. La **figure A.12** montre un exemple d'affichage *SunBurst*. Cette technique permet aussi d'avoir une vue plus détaillée sur des régions d'intérêt grâce à 3 techniques de *focus* : *angular detail*, *detail outside*, *detail inside* [Stasko et Zhang 2000].

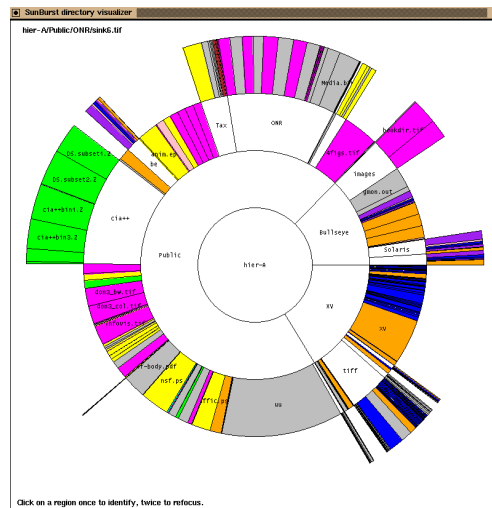


FIG. A.12 – *SunBurst*

A.2.2 Visualisations 3D

On rappelle que le terme de « visualisations 3D » est utilisé ici pour parler des visualisations proposant un affichage des résultats selon trois axes, indépendamment du fait qu'elles proposent ou non une simple manipulation de la représentation selon le troisième axe, ou une navigation selon le troisième axe, ou encore un environnement immersif.

Arbre conique [Roberston *et al.* 1991]
— *Cone Tree* —

Cone Tree est une des techniques les plus célèbres du domaine de l'IV. Cette technique est utilisée pour visualiser des structures d'informations hiérarchiques, et plus précisément pour afficher des arbres en 3D. Elle a été directement développée pour la 3D, contrairement à la généralisation d'algorithmes 2D. La hiérarchie est alors présentée dans un espace 3D afin de maximiser efficacement l'utilisation de l'espace disponible sur l'écran, et de permettre la visualisation de l'ensemble de la structure. La méthode *Cone Tree* consiste à afficher l'arbre sous forme de cônes. Les nœuds sont placés au sommet des cônes et leurs fils sont placés le long de des bases des cônes. Chaque couche possède des cônes de même taille, et les diamètres de la base du cône pour chaque niveau sont réduits de façon à ajuster la couche du fond dans la largeur de la boîte contenant l'arbre en entier. Une animation interactive est utilisée pour transférer une partie de la charge cognitive de l'utilisateur sur le système de perception humain. Il existe aussi des améliorations à cet algorithme, notamment pour réaliser une meilleure utilisation de l'espace disponible (approximation du diamètre de chaque cône en fonction du nombre de descendants). La **figure A.13** présente un exemple de visualisation de la structure d'un système de fichiers.

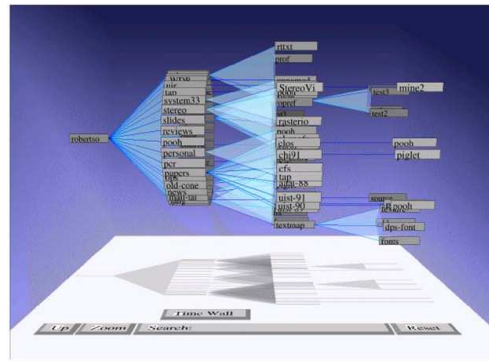


FIG. A.13 – *Cone Tree*

Le cube d'information [Rekimoto et Green 1993]

— *InfoCube* —

InfoCube propose une visualisation 3D de données hiérarchiques basée sur la notion de boîtes transparentes imbriquées les unes dans les autres. L'utilisation d'un rendu semi-transparent permet au système de contrôler la complexité de l'information présentée à l'utilisateur. Grâce aux techniques d'interaction fournies par le système, l'utilisateur peut reconnaître et inspecter intuitivement la structure de l'information. Le système peut être utilisé *via* un équipement de réalité virtuelle, mais il est aussi accessible avec un affichage conventionnel. La **figure A.14** présente un exemple de visualisation de la structure d'un système de fichiers.



FIG. A.14 – *InfoCube*

***StepTree* [Bladh et al. 2004]**

StepTree est essentiellement un *treemap* étendu à trois dimensions par le simple expédient d'empiler les niveaux de l'arbre les uns sur les autres dans l'espace 3D. Cette technique a été spécifiquement développée pour l'affichage de hiérarchies d'un système de fichiers. Elle affiche les associations visuelles des métriques du système de fichiers telles que la taille du fichier, les changements dans les fichiers ou dans les répertoires, le type du fichier. *StepTree* a été en partie développé pour étudier les moyens d'enrichir la visualisation dite de « remplissage de l'espace » de telle façon que la taille soit une information moins dominante. En effet, les relations représentées par l'association de la taille de l'objet à une région de l'espace, viennent souvent aux dépens de toutes les autres associations. Ainsi, comme les régions de nœuds tendent à être liées directement à cette relation,

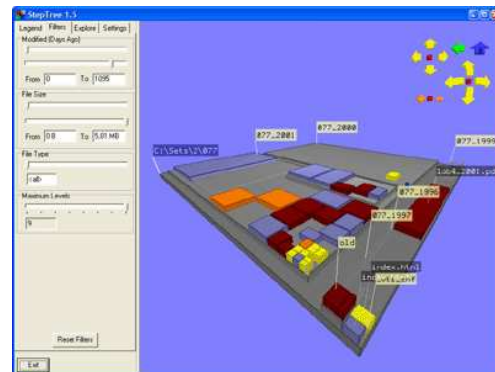


FIG. A.15 – *StepTree*

certaines nœuds peuvent dominer la vue tandis que d'autres peuvent être complètement masqués. Et pourtant, si quelqu'un veut afficher les changements dans la hiérarchie, les modifications des petits fichiers sont aussi importantes que celles des gros fichiers. *StepTree* répond à ce problème en proposant notamment une technique d'affichage des modifications (**figure A.15**).

Annexe B

Algorithme *batch* SOM

L'objectif de cette annexe est de donner un aperçu du code et surtout de la complexité (en nombre d'opérations) des différentes étapes de l'algorithme *batch* SOM. En langage *Java*, **une itération** de l'algorithme *batch* SOM peut s'écrire de la façon suivante :

```
(...)  
  
5 double[][] sumVQ;           // matrice  $c \times j$  permettant de stocker,  
                               // pour chaque unité de la carte, la somme  
                               // des vecteurs qui représentent les données  
                               // associées à cette unité  
  
10 double[] w;                // tableau de taille  $c$  permettant de stocker  
                               // le nombre de données affectées à chaque  
                               // unité de la carte  
  
double sigma;                // valeur de la largeur de la fonction de  
                               // voisinage pour l'itération en cours  
15 (...)  
  
/*-----*/  
20 /* Quantification vectorielle (vector quantization) */  
/*-----*/  
for (int i = 0 ; i < n ; i++){           // Itération sur les  
    int bmu = -1;                         // documents :  $O(n)$   
    double min = Double.MAX_VALUE;  
25    for (int k = 0 ; k < c ; k++){       // Recherche de la  
        double dist = 0;                  // BMU :  $O(3cp)$   
        for (int j = 0 ; j < p ; j++){  
            double tmp = x[i][j] - m[k][j];  
            dist += tmp * tmp;  
30    }  
}
```

```

        if (dist < min){
            min = dist;
            bmu = k;
        }
35     }
        for (int j = 0 ; j < p ; j++){
            sumVQ[bmu][j] += x[i][j];
            w[bmu]++;
        }
40 }
/*-----*/
/* TOTAL "Vector Quantization" =  $O(3ncp + 2np)$  */
/*-----*/

45 /*-----*/
/* Étape de lissage (smoothing step) */
/*-----*/
50 for (int k = 0 ; k < c ; k++){
    double den = 0;
    // Itération sur les neurones :  $O(c)$ 

    for (int j = 0 ; j < p ; j++) m[k][j] = 0; // Initialisation :  $O(p)$ 
    for (int k' = 0 ; k' < c ; k'++){
        // Calcul :  $O(3cp + 2c)$ 

55         double h = Math.exp(-delta(k,k') / (2 * Math.pow(sigma,2)));
        for (int j = 0 ; j < p ; j++){
            m[k][j] += sumVQ[k'][j] * h * w[k'];
        }
        den += h * w[k'];
60     }
    if (den!=0){
        for (int j = 0 ; j < p ; j++){
            // Calcul :  $O(p)$ 
            m[k][j] /= den;
        }
65     }
}
/*-----*/
/* TOTAL "Smoothing step" =  $O(3c^2p + 2c^2 + 2cp)$  */
/*-----*/

70 /*-----*/
/* TOTAL =  $O(3ncp + 2np + 3c^2p + 2c^2 + 2cp)$  */
/*-----*/

```

Dans l'algorithme ci-dessus, $x[i][j]$ est la j^e composante du i^e document et $m[k][j]$ est la j^e composante de la k^e unité de la carte. $\text{delta}(k, k')$ est une fonction qui rend la distance au carré, sur la carte (*i.e.* dans \mathbb{R}^2), entre les unités k et k' , *i.e.* $\text{delta}(k, k') = \|\mathbf{r}_k - \mathbf{r}_{k'}\|^2$.

Annexe C

Informations sur l'implémentation du prototype

Des informations complémentaires sur l'implémentation du prototype *Smart-Web* sont données dans cette annexe. Ces informations portent sur la base de données (section C.1) ainsi que sur l'utilisation du langage X-VRML (section C.2).

C.1 Base de données

Cette section donne des informations sur la base de données utilisée par le prototype. Il s'agit d'une base de données `MySQL` qui est remplie lors d'une étape hors ligne. L'objectif est simplement de disposer, en local, d'une base de données de test pour mettre en place le prototypage des travaux proposés dans la thèse. Pour des raisons de simplicité technique, notre base de données est remplie à partir d'une des bases d'index de *Google*, grâce à l'API fournie par *Google*. Notre base, appelée `SMARTDB`, contient différentes tables décrites ci-après. Tout d'abord, les requêtes sont stockées dans la table `QUERIES` (décrite par le **tableau 1**), ce qui permet de mémoriser les requêtes pouvant être réalisées sur notre prototype. Les pages Web, retournées par les différentes requêtes contenues dans la base, sont décrites par les champs de la table `DOCUMENTS` (cf. **tableau 3**). Les résultats d'une requête (*i.e.* l'association documents - requête) sont mémorisés grâce à la table `QUERIES_RESULTS` (décrite par le **tableau 2**). Par ailleurs, il faut préciser que seuls les 200 premiers résultats de chaque requête sont stockés dans notre base locale. Ensuite, la table `DOCUMENTS_CONTENTS` (décrite par le **tableau 4**) permet de faire le lien entre une page Web et les mots la constituant. Et l'ensemble des mots présents dans le corpus est stocké dans la table `WORDS` (décrite par le **tableau 5**). Ces informations n'étant malheureusement pas disponibles *via* l'API de *Google*, nous les avons recalculées. La sélection des mots et le calcul du champ `TF` sont expliqués dans le **chapitre 2**. Enfin, la table `IMAGES_OF_DOCUMENTS` (décrite par le **tableau 7**) permet d'associer les images aux documents et ces images sont décrites par les différents champs de la table `IMAGES` (cf. **tableau 6**).

Champs	Description	Type 'MySQL'	Caractéristiques	Exemple
ID	identificateur de la requête	int(11)	primary key, not null, auto increment	0
VALUE	requête textuelle	varchar(255)		orange

TAB. C.1 – Table QUERIES

Champs	Description	Type 'MySQL'	Caractéristiques	Exemple
QUERY	identificateur de la requête	int(11)	not null	0
DOCUMENT	identificateur de la page	int(11)	not null	10
RANK	rang du document dans les réponses	int(11)	not null	10
SNIPPET	extrait de la page par rapport à la requête	varchar(255)		*

* Ville d'art et d'histoire, **Orange** brille par son patrimoine historique exceptionnel. . . Le coeur d'**Orange** s'apparente à une sorte de gros village provençal. . .

TAB. C.2 – Table QUERIES_RESULTS

Champs	Description	Type 'MySQL'	Caractéristiques	Exemple
ID	identificateur de la page	int(11)	primary key, not null, auto increment	10
URL	URL complète du document	varchar(255)	not null	*
TITLE	titre de la page	varchar(255)		Orange Accueil Tourisme - Provence Web
SUMMARY	résumé de la page	varchar(255)		

* <http://www.provenceweb.fr/f/vaucluse/orange/orange.htm>

TAB. C.3 – Table DOCUMENTS

Champs	Description	Type 'MySQL'	Caractéristiques		Exemple
DOC	identificateur de la page	int(11)	not null		10
WORD	identificateur du mot	int(11)	not null		2646
TF	<i>term frequency</i>	int(11)	not null		3
TFIDF	<i>term frequency - inverse document frequency</i>			non utilisé	

TAB. C.4 – Table DOCUMENTS_CONTENTS

Champs	Description	Type 'MySQL'	Caractéristiques		Exemple
ID	identificateur du mot	int(11)	primary key, not null, auto increment		2646
WORD	mot	varchar(255)			théâtre

TAB. C.5 – Table WORDS

Champs	Description	Type 'MySQL'	Caractéristiques		Exemple
ID	identificateur de l'image	int(11)	primary key, not null, auto increment		30
URL	URL complète de l'image	varchar(255)	not null		*
HEIGHT	hauteur de l'image	int(11)			139
WIDTH	largeur de l'image	int(11)			202
SIZE	taille de l'image (en Ko)	int(11)			9
ALT	description de l'image	varchar(255)			

* <http://www.provenceweb.fr/grafiq/villes84/orange/theatre.jpg>

TAB. C.6 – Table IMAGES

Champs	Description	Type 'MySQL'	Caractéristiques	Exemple
IMAGE	identificateur de l'image	int(11)	not null	30
DOCUMENT	identificateur du document	int(11)	not null	10

TAB. C.7 – Table IMAGES_OF_DOCUMENTS

C.2 Exemple de fichier X-VRML

Dans cette section, nous donnons en guise d'exemple un extrait du code source du fichier `cityViewer.xvrm1`. Il s'agit du fichier permettant de réaliser la métaphore de la ville. Les balises X-VRML sont surlignées en jaune pour plus de visibilité. Par ailleurs, les parties du fichier supprimées pour obtenir cet extrait sont situées à l'endroit où apparaissent les caractères « #(...) » surlignés en vert.

```
<?xml version="1.0"?>< XVRML_DOCUMENT type="XVRML">< VRML_HEADER />
< model_interface background="0,0,0">
5   < component dataType="button" value="'main_button_2.jpg'" name="button_main" display="IMAGE"
      actions="clear3DScene;changeModel[@MODELS_PATH+'init.xvrm1']" posY="0.05" sizeX="0.07" sizeY="0.27" posX="0.01"/>
      < component dataType="button" value="'back_button_2.jpg'" name="button_back" display="IMAGE"
      actions="clear3DScene;changeModel['previousModel'];processCurrentModel3D" posY="0.05" sizeX="0.07" sizeY="0.27" posX="0.0795"/>
      < component dataType="button" value="'next_button_2.jpg'" name="button_next" display="IMAGE"
10  actions="clear3DScene;changeModel['nextModel'];processCurrentModel3D" posY="0.05" sizeX="0.07" sizeY="0.27" posX="0.1495"/>
      < component dataType="button" value="'help_button_2.jpg'" name="button_help" display="IMAGE"
      actions="loadURL[@MODELS_PATH+'aide/help1.html']" posY="0.05" sizeX="0.039" sizeY="0.27" posX="0.88"/>
      < component dataType="label" value="'logoft.jpg'" name="label_logoft" display="IMAGE" posY="0.02" sizeX="0.2" sizeY="0.71" posX="0.94"/>
15  < component dataType="label" value="'Query'" name="label_request" display="TEXT" fontName="Arial" fontStyle="bold"
      fontSize="15" posY="0.05" sizeX="0.14" sizeY="0.27" posX="0.26"/>
      < component dataType="text_field" value="''" name="text_request" actions="clear3DScene;changeModel[@MODELS_PATH+'cityViewer-mm.xvrm1']"
      fontName="Arial" fontStyle="plain" fontSize="15" posY="0.05" sizeX="0.2" sizeY="0.27" posX="0.3"/>
      < component dataType="button" value="'search_button_2.jpg'" name="button_search" display="IMAGE"
20  actions="clear3DScene;changeModel[@MODELS_PATH+'cityViewer-mm.xvrm1']" posY="0.05" sizeX="0.07" sizeY="0.27" posX="0.51"/>
      < component dataType="label" value="@info_query" name="label_results" display="TEXT" fontName="Arial" fontStyle="bold"
      fontSize="15" posY="0.4" sizeX="0.9" sizeY="0.27" posX="0.30" textColor="255,0,0"/>
      < component dataType="label" value="'Specific actions'" name="label_specific" display="TEXT" fontName="Arial" fontStyle="bold"
      fontSize="15" posY="0.05" sizeX="0.39" sizeY="0.27" posX="0.64"/>
25  < component dataType="combo" value="'none,add bookmarks,remove bookmarks'" data="'none;add;remove'" state="0" name="combo_specific"
      fontName="Arial" fontStyle="plain" fontSize="15" posY="0.05" sizeX="0.1" sizeY="0.27" posX="0.75"/>
      < component dataType="label" value="'Descriptive words'" name="label_descriptiveWords" display="TEXT" fontName="Arial"
      fontStyle="bold" fontSize="15" posY="0.4" sizeX="0.43" sizeY="0.27" posX="0.64"/>
30  < component dataType="combo" value="'5,10,20,50,100,200,300,500'" data="'5;10;20;50;100;200;300;500'" state="4" name="nk"
      fontName="Arial" fontStyle="plain" fontSize="15" posY="0.4" sizeX="0.1" sizeY="0.27" posX="0.75"/>
      < component dataType="label" value="'CityViewer'" name="label_city" display="TEXT" fontName="Arial" fontStyle="bold"
      fontSize="15" posY="0.4" sizeX="0.25" sizeY="0.27" posX="0.01"/>
35  < initial_actions >
      < action actions="processCurrentModel3D"/>
    </ initial_actions >
</ model_interface >
```

```

40 </ model_interface >

< model_body >

< SW_CONNECT >

45 <@INLINE FILE="/protos/cityViewer_protos.wrl"/>

< BINDABLENODE NAME="'NAVINFO'" >
NavigationInfo
50 {
    type
    [
        "FLY"
55 ]
}
</ BINDABLENODE >

DEF cam Camera
60 {
}

*****
#* Front light *
#*****
65 DirectionalLight
{
    direction 0 -1 0
    intensity 0.7
70 }

*****
#* TRANSPORT INFO *
#*****
75 DEF T_INFO TransportInfo
{
}

*****
80 #* DATABASE QUERY *
#*****

< SET NAME="'total'" VALUE="-1"/>
< SW_DOCUMENT_QUERY KEYWORDS="@text_request" RESULT="'res'" COUNTER="'total'" MAX="'-1'"/>
85 < SET NAME="'total'" VALUE="$total + 1"/>

< IF CONDITION="$total > 0">
< THEN >

90 *****
#* MAPPING *
#*****

< SW_CITY_MAP INPUT="'res'" OUTPUT="'res'" TYPE="'kohonen'" PARAMS="gridx = ##5; gridz = ##5; kwsnb = ##($nkw);
95 size = 30; min = 'min'; max = 'max'; xlabel = 'xlabel'; ylabel = 'ylabel'; width = 'width'; depth = 'depth';
specific=@combo_specific; query=@text_request; offline=##1; hacLabels='hacLabels'; time=##1"/>
< CONSOLE VALUE="@hacLabels"/>
< CONSOLE VALUE="'time'"/>
< CONSOLE VALUE="@time"/>
100 *****
#* Background BACK definition *
#*****
< BINDABLENODE NAME="'BACK'" >
105 < IF CONDITION="$time==0">
< THEN >
Background
{
110    groundColor 0 0.65 0.1
    skyColor 0.3 0.6 1
    bottomUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_bottom.png"

