

VERNIER Frédéric
Maîtrise d'Informatique & Magistère
U.F.R. I.M.A. de l'Université de Grenoble
E-mail : Frederic.Vernier@imag.fr

1995-1996

Rapport de
Magistère d'Informatique
2^{ème} année

*Etude et mise en œuvre de
techniques d'interaction pour les moteurs de
recherche du World Wide Web*



Laboratoire CLIPS,
Communication Langagière et Interaction Personne-Système

Equipe Ingénierie de l'Interaction Homme-Machine

Responsable de stage : **Laurence Nigay**

Table des matières

1. Introduction	4
1.1. Le sujet du stage et son domaine.....	4
1.2. Magistère d'informatique.....	4
1.3. Laboratoire CLIPS	4
1.4. Système interactif	5
1.5. Sujet de l'étude et du stage.....	6
1.6. Motivations	7
2. Etude bibliographique	9
2.1. Démarche de travail	9
2.2. Etude des problèmes de l'utilisateur	9
2.2.1. Le temps de réponse	9
2.2.2. Le manque de vue globale de l'espace informationnel.....	10
2.2.3. L'excès de réponses	10
2.2.4. La perte du contexte de la recherche	10
2.3. Sélection de critères d'ergonomie	10
2.3.1. Critères d'ergonomie	10
2.3.1.1. Critères de souplesse	11
2.3.1.2. Critères de robustesse.....	12
2.3.2. Les critères de robustesse pertinents.....	14
2.3.3. Les critères de souplesse pertinents	14
2.4. Les espaces informationnelles de grande taille	15
2.4.1. La problématique des mines d'information trop importantes.....	15
2.4.2. Différentes techniques d'interaction.....	15
2.4.2.1. Lentille magique.....	15
2.4.2.2. Système « DragMad»	16
2.4.2.3. Mur en perspective.....	17
2.4.2.4. Vue en œil de poisson	17
2.4.2.5. Système de « zoom ».....	18
2.4.2.6. Arbre conique	19
2.5. Etude des techniques d'interaction répondant aux critères	19
2.5.1 Exemples de représentation bien adaptée.....	19
2.5.2. Application des techniques de représentation au WWW.....	20
2.5.2.1. Etude des cas existants	20
2.5.2.2. Particularités du WWW.....	20
2.5.3. Une solution de technique d'interaction.....	21
2.6. Conclusion.....	22
3. Réalisation logicielle	23
3.1. Mise en œuvre des résultats de mon étude.....	23
3.2. Démarche de travail	23
3.2.1. Les phases de conception de l'interface	23
3.2.1.1. Identification de l'utilisateur.....	23
3.2.1.2. Définition des tâches et des objets spécifiques au domaine	23
3.2.1.3. Définition des objets et fonctions informatiques	23

3.2.2. Les phases de conception logicielle de l'interface.....	24
3.2.3. Plan : les résultats de chaque étape de conception.....	24
3.3. Les critères de qualité	25
3.3.1. Les critères de génie logiciel.....	25
3.3.2. Les critères d'ergonomie.....	25
3.4. Les spécifications externes du système	26
3.4.1. Les techniques d'interaction.....	26
3.4.2. L'interface	26
3.4.2.1. L'interface à réaliser.....	26
3.4.2.2. L'interface réalisée	28
3.5. Réalisation logicielle : spécification globale	31
3.5.1. Les composants du modèle d'architecture PAC-Amodeus	31
3.5.2. Architecture logicielle de mon interface selon PAC-Amodeus.....	32
3.6. Réalisation logicielle : spécification détaillée.....	34
3.6.1. La partie client	34
3.6.1.1. Les outils mathématiques pour les vues oeil de poisson.....	34
3.6.1.2. Les communications en JAVA.....	35
3.6.1.3. Les modules mis en œuvre.....	35
3.6.2. La partie serveur	36
3.6.2.1. Les communications sur le réseau.....	36
3.6.2.2. Le traitement des fichiers ASCII.....	36
3.6.2.3. Les modules mis en œuvre.....	37
3.6.3. Le couple client-serveur	37
3.6.3.1. Avantages	38
3.6.3.2. Inconvénients	38
4. Conclusion et perspectives	39
4.1. Bilan du travail réalisé.....	39
4.1.1. Les méthodes et modèles appliqués	39
4.1.2. Les vues en oeil de poisson	39
4.1.3. L'interface	40
4.1.4. L'utilisabilité générale	40
4.2. Perspectives.....	40

Chapitre 1

Introduction

1.1. Le sujet du stage et son domaine

Mon stage de Magistère en informatique s'effectue au sein du laboratoire CLIPS (Communication Langagière et Interaction Personne-Système) dans l'équipe Ingénierie de l'Interaction Homme-Machine. Ce stage consiste en une étude sur les techniques d'interaction et une réalisation logicielle d'un système interactif. Les résultats théoriques de mon travail d'étude ont été exploités comme fondements à la conception du système interactif. Dans ce chapitre, je présente le magistère d'informatique et le laboratoire CLIPS, puis je définis ce qu'est un système interactif. Je pose ensuite le sujet de mon travail et enfin j'expose mes motivations face à ce sujet.

1.2. Magistère d'informatique

Le magistère d'informatique de Grenoble est un diplôme de l'université Joseph Fourier permettant aux étudiants de se familiariser avec l'environnement de la recherche. J'ai donc effectué, en plus des cours de Maîtrise, un stage en laboratoire de recherche durant l'année scolaire afin d'y mener une recherche bibliographique sur les techniques d'interaction. De plus j'ai effectué le stage de fin d'année dans le même laboratoire afin de réaliser une partie des propositions qui concluaient l'étude bibliographique. Les étudiants de Grenoble ont la chance de côtoyer un grand nombre de chercheurs, de laboratoires et donc de thèmes de recherche sur le campus. Parmi les thèmes qui m'intéressent plus particulièrement figure l'interface graphique des logiciels. J'ai donc contacté l'équipe d'Interface Homme-Machine (IHM) qui s'occupe entre autre d'étudier les interfaces. Laurence Nigay m'a proposé un thème de recherche sur les techniques d'interaction que j'ai accepté.

1.3. Laboratoire CLIPS

Le laboratoire "Communication Langagière et Interaction Personne Système" (CLIPS) est un laboratoire de recherche de l'IMAG créé en Septembre 1995. Trois thèmes de recherche complémentaires ont suscité sa création :

- 1- la langue comme mode de communication dans le dialogue homme-machine et la traduction automatique.
- 2- les systèmes d'interaction (interfaces multimodales, réalités virtuelles, télé-présence, etc.) pour des usages finalisés (création artistique, conception assistée, etc.).
- 3- les systèmes multimédias (systèmes à base de connaissance ou les systèmes d'information utilisant la langue naturelle etc.) fournissant les modèles et les outils de base.