```



```

frontUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side.png"
leftUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side.png"
rightUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side.png"
115 backUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side.png"
topUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_top.png"
}
</ THEN >
< ELSE >
120 Background
{
groundColor 0 0.65 0.1
skyColor 0.3 0.6 1
bottomUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_bottom_dark.png"
125 frontUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side_dark.png"
leftUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side_dark.png"
rightUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side_dark.png"
backUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_side_dark.png"
topUrl "< INSERT VALUE="@MODELS_PATH"/>/textures/sky_top_dark.png"
130 }
</ ELSE >
</ IF >
</ BINDABLENODE >
135 DEF VIEW1 Viewpoint
{
orientation 0 0 1 0
description "Front"
}
140 DEF VIEW2 Viewpoint
{
orientation 1 0 0 -1.57
description "Top"
145 }

#*****
#* Bouton extensible de navigation (barres de navigation) *
#*****
150 #(. . .)

#*****

155 DrawGroup
{
children
[
#*****
160 #* Ground *
#*****
Transform
{
translation 0 -0.05 0
165 children
[
Shape
{
geometry Box
170 {
size < INSERT VALUE="$width + 200"/> 0.0001 < INSERT VALUE="$depth + 200"/>
}
appearance Appearance
175 {
texture ImageTexture
{
url ["< INSERT VALUE="@MODELS_PATH"/>/textures/road2.jpg"]
repeatS TRUE
repeatT TRUE
180 }
textureTransform TextureTransform
{
scale 20 20
185 }
}
}
}
}

```

```

    }
  }
}
190 DEF Domaines Group
  {
    children
    [
195     < SET NAME='stepHeight' VALUE="0.2"/>

     < SET NAME='lastx' VALUE="##(-1)"/>
     < SET NAME='xmin' VALUE="##(-1)"/>
     < SET NAME='zmin' VALUE="##(-1)"/>
200     < SET NAME='xmin0' VALUE="##(-1)"/>
     < SET NAME='zmin0' VALUE="##(-1)"/>
     < SET NAME='xmax' VALUE="##(-1)"/>
     < SET NAME='zmax' VALUE="##(-1)"/>

205     < PTRD_UTIL OP="iterate" PARAMS="input = 'res'; counter = 'cl' ">

     < IF CONDITION="$lastx!=$gridx">
     < THEN >
       < IF CONDITION="$lastx!=-1">
210       < THEN >
         ]
         pos_c < INSERT VALUE="($xmin+$xmax)/2"/> 0 < INSERT VALUE="($zmin+$zmax)/2"/>
         size < INSERT VALUE="$xmax - $xmin"/> 0.01 < INSERT VALUE="$zmin - $zmax"/>
         }#fermeture Domaine
215       </ THEN >
     < ELSE >
       < SET NAME='xmin0' VALUE="$x1"/>
       < SET NAME='zmin0' VALUE="$z1"/>
     </ ELSE >
220     </ IF >
     < SET NAME='lastx' VALUE="$gridx"/>
     < SET NAME='xmin' VALUE="$x1"/>
     < SET NAME='zmin' VALUE="$z1"/>
     < STR_GET_TOKEN NAME='tmp' TEXT="@xlabels" CHAR=",'" N="##($gridx + 1)"/>

225     DEF Domaine< INSERT VALUE="##($gridx)"/> Domaine
     {
       url "< INSERT VALUE="@tmp"/>"
       translation < INSERT VALUE="($x1 + $x2) / 2"/> 0 0
230       children
       [
         </ THEN >
         </ IF >

235         # pavement
         < SET NAME='xmax' VALUE="$x2"/>
         < SET NAME='zmax' VALUE="$z2"/>

         < IF CONDITION="$neuron_doc_number != 0">
240         < THEN >

           DEF BLOCK_< INSERT VALUE="#gridx + '_' + #gridz"/> Block
           {
             translation 0 0 < INSERT VALUE="($z1 + $z2) / 2"/>
245             size < INSERT VALUE="$x2 - $x1"/> 0.001 < INSERT VALUE="$z1 - $z2"/>
             position 0 0 < INSERT VALUE="(($z1 + $z2) / 2)+(($z1 - $z2)/2)"/>
             type < INSERT VALUE="##($neuron_clusterValue-1)"/>

```

```

children
[
# buildings
250 < FTRD_UTIL OP="iterate" PARAMS="input = 'buildings'; counter = 'c2'">
  < IF CONDITION="$max != $min">
    < THEN >
      < SET NAME="'height'" VALUE="((($size - $min) / ($max - $min)) * 15) + 4"/>
255 </ THEN >
    < ELSE >
      < SET NAME="'height'" VALUE="6"/>
    </ ELSE >
  </ IF >

260 Transform
{
  translation < INSERT VALUE="$x - ($x1 + $x2)/2"/> < INSERT VALUE="$y"/>
  < INSERT VALUE="$z - ($z1 + $z2)/2"/>
  children
  [
    < SET NAME="'info'" VALUE="'INFO_' + #c1 + '_' + #c2"/>

    DEF < INSERT VALUE="@info"/> Info
270 {
      text
      [
        < IF CONDITION="$desc != null">
          < THEN >
275 " < INSERT VALUE="@desc"/>
          </ THEN >
          < ELSE >
            " < INSERT VALUE="'No description available.'"/>
          </ ELSE >
        </ IF >
      ]
      info
      [
285 < IF CONDITION="$keywords != null">
          < THEN >
            " < INSERT VALUE="@keywords"/>
          </ THEN >
          < ELSE >
290 " < INSERT VALUE="'No keyword.'"/>
          </ ELSE >
        </ IF >
      ]
      address
      [
295 " < INSERT VALUE="@URL"/>
      ]

      < SET NAME="'i'" VALUE="##1"/>
      < STR_GET_TOKEN NAME="'tmp'" TEXT="@neighbors_urls" CHAR="';'" N="##($i)"/>
300 < WHILE CONDITION="@tmp!=''">
        address< INSERT VALUE="##($i+1)"/> [ "< INSERT VALUE="@tmp"/> ]
        < INC NAME="'i'"/>
        < STR_GET_TOKEN NAME="'tmp'" TEXT="@neighbors_urls" CHAR="';'" N="##($i)"/>
      </ WHILE >

305 < SET NAME="'i'" VALUE="##1"/>
      < STR_GET_TOKEN NAME="'tmp'" TEXT="@neighbors_desc" CHAR="'|'" N="##($i)"/>
      < WHILE CONDITION="@tmp!=''">
        text< INSERT VALUE="##($i+1)"/>

```

```

310         [
            < IF CONDITION="$tmp != null">
                < THEN >
                    "< INSERT VALUE="@tmp"/>"
                </ THEN >
315            < ELSE >
                "No description available."
            </ ELSE >
        </ IF >
    ]
320    < INC NAME="'i'"/>
    < STR_GET_TOKEN NAME="'tmp' " TEXT="@neighbors_desc" CHAR="|" N="##($i)"/>
</ WHILE >

< SET NAME="'i' " VALUE="##1"/>
325 < STR_GET_TOKEN NAME="'tmp' " TEXT="@neighbors_keywords" CHAR=";" N="##($i)"/>
< WHILE CONDITION="@tmp!="">
    info< INSERT VALUE="##($i+1)"/>
    [
330        < IF CONDITION="$tmp != null">
            < THEN >
                "< INSERT VALUE="@tmp"/>"
            </ THEN >
            < ELSE >
                "No keyword."
335        </ ELSE >
        </ IF >
    ]
    < INC NAME="'i'"/>
    < STR_GET_TOKEN NAME="'tmp' " TEXT="@neighbors_keywords" CHAR=";" N="##($i)"/>
340 </ WHILE >

children
[
345 < SET NAME="'building' " VALUE="'BUILDING_' + #c1 + '_' + #c2"/>
< SET NAME="'color' " VALUE="'0.6 0.6 0.6'"/>
< IF CONDITION="($videos != 0) || ($audios != 0)">
    < THEN >
        < SET NAME="'color' " VALUE="'0.8 0.5 0.2'"/>
    </ THEN >
350 </ IF >

< NAMEDNODE NAME="'ACTION_' + #c1 + '_' + #c2">
DEF < INSERT VALUE="@building"/> Building
{
355    type < INSERT VALUE="##($lang - 1)"/>
    tall 20
    rank < INSERT VALUE="##($rank)"/>
    height 15
    roofColor < INSERT VALUE="@color"/>
360    url [ "< INSERT VALUE="@URL"/>" ]
    snapshot [ "< INSERT VALUE="@snapshot"/>" ]
    images < INSERT VALUE="##($images-1)"/>
    bookmark < INSERT VALUE="##($bookmark-1)"/>
}
365 </ NAMEDNODE >
]
}

< listener_action node = "'ACTION_' + #c1 + '_' + #c2" event="'action'"

```

```

370      actions="clear3DScene;changeModel[@MODELS_PATH + 'image-mm.xvrm', 'addr' = @URL,
      'text_request' = @text_request]"/>
      ROUTE < INSERT VALUE="@info"/>.text_changed TO T_INFO.text_text_in
      ROUTE < INSERT VALUE="@info"/>.address_changed TO T_INFO.text_address_in
375      ROUTE < INSERT VALUE="@info"/>.info_changed TO T_INFO.text_info_in
      ROUTE < INSERT VALUE="@info"/>.text_default TO T_INFO.text_default
      ROUTE < INSERT VALUE="@info"/>.info_default TO T_INFO.info_default
      ROUTE < INSERT VALUE="@info"/>.address_default TO T_INFO.address_default
380      #(. . .)
      ROUTE < INSERT VALUE="@building"/>.isVisited TO <INSERT VALUE="@info"/>.isVisited
      ROUTE < INSERT VALUE="@building"/>.isSelected TO <INSERT VALUE="@info"/>.isSelected
385      ROUTE < INSERT VALUE="@building"/>.isOver TO <INSERT VALUE="@info"/>.isOver
      ROUTE < INSERT VALUE="@building"/>.isOver TO T_INFO.isOver
390      ]
      </ FTRD_UTIL >
    }
  </ THEN >
  </ IF >
395 </ FTRD_UTIL >
#(. . .)
400

```

Il n'existe pas de document contenant les spécifications du langage X-VRML dans sa version évoluée telle qu'elle est utilisée dans le prototype *SmartWeb*. Certaines informations sont toutefois données dans la section 4.3.1. Par ailleurs, de nombreuses balises du langage n'ont pas évolué et sont issues des premières spécifications du langage [Poznan 2001].

Annexe D

Résultats de l'ACM

Cette annexe contient l'ensemble des résultats issus de l'analyse des correspondances multiples (ACM) sur un sous-ensemble du questionnaire utilisateur proposé dans le chapitre 5. L'interprétation des réponses des utilisateurs à ce sous-ensemble de questions, proposée dans la section 5.1.3.3, est en partie fondée sur les informations contenues dans cette annexe (figures D.1 à D.10).

```
SELECTION DES INDIVIDUS ET DES VARIABLES UTILES
VARIABLES NOMINALES ACTIVES
  16 VARIABLES      80 MODALITES ASSOCIEES
-----
  2 . satisfaction / SmartWeb          ( 5 MODALITES )
  3 . intérêt / SmartWeb              ( 5 MODALITES )
  5 . utilisation du prototype         ( 5 MODALITES )
  6 . vitesse du système               ( 5 MODALITES )
  8 . lecture des caractères          ( 5 MODALITES )
  9 . affichage contextuel            ( 5 MODALITES )
 11 . séquence des affichages         ( 5 MODALITES )
 17 . apprentissage système           ( 5 MODALITES )
 21 . destiné à tous                  ( 5 MODALITES )
 24 . pertinence métaphore            ( 5 MODALITES )
 25 . compréhension métaphore        ( 5 MODALITES )
 27 . déplacement dans la ville       ( 5 MODALITES )
 31 . pertinence de la représentation ( 5 MODALITES )
 33 . navigation dans la représentation ( 5 MODALITES )
 35 . utilité de la galerie 3D         ( 5 MODALITES )
 39 . visualisation plus abstraite    ( 5 MODALITES )
-----
VARIABLES NOMINALES ILLUSTRATIVES
  4 VARIABLES      13 MODALITES ASSOCIEES
-----
 40 . âge                             ( 4 MODALITES )
 41 . sexe                             ( 2 MODALITES )
 42 . jeux 3D                          ( 2 MODALITES )
 43 . temps par semaine sur un navigateur ( 5 MODALITES )
-----
INDIVIDUS
----- NOMBRE ----- POIDS -----
POIDS DES INDIVIDUS: Poids des individus, uniforme egal a 1.          UNIF
RETENUS ..... NITOT = 60      PITOT = 60.000
ACTIFS ..... NIACT = 60      PIACT = 60.000
SUPPLEMENTAIRES .... NISUP = 0      PISUP = 0.000
-----
```

FIG. D.1 – Sélection des individus et des variables utiles

ANALYSE DES CORRESPONDANCES MULTIPLES
 APUREMENT DES MODALITES ACTIVES
 SEUIL (PCMIN) : 7.00 % POIDS: 4.20
 AVANT APUREMENT : 16 QUESTIONS ACTIVES 80 MODALITES ASSOCIEES
 APRES : 16 QUESTIONS ACTIVES 64 MODALITES ASSOCIEES
 POIDS TOTAL DES INDIVIDUS ACTIFS : 60.00
 TRI-A-PLAT DES QUESTIONS ACTIVES

IDENT	MODALITES LIBELLE	AVANT APUREMENT		APRES APUREMENT		HISTOGRAMME DES POIDS RELATIFS
		EFF.	POIDS	EFF.	POIDS	

2 . satisfaction / SmartWeb						
1	- 32-1 (mauvais)	15	15.00	17	17.00	*****
2	- 32-2	19	19.00	21	21.00	*****
3	- 32-3	18	18.00	22	22.00	*****
4	- 32-4 (excellent)	4	4.00	===	VENTILEE	===
5	- 32-NA	4	4.00	===	VENTILEE	===

3 . intérêt / SmartWeb						
1	- 33-1 (mauvais)	5	5.00	6	6.00	*****
2	- 33-2	20	20.00	20	20.00	*****
3	- 33-3	22	22.00	23	23.00	*****
4	- 33-4 (excellent)	11	11.00	11	11.00	*****
5	- 33-NA	2	2.00	===	VENTILEE	===

5 . utilisation du prototype						
1	- 35-1 (mauvais)	15	15.00	15	15.00	*****
2	- 35-2	21	21.00	21	21.00	*****
3	- 35-3	13	13.00	15	15.00	*****
4	- 35-4 (excellent)	9	9.00	9	9.00	*****
5	- 35-NA	2	2.00	===	VENTILEE	===