L'équipe IIHM s'inscrit dans le thème intitulé "système d'interaction". Comme son nom l'indique, le domaine de recherche de cette équipe est l'Interaction Homme-Machine (IHM) que je présente succinctement au paragraphe suivant.

1.4. Système interactif

L'Interaction Homme-Machine (IHM) a pour objet l'étude de la mise en œuvre de systèmes interactifs. L'objectif est de produire des systèmes faciles à utiliser. L' "utilisabilité" d'un système est une propriété difficile à vérifier. Les raisons à cela sont multiples :

- L'attitude de l'utilisateur change. A l'usager inquiet et soumis aux « ordres de la machine », succède un consommateur mieux averti qui veut voir en l'ordinateur un compagnon de travail productif.
- L'innovation technologique accélère l'allure. Elle se manifeste par le progrès des performances des ordinateurs. Elle se constate aussi par l'apparition de nouveaux modes de communication comme la parole ou le geste (capté par une caméra ou un gant numérique).

Les systèmes interactifs doivent répondre à cette évolution par un double effort d'adaptation : satisfaire aux exigences grandissantes des utilisateurs et tenir compte d'une technologie hautement évolutive.

Comme le montre la figure 1, un système interactif, objet du domaine de recherche de l'IHM, comprend le système informatique et l'utilisateur.

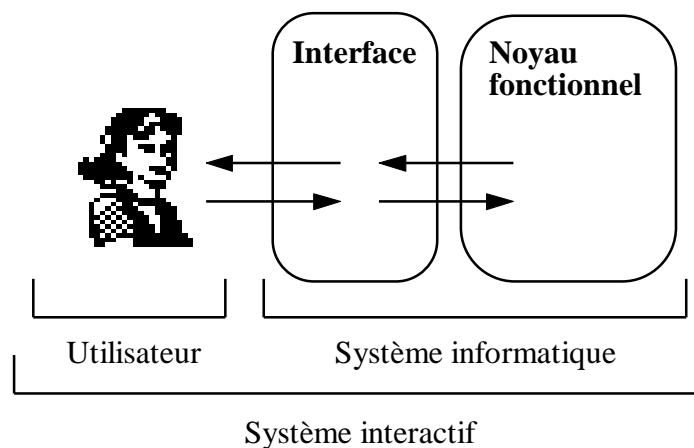


Figure 1 : Un système interactif

Le système informatique peut être divisé en deux composants :

- L'interface :

L'interface correspond à la partie du système informatique qui définit la perception du noyau fonctionnel par l'utilisateur. Elle gère les entrées spécifiées par l'utilisateur selon une modalité d'interaction mais aussi les sorties du système.

(Si l'interface offre la possibilité à plusieurs utilisateurs de travailler ensemble, le système interactif est alors appelé "collecticiel".)

- Le noyau fonctionnel :

Le noyau fonctionnel définit et maintient tous les concepts propres au domaine applicatif (Base de donnée, Algorithme mathématique, etc.)

La distinction entre l'interface et son noyau fonctionnel est un principe de base, admis de tous, pour la conception logicielle d'un système interactif : le logiciel qui réalise le noyau fonctionnel doit être clairement distingué de celui de l'interface.

Cependant il n'est pas souhaitable que la conception d'un système interactif se fasse selon ce principe. L'approche qui consiste à considérer l'interface comme un frontal d'un noyau fonctionnel n'est pas appropriée, et les deux parties doivent être conçues en symbiose. Typiquement la conception de l'interface impose des contraintes sur la conception du noyau fonctionnel et vice-versa. Néanmoins il est envisageable de développer une interface pour un noyau fonctionnel déjà mis en oeuvre : la situation n'est certes pas idéal. Une partie de l'interface est alors dédiée à adapter le noyau fonctionnel aux besoins de l'utilisateur. Dans le cadre de mon stage, le noyau fonctionnel du système interactif à développer est déjà mis en oeuvre : la situation est donc particulière. Le sujet de mon stage fait l'objet du paragraphe suivant.

1.5. Sujet de l'étude et du stage

- L'étude porte sur les techniques d'interaction en général et leurs applications possibles aux moteurs de recherche. Les techniques d'interaction et de visualisation font l'objet d'études approfondies. Ces techniques permettent de visualiser et d'interagir avec un grand nombre d'informations. Afin d'appliquer ces techniques au Web, il convient d'abord d'étudier les différentes techniques qui existent puis de choisir et justifier celle qui convient le mieux aux moteurs de recherche.
- La réalisation logicielle consiste à concevoir et développer un système interactif englobant un moteur de recherche du World Wide Web (WWW) qui puisse facilement être utilisé par tous. La réalisation porte essentiellement sur l'interface du moteur de recherche. Une contrainte du sujet est l'indépendance du code produit par rapport à celui du noyau fonctionnel existant, le moteur de recherche. Aussi le sujet consiste à développer plusieurs systèmes interactifs : une même interface avec des noyaux fonctionnels différents. La figure 2 présente les systèmes interactifs à concevoir et développer.

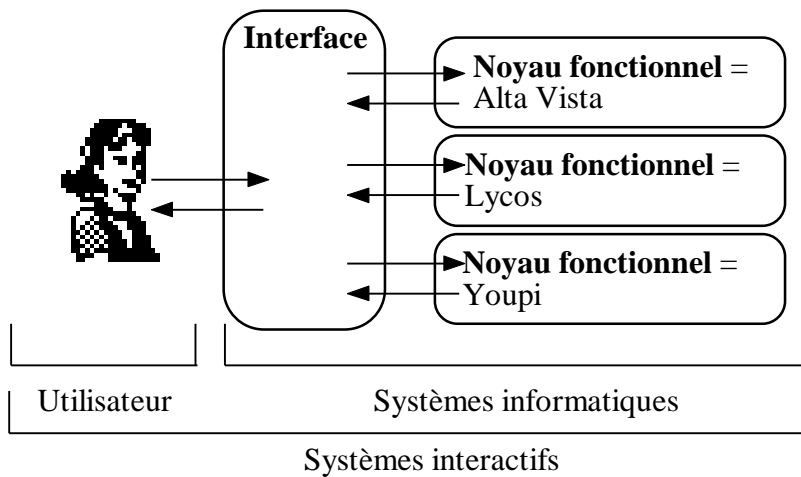


Figure 2 : Les systèmes interactifs à concevoir et développer

1.6. Motivations

Avant de présenter ma démarche de travail et les résultats de mon Étude, il me semble important de comprendre pourquoi les interfaces des moteurs de recherche doivent être améliorées, c'est-à-dire, rendues plus facile à utiliser. Pour cela, nous présentons les interfaces actuelles des moteurs de recherche et leurs problèmes d'utilisabilité. Les figures 3 présentent des copies d'écran de l'interface du moteur de recherche Alta Vista [ALTAVISTA].

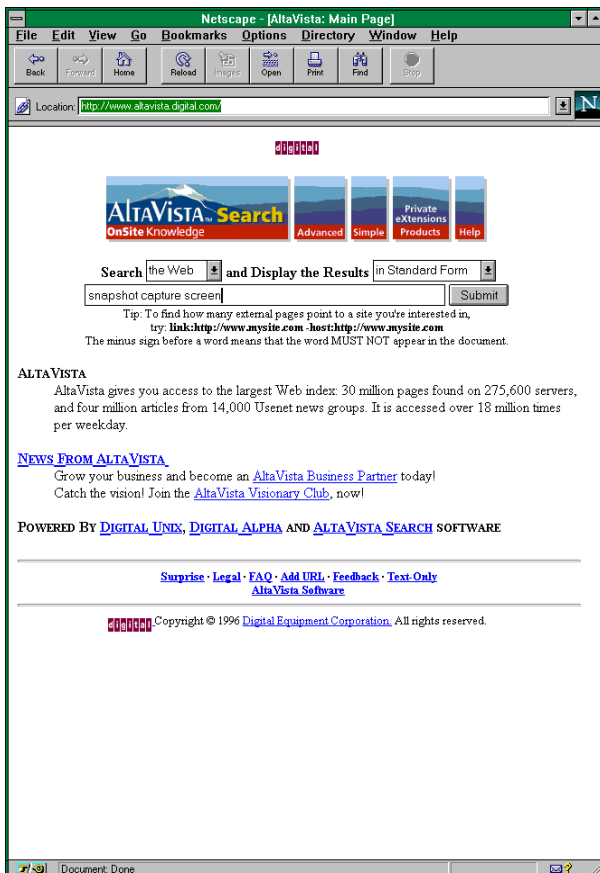


Figure 3a : Page d'accueil d'AltaVista



Figure 3b : Page de résultats d'une requête

Comme le montre la figure 3a, l'utilisateur saisit les mots-clés dans la page d'accueil : il s'agit d'une suite de mots séparés par des espaces, comme "computer human interface 1996 visualisation technique". Pour envoyer sa requête au moteur de recherche, l'utilisateur doit sélectionner le bouton "Submit" puis patiente alors quelques secondes pour obtenir les résultats. Une page de résultats est présentée à la figure 3b. Cette dernière contient une liste de liens vers d'autres pages HTML. Le résultat d'une requête peut ainsi contenir plusieurs milliers de réponses. Dans l'exemple ci-dessus (figure 3b), 40000 réponses ont été trouvées et celles-ci sont affichées dix par dix. A la fin de chaque page se trouvent des pointeurs vers les autres pages de réponses. L'utilisateur peut alors parcourir les différentes pages pour tenter d'obtenir une vue d'ensemble des réponses. L'utilisateur peut aussi parcourir séquentiellement les réponses jusqu'à ce qu'il ait trouvé l'objet de sa recherche. Dans ce contexte, le problème est de visualiser et de trouver les réponses les plus pertinentes à une requête.

Plus généralement, le problème n'est plus tant l'accès à l'information mais la capacité d'élicitation, à partir des représentations perçues, d'une information qui fasse sens et qui réponde à la tâche de recherche. L'information est disponible mais souvent inaccessible.

Dans ce contexte, il convient donc d'étudier les techniques d'interaction qui permettent de donner l'accès à un grand nombre d'informations. Les techniques d'interaction doivent aider et encourager l'utilisateur dans sa tâche de recherche. Cette dernière est variée et dépend du domaine applicatif :

- Dans le cas d'un système d'analyse statistique, la tâche consiste par exemple à trouver des tendances qui se répètent dans un ensemble d'information ou à mettre en valeur l'évolution d'une variable par rapport à une autre.
- Dans le cas d'un système de recherche d'information, la tâche consiste à trouver un ensemble de documents qui correspond le mieux à une requête. La tâche est donc de trouver des éléments dans un ensemble.

Chapitre 2

Etude bibliographique

2.1. Démarche de travail

Mon approche de travail est centrée sur l'utilisateur. Elle se décompose en 4 étapes :

- J'ai tout d'abord étudié l'activité de l'utilisateur pour tenter d'identifier les problèmes. Cette étude est dépendante du domaine applicatif choisi, les butineurs du WWW.
- J'ai ensuite généralisé les problèmes identifiés et sélectionné un ensemble de critères d'ergonomie qu'il me semble crucial de vérifier dans l'interface afin de minimiser les problèmes d'utilisation.
- En vue de la conception d'une interface adaptée, l'étape suivante consiste à étudier les techniques d'interaction d'une grande quantité d'information selon les critères d'ergonomie choisis.
- Ma dernière étape consiste à proposer une technique d'interaction en la justifiant selon les critères d'ergonomie.

Mon approche de travail repose sur des critères d'ergonomie qui permettent de prendre en compte l'utilisateur dans la conception. Mon travail est donc plus empirique qu'analytique car l'identification des critères que nous allons utiliser s'appuie en grande partie sur des expérimentations conduites par des ergonomes. Une autre approche, plus analytique, pour la prise en compte de l'utilisateur consiste à s'appuyer sur des théories de psychologie cognitive qui modélise l'activité mentale de l'utilisateur. L'approche que nous avons adoptée est sans nul doute la plus abordable par des informaticiens.

2.2. Etude des problèmes de l'utilisateur

L'utilisateur d'un moteur de recherche peut rencontrer de nombreux obstacles pendant la recherche d'information. Certains s'en sortent mieux que d'autres mais nombreux sont ceux qui sont perdus dans l'espace d'informations ("lost in the hyperspace") [Foley 95]. Cette impression d'être perdu dans cet hypertexte multimédia est le résultat de plusieurs facteurs que je présente aux paragraphes suivants.

2.2.1. Le temps de réponse

Les utilisateurs se plaignent très souvent d'un temps de réponse trop lent. En effet rares sont les utilisateurs ayant accès pleinement à Internet. La plupart se contente d'un accès via un

modem à 14400 bauds chez un prestataire de service dont les liaisons ne sont pas toujours excellentes. De plus les grands moteurs de recherche (“search engines”) sont presque tous aux Etats-Unis et les liaisons s’en trouvent ralenties. L’utilisateur n’est donc pas toujours très enclin à utiliser les moteurs de recherche mais force est de constater qu’ils sont essentiels à l’utilisation d’internet.

2.2.2. Le manque de vue globale de l’espace informationnel

Le manque de repères dans le World Wide Web rend très importante l’idée globale que l’on se fait de l’espace informationnel. Pourtant peu d’aide est fournie à l’utilisateur des moteurs de recherche pour se guider dans le dédale des réponses. En effet les réponses sont souvent affichées par groupe de 10 ou 20. Selon le moteur de recherche que l’on utilise, les réponses sont parfois listées selon leur pertinence (résultat qu’elles atteignent par rapport aux mots-clés). De plus pour chaque réponse, le passage contenant les mots clés de la recherche (1 ou 2 phrases généralement) est affiché. Mais souvent les moteurs de recherche ne proposent pas une vue globale sur la liste totale des réponses. Seul Ecila [ECILA], moteur de recherche français, donne 40 réponses maximum en 1 seule liste.