6 . vitesse du système						
1	- 36-1 (mauvais)	18	18.00	18	18.00	*****
2	- 36-2	14	14.00	15	15.00	*****
3	- 36-3	14	14.00	15	15.00	*****
4	- 36-4 (excellent)	11	11.00	12	12.00	*****
5	- 36-NA	3	3.00	===	VENTILEE	===

8 . lecture des caractères						
1	- 41-1 (mauvais)	13	13.00	13	13.00	*****
2	- 41-2	15	15.00	16	16.00	*****
3	- 41-3	15	15.00	15	15.00	*****
4	- 41-4 (excellent)	14	14.00	16	16.00	*****
5	- 41-NA	3	3.00	===	VENTILEE	===

9 . affichage contextuel						
1	- 42-1 (mauvais)	10	10.00	10	10.00	*****
2	- 42-2	10	10.00	10	10.00	*****
3	- 42-3	25	25.00	27	27.00	*****
4	- 42-4 (excellent)	12	12.00	13	13.00	*****
5	- 42-NA	3	3.00	===	VENTILEE	===

11 . séquence des affichages						
1	- 44-1 (mauvais)	9	9.00	10	10.00	*****
2	- 44-2	20	20.00	20	20.00	*****
3	- 44-3	19	19.00	19	19.00	*****
4	- 44-4 (excellent)	10	10.00	11	11.00	*****
5	- 44-NA	2	2.00	===	VENTILEE	===

FIG. D.2 – Apurement des modalités actives (partie 1)

17 . apprentissage système						
1	- 61-1 (mauvais)	13	13.00	14	14.00	*****
2	- 61-2	12	12.00	12	12.00	*****
3	- 61-3	17	17.00	17	17.00	*****
4	- 61-4 (excellent)	17	17.00	17	17.00	*****
5	- 61-NA	1	1.00	=== VENTILEE ===		

21 . destiné à tous						
1	- 65-1 (mauvais)	19	19.00	19	19.00	*****
2	- 65-2	17	17.00	18	18.00	*****
3	- 65-3	5	5.00	5	5.00	*****
4	- 65-4 (excellent)	17	17.00	18	18.00	*****
5	- 65-NA	2	2.00	=== VENTILEE ===		

♀						

IDENT	MODALITES LIBELLE	AVANT APUREMENT EFF.	POIDS	APRES APUREMENT EFF.	POIDS	HISTOGRAMME DES POIDS RELATIFS

24 . pertinence métaphore						
1	- 81-1 (mauvais)	9	9.00	9	9.00	*****
2	- 81-2	13	13.00	14	14.00	*****
3	- 81-3	24	24.00	25	25.00	*****
4	- 81-4 (excellent)	12	12.00	12	12.00	*****
5	- 81-NA	2	2.00	=== VENTILEE ===		

25 . compréhension métaphore						
1	- 82-1 (mauvais)	11	11.00	13	13.00	*****
2	- 82-2	11	11.00	11	11.00	*****
3	- 82-3	21	21.00	21	21.00	*****
4	- 82-4 (excellent)	15	15.00	15	15.00	*****
5	- 82-NA	2	2.00	=== VENTILEE ===		

27 . déplacement dans la ville						
1	- 84-1 (mauvais)	23	23.00	25	25.00	*****
2	- 84-2	11	11.00	12	12.00	*****
3	- 84-3	9	9.00	9	9.00	*****
4	- 84-4 (excellent)	14	14.00	14	14.00	*****
5	- 84-NA	3	3.00	=== VENTILEE ===		

31 . pertinence de la représentation						
1	- 92-1 (mauvais)	5	5.00	5	5.00	*****
2	- 92-2	12	12.00	12	12.00	*****
3	- 92-3	24	24.00	26	26.00	*****
4	- 92-4 (excellent)	15	15.00	17	17.00	*****
5	- 92-NA	4	4.00	=== VENTILEE ===		

33 . navigation dans la représentation						
1	- 94-1 (mauvais)	5	5.00	6	6.00	*****
2	- 94-2	9	9.00	11	11.00	*****
3	- 94-3	25	25.00	25	25.00	*****
4	- 94-4 (excellent)	17	17.00	18	18.00	*****
5	- 94-NA	4	4.00	=== VENTILEE ===		

35 . utilité de la galerie 3D						
1	- 96-1 (mauvais)	12	12.00	12	12.00	*****
2	- 96-2	7	7.00	7	7.00	*****
3	- 96-3	13	13.00	13	13.00	*****
4	- 96-4 (excellent)	23	23.00	23	23.00	*****
5	- 96-NA	5	5.00	5	5.00	*****

39 . visualisation plus abstraite						
1	- 102-1 (mauvais)	2	2.00	=== VENTILEE ===		
2	- 102-2	17	17.00	18	18.00	*****
3	- 102-3	11	11.00	11	11.00	*****
4	- 102-4 (excellent)	23	23.00	24	24.00	*****
5	- 102-NA	7	7.00	7	7.00	*****

FIG. D.3 – Apurement des modalités actives (partie 2)

VALEURS PROPRES
 APERCU DE LA PRECISION DES CALCULS : TRACE AVANT DIAGONALISATION .. 3.0000
 SOMME DES VALEURS PROPRES 3.0000

HISTOGRAMME DES 48 PREMIERES VALEURS PROPRES

NUMERO	VALEUR PROPRE	POURCENTAGE	POURCENTAGE CUMULE	
1	0.2829	9.43	9.43	*****
2	0.2125	7.08	16.51	*****
3	0.1783	5.94	22.46	*****
4	0.1719	5.73	28.19	*****
5	0.1671	5.57	33.76	*****
6	0.1548	5.16	38.92	*****
7	0.1422	4.74	43.66	*****
8	0.1286	4.29	47.94	*****
9	0.1165	3.88	51.83	*****
10	0.1040	3.47	55.30	*****
11	0.0990	3.30	58.60	*****
12	0.0952	3.17	61.77	*****
13	0.0897	2.99	64.76	*****
14	0.0774	2.58	67.34	*****
15	0.0718	2.39	69.73	*****
16	0.0708	2.36	72.09	*****
17	0.0700	2.33	74.43	*****
18	0.0691	2.30	76.73	*****
19	0.0589	1.96	78.70	*****
20	0.0580	1.93	80.63	*****
21	0.0530	1.77	82.40	*****
22	0.0496	1.65	84.05	*****
23	0.0479	1.60	85.65	*****
24	0.0452	1.51	87.16	*****
25	0.0418	1.39	88.55	*****
26	0.0386	1.29	89.83	*****
27	0.0352	1.17	91.01	*****
28	0.0315	1.05	92.06	*****
29	0.0296	0.99	93.04	*****
30	0.0272	0.91	93.95	*****
31	0.0257	0.86	94.80	*****
32	0.0212	0.71	95.51	*****
33	0.0198	0.66	96.17	*****
34	0.0174	0.58	96.75	*****
35	0.0159	0.53	97.28	*****
36	0.0135	0.45	97.74	****
37	0.0121	0.40	98.14	****
38	0.0114	0.38	98.52	****
39	0.0100	0.33	98.85	***
40	0.0086	0.29	99.14	***
41	0.0064	0.21	99.35	**
42	0.0051	0.17	99.52	**
43	0.0044	0.15	99.67	**
44	0.0032	0.11	99.77	*
45	0.0025	0.08	99.86	*
46	0.0022	0.07	99.93	*
47	0.0016	0.05	99.99	*
48	0.0004	0.01	100.00	*

FIG. D.4 – Histogramme des valeurs propres

RECHERCHE DE PALIERS (DIFFERENCES TROISIEMES)		
PALIER ENTRE	VALEUR DU PALIER	
2 -- 3	-25.98	*****
18 -- 19	-13.41	*****
5 -- 6	-9.23	*****
1 -- 2	-8.59	*****
9 -- 10	-6.21	*****
13 -- 14	-4.97	*****
14 -- 15	-4.30	*****
31 -- 32	-4.19	*****
12 -- 13	-3.04	*****
21 -- 22	-2.64	*****
27 -- 28	-2.38	*****
35 -- 36	-1.83	****
7 -- 8	-1.82	****
36 -- 37	-1.46	***
41 -- 42	-1.20	***
40 -- 41	-0.65	**
24 -- 25	-0.49	*
15 -- 16	-0.32	*
RECHERCHE DE PALIERS ENTRE (DIFFERENCES SECONDES)		
PALIER ENTRE	VALEUR DU PALIER	
1 -- 2	36.25	*****
2 -- 3	27.66	*****
18 -- 19	9.29	*****
9 -- 10	7.46	*****
13 -- 14	6.81	*****
14 -- 15	4.51	*****
31 -- 32	3.16	*****
27 -- 28	1.85	***
3 -- 4	1.69	***
21 -- 22	1.69	***
20 -- 21	1.61	***
7 -- 8	1.45	***
10 -- 11	1.25	**
29 -- 30	0.96	**
35 -- 36	0.93	**
33 -- 34	0.91	**
40 -- 41	0.83	**
36 -- 37	0.80	**
41 -- 42	0.68	*
24 -- 25	0.26	*
15 -- 16	0.21	*

FIG. D.5 – Valeurs propres — recherche de paliers

COORDONNEES, CONTRIBUTIONS ET COSINUS CARRÉS DES MODALITES ACTIVES																		
AXES 1 A 5																		
MODALITES		COORDONNEES					CONTRIBUTIONS					COSINUS CARRÉS						
IDEN - LIBELLE	P.REL	DISTO	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
2 . satisfaction / SmartWeb																		
1	- 32-1 (mauvais)	1.77	2.53	-0.64	-0.27	0.09	-0.64	-0.38	2.6	0.6	0.1	4.2	1.5	0.16	0.03	0.00	0.16	0.06
2	- 32-2	2.19	1.86	-0.26	0.14	0.15	0.51	0.60	0.5	0.2	0.3	3.3	4.7	0.04	0.01	0.01	0.14	0.19
3	- 32-3	2.29	1.73	0.75	0.08	-0.22	0.01	-0.27	4.5	0.1	0.6	0.0	1.0	0.32	0.00	0.03	0.00	0.04
									CONTRIBUTION CUMULEE = 7.6 0.9 1.0 7.5 7.3									
3 . intérêt / SmartWeb																		
1	- 33-1 (mauvais)	0.63	9.00	-0.87	-1.27	0.02	-0.28	-0.61	1.7	4.8	0.0	0.3	1.4	0.08	0.18	0.00	0.01	0.04
2	- 33-2	2.08	2.00	-0.03	0.11	-0.38	-0.64	0.29	0.0	0.1	1.7	4.9	1.0	0.00	0.01	0.07	0.20	0.04
3	- 33-3	2.40	1.61	-0.10	0.29	0.23	0.32	0.32	0.1	0.9	0.7	1.4	1.5	0.01	0.05	0.03	0.06	0.06
4	- 33-4 (excellent)	1.15	4.45	0.75	-0.11	0.20	0.64	-0.86	2.3	0.1	0.3	2.7	5.1	0.13	0.00	0.01	0.09	0.17
									CONTRIBUTION CUMULEE = 4.1 5.9 2.7 9.3 9.0									
5 . utilisation du prototype																		
1	- 35-1 (mauvais)	1.56	3.00	-0.90	-0.42	-0.47	-0.22	0.35	4.5	1.3	1.9	0.4	1.2	0.27	0.06	0.07	0.02	0.04
2	- 35-2	2.19	1.86	0.07	0.11	0.45	0.44	-0.71	0.0	0.1	2.5	2.4	6.6	0.00	0.01	0.11	0.10	0.27
3	- 35-3	1.56	3.00	-0.09	0.47	-0.34	-0.07	0.19	0.0	1.7	1.0	0.1	0.3	0.00	0.08	0.04	0.00	0.01
4	- 35-4 (excellent)	0.94	5.67	1.49	-0.36	0.28	-0.53	0.75	7.4	0.6	0.4	1.5	3.2	0.39	0.02	0.01	0.05	0.10
									CONTRIBUTION CUMULEE = 11.9 3.6 5.9 4.4 11.3									
6 . vitesse du système																		
1	- 36-1 (mauvais)	1.88	2.33	-0.66	-0.52	0.34	-0.38	-0.29	2.9	2.4	1.2	1.6	1.0	0.19	0.12	0.05	0.06	0.04
2	- 36-2	1.56	3.00	-0.33	0.74	-0.02	0.24	0.13	0.6	4.0	0.0	0.5	0.2	0.04	0.18	0.00	0.02	0.01
3	- 36-3	1.56	3.00	0.25	0.02	-0.63	0.33	-0.07	0.3	0.0	3.5	1.0	0.0	0.02	0.00	0.13	0.04	0.00
4	- 36-4 (excellent)	1.25	4.00	1.09	-0.16	0.30	-0.14	0.37	5.3	0.2	0.6	0.1	1.0	0.30	0.01	0.02	0.00	0.03
									CONTRIBUTION CUMULEE = 9.1 6.6 5.4 3.2 2.2									
8 . lecture des caractères																		
1	- 41-1 (mauvais)	1.35	3.62	-0.48	-0.46	0.47	0.25	0.67	1.1	1.3	1.7	0.5	3.7	0.06	0.06	0.06	0.02	0.13
2	- 41-2	1.67	2.75	-0.31	0.26	-0.97	-0.68	0.09	0.6	0.5	8.9	4.5	0.1	0.04	0.02	0.35	0.17	0.00
3	- 41-3	1.56	3.00	0.17	0.71	0.33	0.05	0.01	0.2	3.7	1.0	0.0	0.0	0.01	0.17	0.04	0.00	0.00
4	- 41-4 (excellent)	1.67	2.75	0.55	-0.56	0.28	0.43	-0.65	1.8	2.4	0.8	1.8	4.2	0.11	0.11	0.03	0.07	0.15
									CONTRIBUTION CUMULEE = 3.6 8.0 12.3 6.8 7.9									
9 . affichage contextuel																		
1	- 42-1 (mauvais)	1.04	5.00	-0.73	-0.16	0.87	-0.85	-0.52	2.0	0.1	4.4	4.4	1.7	0.11	0.00	0.15	0.15	0.05
2	- 42-2	1.04	5.00	-0.32	0.09	-0.23	0.82	-0.85	0.4	0.0	0.3	4.1	4.5	0.02	0.00	0.01	0.14	0.14
3	- 42-3	2.81	1.22	0.05	0.29	-0.14	-0.05	0.49	0.0	1.1	0.3	0.0	4.1	0.00	0.07	0.02	0.00	0.20
4	- 42-4 (excellent)	1.35	3.62	0.70	-0.55	-0.21	0.13	0.03	2.4	1.9	0.3	0.1	0.0	0.14	0.08	0.01	0.00	0.00
									CONTRIBUTION CUMULEE = 4.8 3.2 5.4 8.7 10.3									
11 . séquence des affichages																		
1	- 44-1 (mauvais)	1.04	5.00	-0.76	-0.99	0.30	-0.16	0.51	2.1	4.8	0.5	0.2	1.6	0.11	0.19	0.02	0.01	0.05
2	- 44-2	2.08	2.00	-0.17	0.26	0.09	0.00	0.60	0.2	0.7	0.1	0.0	4.4	0.01	0.03	0.00	0.00	0.18
3	- 44-3	1.98	2.16	0.05	0.51	-0.09	0.07	-0.82	0.0	2.4	0.1	0.1	7.9	0.00	0.12	0.00	0.00	0.31
4	- 44-4 (excellent)	1.15	4.45	0.91	-0.46	-0.28	0.02	-0.13	3.3	1.1	0.5	0.0	0.1	0.18	0.05	0.02	0.00	0.00
									CONTRIBUTION CUMULEE = 5.7 9.0 1.2 0.2 14.0									
17 . apprentissage système																		
1	- 61-1 (mauvais)	1.46	3.29	-0.35	-0.16	-0.68	0.54	0.19	0.6	0.2	3.7	2.5	0.3	0.04	0.01	0.14	0.09	0.01
2	- 61-2	1.25	4.00	-0.49	-0.73	-0.09	0.34	0.15	1.1	3.1	0.1	0.9	0.2	0.06	0.13	0.00	0.03	0.01
3	- 61-3	1.77	2.53	-0.33	0.87	0.08	-0.36	-0.66	0.7	6.3	0.1	1.3	4.7	0.04	0.30	0.00	0.05	0.17
4	- 61-4 (excellent)	1.77	2.53	0.97	-0.22	0.54	-0.33	0.40	5.9	0.4	2.9	1.1	1.7	0.37	0.02	0.12	0.04	0.06
									CONTRIBUTION CUMULEE = 8.3 10.0 6.8 5.8 6.9									
21 . destiné à tous																		
1	- 65-1 (mauvais)	1.98	2.16	-0.56	-0.27	-0.54	-0.40	-0.43	2.2	0.7	3.2	1.8	2.2	0.15	0.03	0.13	0.07	0.09
2	- 65-2	1.88	2.33	0.33	0.08	-0.13	-0.19	0.38	0.7	0.0	0.2	0.4	1.6	0.05	0.00	0.01	0.02	0.06
3	- 65-3	0.52	11.00	-0.91	0.47	0.44	1.81	0.78	1.5	0.5	0.6	9.9	1.9	0.08	0.02	0.02	0.30	0.05
4	- 65-4 (excellent)	1.88	2.33	0.52	0.08	0.57	0.11	-0.14	1.8	0.1	3.4	0.1	0.2	0.11	0.00	0.14	0.00	0.01
									CONTRIBUTION CUMULEE = 6.3 1.3 7.3 12.2 5.9									