2.2.3. L’excès de réponses

Les réponses apparaissent souvent en trop grand nombre. Les moteurs proposent des milliers (voir des millions sur les mots les plus courants) de réponses et en affichent des centaines. Parfois même on retrouve plusieurs fois la même adresse URL avec un “/” ou bien le numéro de port explicite en plus. C’est dû au traitement “aveugle” des butineurs du WWW.

L’utilisateur ne se sert que rarement des réponses au delà des dix premières car bien souvent il est découragé devant la masse des réponses à sa requête. Malheureusement il arrive souvent que le site WWW recherché se trouve au delà de cette limite.

2.2.4. La perte du contexte de la recherche

Enfin de nombreux utilisateurs, pour ne pas dire presque tous, se plaignent de la confusion qui règne dans le WWW. Les moteurs de recherche sont à l’image de l’espace informationnel qu’ils représentent c’est-à-dire assez confuse. Cette confusion vient essentiellement de la perte du contexte qui s’opère à chaque fois que l’utilisateur va regarder l’une des adresse qui lui est proposée. Il perd alors la liste des réponses et donc le contexte de sa recherche. Pour revenir à la réponse du moteur de recherche, la plupart des fureteurs (netscape, omniweb, internet explorer, etc.) proposent un bouton “marche arrière”qui permet de retourner à la page précédente.

2.3. Sélection de critères d’ergonomie

Les critères d’ergonomie sont un raffinement du critère d’utilisabilité de Génie logiciel. Les critères d’ergonomie ne font pas l’objet d’un consensus et chaque grand organisme en informatique possède sa propre liste de critères.

2.3.1. Critères d'ergonomie

Les critères d'ergonomie constituent un raffinement du critère d'utilisabilité. Ce sont les ergonomes qui, en essayant de qualifier l'utilisabilité d'un système, ont introduit ces critères. Ils ne sont pas encore définitivement établis et il existe plusieurs listes de critères. En adoptant l'approche décrite dans [Unger 92], les critères sont classifiables en deux catégories : les critères de robustesse et de souplesse.

Le paragraphe suivant présente la liste complète des critères et de leurs définitions dans [Unger 92].

2.3.1.1. Critères de souplesse

Propriété	Définition	Propriétés liées, remarques
Atteignabilité	Capacité du système à permettre à l'utilisateur d'atteindre le but désiré depuis l'état actuel.	Propriété non vérifiée si analyse de tâche déficiente. Métrique : longueur de la trace d'interaction pour atteindre le but.
Non-préemption	Le prochain but souhaité par l'utilisateur est directement atteignable. (pour l'utilisateur, il n'y a pas de contrainte artificielle)	- Contre-exemple : boîte de dialogue modale (à utiliser à bon escient : prévenir des conséquences d'une action irrévocable) ; opération ininterrompible. - l'utilisateur peut être perdu suite à trop de liberté : concept de tableau de bord.
Interaction à plusieurs fils (multithreading)	Capacité du système à permettre à l'utilisateur de réaliser plusieurs tâches de manière entrelacée.	Exemples : - édition de plusieurs documents ouverts à la fois Métrique : nombre de tâches exécutables de manière entrelacée -> s'interroger sur la surcharge potentielle
Multiplicité de la représentation	Capacité du système à fournir plusieurs représentations d'un même concept (variable d'état)	Différentes déclinaisons : <u>En sortie</u> : - formes différentes : ex. concept de température : thermomètre/un entier. - contenus différents : détail/ ensemble. Attention à la discontinuité visuelle. <u>En entrée</u> : - formes différentes : 6*4 et 24 <u>Principe d'égalité d'opportunité</u> : l'utilisateur choisit la nature de l'entrée et le système en déduit la sortie (tableaux).
Réutilisabilité des données d'entrée et de sortie	Les sorties du système peuvent être utilisées comme données d'entrée (couper-coller) Les entrées de l'utilisateur peuvent être réutilisées par le système en sortie (valeurs par défaut)	Attention aux effets de bord suite aux conversions de types de données. Cas du cercle Macdraw par aller-retour via MacPaint au moyen d'un copier-coller.

Adaptabilité	Personnalisation du système sur intervention explicite de l'utilisateur	<ul style="list-style-type: none"> - Menus et formulaires d'options et de préférences. Macros d'encapsulation de commandes élémentaires dont le niveau d'abstraction est trop bas à caractère répétitif. - Risque de perte de cohérence entre les systèmes d'une communauté d'utilisateurs sensés travailler ensemble. - Toute information du niveau lexical de l'interaction doit être définie dans un fichier de ressources (pas dans le code du programme!)
Adaptativité	Capacité du système à s'adapter à l'utilisateur sans intervention explicite de la part de l'utilisateur	<p>L'adaptativité s'appuie sur un modèle "embarqué" de l'utilisateur. Veiller à ce que le système ait un comportement prévisible. Ne pas surprendre l'utilisateur.</p>
Migration de tâche	Capacité de délégation dynamique de tâches entre le système et l'utilisateur (c.-à-d. changement de l'acteur responsable de la tâche)	<p>Se manifeste à différents niveaux de granularité:</p> <ul style="list-style-type: none"> - 6*4 et 24 - valeurs par défaut - détection de tâche répétitives puis prise en charge (Eager) - sauvegarde automatique

2.3.1.2. Critères de robustesse

Prévisibilité	Capacité pour l'utilisateur de prévoir, pour un état donné, l'effet d'une action.	<p>Cohérence : conformité aux règles/usages de l'utilisateur.</p> <ul style="list-style-type: none"> - Les règles/usages de l'utilisateur ne sont pas nécessairement les vôtres. - Cohérence interne : cohérence lexicale, syntaxique, sémantique. - Cohérence externe : <ul style="list-style-type: none"> . conformité à des normes d'interface, à un autre système, . conformité à l'expérience dans le monde réel (analogie, métaphore du monde réel). - Retour d'information pro-actif : principe du "do nothing", résistance passive : éléments interdits en grisé. - Prévisibilité et stabilité des temps de réponse <ul style="list-style-type: none"> . rassurer si temps de réponse long . attention aux solutions système de type Ramasse-miette à la volée sur la stabilité des tps de réponse.
Tolérance du rythme	L'utilisateur, plutôt que le système, décide quand il peut agir	<ul style="list-style-type: none"> - Ex. : saisie anticipée <p>Métrique : nombre d'actions anticipables</p> <ul style="list-style-type: none"> - Contre-exemple : logout, certains distributeurs de café! <p>Métrique : durée de tolérance.</p>

Propriété	Définition	Propriétés liées, remarques
Observabilité	Capacité pour l'utilisateur d'évaluer (de déterminer) l'état interne du système. L'utilisateur PEUT PERCEVOIR	Inspectabilité (browsability) : capacité pour l'utilisateur d'explorer l'état interne du système au moyen de commandes passives (c.-à-d. qui ne modifient pas cet état) : défilement, zoom, etc. Contre-exemple : "defense in depth design" (Rasmussen)
Insistance	Capacité du système à forcer la perception de l'état du système. L'utilisateur DEVRA PERCEVOIR	Le retour d'information du système peut être : - éphémère (ex. audio, vidéo) ou non (ex. écrit statique) - évitable (retour visuel) ou inévitable (audio) - entretenu par le système (clignotement) ou par l'utilisateur (maintien d'un bouton enfoncé qui minimise les oublis) Attributs additifs : l'intensité sonore et l'intensité des couleurs ont un effet additif ; le ton sonore et la teinte ne sont pas additifs.
Honnêteté	Capacité du système à rendre observable l'état du système sous une forme conforme à cet état et qui engendre une interprétation correcte de la part de l'utilisateur. L'utilisateur INTERPRETERA CORRECTEMENT L'ETAT DU SYSTEME.	- WYSIWYG (What You See is What you Get) - Distorsion des informations (ex. les données linéaires ne doivent pas être présentées en 2 dimensions) - Conformité état interne et présentation pas toujours compatible avec les temps de réponse attendus : utiliser un indicateur pour notifier l'utilisateur que l'info a changé et qu'elle n'a pas été réactualisée. - Formulation des unités de mesure, du format des données de saisie. - Terminologie précise.
Curabilité	Capacité pour l'utilisateur de corriger une situation non désirée.	- Curabilité arrière : capacité de défaire. - Curabilité avant : reconnaissance de l'état actuel et capacité de négociation pour atteindre le but désiré (= atteignabilité suite à détection d'erreur) Messages d'erreur explicatifs et correctifs La procédure corrective doit refléter le travail à corriger = principe de l'effort incommensurable : si difficile à corriger, alors difficile à faire (ex. destruction de fichier)

Parmi ces critères j'en ai retenu six qui me semblent pertinents pour notre étude. Certes tous ces critères sont pertinents mais certains me semblent plus cruciaux dans le cas de l'interaction pour une grande quantité d'information.

2.3.2. Les critères de robustesse pertinents

J'ai retenu trois critères de robustesse pertinents pour notre étude :

- L'observabilité est la capacité de l'utilisateur à évaluer l'état interne du système à partir de sa présentation. Toutefois l'observabilité totale n'est pas le remède au manque d'observabilité d'un système. Il faut bien discerner les variables internes qu'il est pertinent d'observer. Dans le cadre des moteurs de recherche j'ai retenue les variables suivantes :
 - l'ensemble des réponses pour ne pas perdre le contexte,
 - les liens hypermédias entre les réponses comme le préconise [Foley 95],
 - le nombre de réponses pour renseigner de la taille de l'espace manipulé,
 - le centre d'intérêt, c'est-à-dire une des réponses à la requête.
- L'honnêteté est la capacité de l'utilisateur à évaluer correctement l'état interne du système. Elle est garante de la bonne interprétation des données quand le système représente son état interne de façon inhabituelle.
- La prévisibilité est le critère qui indique que pour chaque action entreprise, le résultat est prévisible. Pour notre cas elle est pertinente dans la navigation de l'espace des informations.

2.3.3. Les critères de souplesse pertinents

J'ai aussi retenu trois critères de robustesse pertinents pour notre étude :

- La multiplicité des représentations reflète la capacité du système à afficher ses données de différentes manières. Dans le cas des techniques d'interaction, elle permet de faire face à la diversité des utilisateurs. Une seule représentation ne pourra satisfaire tous les utilisateurs. La grande variété dans les utilisateurs incite à proposer des techniques différentes que l'utilisateur pourra choisir.
- La réutilisabilité des données en entrée et en sortie est aussi importante pour une bonne utilisation des systèmes de technique d'interaction. Elle permet d'insérer la technique d'interaction dans une démarche plus globale de l'utilisateur.
- L'adaptabilité est le critère de souplesse qui indique la possibilité de paramétrer le système. Les techniques d'interaction nécessitent la vérification de ce critère car elle permet à l'utilisateur d'adapter la technique d'interaction à ses préférences.

2.4. Les espaces informationnelles de grande taille

Pour répondre aux problèmes d'observabilité de nombreux scientifiques et chercheurs en informatique ont tenté d'apporter des solutions par des techniques de visualisation ou d'interaction.

2.4.1. La problématique des mines d'information trop importantes

La taille des systèmes d'information croît beaucoup plus vite que la taille des outils permettant de les représenter. De plus à partir d'une certaine taille le cerveau humain et la mémoire en particulier ne peuvent plus palier au manque d'information visible. Il faut alors utiliser la capacité de synthèse de l'homme en lui fournissant une information moins brute dans laquelle il trouvera les repères adéquates associés à la sémantique de l'information. Le repère le plus naturel est le contexte de l'information. En effet il permet de ne pas se perdre puisque à tout moment l'utilisateur peut connaître ce qui l'entoure et donc savoir où il va. Néanmoins ce contexte n'est pas toujours facilement représentable. Il est aussi très important de pouvoir localiser ce contexte dans l'espace d'information que l'on nous donne sinon il perd toute sa valeur. On cherche à ce que cette représentation lie l'information et le contexte de façon **cohérente et intuitive**. Il est donc du domaine des professionnels de l'interface Homme-Machine de définir avec précision la qualité d'une représentation. Pour cela un travail d'évaluation et de classification des moyens de représentation est une étape indispensable. Il existe déjà de nombreuses idées qu'il faut classer et évaluer afin d'avoir une meilleure vue du problème et de trouver de nouvelles représentations permettant la visualisation du contexte. Il est intéressant de noter que de telles techniques ne nécessitent pas forcément un ordinateur mais ce dernier permet d'envisager des représentations d'un type nouveau grâce au couplage avec l'utilisateur. Les informations peuvent alors prendre une dimension malléable et évolutive quant aux attentes de l'utilisateur.

2.4.2. Différentes techniques d'interaction

De nombreuses techniques d'interaction ont déjà été conçues pour répondre à d'autres problèmes d'observabilité. Elles peuvent être réutilisées pour d'autres types de données que ceux initialement prévus. C'est pourquoi nous allons passer en revue les techniques d'interaction les plus courantes afin de voir si l'une d'elles pourrait être adaptée à notre cas d'étude.

2.4.2.1. Lentille magique

Les lentilles magiques (« magic lenses ») ne déforment pas le champ des informations mais filtrent son contenu. Sans lentille magique toutes les informations sont représentées à importance égale (le treillis est donc très confus). Les lentilles magiques permettent de filtrer certaines zones de l'espace (zooms, filtres sémantiques, etc.). L'intérêt de ces filtres est de les combiner pour laisser apparaître un nombre minimal d'information. Le déplacement des lentilles permet de visualiser la totalité des informations et la superposition des lentilles permet de dégager une information particulière. Ce système est assez efficace sur les espaces denses d'information dense comme les cartes géographiques ou météorologiques.