FIG. D.6 – Coordonnées, contributions et qualité de représentation des modalités actives (partie 1)

21 . destiné à tous																			
1	- 65-1 (mauvais)	1.98	2.16	-0.56	-0.27	-0.54	-0.40	-0.43	2.2	0.7	3.2	1.8	2.2	0.15	0.03	0.13	0.07	0.09	
2	- 65-2	1.88	2.33	0.33	0.08	-0.13	-0.19	0.38	0.7	0.0	0.2	0.4	1.6	0.05	0.00	0.01	0.02	0.06	
3	- 65-3	0.52	11.00	-0.91	0.47	0.44	1.81	0.78	1.5	0.5	0.6	9.9	1.9	0.08	0.02	0.02	0.30	0.05	
4	- 65-4 (excellent)	1.88	2.33	0.52	0.08	0.57	0.11	-0.14	1.8	0.1	3.4	0.1	0.2	0.11	0.00	0.14	0.00	0.01	
+-----CONTRIBUTION CUMULEE = 6.3 1.3 7.3 12.2 5.9 +-----																			
24 . pertinence métaphore																			
1	- 81-1 (mauvais)	0.94	5.67	-0.82	-0.41	0.00	-1.12	0.16	2.2	0.7	0.0	6.8	0.1	0.12	0.03	0.00	0.22	0.00	
2	- 81-2	1.46	3.29	-0.31	-0.03	0.31	0.92	0.55	0.5	0.0	0.8	7.2	2.7	0.03	0.00	0.03	0.26	0.09	
3	- 81-3	2.60	1.40	0.12	0.08	-0.38	0.04	-0.25	0.1	0.1	2.1	0.0	1.0	0.01	0.00	0.10	0.00	0.04	
4	- 81-4 (excellent)	1.25	4.00	0.73	0.17	0.42	-0.31	-0.24	2.4	0.2	1.3	0.7	0.4	0.13	0.01	0.04	0.02	0.01	
+-----CONTRIBUTION CUMULEE = 5.2 1.0 4.2 14.8 4.2 +-----																			
25 . compréhension métaphore																			
1	- 82-1 (mauvais)	1.35	3.62	-0.54	-0.77	-0.57	0.31	0.49	1.4	3.7	2.4	0.7	1.9	0.08	0.16	0.09	0.03	0.07	
2	- 82-2	1.15	4.45	-0.50	0.66	-0.19	-0.53	-0.16	1.0	2.3	0.2	1.9	0.2	0.06	0.10	0.01	0.06	0.01	
3	- 82-3	2.19	1.86	-0.19	0.22	0.07	0.34	-0.37	0.3	0.5	0.1	1.5	1.8	0.02	0.03	0.00	0.06	0.07	
4	- 82-4 (excellent)	1.56	3.00	1.10	-0.13	0.53	-0.35	0.21	6.7	0.1	2.5	1.1	0.4	0.40	0.01	0.09	0.04	0.02	
+-----CONTRIBUTION CUMULEE = 9.4 6.7 5.2 5.2 4.3 +-----																			
27 . déplacement dans la ville																			
1	- 84-1 (mauvais)	2.60	1.40	-0.57	-0.33	-0.02	0.08	-0.19	3.0	1.4	0.0	0.1	0.5	0.23	0.08	0.00	0.01	0.02	
2	- 84-2	1.25	4.00	-0.27	0.52	0.57	-0.24	-0.20	0.3	1.6	2.3	0.4	0.3	0.02	0.07	0.08	0.01	0.01	
3	- 84-3	0.94	5.67	0.17	0.79	-0.32	-0.02	0.91	0.1	2.8	0.5	0.0	4.6	0.01	0.11	0.02	0.00	0.15	
4	- 84-4 (excellent)	1.46	3.29	1.13	-0.35	-0.24	0.07	-0.08	6.6	0.9	0.5	0.0	0.1	0.39	0.04	0.02	0.00	0.00	
+-----CONTRIBUTION CUMULEE = 10.0 6.6 3.3 0.6 5.5 +-----																			
♀																			
+-----																			
MODALITES COORDONNEES CONTRIBUTIONS COSINUS CARRES																			
IDEN - LIBELLE P.REL DISTO 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5																			
+-----																			
31 . pertinence de la représentation																			
1	- 92-1 (mauvais)	0.52	11.00	-1.06	-0.70	1.84	-0.11	-0.17	2.1	1.2	9.9	0.0	0.1	0.10	0.04	0.31	0.00	0.00	
2	- 92-2	1.25	4.00	-0.06	0.58	-0.14	0.58	0.54	0.0	2.0	0.1	2.5	2.2	0.00	0.09	0.00	0.08	0.07	
3	- 92-3	2.71	1.31	0.29	0.31	-0.03	-0.20	-0.07	0.8	1.2	0.0	0.6	0.1	0.06	0.07	0.00	0.03	0.00	
4	- 92-4 (excellent)	1.77	2.53	-0.09	-0.68	-0.40	-0.08	-0.23	0.1	3.9	1.6	0.1	0.5	0.00	0.18	0.06	0.00	0.02	
+-----CONTRIBUTION CUMULEE = 2.9 8.3 11.6 3.2 2.9 +-----																			
33 . navigation dans la représentation																			
1	- 94-1 (mauvais)	0.63	9.00	-0.64	-0.77	0.27	0.46	0.08	0.9	1.7	0.3	0.8	0.0	0.05	0.07	0.01	0.02	0.00	
2	- 94-2	1.15	4.45	-0.30	-0.01	0.63	-0.05	0.03	0.4	0.0	2.6	0.0	0.0	0.02	0.00	0.09	0.00	0.00	
3	- 94-3	2.60	1.40	-0.25	0.70	0.01	-0.04	0.09	0.6	6.0	0.0	0.0	0.1	0.05	0.35	0.00	0.00	0.01	
4	- 94-4 (excellent)	1.88	2.33	0.75	-0.71	-0.49	-0.06	-0.17	3.7	4.4	2.5	0.0	0.3	0.24	0.21	0.10	0.00	0.01	
+-----CONTRIBUTION CUMULEE = 5.6 12.1 5.3 0.9 0.5 +-----																			
35 . utilité de la galerie 3D																			
1	- 96-1 (mauvais)	1.25	4.00	-0.85	-0.22	0.93	-0.24	0.06	3.2	0.3	6.1	0.4	0.0	0.18	0.01	0.22	0.01	0.00	
2	- 96-2	0.73	7.57	0.03	0.11	0.18	-0.24	0.99	0.0	0.0	0.1	0.2	4.3	0.00	0.00	0.00	0.01	0.13	
3	- 96-3	1.35	3.62	-0.12	0.83	-0.42	0.43	-0.38	0.1	4.4	1.3	1.5	1.2	0.00	0.19	0.05	0.05	0.04	
4	- 96-4 (excellent)	2.40	1.61	0.47	-0.31	-0.27	-0.33	-0.04	1.9	1.1	1.0	1.5	0.0	0.14	0.06	0.05	0.07	0.00	
5	- 96-NA	0.52	11.00	0.15	-0.35	-0.14	1.31	-0.34	0.0	0.3	0.1	5.2	0.4	0.00	0.01	0.00	0.16	0.01	
+-----CONTRIBUTION CUMULEE = 5.2 6.1 8.6 8.9 5.9 +-----																			
39 . visualisation plus abstraite																			
2	- 102-2	1.88	2.33	-0.05	-0.03	-0.70	-0.24	0.21	0.0	0.0	5.1	0.7	0.5	0.00	0.00	0.21	0.03	0.02	
3	- 102-3	1.15	4.45	-0.09	1.15	0.13	-0.38	-0.33	0.0	7.1	0.1	1.0	0.7	0.00	0.29	0.00	0.03	0.02	
4	- 102-4 (excellent)	2.50	1.50	0.15	-0.24	0.68	-0.01	0.10	0.2	0.7	6.5	0.0	0.1	0.02	0.04	0.31	0.00	0.01	
5	- 102-NA	0.73	7.57	-0.25	-0.92	-0.74	1.26	-0.35	0.2	2.9	2.2	6.7	0.5	0.01	0.11	0.07	0.21	0.02	
+-----CONTRIBUTION CUMULEE = 0.4 10.7 14.0 8.3 1.9 +-----																			

FIG. D.7 – Coordonnées, contributions et qualité de représentation des modalités actives (partie 2)

MODALITES				VALEURS-TEST					COORDONNEES					DISTO.
IDEN	LIBELLE	EFF.	P.ABS	1	2	3	4	5	1	2	3	4	5	
25 . compréhension métaphore														
1	- 82-1 (mauvais)	11	11.00	-2.3	-2.7	-2.3	0.2	1.8	-0.64	-0.75	-0.63	0.05	0.50	4.45
2	- 82-2	11	11.00	-1.8	2.4	-0.7	-1.9	-0.6	-0.50	0.66	-0.19	-0.53	-0.16	4.45
3	- 82-3	21	21.00	-1.1	1.3	0.4	1.9	-2.1	-0.19	0.22	0.07	0.34	-0.37	1.86
4	- 82-4 (excellent)	15	15.00	4.9	-0.6	2.4	-1.6	0.9	1.10	-0.13	0.53	-0.35	0.21	3.00
5	- 82-NA	2	2.00	0.1	-1.2	-0.3	2.5	0.6	0.04	-0.85	-0.23	1.73	0.43	29.00
27 . déplacement dans la ville														
1	- 84-1 (mauvais)	23	23.00	-3.5	-1.7	0.1	-0.7	-1.0	-0.58	-0.29	0.02	-0.11	-0.17	1.61
2	- 84-2	11	11.00	-0.9	2.2	2.7	-0.5	-0.8	-0.23	0.61	0.73	-0.14	-0.23	4.45
3	- 84-3	9	9.00	0.5	2.6	-1.0	-0.1	2.9	0.17	0.79	-0.32	-0.02	0.91	5.67
4	- 84-4 (excellent)	14	14.00	4.8	-1.5	-1.0	0.3	-0.3	1.13	-0.35	-0.24	0.07	-0.08	3.29
5	- 84-NA	3	3.00	-0.9	-1.3	-1.3	2.0	-0.5	-0.49	-0.75	-0.75	1.12	-0.26	19.00
31 . pertinence de la représentation														
1	- 92-1 (mauvais)	5	5.00	-2.4	-1.6	4.3	-0.3	-0.4	-1.06	-0.70	1.84	-0.11	-0.17	11.00
2	- 92-2	12	12.00	-0.2	2.2	-0.5	2.2	2.1	-0.06	0.58	-0.14	0.58	0.54	4.00
3	- 92-3	24	24.00	2.2	2.2	-0.4	-2.2	-0.2	0.35	0.35	-0.07	-0.35	-0.03	1.50
4	- 92-4 (excellent)	15	15.00	-0.6	-3.1	-1.8	-0.9	-0.9	-0.13	-0.69	-0.41	-0.21	-0.21	3.00
5	- 92-NA	4	4.00	-0.2	-0.8	0.2	2.7	-0.9	-0.09	-0.37	0.09	1.29	-0.43	14.00
33 . navigation dans la représentation														
1	- 94-1 (mauvais)	5	5.00	-1.7	-1.6	1.4	0.6	0.5	-0.73	-0.67	0.59	0.26	0.20	11.00
2	- 94-2	9	9.00	-0.9	0.0	2.1	-1.4	0.5	-0.28	0.01	0.66	-0.43	0.14	5.67
3	- 94-3	25	25.00	-1.6	4.5	0.1	-0.3	0.6	-0.25	0.70	0.01	-0.04	0.09	1.40
4	- 94-4 (excellent)	17	17.00	3.7	-3.5	-2.1	-0.8	-0.2	0.76	-0.72	-0.43	-0.17	-0.04	2.53
5	- 94-NA	4	4.00	-0.2	-1.0	-0.9	3.4	-2.0	-0.11	-0.51	-0.43	1.64	-0.98	14.00
35 . utilité de la galerie 3D														
1	- 96-1 (mauvais)	12	12.00	-3.3	-0.8	3.6	-0.9	0.2	-0.85	-0.22	0.93	-0.24	0.06	4.00
2	- 96-2	7	7.00	0.1	0.3	0.5	-0.7	2.8	0.03	0.11	0.18	-0.24	0.99	7.57
3	- 96-3	13	13.00	-0.5	3.4	-1.7	1.7	-1.6	-0.12	0.83	-0.42	0.43	-0.38	3.62
4	- 96-4 (excellent)	23	23.00	2.8	-1.9	-1.6	-2.0	-0.3	0.47	-0.31	-0.27	-0.33	-0.04	1.61
5	- 96-NA	5	5.00	0.3	-0.8	-0.3	3.0	-0.8	0.15	-0.35	-0.14	1.31	-0.34	11.00
39 . visualisation plus abstraite														
1	- 102-1 (mauvais)	2	2.00	1.1	0.6	-1.1	-0.4	1.5	0.77	0.43	-0.75	-0.28	1.02	29.00
2	- 102-2	17	17.00	-0.1	-0.5	-3.3	-1.4	0.7	-0.03	-0.11	-0.68	-0.30	0.15	2.53
3	- 102-3	11	11.00	-0.3	4.2	0.5	-1.4	-1.2	-0.09	1.15	0.13	-0.38	-0.33	4.45
4	- 102-4 (excellent)	23	23.00	0.4	-1.4	4.4	0.3	0.4	0.07	-0.22	0.73	0.04	0.07	1.61
5	- 102-NA	7	7.00	-0.7	-2.6	-2.1	3.5	-1.0	-0.25	-0.92	-0.74	1.26	-0.35	7.57
40 . âge														
A-1	- NA	1	1.00	-0.1	0.0	0.0	0.2	0.4	-0.06	0.05	-0.03	0.21	0.38	59.00
A-2	- 12-16	4	4.00	-0.4	-2.3	-0.5	-0.8	1.0	-0.21	-1.14	-0.26	-0.40	0.48	14.00
A-3	- 20-26	50	50.00	-0.5	1.6	-0.7	0.5	-1.5	-0.03	0.09	-0.04	0.03	-0.09	0.20
A-4	- 31-46	5	5.00	1.1	0.0	1.4	0.0	1.0	0.48	0.00	0.62	0.01	0.43	11.00
41 . sexe														
S-1	- Masculin	47	47.00	0.4	-0.1	-1.7	-0.5	1.9	0.03	-0.01	-0.11	-0.03	0.13	0.28
S-2	- Féminin	13	13.00	-0.4	0.1	1.7	0.5	-1.9	-0.11	0.03	0.41	0.12	-0.46	3.62
42 . jeux 3D														
J-1	- NON	19	19.00	0.1	0.8	1.0	0.3	-1.5	0.02	0.14	0.19	0.06	-0.29	2.16
J-2	- OUI	41	41.00	-0.1	-0.8	-1.0	-0.3	1.5	-0.01	-0.07	-0.09	-0.03	0.13	0.46
43 . temps par semaine sur un navigateur														
N-1	- NA	3	3.00	-0.6	-0.2	-0.2	0.2	0.4	-0.34	-0.09	-0.11	0.14	0.22	19.00
N-2	- moins d'1h	3	3.00	0.1	-0.4	-0.4	1.7	-0.1	0.06	-0.21	-0.24	0.99	-0.08	19.00
N-3	- entre 1 et 4h	25	25.00	0.8	0.1	0.7	-1.1	1.9	0.13	0.02	0.11	-0.17	0.30	1.40
N-4	- entre 4 et 10h	16	16.00	0.0	2.4	0.7	-0.8	-1.9	0.01	0.51	0.14	-0.18	-0.41	2.75
N-5	- plus de 10h	13	13.00	-0.8	-2.4	-1.2	1.1	-0.4	-0.20	-0.59	-0.30	0.28	-0.10	3.62

FIG. D.8 – Coordonnées et valeurs-test des modalités (partie 1)