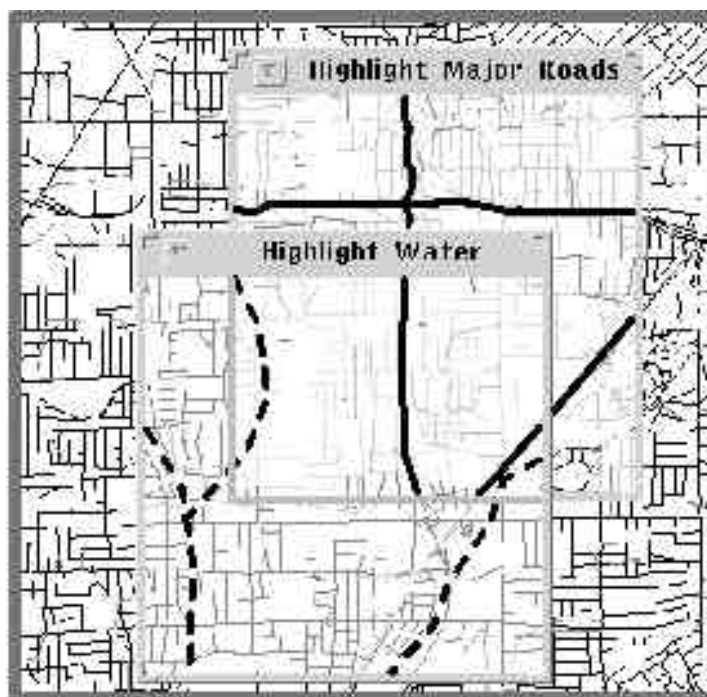


Figure 4 : Deux lentilles magiques superposées sur le plan d'une ville

2.4.2.2. Système « DragMad »

Le système « DragMad » [Ware 95] est une amélioration des lentilles magiques agrandissantes qui permet de mieux situer le centre d'intérêt dans le contexte. En effet les lentilles magiques cachent l'emplacement du centre d'intérêt à l'utilisateur. Le « DragMad » quant à lui déplace la zone « agrandie » et la situe dans le contexte en reliant la zone à agrandir et la zone agrandie par la vue en perspective d'un cube. On a l'impression de voir un bâtiment vu du ciel dont la base (très lointaine) serait copiée sur le toit (plus proche). La face avant peut être glissée sur l'écran. Le facteur d'agrandissement est proportionnel à la distance de la face avant et de la base. La base aussi peut être déplacée (toujours en glissant la souris) afin de visualiser une autre partie de l'espace d'information.



Figure 5 : Un « Drag Mad » sur la région lyonnaise

2.4.2.3. Mur en perspective

Le mur en perspective (« Perspective Wall ») est une technique d'interaction qui permet de visualiser une grande quantité d'information. Ces informations sont représentées sur un mur infini. Ce mur contient en général trois faces comme le montre la figure 6 :

- la face avant qui n'est pas déformée mais qui permet de visualiser une petite quantité d'information sans distorsions,
- deux faces périphériques vues en perspective (déformation des informations) mais qui offrent une surface quasi infinie pour représenter le contexte.

Cette représentation est bien adaptée aux listes trop longues et aux diagrammes trop longs pour être représentés d'un seul tenant. Elle s'adapte aussi très bien aux représentations du temps puisqu'elle permet de rendre compte du passé et du futur.

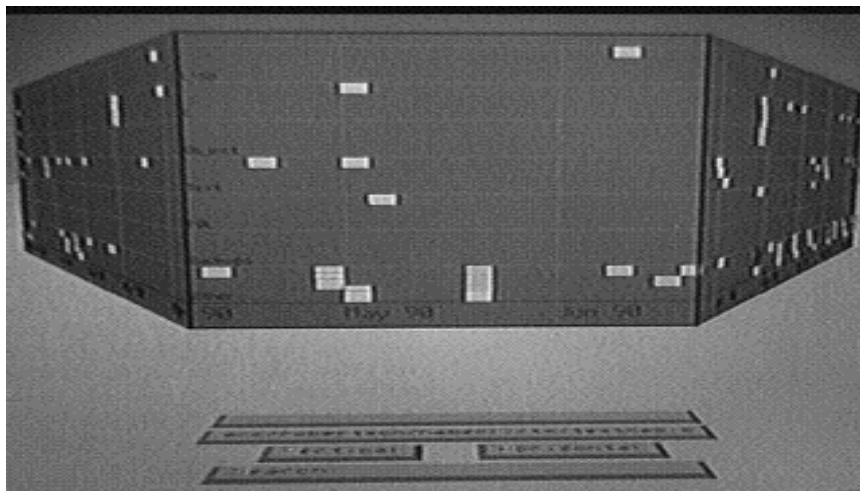


Figure 6 : Mur en perspective

2.4.2.4. Vue en œil de poisson

Une vue en œil de poisson (« Fisheye View ») est une vue 2D déformée d'un ensemble d'informations consistant à faire apparaître une information au premier plan (centre d'intérêt), de taille satisfaisante, et de faire apparaître toutes les autres informations de taille inversement proportionnelle à la distance à l'information principale (centre d'intérêt). On obtient ainsi une vue similaire à celle d'un objectif « grand angle » d'un appareil photographique. L'utilisateur ne perd pas le contexte mais déforme sa vision globale (comme la vue d'un poisson ou d'une mouche). Cette technique peut par exemple rendre méconnaissable une carte des Etats-Unis comme le montre la figure 7, mais clairement faire apparaître des informations sur St Louis et sa région qui est pourtant petite à l'échelle nationale. La vue en œil de poisson ne permet pas une comparaison aisée entre deux éléments détaillés (à moins qu'ils ne soient très proches). Il faudrait avoir une autre déformation centrée sur deux points d'intérêts, ce qui reste envisageable mais difficile à réaliser.

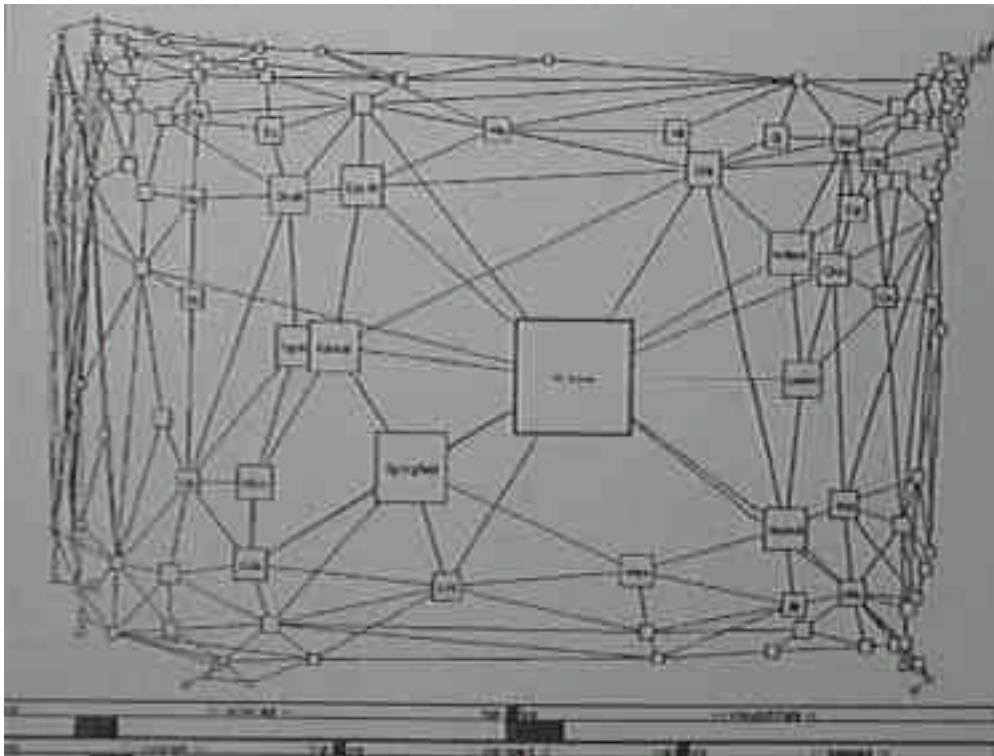


Figure 7 : Vue en œil de poisson sur une carte des Etats-Unis

2.4.2.5. Système de « zoom »

La technique du zoom permet de s'éloigner de l'ensemble des informations pour changer de centre d'intérêt puis de se rapprocher à nouveau comme dans le système ProjectX [PROJECTX 96]. Il permet de se situer mais ne permet pas la visualisation simultanée du détail et du contexte. A l'inverse de la vue en œil de poisson dans laquelle c'est l'information qui se déforme pour mettre en évidence le centre d'intérêt, la technique de zoom permet de se déplacer dans l'espace d'information pour se rapprocher de son centre d'intérêt.



Figure 8 : Technique de zoom, ProjectX

2.4.2.6. Arbre conique

Cette technique du sapin ou de l'arbre conique, présentée à la figure 9, vise à ranger les éléments non pas linéairement mais sur un carrousel composé de cercles. On regarde alors les cercles de profil et on les fait tourner pour amener une information devant soi. Il est alors possible d'introduire une hiérarchisation des données ce qui permet lorsqu'on a sélectionné un objet de voir apparaître un nouveau cercle affilié au précédent. Une telle organisation fait apparaître un cône car à chaque niveau l'ensemble des informations représentées augmente. Cette représentation spatiale, à première vue peu commode, est parfois le seul recours devant un très grand nombre d'informations hiérarchisées.

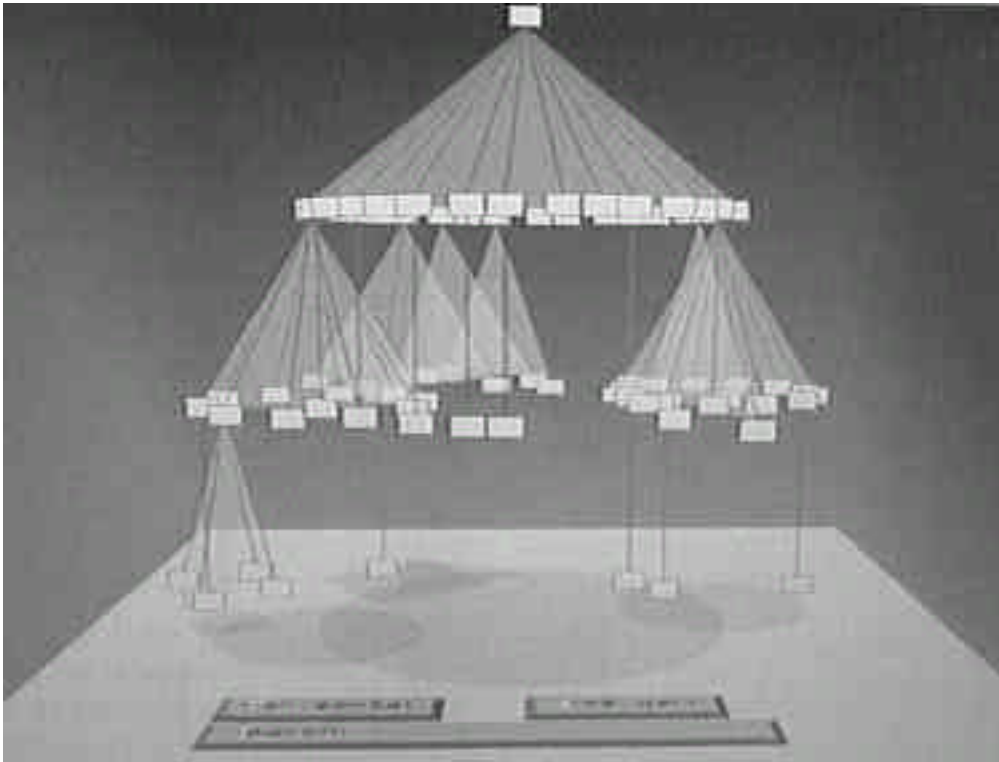


Figure 9 : L'arbre conique sur une arborescence de fichiers

2.5. Etude des techniques d'interaction répondant aux critères

Une technique de représentation ne convient pas à n'importe quel type d'information et à n'importe quelle tâche liée au domaine applicatif. En particulier l'information que l'on veut représenter comporte des contraintes qui peuvent être incompatibles avec un système de représentation. Ainsi le nombre de dimensions ou la nature des médias mis en jeu sont décisifs dans l'établissement d'une technique appropriée.

2.5.1 Exemples de représentation bien adaptée

Dans le cadre d'une présentation d'une carte des États-Unis, la représentation des lignes aériennes entre les aéroports des grandes villes américaines par une vue en œil de poisson donne un bon résultat puisque les lignes lointaines du centre d'intérêt n'interfèrent pas avec le centre d'intérêt. De plus les noms des villes sont de bons repères car ils sont bien connus de l'utilisateur. Par contre la représentation du réseau routier serait déformée par une vue en œil de poisson et une

route droite apparaîtrait courbe si elle était située près du centre d'intérêt. La géométrie du réseau constitue ici un repère important pour l'utilisateur. On remarque que c'est l'utilisateur qui détermine la représentation car la distinction entre le réseau routier et aérien peut paraître minime en conception : cependant l'utilisateur s'en sert de deux manières différentes, impliquant deux représentations différentes.

2.5.2. Application des techniques de représentation au WWW

Toutes les techniques d'interaction que nous venons de présenter ont pour but d'améliorer l'observabilité des systèmes. Pour définir une technique adaptée, il convient aussi de prendre en compte les caractéristiques propres au domaine applicatif et les autres critères retenus : la multiplicité des représentations et l'adaptabilité. Ceci fait l'objet de ce paragraphe.