MODALITES			VALEURS-TEST					COORDONNEES					
IDEN - LIBELLE	EFF.	P.ABS	1	2	3	4	5	1	2	3	4	5	DISTO.
25 . compréhension métaphore													
1 - 82-1 (mauvais)	11	11.00	-2.3	-2.7	-2.3	0.2	1.8	-0.64	-0.75	-0.63	0.05	0.50	4.45
2 - 82-2	11	11.00	-1.8	2.4	-0.7	-1.9	-0.6	-0.50	0.66	-0.19	-0.53	-0.16	4.45
3 - 82-3	21	21.00	-1.1	1.3	0.4	1.9	-2.1	-0.19	0.22	0.07	0.34	-0.37	1.86
4 - 82-4 (excellent)	15	15.00	4.9	-0.6	2.4	-1.6	0.9	1.10	-0.13	0.53	-0.35	0.21	3.00
5 - 82-NA	2	2.00	0.1	-1.2	-0.3	2.5	0.6	0.04	-0.85	-0.23	1.73	0.43	29.00
27 . déplacement dans la ville													
1 - 84-1 (mauvais)	23	23.00	-3.5	-1.7	0.1	-0.7	-1.0	-0.58	-0.29	0.02	-0.11	-0.17	1.61
2 - 84-2	11	11.00	-0.9	2.2	2.7	-0.5	-0.8	-0.23	0.61	0.73	-0.14	-0.23	4.45
3 - 84-3	9	9.00	0.5	2.6	-1.0	-0.1	2.9	0.17	0.79	-0.32	-0.02	0.91	5.67
4 - 84-4 (excellent)	14	14.00	4.8	-1.5	-1.0	0.3	-0.3	1.13	-0.35	-0.24	0.07	-0.08	3.29
5 - 84-NA	3	3.00	-0.9	-1.3	-1.3	2.0	-0.5	-0.49	-0.75	-0.75	1.12	-0.26	19.00
31 . pertinence de la représentation													
1 - 92-1 (mauvais)	5	5.00	-2.4	-1.6	4.3	-0.3	-0.4	-1.06	-0.70	1.84	-0.11	-0.17	11.00
2 - 92-2	12	12.00	-0.2	2.2	-0.5	2.2	2.1	-0.06	0.58	-0.14	0.58	0.54	4.00
3 - 92-3	24	24.00	2.2	2.2	-0.4	-2.2	-0.2	0.35	0.35	-0.07	-0.35	-0.03	1.50
4 - 92-4 (excellent)	15	15.00	-0.6	-3.1	-1.8	-0.9	-0.9	-0.13	-0.69	-0.41	-0.21	-0.21	3.00
5 - 92-NA	4	4.00	-0.2	-0.8	0.2	2.7	-0.9	-0.09	-0.37	0.09	1.29	-0.43	14.00
33 . navigation dans la représentation													
1 - 94-1 (mauvais)	5	5.00	-1.7	-1.6	1.4	0.6	0.5	-0.73	-0.67	0.59	0.26	0.20	11.00
2 - 94-2	9	9.00	-0.9	0.0	2.1	-1.4	0.5	-0.28	0.01	0.66	-0.43	0.14	5.67
3 - 94-3	25	25.00	-1.6	4.5	0.1	-0.3	0.6	-0.25	0.70	0.01	-0.04	0.09	1.40
4 - 94-4 (excellent)	17	17.00	3.7	-3.5	-2.1	-0.8	-0.2	0.76	-0.72	-0.43	-0.17	-0.04	2.53
5 - 94-NA	4	4.00	-0.2	-1.0	-0.9	3.4	-2.0	-0.11	-0.51	-0.43	1.64	-0.98	14.00
35 . utilité de la galerie 3D													
1 - 96-1 (mauvais)	12	12.00	-3.3	-0.8	3.6	-0.9	0.2	-0.85	-0.22	0.93	-0.24	0.06	4.00
2 - 96-2	7	7.00	0.1	0.3	0.5	-0.7	2.8	0.03	0.11	0.18	-0.24	0.99	7.57
3 - 96-3	13	13.00	-0.5	3.4	-1.7	1.7	-1.6	-0.12	0.83	-0.42	0.43	-0.38	3.62
4 - 96-4 (excellent)	23	23.00	2.8	-1.9	-1.6	-2.0	-0.3	0.47	-0.31	-0.27	-0.33	-0.04	1.61
5 - 96-NA	5	5.00	0.3	-0.8	-0.3	3.0	-0.8	0.15	-0.35	-0.14	1.31	-0.34	11.00
39 . visualisation plus abstraite													
1 - 102-1 (mauvais)	2	2.00	1.1	0.6	-1.1	-0.4	1.5	0.77	0.43	-0.75	-0.28	1.02	29.00
2 - 102-2	17	17.00	-0.1	-0.5	-3.3	-1.4	0.7	-0.03	-0.11	-0.68	-0.30	0.15	2.53
3 - 102-3	11	11.00	-0.3	4.2	0.5	-1.4	-1.2	-0.09	1.15	0.13	-0.38	-0.33	4.45
4 - 102-4 (excellent)	23	23.00	0.4	-1.4	4.4	0.3	0.4	0.07	-0.22	0.73	0.04	0.07	1.61
5 - 102-NA	7	7.00	-0.7	-2.6	-2.1	3.5	-1.0	-0.25	-0.92	-0.74	1.26	-0.35	7.57
40 . âge													
A-1 - NA	1	1.00	-0.1	0.0	0.0	0.2	0.4	-0.06	0.05	-0.03	0.21	0.38	59.00
A-2 - 12-16	4	4.00	-0.4	-2.3	-0.5	-0.8	1.0	-0.21	-1.14	-0.26	-0.40	0.48	14.00
A-3 - 20-26	50	50.00	-0.5	1.6	-0.7	0.5	-1.5	-0.03	0.09	-0.04	0.03	-0.09	0.20
A-4 - 31-46	5	5.00	1.1	0.0	1.4	0.0	1.0	0.48	0.00	0.62	0.01	0.43	11.00
41 . sexe													
S-1 - Masculin	47	47.00	0.4	-0.1	-1.7	-0.5	1.9	0.03	-0.01	-0.11	-0.03	0.13	0.28
S-2 - Féminin	13	13.00	-0.4	0.1	1.7	0.5	-1.9	-0.11	0.03	0.41	0.12	-0.46	3.62
42 . jeux 3D													
J-1 - NON	19	19.00	0.1	0.8	1.0	0.3	-1.5	0.02	0.14	0.19	0.06	-0.29	2.16
J-2 - OUI	41	41.00	-0.1	-0.8	-1.0	-0.3	1.5	-0.01	-0.07	-0.09	-0.03	0.13	0.46
43 . temps par semaine sur un navigateur													
N-1 - NA	3	3.00	-0.6	-0.2	-0.2	0.2	0.4	-0.34	-0.09	-0.11	0.14	0.22	19.00
N-2 - moins d'1h	3	3.00	0.1	-0.4	-0.4	1.7	-0.1	0.06	-0.21	-0.24	0.99	-0.08	19.00
N-3 - entre 1 et 4h	25	25.00	0.8	0.1	0.7	-1.1	1.9	0.13	0.02	0.11	-0.17	0.30	1.40
N-4 - entre 4 et 10h	16	16.00	0.0	2.4	0.7	-0.8	-1.9	0.01	0.51	0.14	-0.18	-0.41	2.75
N-5 - plus de 10h	13	13.00	-0.8	-2.4	-1.2	1.1	-0.4	-0.20	-0.59	-0.30	0.28	-0.10	3.62

FIG. D.9 – Coordonnées et valeurs-test des modalités (partie 2)

COORDONNEES, CONTRIBUTIONS ET COSINUS CARRES DES INDIVIDUS
AXES 1 A 5

INDIVIDUS		COORDONNEES					CONTRIBUTIONS					COSINUS CARRES					
IDENTIFICATEUR	P.REL	DISTO	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Ind1	1.67	2.88	-0.36	-0.23	-0.51	-0.54	0.02	0.8	0.4	2.4	2.9	0.0	0.04	0.02	0.09	0.10	0.00
Ind2	1.67	2.41	0.07	-0.08	-0.69	-0.26	0.07	0.0	0.1	4.4	0.7	0.1	0.00	0.00	0.20	0.03	0.00
Ind3	1.67	2.55	-0.09	0.43	0.26	0.08	0.55	0.1	1.5	0.6	0.1	3.0	0.00	0.07	0.03	0.00	0.12
Ind4	1.67	2.25	0.22	-0.33	-0.62	0.10	-0.53	0.3	0.9	3.6	0.1	2.8	0.02	0.05	0.17	0.00	0.13
Ind5	1.67	3.00	-0.50	-0.63	-0.27	0.29	0.16	1.5	3.1	0.7	0.8	0.3	0.08	0.13	0.02	0.03	0.01
Ind6	1.67	3.23	-0.51	0.34	-0.51	-0.12	-0.06	1.6	0.9	2.4	0.1	0.0	0.08	0.04	0.08	0.00	0.00
Ind7	1.67	2.46	-0.23	0.01	0.28	0.18	-0.25	0.3	0.0	0.7	0.3	0.6	0.02	0.00	0.03	0.01	0.03
Ind8	1.67	3.43	0.34	0.01	0.27	0.16	-0.06	0.7	0.0	0.7	0.2	0.0	0.03	0.00	0.02	0.01	0.00
Ind9	1.67	3.04	1.45	-0.48	0.31	-0.13	-0.12	12.4	1.8	0.9	0.2	0.1	0.69	0.07	0.03	0.01	0.00
Ind10	1.67	2.79	1.22	-0.26	-0.09	-0.36	0.17	8.8	0.5	0.1	1.3	0.3	0.53	0.02	0.00	0.05	0.01
Ind11	1.67	2.89	0.64	-0.30	0.00	0.10	0.10	2.4	0.7	0.0	0.1	0.1	0.14	0.03	0.00	0.00	0.00
Ind12	1.67	2.58	-0.26	0.63	-0.46	0.28	0.52	0.4	3.1	2.0	0.7	2.7	0.03	0.15	0.08	0.03	0.10
Ind13	1.67	2.71	0.83	0.33	0.14	-0.36	0.63	4.0	0.9	0.2	1.2	4.0	0.25	0.04	0.01	0.05	0.15
Ind14	1.67	2.57	1.09	-0.24	-0.17	-0.51	0.32	7.0	0.4	0.3	2.5	1.0	0.46	0.02	0.01	0.10	0.04
Ind15	1.67	3.19	-0.03	0.46	-0.39	0.51	-0.23	0.0	1.7	1.4	2.5	0.5	0.00	0.07	0.05	0.08	0.02
Ind16	1.67	2.81	0.77	0.23	-0.40	-0.08	-0.31	3.5	0.4	1.5	0.1	0.9	0.21	0.02	0.06	0.00	0.03
Ind17	1.67	2.46	-0.08	-0.20	0.83	0.08	0.03	0.0	0.3	6.4	0.1	0.0	0.00	0.02	0.28	0.00	0.00
Ind18	1.67	3.18	-0.41	1.04	0.17	0.62	0.04	1.0	8.5	0.3	3.7	0.0	0.05	0.34	0.01	0.12	0.00
Ind19	1.67	3.16	0.05	-0.11	0.56	0.00	-0.56	0.0	0.1	2.9	0.0	3.1	0.00	0.00	0.10	0.00	0.10
Ind20	1.67	2.75	-0.35	-0.30	0.24	-0.18	-0.41	0.7	0.7	0.5	0.3	1.7	0.04	0.03	0.02	0.01	0.06
Ind21	1.67	3.49	-0.19	-0.15	0.23	-0.86	0.31	0.2	0.2	0.5	7.1	1.0	0.01	0.01	0.01	0.21	0.03
Ind22	1.67	2.51	0.38	-0.10	0.51	0.00	-0.68	0.8	0.1	2.5	0.0	4.6	0.06	0.00	0.11	0.00	0.18
Ind23	1.67	2.76	0.84	-0.19	0.06	0.40	-0.31	4.2	0.3	0.0	1.5	0.9	0.26	0.01	0.00	0.06	0.03
Ind24	1.67	3.04	-0.23	0.58	-0.61	-0.37	-0.16	0.3	2.7	3.5	1.3	0.3	0.02	0.11	0.12	0.04	0.01
Ind25	1.67	2.65	-0.26	0.33	-0.37	-0.74	-0.15	0.4	0.9	1.3	5.3	0.2	0.02	0.04	0.05	0.21	0.01
Ind26	1.67	2.52	0.01	0.57	-0.44	-0.23	0.01	0.0	2.5	1.8	0.5	0.0	0.00	0.13	0.08	0.02	0.00
Ind27	1.67	2.38	-0.50	0.07	-0.43	-0.27	0.21	1.5	0.0	1.7	0.7	0.4	0.11	0.00	0.08	0.03	0.02
Ind28	1.67	2.88	-0.17	-0.55	-0.29	-0.09	-0.88	0.2	2.4	0.8	0.1	7.7	0.01	0.10	0.03	0.00	0.27
Ind29	1.67	3.22	0.06	-0.31	-0.03	0.40	0.37	0.0	0.7	0.0	1.5	1.4	0.00	0.03	0.00	0.05	0.04
Ind30	1.67	3.18	-0.68	0.45	0.30	-0.74	-0.49	2.7	1.6	0.8	5.3	2.4	0.14	0.06	0.03	0.17	0.08
Ind31	1.67	2.88	-0.18	-0.25	-0.15	0.37	0.78	0.2	0.5	0.2	1.3	6.0	0.01	0.02	0.01	0.05	0.21
Ind32	1.67	4.28	-0.31	-0.23	0.11	1.34	-0.11	0.6	0.4	0.1	17.3	0.1	0.02	0.01	0.00	0.42	0.00
Ind33	1.67	2.92	0.12	0.91	0.44	-0.01	-0.82	0.1	6.5	1.8	0.0	6.7	0.00	0.29	0.06	0.00	0.23
Ind34	1.67	4.25	-0.12	-0.57	-0.55	0.61	-0.23	0.1	2.6	2.8	3.6	0.5	0.00	0.08	0.07	0.09	0.01
Ind35	1.67	2.71	-0.68	-0.04	-0.57	-0.23	0.42	2.7	0.0	3.0	0.5	1.8	0.17	0.00	0.12	0.02	0.07
Ind36	1.67	3.82	-0.71	-0.17	0.08	0.83	0.24	2.9	0.2	0.1	6.7	0.6	0.13	0.01	0.00	0.18	0.02
Ind37	1.67	3.23	-0.20	0.42	-0.12	0.38	0.78	0.2	1.4	0.1	1.4	6.1	0.01	0.06	0.00	0.05	0.19
Ind38	1.67	2.45	-0.10	0.92	0.14	0.11	0.15	0.1	6.6	0.2	0.1	0.2	0.00	0.35	0.01	0.00	0.01
Ind39	1.67	2.99	-0.53	0.28	-0.01	-0.44	-0.06	1.6	0.6	0.0	1.9	0.0	0.09	0.03	0.00	0.06	0.00
Ind40	1.67	3.06	0.30	-0.25	-0.60	0.73	-0.96	0.5	0.5	3.3	5.2	9.2	0.03	0.02	0.12	0.18	0.30
Ind41	1.67	2.29	0.70	-0.04	0.32	-0.07	0.11	2.9	0.0	1.0	0.1	0.1	0.22	0.00	0.04	0.00	0.01
Ind42	1.67	2.92	-0.02	0.50	-0.21	0.61	-0.41	0.0	1.9	0.4	3.6	1.7	0.00	0.08	0.01	0.13	0.06
Ind43	1.67	2.71	0.06	0.87	0.22	-0.14	-0.20	0.0	6.0	0.4	0.2	0.4	0.00	0.28	0.02	0.01	0.02
Ind44	1.67	2.48	0.77	0.30	0.17	0.10	-0.50	3.5	0.7	0.3	0.1	2.4	0.24	0.04	0.01	0.00	0.10
Ind45	1.67	4.17	-1.07	-0.95	0.65	-0.27	-0.20	6.8	7.1	3.9	0.7	0.4	0.28	0.22	0.10	0.02	0.01
Ind46	1.67	3.31	0.35	-0.55	-0.30	0.10	0.46	0.7	2.4	0.9	0.1	2.1	0.04	0.09	0.03	0.00	0.06
Ind47	1.67	2.23	-0.30	0.32	-0.45	-0.48	-0.38	0.5	0.8	1.9	2.2	1.4	0.04	0.05	0.09	0.10	0.06
Ind48	1.67	2.92	0.97	-0.21	-0.11	-0.12	0.45	5.5	0.3	0.1	0.1	2.1	0.32	0.01	0.00	0.01	0.07
Ind49	1.67	3.07	-0.41	0.28	0.22	-0.62	0.02	1.0	0.6	0.4	3.7	0.0	0.05	0.03	0.02	0.12	0.00
Ind50	1.67	3.31	0.62	0.27	0.53	0.15	0.97	2.2	0.6	2.6	0.2	9.4	0.11	0.02	0.08	0.01	0.28
Ind51	1.67	3.79	-0.80	0.02	0.70	0.58	0.63	3.8	0.0	4.5	3.3	3.9	0.17	0.00	0.13	0.09	0.10
Ind52	1.67	3.12	-0.40	0.20	0.76	0.01	-0.07	1.0	0.3	5.3	0.0	0.1	0.05	0.01	0.18	0.00	0.00
Ind53	1.67	2.81	-0.03	0.02	-0.01	0.09	0.15	0.0	0.0	0.0	0.1	0.2	0.00	0.00	0.00	0.00	0.01
Ind54	1.67	2.87	-0.05	-0.32	-0.19	0.18	0.11	0.0	0.8	0.3	0.3	0.1	0.00	0.03	0.01	0.01	0.00
Ind55	1.67	2.83	-0.10	0.11	0.31	0.04	-0.30	0.1	0.1	0.9	0.0	0.9	0.00	0.00	0.03	0.00	0.03
Ind56	1.67	3.45	0.13	0.19	0.30	-0.23	0.41	0.1	0.3	0.9	0.5	1.7	0.01	0.01	0.03	0.02	0.05
Ind57	1.67	4.08	-0.58	-0.78	1.22	-0.55	-0.14	2.0	4.7	13.9	3.0	0.2	0.08	0.15	0.36	0.07	0.00
Ind58	1.67	3.52	0.51	-0.48	0.25	0.12	-0.06	1.5	1.8	0.6	0.1	0.0	0.07	0.07	0.02	0.00	0.00
Ind59	1.67	2.83	-0.46	-0.57	-0.63	-0.52	0.34	1.3	2.5	3.7	2.6	1.2	0.08	0.11	0.14	0.09	0.04
Ind60	1.67	3.71	-0.63	-1.25	-0.37	-0.02	0.10	2.3	12.2	1.3	0.0	0.1	0.11	0.42	0.04	0.00	0.00

FIG. D.10 – Coordonnées, contributions et qualité de représentation des individus

Caractérisation par les modalités des variables nominales

De la classe : Coupure 'a' de l'arbre en 3 classes - Classe 1 / 3 Effectif = 37 Pourcentage = 61.67

Histogramme des : Valeurs-test

Libellés des variables	Modalités caractéristiques	Valeurs-test	% de la modalité dans la classe	% de la modalité dans l'échantillon	% de la classe dans la modalité	Probabilités	Poids
destiné à tous	65-1 (mauvais)	3,53	48,64	31,66	94,73	0,0002	19,00
entrées textuelles	3	-3,00	32,43	20,00	100,00	0,0013	12,00
compréhension métaphore	82-2	2,80	29,72	18,33	100,00	0,0024	11,00
apprentissage système	61-3	2,46	40,54	28,33	88,23	0,0069	17,00
ajout d'un historique	4 (excellent)	-2,33	27,02	40,00	41,66	0,0098	24,00
essai-erreur	4 (excellent)	-2,66	8,10	20,00	25,00	0,0051	12,00
déplacement dans la ville	84-4 (excellent)	-2,57	10,81	23,33	28,57	0,0050	14,00
navigation dans la métaphore	4 (excellent)	-2,69	5,40	16,66	20,00	0,0047	10,00
temps par semaine sur un navigateur	entre 1 et 4h	-2,65	27,02	41,66	40,00	0,0039	25,00
destiné à tous	65-4 (excellent)	-2,92	13,51	28,33	29,41	0,0017	17,00
utilisation du prototype	35-4 (excellent)	-3,01	2,70	15,00	11,11	0,0012	9,00
visualisation plus abstraite	102-4 (excellent)	-3,11	21,62	38,33	34,78	0,0069	23,00
apprentissage système	61-4 (excellent)	-4,78	5,40	28,33	11,76	0,0000	17,00
compréhension métaphore	82-4 (excellent)	-4,86	2,70	25,00	6,66	0,0000	15,00

Caractérisation par les modalités des variables nominales

De la classe : Coupure 'a' de l'arbre en 3 classes - Classe 2 / 3 Effectif = 10 Pourcentage = 16.67

Histogramme des : Valeurs-test

Libellés des variables	Modalités caractéristiques	Valeurs-test	% de la modalité dans la classe	% de la modalité dans l'échantillon	% de la classe dans la modalité	Probabilités	Poids
pertinence de la représentation	92-1 (mauvais)	2,88	40,00	6,33	80,00	0,0019	5,00
utilité de la galerie 3D	95-1 (mauvais)	2,79	60,00	20,00	60,00	0,0025	12,00
séquence des affichages	44-1 (mauvais)	2,63	50,00	15,00	55,55	0,0042	9,00
lecture des caractères	41-1 (mauvais)	2,61	60,00	21,66	46,15	0,0044	13,00
visualisation plus abstraite	102-4 (excellent)	2,59	80,00	38,33	34,78	0,0047	23,00
terminologie	1 (mauvais)	2,54	40,00	10,00	66,66	0,0053	6,00

Caractérisation par les modalités des variables nominales

De la classe : Coupure 'a' de l'arbre en 3 classes - Classe 3 / 3 Effectif = 13 Pourcentage = 21.67

Histogramme des : Valeurs-test

Libellés des variables	Modalités caractéristiques	Valeurs-test	% de la modalité dans la classe	% de la modalité dans l'échantillon	% de la classe dans la modalité	Probabilités	Poids
compréhension métaphore	82-4 (excellent)	5,01	84,81	25,00	73,33	0,0000	15,00
utilisation du prototype	35-4 (excellent)	4,45	61,53	15,00	88,88	0,0000	9,00
apprentissage système	61-4 (excellent)	3,89	76,92	28,33	58,82	0,0000	17,00
navigation dans la métaphore	4 (excellent)	3,36	53,84	16,66	70,00	0,0003	10,00
navigation dans la représentation	94-4 (excellent)	3,22	69,23	28,33	52,94	0,0006	17,00
déplacement dans la ville	84-4 (excellent)	3,13	61,53	23,33	57,14	0,0008	14,00
vitesse du système	36-4 (excellent)	3,10	53,84	18,33	63,63	0,0009	11,00
entrées textuelles	4 (excellent)	3,04	76,92	36,66	45,45	0,0011	22,00
essai-erreur	4 (excellent)	2,86	53,84	20,00	58,33	0,0020	12,00
affichage contextuel	42-4 (excellent)	2,86	53,84	20,00	58,33	0,0020	12,00
utilité de la suppression	4 (excellent)	2,66	53,84	21,66	53,84	0,0039	13,00
destiné à tous	65-4 (excellent)	2,56	61,53	28,33	47,05	0,0051	17,00
nom et usage	4 (excellent)	2,46	53,84	23,33	50,00	0,0070	14,00
satisfaction / SmartWeb	32-3	2,39	61,53	30,00	44,44	0,0083	18,00
organisation de l'info	43-4 (excellent)	2,36	38,46	13,33	62,50	0,0089	8,00
informe sur l'avancement	4 (excellent)	2,36	38,46	13,33	62,50	0,0089	8,00
déplacement dans la ville	84-1 (mauvais)	-2,36	7,69	38,33	4,34	0,0089	23,00
fiabilité du système	1 (mauvais)	-2,45	0,00	28,33	0,00	0,0070	17,00
destiné à tous	65-1 (mauvais)	-2,70	0,00	31,66	0,00	0,0034	19,00

FIG. D.12 – Caractérisation des trois classes en utilisant les valeurs-test des variables actives et illustratives

Références

- AGARAWALA A. et BALAKRISHNAN R. (2006). Keepin' It Real : Pushing the Desktop Metaphor with Physics, Piles and the Pen. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1283–1292, Montréal, Québec, Canada. ACM Press.
- ANKERST M., KEIM D. A. et KRIEGEL H.-P. (1996). 'Circle Segments' : A Technique for Visually Exploring Large Multidimensional Data Sets. *Proceedings of the IEEE Visualization Conference (VIS), Late Breaking Hot Topics*, San Francisco, CA, USA.
- AZZAG H., DA COSTA D., GUINOT C. et VENTURINI G. (2006). Generating Maps of Web Pages using Cellular Automata. *15th International World Wide Web Conference (WWW)*, Edinburgh, Scotland.
- BAEZ-MONROY V. O. et O'KEEFE S. (2005). Modelling Incremental Learning With The Batch SOM Training Method. *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS)*, pages 542–544, Rio de Janeiro, Brazil. IEEE Computer Society.
- BAEZA-YATES R. A. et RIBEIRO-NETO B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.
- BEAUDOUIN-LAFON M. (2004). Designing Interaction, not Interfaces. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 15–22, Gallipoli, Italy. ACM Press.
- BELL S. J. (2003). Is More Always Better? *American Libraries*, 34(1):44–46. American Library Association.
- BELL S. J. (2004). The Infodiet : How Libraries Can Offer an Appetizing Alternative to Google. *The Chronicle of Higher Education, The Chronicle Review*, 50(24):B15.
- BELL S. J. (2005). Submit or Resist : Librarianship in the Age of Google. *American Libraries*, 36(9):68–71. American Library Association.
- BENFORD S., SNOWDON D., GREENHALGH C., INGRAM R., KNOX I. et BROWN C. (1995). VR-VIBE : A Virtual Environment for Co-operative Information Retrieval. *Computer Graphics Forum (Proceedings of Eurographics'95)*, 14(3):349–360. Blackwell Publishers.

- BENFORD S., TAYLOR I., BRAILSFORD D., KOLEVA B., CRAVEN M., FRASER M., REYNARD G. et GREENHALGH C. (1999). Three Dimensional Visualization of the World Wide Web. *ACM Computing Surveys (CSUR)*, 31(4es):25–40. ACM Press.
- BERTIN J. (1999). *Sémiologie graphique : Les diagrammes - Les réseaux - Les cartes*. Éditions de l'École des Hautes Études en Sciences.
- BLACKMORE J. et MIIKKULAINEN R. (1993). Incremental Grid Growing : Encoding High-Dimensional Structure into a Two-Dimensional Feature Map. *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, volume 1, pages 450–455. IEEE Computer Society.
- BLADH T., CARR D. A. et SCHOLL J. (2004). Extending Tree-Maps to Three Dimensions : A Comparative Study. *Proceedings of the 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI)*, pages 50–59, Rotorua, New Zealand. Springer-Verlag.
- BONNEL N. et CHEVALIER M. (2006a). Critères d'évaluation pour les interfaces des systèmes de recherche d'information. *Actes de la 3^e Conférence en Recherche d'Information et Applications (CORIA)*, pages 109–114, Lyon, France. ARIA.
- BONNEL N. et CHEVALIER M. (2006b). Évaluation des Interfaces Utilisateur d'Information. *Actes du 4^e Atelier Visualisation et Extraction de Connaissances (associé aux 6^e Journées Francophones Extraction et Gestion des Connaissances - EGC'06)*, pages 23–38, Lille, France.
- BONNEL N., COTARMANAC'H A. et MORIN A. (2005a). Gestion et visualisation des résultats d'une requête. *Actes du 3^e Atelier Visualisation et Extraction de Connaissances (associé aux 5^e Journées Francophones Extraction et Gestion des Connaissances - EGC'05)*, pages 37–47, Paris, France.
- BONNEL N., COTARMANAC'H A. et MORIN A. (2005b). Meaning Metaphor for Visualizing Search Results. *Proceedings of the 9th International Conference on Information Visualisation (IV)*, pages 467–472, London, England, UK. IEEE Computer Society.
- BONNEL N., COTARMANAC'H A. et MORIN A. (2005c). Visualisation 3D des résultats de recherche : quel avenir ? *Créer, jouer, échanger : expériences de réseaux - Actes de la 8^e Conférence Hypermedias Hypertexts, Products, Tools and Methods (H²PTM)*, pages 325–339, Paris, France. Hermes Science Publications.
- BONNEL N. et LE GUERNIC G. (2006). Système de recherche de méthodes *Java* basé sur leur signature. *Actes de la 4^e Manifestation des Jeunes Chercheurs Francophones en Sciences et Technologies de l'Information et de la Communication (MajecSTIC)*, page à paraître, Lorient, France.
- BONNEL N., LEMAIRE V., COTARMANAC'H A. et MORIN A. (2006). Effective Organization and Visualization of Web Search Results. *Proceedings of the IASTED Internatio-*

- nal Conference on Internet and Multimedia Systems and Applications (EuroIMSA)*, pages 209–216, Innsbruck, Austria. ACTA Press.
- BONNEL N. et MOREAU F. (2005). Quel avenir pour les moteurs de recherche? *Actes de la 3^e Manifestation des Jeunes Chercheurs Francophones en Sciences et Technologies de l'Information et de la Communication (MajecSTIC)*, pages 291–299, Rennes, France.
- BONNEL N., MORIN A. et COTARMANAC'H A. (2005d). Classification non supervisée et visualisation 3D de documents. *Revue des Nouvelles Technologies de l'Information (RNTI-E-3), Numéro spécial sur les actes des 5^e Journées Francophones Extraction et Gestion des Connaissances (EGC)*, volume 2, pages 557–562, Paris, France. Cépaudès-Éditions.
- BOYACK K. W., WYLIE B. N. et DAVIDSON G. S. (2002a). Domain Visualization Using VxInsight for Science and Technology Management. *Journal of the American Society for Information Science and Technology (JASIST)*, 53(9):764–774. John Wiley & Sons, Inc.
- BOYACK K. W., WYLIE B. N. et DAVIDSON G. S. (2002b). Information Visualization, Human-Computer Interaction, and Cognitive Psychology : Domain Visualizations. Dans BÖRNER K. et CHEN C., éditeurs : *Visual Interfaces to Digital Libraries (JCDL Workshop)*, volume 2539 de *Lecture Notes in Computer Science*, pages 145–160. Springer.
- BRAJNIK G., MIZZARO S. et TASSO C. (1996). Evaluating User Interfaces to Information Retrieval Systems : A Case Study on User Support. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 128–136, Zurich, Switzerland. ACM Press.
- BRIN S. et PAGE L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the 7th International World Wide Web Conference (WWW7)*, pages 107–117, Brisbane, Australia. Elsevier Science Publishers B. V.
- CARD S. K., ROBERTSON G. G. et YORK W. (1996). The WebBook and the Web Forager : An Information Workspace for the World-Wide Web. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 111–117, Vancouver, British Columbia, Canada. ACM Press.
- CHALMERS M. (1993). Using a Landscape Methaphor to Represent a Corpus of Documents. *Proceedings of the International Conference on Spatial Information Theory (CO-SIT)*, pages 377–390, Marciana Marina, Elba Island, Italy. Springer.
- CHERNOFF H. (1973). The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal of the American Statistical Association (JASA)*, 68:361–368. American Statistical Association.
- CHERNOFF H. et RIZVI M. H. (1975). Effect on classification error or random permutations of features in representing multivariate data by faces. *Journal of the American Statistical Association (JASA)*, 70:548–554. American Statistical Association.

- CHEVALIER M. (2002). *Interface adaptative pour l'aide à la recherche d'information sur le web*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, France.
- CHEVALIER M. et HUBERT G. (2005). Évaluation d'une interface de restitution de recherche : Quelles conclusions en tirer ? *Actes du 3^e Atelier Visualisation et Extraction de Connaissances (associé aux 5^e Journées Francophones Extraction et Gestion des Connaissances - EGC'05)*, pages 15–27, Paris, France.
- CLEVELAND W. S. (1993). *Visualizing Data*. Hobart Press.
- COCKBURN A. (2004). Revisiting 2D vs 3D Implications on Spatial Memory. *Proceedings of the 5th Conference on Australasian User Interface (AUIC)*, pages 25–31, Dunedin, New Zealand. Australian Computer Society, Inc.
- COCKBURN A. et MCKENZIE B. (2001). 3D or not 3D? Evaluating the Effect of the Third Dimension in a Document Management System. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI) : Changing our world, changing ourselves*, pages 434–441, Seattle, Washington, USA. ACM Press.
- COCKBURN A. et MCKENZIE B. (2002). Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 203–210, Minneapolis, Minnesota, USA. ACM Press.
- CUGINI J., LASKOWSKI S. et SEBRECHTS M. (2000). Design of 3D Visualization of Search Results : Evolution and Evaluation. *Proceedings of IST/SPIE's International Symposium : Electronic Imaging 2000 : Visual Data Exploration and Analysis*.
- CUGINI J., PIATKO C. et LASKOWSKI S. (1996). Interactive 3D Visualization for Document Retrieval. *Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation, in conjunction with the 5th International Conference on Information and Knowledge Management (CIKM)*, Rockville, Maryland, USA.
- CUTTING D. R., KARGER D. R. et PEDERSEN J. O. (1993). Constant Interaction-Time Scatter/Gather Browsing of Large Document Collections. *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 126–134, Pittsburgh, Pennsylvania, USA. ACM Press.
- CUTTING D. R., KARGER D. R., PEDERSEN J. O. et TUKEY J. W. (1992). Scatter/Gather : A Cluster-based Approach to Browsing Large Document Collections. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 318–329, Copenhagen, Denmark. ACM Press.
- DAKKA W., IPEIROTIS P. G. et WOOD K. R. (2005). Automatic Construction of Multifaceted Browsing Interfaces. *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 768–775, Bremen, Germany. ACM Press.

- DAVIDSON G. S., WYLIE B. N. et BOYACK K. W. (2001). Cluster Stability and the Use of Noise in Interpretation of Clustering. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 23–30, San Diego, California. IEEE Computer Society.
- DIDAY E. (1971). La méthode des nuées dynamiques. *Revue de Statistique Appliquée*, XIX(2):19–34.
- DIEBERGER A. (1994). *Navigation in Textual Virtual Environments using a City Metaphor*. PhD Thesis, Vienna University of Technology, Austria.
- DIEBERGER A. (1997). A City Metaphor to Support Navigation in Complex Information Spaces. *Proceedings of the International Conference on Spatial Information Theory (COSIT)*, pages 53–67, Laurel Highlands, Pennsylvania, USA. Springer-Verlag.
- DIEBERGER A. et FRANK U. (1998). A city metaphor to support navigation in complex information spaces. *Journal of Visual Languages and Computing*, 9:597–622.
- DITTENBACH M., MERKL D. et RAUBER A. (2000a). The Growing Hierarchical Self-Organizing Map. *Proceedings of the International Joint Conference on Neural Networks (IJCNN), Neural Computing : New Challenges and Perspectives for the New Millennium*, volume 6, pages 15–19, Como, Italy. IEEE Computer Society.
- DITTENBACH M., MERKL D. et RAUBER A. (2000b). Using Growing Hierarchical Self-Organizing Maps for Document Classification. *Proceedings of the 8th European Symposium on Artificial Neural Networks (ESANN)*, pages 7–12, Bruges, Belgium. D-Facto publications.
- DITTENBACH M., RAUBER A. et MERKL D. (2001). Recent Advances with the Growing Hierarchical Self-Organizing Map. *Advances in Self-Organising Maps, Proceedings of the 3rd Workshop on Self-Organizing Maps (WSOM)*, pages 140–145, Lincoln, England, UK. Springer.
- DITTENBACH M., RAUBER A. et MERKL D. (2002). Uncovering Hierarchical Structure in Data Using the Growing Hierarchical Self-Organizing Map. *Neurocomputing*, 48(1-4):199–216. Elsevier.
- FEKETE J.-D. (2005). Nouvelle génération d’Interfaces Homme-Machine pour mieux agir et mieux comprendre.
- FEKETE J.-D. et PLAISANT C. (2004). Les leçons tirées de deux compétitions de visualisation d’information. *Actes de la 16^e Conférence Francophone sur l’Interaction Homme-Machine (IHM)*, pages 7–12, Namur, Belgique. ACM Press.
- FISHKIN K. et STONE M. C. (1995). Enhanced Dynamic Queries via Movable Filters. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 415–420, Denver, Colorado, USA. ACM Press/Addison-Wesley Publishing Co.

- FLURY B. et RIEDWYL H. (1981). Graphical Representation of Multivariate Data by Means of Asymmetrical Faces. *Journal of the American Statistical Association (JASA)*, 76(376):757–765. American Statistical Association.
- FRITZKE B. (1995). Growing Grid - a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 2(5):9–13. D Facto Publishing.
- FURNAS G. W. (1986). Generalized Fisheye Views. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 16–23, Boston, Massachusetts, USA. ACM Press.
- FURNAS G. W. et BUJA A. (1994). Prosections Views : Dimensional Inference through Sections and Projections. *Journal of Computational and Graphical Statistics*, 3(4):323–353. American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of America.
- GERIOMENKO V. et CHEN C., éditeurs (2003). *Visualizing the Semantic Web — XML-based Internet and Information Visualization*. Springer.
- GULLI A. et SIGNORINI A. (2005). The Indexable Web is More than 11.5 Billion Pages. *Special Interest Tracks and Posters of the 14th International World Wide Web Conference (WWW)*, pages 902–903, Chiba, Japan. ACM Press.
- HAN J. Y. (2005). Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 115–118, Seattle, WA, USA. ACM Press.
- HASCOËT M. et BEAUDOUIN-LAFON M. (2001). Visualisation interactive d’information. *Revue Information - Interaction - Intelligence (I3)*, 1(1):77–108. Cépaduès-Éditions.
- HASCOËT M., COLLOMB M. et BLANCH R. (2004). Évolution du drag-and-drop : du modèle d’interaction classique aux surfaces multi-supports. *Revue Information, Interaction, Intelligence (I3)*, 4(2):9–38. Cépaduès-Éditions.
- HATANO K., SANO R., DUAN Y. et TANAKA K. (1999). An Interactive Classification of Web Documents by Self-Organizing Maps and Search Engines. *Proceedings of the 6th International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 35–42, Hsinchu, Taiwan. IEEE Computer Society.
- HEALEY C. G. et ENNS J. T. (2002). Perception and Painting : A Search for Effective, Engaging Visualizations. *IEEE Computer Graphics and Applications*, 22(2):10–15. IEEE Computer Society.
- HEARST M. et KARADI C. (1997). Cat-a-Cone : An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 246–255, Philadelphia, PA, USA. ACM Press.

- HEARST M. A. (1995). TileBars : Visualization of Term Distribution Information in Full Text Information Access. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 59–66, Denver, Colorado, USA. ACM Press/Addison-Wesley Publishing Co.
- HEARST M. A. (2006). Clustering versus Faceted Categories for Information Exploration. *Communications of the ACM*, 49(4):59–61. ACM Press.
- HEARST M. A., KARGER D. R. et PEDERSEN J. O. (1995). Scatter/Gather as a Tool for the Navigation of Retrieval Results. Dans BURKE R., éditeur : *Working Notes of the AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, pages 65–71. AAAI Press.
- HEARST M. A. et PEDERSEN J. O. (1996). Reexamining the Cluster Hypothesis : Scatter/Gather on Retrieval Results. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 76–84, Zurich, Switzerland. ACM Press.
- HERMAN I., MELANÇON G. et MARSHALL M. S. (2000). Graph Visualization and Navigation in Information Visualization : A Survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 6(1):24–43. IEEE Computer Society.
- HILL D. R. (1968). A vector clustering technique. *Mechanised Information Storage, Retrieval and Dissemination*. North-Holland.
- HÖLSCHER C. et STRUBE G. (2000). Web Search Behavior of Internet Experts and Newbies. *Computer Networks (The International Journal of Computer and Telecommunications Networking)*, *Proceedings of 9th International World Wide Web Conference (WWW9)*, 33(1-6):337–346, Amsterdam, The Netherlands. North-Holland Publishing Co.
- INSELBERG A. (1998). Visual Data Mining with Parallel Coordinates. *Journal of Computational Statistics*, 13(1):47–63. Springer-Verlag.
- IProspect (2004). iProspect’s Search Engine User Attitudes Survey Results. White paper.
- IProspect (2006). iProspect Search Engine User Behavior Study, White Paper. http://www.iprospect.com/premiumPDFs/WhitePaper_2006_SearchEngineUserBehavior.pdf.
- JARDINE N. et van RIJSBERGEN C. (1971). The Use of Hierarchic Clustering in Information Retrieval. *Information Storage and Retrieval*, 7:217–240.
- KANDOGAN E. (2000). Star Coordinates : A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis), Late Breaking Hot Topics*, pages 4–8, Salt Lake City, Utah, USA. IEEE Computer Society.

- KASKI S. (1998). Dimensionality Reduction by Random Mapping : Fast Similarity Computation for Clustering. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, volume 1, pages 413–418. IEEE Computer Society.
- KEIM D. et KRIEGEL H.-P. (1994). VisDB : Database Exploration Using Multidimensional Visualization. *IEEE Computer Graphics and Applications*, 14(5):40–49. IEEE Computer Society.
- KEIM D. A. (2000). Designing Pixel-Oriented Visualization Techniques : Theory and Applications. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 6(1): 59–78. IEEE Computer Society.
- KEIM D. A., KRIEGEL H.-P. et ANKERST M. (1995). Recursive Pattern : A Technique for Visualizing Very Large Amounts of Data. *Proceedings of the IEEE Visualization Conference (VIS)*, pages 279–286, Atlanta, GA, USA. IEEE Computer Society.
- KEIM D. A., MÜLLER W. et SCHUMANN H. (2002). Visual Data Mining. *State of The Art Report, Eurographics*, pages 215–229, Germany, Saarbrücken.
- KENT A., BERRY M. M., JR. F. U. L. et PERRY J. W. (1955). Operational Criteria for Designing Information Retrieval Systems. *American Documentation*, 6:93–101.
- KOHONEN T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, pages 59–69.
- KOHONEN T. (1995). *Self-Organizing Maps*. Springer.
- KOHONEN T., KASKI S., LAGUS K., SALOJÄRVI J., HONKELA J., PAATERO V. et SAARELA A. (2000). Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585. IEEE Computer Society.
- KOSARA R., HEALEY C. G., INTERRANTE V., LAIDLAW D. H. et WARE C. (2003). User Studies : Why, How, and When? *IEEE Computer Graphics and Applications*, 23(4):20–25. IEEE Computer Society.
- KÄKI M. (2005). *Enhancing Web Search Result Access with Automatic Categorization*. Academic dissertation in dissertations in interactive technology, Department of Computer Sciences, University of Tampere, Finland.
- LAGUS K., KASKI S., HONKELA T. et KOHONEN T. (1996). Browsing Digital Libraries with the Aid of Self-Organizing Maps. *Proceedings of the 5th International World Wide Web Conference (WWW5)*, pages 71–79, Paris, France.
- LAINÉ-CRUZEL S. (1999). ProfilDoc - Filtrer une information exploitable. *Bulletin des bibliothèques de France (BBF)*, 44(5):60–64. ENSSIB.

- LAMIREL J.-C., TOUSSAINT Y., DUCLOY J., CZYSZ C. et FRANÇOIS C. (2001). Réseaux neuronaux avancés pour la cartographie de la science et de la technologie : application à l'analyse des brevets. *Actes du Colloque Veille Stratégique Scientifique et Technologique (VSST)*, volume 2, pages 215–229, Barcelone, Espagne. FPC/UPC - SFBA - IRIT.
- LAMPING J. et RAO R. (1994). Laying out and Visualizing Large Trees Using a Hyperbolic Space. *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 13–14, Marina del Rey, California, USA. ACM Press.
- LAMPING J., RAO R. et PIROLI P. (1995). A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 401–408, Denver, Colorado, USA. ACM Press/Addison-Wesley Publishing Co.
- LAMY J.-B., DUCLOS C., RIALLE V. et VENOT A. (2006). Vers une méthodologie rigoureuse de conception des langages graphiques s'appuyant sur les sciences cognitives. *Actes du 4^e Atelier Visualisation et Extraction de Connaissances (associé aux 6^e Journées Francophones Extraction et Gestion des Connaissances - EGC'06)*, pages 49–54, Lille, France.
- LE GRAND B. et SOTO M. (2002). Visualisation of the Semantic Web : Topic Maps Visualisation. *Proceedings of the 6th International Conference on Information Visualisation (IV)*, pages 344–349, London, England, UK. IEEE Computer Society.
- LEBART L. (2005). Visualization of textual data : unfolding the Kohonen maps. *Proceedings of the XIth International Symposium on Applied Stochastic Models and Data Analysis (ASMDA)*, Brest, France.
- LEBART L., MORINEAU A. et PIRON M. (2000). *Statistique exploratoire multidimensionnelle, 3^e édition*. Dunod.
- LEMAIRE V. et CLÉROT F. (2005). The Many Faces of a Kohonen Map. Dans HALGAMUGE S. K. et WANG L., éditeurs : *Classification and Clustering for Knowledge Discovery*, volume 4 de *Studies in Computational Intelligence*. Springer-Verlag.
- LESOT M.-J., d'Alché BUC F. et SIOLAS G. (2003). Évaluation des cartes auto-organisatrices et de leur variante à noyaux. *Actes de la Conférence d'Apprentissage (CAp)*, pages 139–154, Laval, France. Presses Universitaires de Grenoble.
- LEVKOWITZ H. (1991). Color Icons : Merging Color and Texture Perception for Integrated Visualization of Multiple Parameters. *Proceedings of the 2nd IEEE Conference on Visualization (VIS)*, pages 164–170, San Diego, CA, USA. IEEE Computer Society Press.
- LIN X., SOERGEL D. et MARCHIONINI G. (1991). A Self-Organizing Semantic Map for Information Retrieval. *Proceedings of the 14th Annual International ACM SIGIR*

- Conference on Research and Development in Information Retrieval (SIGIR)*, pages 262–269, Chicago, Illinois, USA. ACM Press.
- LYMAN P. et VARIAN H. R. (2000). How Much Information? <http://www.sims.berkeley.edu/how-much-info>.
- LYMAN P. et VARIAN H. R. (2003). How Much Information? <http://www.sims.berkeley.edu/how-much-info-2003>.
- MACKINLAY J. D., ROBERTSON G. G. et CARD S. K. (1991). The Perspective Wall : Detail and Context Smoothly Integrated. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 173–176, New Orleans, Louisiana, USA. ACM Press.
- MACQUEEN J. B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, California, USA. University of California Press.
- MANN T. M. (1999). Visualization of WWW-Search Results. *Proceedings of the 10th International Workshop on Database & Expert Systems Applications (DEXA)*, pages 264–268, Florence, Italy. IEEE Computer Society.
- MANN T. M. (2002). *Visualization of Search Results from the World Wide Web*. PhD Thesis, University of Konstanz, Germany.
- MERKL D., HE S. H., DITTENBACH M. et RAUBER A. (2003). Adaptive Hierarchical Incremental Grid Growing : An architecture for high-dimensional data visualization. *Proceedings of the 4th Workshop on Self-Organizing Maps (WSOM), Advances in Self-Organizing Maps*, pages 293–298, Kitakyushu, Japon.
- MERKL D. et RAUBER A. (1997). Alternative Ways for Cluster Visualization in Self-Organizing Maps. *Proceedings of the 1st Workshop on Self-Organizing Maps (WSOM)*, pages 106–111, Espoo, Finlande. Helsinki University of Technology, Neural Networks Research Centre.
- MERKL D. et RAUBER A. (1999). Automatic Labeling of Self-Organizing Maps for Information Retrieval. *Proceedings of the 6th International Conference on Neural Information Processing (ICONIP)*, volume 1, pages 37–42, Perth, WA, Australia. IEEE Computer Society Press.
- MIKKULAINEN R. (1990). Script Recognition with Hierarchical Feature Maps. *Connection Science*, 2(1&2):83–101.
- MIZZARO S. (1997). Relevance : The whole history. *Journal of the American Society for Information Science (JASIS)*, 48(9):810–832. John Wiley & Sons, Inc.
- MIZZARO S. (1998). How many relevances in information retrieval? *Interacting with Computers (IWC)*, 10(3):303–320. Elsevier BV.

- MOTHE J., CHRISMENT C., DOUSSET B. et ALAUX J. (2003). DocCube : multi-dimensional visualisation and exploration of large document sets. *Journal of the American Society for Information Science and Technology (JASIST)*, 54(7):650–659. John Wiley & Sons, Inc.
- MUKHERJEA S. et HARA Y. (1999). Visualizing World-Wide Web Search Engine Results. *Proceedings of the IEEE International Conference on Information Visualisation (IV)*, pages 400–405, London, England, UK. IEEE Computer Society.
- NIELSEN J. (2004). When Search Engines Become Answer Engines, Jakob Nielsen's Alertbox. <http://www.useit.com/alertbox/20040816.html>.
- O'REILLY T. (2005). What Is Web 2.0. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1>.
- PAUL G. (1979). *La psychologie de la forme*. Flammarion.
- PICKETT R. et GRINSTEIN G. (1988). Iconographic Displays For Visualizing Multidimensional Data. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, volume 1, pages 514–519, Beijing et Shenyang, Chine. International Academic Publishers.
- PINTE J.-P. (2005). Les outils de la veille pédagogique. *Revue internationale des technologies en pédagogie universitaire*, 2(1):20–29.
- PIROLI P., SCHANK P., HEARST M. et DIEHL C. (1996). Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 213–220, Vancouver, British Columbia, Canada. ACM Press.
- POINÇOT P. (1999). *Classification et recherche d'information bibliographique par l'utilisation des cartes auto-organisatrices, applications en astronomie*. Thèse de Doctorat, Université Louis Pasteur, Strasbourg, France.
- POZNAN (2001). 3-Dimensional Visualization of Search Results. Rapport technique Version 2.11, The Poznan University of Economics, Department of Information Technology.
- RAJMAN M. et LEBART L. (1998). Similarités pour données textuelles. *Actes des 4^e Journées Internationales d'Analyse Statistique des Données Textuelles (JADT)*, pages 545–556, Nice, France.
- RAO R. et CARD S. K. (1994). The Table Lens : Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 318–322, Boston, Massachusetts, USA. ACM Press.

- RAUBER A. (1999). LabelSOM : On the Labeling of Self-Organizing Maps. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 6, pages 3527–3532, Washington, DC, USA. IEEE Computer Society.
- RAUBER A., DITTENBACH M. et MERKL D. (2000). Automatically Detecting and Organizing Documents into Topic Hierarchies : A Neural Network Based Approach to Bookshelf Creation and Arrangement. *Proceedings of the 4th European Conference on Digital Libraries (ECDL)*, pages 348–351, Lisbon, Portugal. Springer.
- RAUBER A. et MERKL D. (1999). Automatic Labeling of Self-Organizing Maps : Making a Treasure-Map Reveal its Secrets. *Proceedings of the 3rd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 228–237, Beijing, China. Springer Verlag.
- RAUBER A. et MERKL D. (2001). Automatic Labeling of Self-Organizing Maps for Information Retrieval. *Journal of Systems Research and Information Systems (JSRIS)*, 10(10):23–45. OPA, Gordon and Breach Science Publishers.
- REKIMOTO J. et GREEN M. (1993). The Information Cube : Using Transparency in 3D Information Visualization. *Proceedings of the 3rd Annual Workshop on Information Technologies & Systems (WITS)*, pages 125–132, Orlando, FL, USA.
- RISDEN K., CZERWINSKI M. P., MUNZNER T. et COOK D. B. (2000). An Initial Examination of Ease of Use for 2D and 3D Information Visualizations of Web Content. *International Journal of Human-Computer Studies (IJHCS)*, 53(5):695–714. Academic Press, Inc.
- ROBERSTON G. G., MACKINLAY J. D. et CARD S. K. (1991). Cone Trees : Animated 3D Visualizations of Hierarchical Information. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 189–194, New Orleans, Louisiana, USA. ACM Press.
- ROBERTSON G., van DANTZICH M., ROBBINS D., CZERWINSKI M., HINCKLEY K., RISDEN K., THIEL D. et GOROKHOVSKY V. (2000). The Task Gallery : A 3D Window Manager. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 494–501, The Hague, The Netherlands. ACM Press.
- ROBERTSON G. G., CZERWINSKI M., LARSON K., ROBBINS D. C., THIEL D. et van DANTZICH M. (1998). Data Mountain : Using Spatial Memory for Document Management. *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 153–162, San Francisco, California, USA. ACM Press.
- ROCCHIO J. J. (1966). *Document Retrieval Systems - Optimization and Evaluation*. PhD Thesis, Harvard University, MA, USA.
- ROSSI A. M. et VARGA M. (1999). Visualization of Massive Retrieved Newsfeeds in Interactive 3D. *Proceedings of the International Conference on Information Visualisation (IV)*, pages 12–17, London, England, UK. IEEE Computer Society.

- SALTON G. et BUCKLEY C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523. Pergamon Press, Inc.
- SALTON G., WONG A. et YANG C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18:613–620. ACM Press.
- SAMMON Jr J. W. (1969). A Non-Linear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, C-18(5):401–409.
- SARKAR M. et BROWN M. H. (1994). Graphical Fisheye Views. *Communications of the ACM*, 37(12):73–83. ACM Press.
- SCOTT D. W. (1992). *Multivariate density estimation : Theory, practice, and visualization*. John, Wiley & Sons, New York, NY, USA.
- SEBRECHTS M. M., CUGINI J. V., LASKOWSKI S. J., VASILAKIS J. et MILLER M. S. (1999). Visualization of Search Results : A Comparative Evaluation of Text, 2D, and 3D Interfaces. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 3–10, Berkeley, California, USA. ACM Press.
- SHNEIDERMAN B. (1992). Tree Visualization with Tree-Maps : 2-d Space-Filling Approach. *ACM Transactions on Graphics (TOG)*, 11(1):92–99. ACM Press.
- SHNEIDERMAN B. (1996). The Eyes Have It : A Task by Data Type Taxonomy for Information Visualizations. *Proceedings of the IEEE Symposium on Visual Languages (VL)*, pages 336–343, Boulder, CO, USA. IEEE Computer Society.
- SHNEIDERMAN B. (1998). *Designing the User Interface : Strategies for Effective Human-Computer Interaction*, 3^e édition. Addison-Wesley Longman Publishing Co., Inc.
- SHNEIDERMAN B. (2003a). Promoting Universal Usability with Multi-Layer Interface Design. *Proceedings of the 2nd ACM Conference on Universal Usability (CUU)*, pages 1–8, Vancouver, British Columbia, Canada. ACM Press.
- SHNEIDERMAN B. (2003b). Why Not Make Interfaces Better than 3D Reality? *IEEE Computer Graphics and Applications*, 23(6):12–15. IEEE Computer Society.
- SHNEIDERMAN B. (2004). Designing for Fun : How Can We Design User Interfaces to Be More Fun? *Interactions*, 11(5):48–50. ACM Press.
- SKUPIN A. (2000). From Metaphor to Method : Cartographic Perspectives on Information Visualization. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 91–97. IEEE Computer Society.
- SKUPIN A. (2002). A Cartographic Approach to Visualizing Conference Abstracts. *IEEE Computer Graphics and Applications*, 22(1):50–58. IEEE Computer Society.

- SPARACINO F., DAVENPORT G. et PENTLAND A. (2000). City of News : cataloguing the World Wide Web through Virtual Architecture. *KOS*, (179-180).
- SPARACINO F., WREN C., AZARBAYEJANI A. et PENTLAND A. (2002). Browsing 3-D spaces with 3-D vision : body-driven navigation through the Internet city. *Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT)*, pages 224–233, Padova, Italy. IEEE Computer Society.
- SPÄRCK JONES K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.
- STASKO J. et ZHANG E. (2000). Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 57–65, Salt Lake City, Utah, USA. IEEE Computer Society.
- STOICA E. et HEARST M. A. (2004). Nearly-Automated Metadata Hierarchy Creation. *Companion Proceedings of Human Language Technology Conference / North American chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL)*, pages 117–120, Boston, Massachusetts, USA. Association for Computational Linguistics.
- SUTCLIFFE A. G. et PATEL U. (1996). 3D or Not 3D : Is It Nobler in the Mind? *People and Computers XI : Proceedings of the British Human-Computer Interaction Conference (HCI)*, pages 79–94. Springer.
- TURNER D. W. et TIDMORE F. E. (1980). FACES-A FORTRAN Program for Generating Chernoff-Type Faces on a Line Printer. *The American Statistician*, 34(3):187. American Statistical Association.
- van RIJSBERGEN C. (1979). *Information Retrieval*, 2nd edition. Butterworths.
- van WIJK J. J. et van de WETERING H. (1999). Cushion Treemaps : Visualization of Hierarchical Information. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 73–78, San Francisco, California, USA. IEEE Computer Society.
- van WIJK J. J. et van LIERE R. (1993). HyperSlice : Visualization of Scalar Functions of Many Variables. *Proceedings of the 4th IEEE Visualization Conference (VIS)*, pages 119–125, San Jose, California, USA. IEEE Computer Society.
- VEGAS J., de la FUENTE P. et CRESTANI F. (2003). WebDocBall : A Graphical Visualization Tool for Web Search Results. *Proceedings of the 25th European Conference on Information Retrieval (ECIR)*, pages 351–362, Pisa, Italy. Springer.
- VESANTO J. (2002). *Data Exploration Process Based on the Self-Organizing Map*. PhD Thesis, Helsinki University of Technology, Finland.

- VOORHEES E. M. (1985). The Cluster Hypothesis Revisited. *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 188–196, Montreal, Quebec, Canada. ACM Press.
- WIZA W., WALCZAK K. et CELLARY W. (2004). Periscope - A System for Adaptive 3D Visualization of Search Results. *Proceedings of the 9th International Conference on 3D Web Technology (Web3D Symposium)*, pages 29–40, Monterey, California, USA. ACM Press.
- YATES F. A. (1966). *The Art of Memory*. University Of Chicago Press.
- YEE K.-P., SWEARINGEN K., LI K. et HEARST M. (2003). Faceted Metadata for Image Search and Browsing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 401–408, Ft. Lauderdale, Florida, USA. ACM Press.
- ZAMIR O. (1998). Visualization of Search Results in Document Retrieval Systems. General Examination Report. University of Washington, USA.
- ZAMIR O. et ETZIONI O. (1998). Web Document Clustering : A Feasibility Demonstration. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 46–54, Melbourne, Australia. ACM Press.
- ZAMIR O. et ETZIONI O. (1999). Grouper : A Dynamic Clustering Interface to Web Search Results. *Computer Networks, Proceeding of the 8th International World Wide Web Conference (WWW8)*, 31(11-16):1361–1374, Toronto, Canada. Elsevier North-Holland, Inc.

Liste des figures

1	Aperçu des contributions	7
2	L'homme face à l'infobésité	13
1.1	Description simplifiée du fonctionnement d'un moteur de recherche	20
1.2	Taxonomie des différentes approches de visualisation des résultats d'une recherche	26
1.3	Approche utilisant plusieurs visualisations simples alternatives	29
1.4	Un extrait de la liste de résultats proposée par le moteur de recherche <i>Google</i>	30
1.5	Listes de résultats proposées par le moteur de recherche <i>Clusty</i>	31
1.6	Liste de résultats proposée par le moteur de recherche <i>Ujiko</i>	33
1.7	Exemples de visualisations basées sur la distribution des termes de la requête	36
1.8	<i>KartOO</i> , méta-moteur de recherche cartographique en 2D	37
1.9	<i>MapStan Search</i> , méta-moteur de recherche cartographique en 2D	39
1.10	Métaphores sous forme de terrain	42
1.11	<i>Cat-a-Cone</i>	44
1.12	<i>Grokker</i>	45
1.13	Taxonomie des métaphores 3D de visualisation pouvant être utilisées par les IUI	45
1.14	Bureaux virtuels en 3D	48
1.15	<i>SphereXPlorer</i> : navigateur Web 3D	48
1.16	<i>Hyperbolic Browser</i> : illustration d'un changement de <i>focus</i>	54
1.17	Nouvelles interfaces, nouvelles applications, nouvelles interactions	56
2.1	Description simplifiée du fonctionnement des cartes auto-organisatrices	66
3.1	Vue de dessus de la métaphore de la ville	88
3.2	Vue panoramique de la métaphore de la ville	88
3.3	Métaphore de la ville	89
3.4	Visualisation 2D des images contenues dans un site Web	92
3.5	Visualisation 2D des images contenues dans un site Web	92
3.6	Visualisation 3D des images contenues dans un site Web	93
3.7	Différents niveaux de la métaphore de la ville	95

3.8	Visualisation des résultats de recherche avec la métaphore de la galerie .	96
4.1	Schéma conceptuel du prototype	102
4.2	Architecture de l'application	103
4.3	Schéma d'implémentation du prototype	104
4.4	Écran d'accueil du prototype	111
4.5	Métaphore de la ville	111
4.6	Représentation, dans la ville virtuelle, de la page Web sélectionnée afin de visualiser les images qu'elle contient	114
4.7	Représentations 2D des images d'un site Web	115
4.8	Représentation, sous la forme d'une galerie, des images d'un site Web .	116
5.1	Questionnaire utilisateur : profil	123
5.2	Questionnaire utilisateur : évaluation du prototype	124
5.3	Âge	128
5.4	Sexe	128
5.5	Temps passé à travailler sur SmartWeb	129
5.6	Temps moyen passé par semaine sur SmartWeb	129
5.7	Nombre de systèmes d'exploitation déjà utilisés	130
5.8	Appareils déjà utilisés et avec lesquels l'utilisateur est familier	130
5.9	Logiciels déjà utilisés et avec lesquels l'utilisateur est familier	131
5.10	Systèmes déjà utilisés et avec lesquels l'utilisateur est familier	131
5.11	Moteur de recherche principalement utilisé par les utilisateurs	131
5.12	Avantage et inconvénient majeurs du moteur de recherche utilisé	132
5.13	Catégorie de sites Web préférée par les utilisateurs	132
5.14	Nombre de <i>bookmarks</i> utilisés par les utilisateurs	133
5.15	Profondeur de la hiérarchie de <i>bookmarks</i>	133
5.16	Temps passé en moyenne par semaine sur un navigateur Web	134
5.17	Réaction générale à SmartWeb	137
5.18	Affichage	138
5.19	Terminologie et informations du système	139
5.20	Apprentissage	139
5.21	Aide en ligne	140
5.22	Métaphore de la ville	140
5.23	Représentation des images	141
5.24	Idées	142
5.25	Plan factoriel 1×2 (variables 3.2, 3.3, 3.5 et 3.6)	145
5.26	Plan factoriel 1×2 (variables 4.1, 4.2 et 4.4)	146
5.27	Plan factoriel 1×2 (variables 6.1, 6.5 et 10.2)	146
5.28	Plan factoriel 1×2 (variables 8.1, 8.2 et 8.4)	147
5.29	Plan factoriel 1×2 (variables 9.2, 9.4 et 9.6)	147
5.30	Plan factoriel 1×2 (variables illustratives)	148
A.1	<i>Scatterplot matrix</i>	173

A.2	Coordonnées parallèles	174
A.3	Coordonnées en étoile	174
A.4	<i>Chernoff Faces</i>	175
A.5	<i>Stick Figures</i>	176
A.6	<i>Color Icons</i>	176
A.7	<i>Today's front pages</i>	177
A.8	Exemple de motif récursif	178
A.9	Exemple d'arrangement selon la technique des segments de cercle	179
A.10	<i>Treemap</i>	180
A.11	<i>Newsmap</i>	180
A.12	<i>SunBurst</i>	181
A.13	<i>Cone Tree</i>	182
A.14	<i>InfoCube</i>	183
A.15	<i>StepTree</i>	183
D.1	Sélection des individus et des variables utiles	197
D.2	Apurement des modalités actives (partie 1)	198
D.3	Apurement des modalités actives (partie 2)	199
D.4	Histogramme des valeurs propres	200
D.5	Valeurs propres — recherche de paliers	201
D.6	Coordonnées, contributions et qualité de représentation des modalités actives (partie 1)	202
D.7	Coordonnées, contributions et qualité de représentation des modalités actives (partie 2)	203
D.8	Coordonnées et valeurs-test des modalités (partie 1)	204
D.9	Coordonnées et valeurs-test des modalités (partie 2)	205
D.10	Coordonnées, contributions et qualité de représentation des individus	206
D.11	Plan factoriel 1×2 avec trois classes d'individus	207
D.12	Caractérisation des trois classes en utilisant les valeurs-test	208

Liste des tableaux

1.1	Propriétés du modèle d'information	26
3.1	Propriétés de la métaphore	87
3.2	Temps de calcul de la ville 3D	89
3.3	Propriétés de la visualisation 2D	91
5.1	Résultats du test utilisateur : définition du profil des utilisateurs	128
5.2	Résultats du test utilisateur : évaluation du prototype	136
5.3	Variables actives pour l'ACM	143
5.4	Distribution des votes sur les 16 questions sélectionnées	144
C.1	Table QUERIES	188
C.2	Table QUERIES_RESULTS	188
C.3	Table DOCUMENTS	188
C.4	Table DOCUMENTS_CONTENTS	189
C.5	Table WORDS	189
C.6	Table IMAGES	189
C.7	Table IMAGES_OF_DOCUMENTS	190

Liste des algorithmes

1	<i>stochastic SOM</i>	67
2	<i>batch SOM</i>	67
3	<i>k-means</i>	68
4	Algorithme <i>batch SOM</i> avec pondération des données.	80

Index

- Bruit, 23
- Cartes auto-organisatrices, 62
- Cat-a-Cone*, 42
- Catégorisation, 14
- CityOfNews*, 39
- Clustering*, 14
- Color Icons*, 164
- Cone Tree*, 170
- Coordonnées en étoile, 162
- Coordonnées parallèles, 162
- corpus, 10
- Data Mountain*, 40
- Easy-DoR*, 34
- F-mesure, 23
- Filtrage interactif, 51
- Fisheye View*, 52
- Focus+Context*, 51
- Google, 11
- Grokker*, 43
- IHM, 14
- Infobésité, 11
- InfoCube*, 171
- Interaction, 50
- Interfaces Homme-Machine, 14
- JMBrowser*, 54
- KartOO*, 35
- métaphore, 10
- MapStan Search*, 35
- Moteurs de recherche, 12
- Motif récursif, 166
- Newsmap*, 168
- Précision, 23
- Rappel, 23
- Scatterplots matrix*, 161
- Segments de cercle, 167
- Silence, 23
- SmartWeb, 91
- SphereXP*, 45
- SphereXPlorer*, 46
- SRI, 14
- StepTree*, 171
- Stick Figures*, 164
- SunBurst*, 169
- Système de recherche d'information, 14
- TaskGallery*, 45
- ThemeView*, 39
- Three-keyword axes display*, 34
- Tilebars*, 33
- Today's Front Pages*, 165
- Treemap*, 168
- ViOS, 40
- Visages de *Chernoff*, 163
- vocabulaire, 10
- VR-VIBE*, 34
- VxInsight*, 40
- Web 2.0, 155
- Webdocball*, 34
- WEBSOM*, 38
- X-VRML, 96
- Zoom, 51

Résumé

La recherche d'information textuelle fait partie des principales tâches liées au Web. Elle se fait majoritairement par des moteurs de recherche qui sont rapidement devenus incontournables. En effet, lorsque les utilisateurs ont une nouvelle tâche à accomplir sur le Web, ils démarrent — 88% du temps — par l'utilisation d'un moteur de recherche. Cependant, face à l'augmentation des informations disponibles sur le Web et à l'absence d'évolution significative du processus de recherche, la quantité de résultats obtenus pour une requête devient très importante. Il est alors difficile pour l'utilisateur d'interpréter efficacement tous ces résultats. Cette problématique est abordée en se plaçant du côté de la restitution des résultats de recherche à l'utilisateur *via* des interfaces utilisateur d'information (IUI). De nombreux travaux ont déjà été réalisés sur la visualisation des résultats de recherche au cours des dernières années, sans réelle influence sur les interfaces grand public. L'objectif de notre approche est de créer dynamiquement des présentations 3D interactives et basées sur des métaphores de visualisation adaptées à la fois à l'utilisateur, à la tâche à réaliser et aux données. Pour cela, nous avons développé un prototype — *SmartWeb* — proposant des interfaces hybrides (2D/3D) de visualisation de résultats de recherche. Il utilise le langage X-VRML qui permet d'exprimer efficacement les métaphores 3D de visualisation et de générer automatiquement des contenus 3D interactifs. Nous proposons alors une métaphore 3D cognitive permettant d'utiliser une représentation spatiale plus riche afin d'aider efficacement l'utilisateur dans sa tâche. Il s'agit d'une ville virtuelle dans laquelle les résultats sont organisés d'après le calcul d'une carte auto-organisatrice. Une étude utilisateur de cette interface est réalisée et une réflexion plus générale sur l'évaluation des IUI est proposée.

Mots-clés : métaphores 3D, visualisation de résultats de recherche, cartes auto-organisatrices, évaluation d'interfaces, recherche d'information, interfaces Homme-Machine, catégorisation visuelle de pages Web.

Abstract

Textual information retrieval is one of the main tasks related to the Web. This task relies generally on search engines which have become an essential tool of the Web. Indeed users start at a search engine 88% of the time when they have a new task to complete on the Web. However, confronted with the huge increase of available information on the Web and the lack of significant evolution of the search process, the amount of documents matching a query becomes awfully important. It's therefore difficult for the user to effectively interpret all these results. This problem of representing search result is addressed through information user interfaces (IUI). Much work has been carried out on search result visualization since a decade, without real impact on the most popular user interfaces. Our approach focuses on the necessity to dynamically create interactive 3D presentations based on visualization metaphors adapted not only to the end-user, but also to the task to complete as well as to the data. We have thus developed the *SmartWeb* prototype that proposes hybrid interfaces (2D/3D) for the visualization of search results. It uses the X-VRML language that enables to effectively design 3D visualization metaphors and automatically generate interactive 3D content. We propose a cognitive 3D metaphor allowing for a richer space representation which efficiently and effectively helps users in their tasks. A user study of this interface was carried out and an evaluation framework for IUI is proposed.

Keywords : 3D Metaphors, Search Result Visualization, Self-Organizing Maps, Interface Evaluation, Information Retrieval, Human-Computer Interfaces, Visual Categorization of Web Pages.