2.5.2.1. Etude des cas existants

Avant de proposer une technique d'interaction, nous examinons d'abord quelles sont les représentations actuelles du WWW. En effet les représentations actuelles constituent la source de problèmes d'utilisation.

Nous constatons tout d'abord que les navigateurs du WWW (netscape, mosaïc, etc.) n'essaient pas de représenter le contexte. Les butineurs qui s'utilisent au travers de ces navigateurs présentent qu'une petite partie des réponses : afin d'améliorer la lisibilité des réponses, ils appliquent des techniques de découpage de la liste complète des réponses. Ainsi la première page résultat d'une requête contient qu'un sous-ensemble de l'ensemble des résultats.

La liste, résultat d'une requête, peut aussi faire apparaître le contexte du mot recherché en affichant la ou les phrases qui entourent le ou les mots recherchés en y soulignant le mot clé. L'utilisateur peut alors voir si le mot qu'il recherche est bien employé dans le sens désiré. Cependant le contexte reste textuel : le mot clé par rapport à un texte. Les autres médias (image, vidéo, sons, etc.) ne sont donc pas pris en compte.

Enfin nous constatons que les réponses sont classées selon leur pertinence par rapport aux mots clés d'une requête uniquement. La situation de la réponse par rapport à l'ensemble des réponses est absente. Or une page fréquemment référencée par les autres pages retrouvées peut intéresser l'utilisateur même si sa pertinence globale par rapport à la requête est faible. Ce type de page serait dans les systèmes existants sans doute classées dans les dernières en vue de sa pertinence et sûrement pas présentés dans la première page de résultats.

2.5.2.2. Particularités du WWW

Le WWW peut être assimilé à un hypertexte multimédia (son, texte, image, etc.). Les représentations habituelles des hypertextes sont les graphes où l'on peut aisément représenter les objets et les liens. Il est cependant difficile d'insérer dans un graphe des éléments aussi différents que du texte, des images, du son et pourquoi pas des scènes 3D ou des animations.

De plus, comme il est souligné dans [Foley 95], l'application d'un graphe pour représenter l'ensemble des pages du WWW est sans utilité en vue de la quantité d'information. Toutefois Apple tente de finaliser un tel projet [ProjectX 96]. Apple s'est servi de la partie hiérarchisée du WWW qu'ont réalisé les auteurs du butineur Yahoo [YAHOO]. Cette hiérarchisation soulève des problèmes :

1. Elle est incomplète.
2. Elle est trop stricte. (Comment juger si une page WWW est plutôt classée dans “Loisir” plutôt que dans “Education” quand celle-ci contient la page personnelle d’un étudiant ?)
3. Elle n’évolue pas assez vite et peut donc contenir des informations erronées.

La particularité essentielle du WWW provient des liens qui unissent les pages entre elles. On représente habituellement les liens dans les graphes par des lignes, des courbes ou des flèches. Le WWW contient évidemment des liens orientés (les pointeurs ou adresses URL sont rarement réciproques).

2.5.3. Une solution de technique d’interaction

La technique que nous envisageons est destinée à représenter l’ensemble des réponses à un butineur et non l’ensemble des pages WWW comme dans le projet ProjectX.

En prenant en compte les particularités du WWW exposés ci-dessus, la technique d’interaction doit pouvoir représenter :

- un ensemble dense d’information puisqu’une requête peut engendrer des centaines de réponses,
- un ensemble d’information de types différents (son, image, texte, liens, animation, applets JAVA, etc.),
- un ensemble d’information dense par le contenu puisqu’une page WWW correspond à des centaines de mots et des dizaines de liens.

Nous avons décidé d’insister sur la représentation en graphe des résultats d’une requête afin d’explicitier les liens entre les réponses. Dans ce contexte, il s’agit donc de choisir une technique d’interaction adaptée aux graphes : nous optons pour une vue en oeil de poisson particulièrement adaptée. Le principal inconvénient de cette technique est de déformer la géométrie de l’espace. Cependant dans le cas des graphes, la géométrie de l’espace n’est pas pertinente car elle ne véhicule aucune sémantique particulière.

Les liens sont des adresses URL faisant référence à des pages de l’ensemble des résultats. J’envisage de pondérer ces liens selon le nombre d’occurrences de l’adresse URL dans la page mais aussi selon le contexte d’apparition de l’adresse URL. Par exemple une adresse URL proche d’un mot clé de la requête pourrait augmenter l’importance de la liaison. Le poids associé au lien serait traduit à l’écran par la proximité des deux pages dans le graphe mais aussi par une présentation différente. Ces liens peuvent être représentés par des flèches car ils sont orientés. Cependant la représentation par une flèche soulève des problèmes : le bout de la flèche a tendance à donner plus d’importance à l’objet pointé qu’à l’objet pointeur. J’envisage donc d’autres solutions comme la représentation des liens sous la forme d’un triangle de largeur variable selon le poids du lien. Seule une expérimentation avec des utilisateurs confirmera ou non ces choix de conception de la présentation (d’ordre lexical).

Il est de plus envisageable de définir plusieurs types de vue en oeil de poisson (sphérique, tronquée, etc.). Cette solution permet de vérifier les critères de multiplicité des représentations et d’adaptabilité.

Pour pouvoir gérer la variété des médias dans le graphe, je propose d’exploiter la technique des lentilles magiques. Combinée avec celle de la vue en oeil de poisson, cette dernière permet par

exemple de présenter un sous-ensemble des informations du graphe selon une ou plusieurs caractéristiques telles que le média de l'information. Ainsi l'utilisateur peut centrer sa recherche dans l'ensemble des réponses sur un média particulier sans perdre une vue globale de l'ensemble des réponses. Par exemple, une lentille magique permet de faire apparaître (ou disparaître) les états du graphes qui ne contiennent pas un média. Ainsi si un utilisateur cherche une photographie d'une "Ford Mustang" il lancera la recherche avec les mots clés "automobile, Ford et Mustang" et ne s'intéressera qu'aux réponses comportant une photographie.

Dans la vue en oeil de poisson, le focus ou centre d'intérêt devra faire apparaître l'information en détail (mot clé recherché dans sa phrase ou la pertinence globale de la page). De plus l'utilisateur doit pouvoir obtenir des détails sur des noeuds éloignés du focus à des fins de comparaison.

2.6. Conclusion

Dans ce chapitre j'ai identifié des critères d'ergonomie qui me semblent pertinents pour l'interaction avec une grande quantité d'information.

A partir de ces critères et des caractéristiques propres au domaine applicatif (les butineurs du WWW), je propose une technique d'interaction qui combine une vue en oeil de poisson et des lentilles magiques.

Il convient maintenant d'en étudier sa mise en oeuvre informatique, deuxième volet de mon étude. La réalisation logicielle de la technique d'interaction fait l'objet du chapitre suivant.

Chapitre 3

Réalisation logicielle

3.1. Mise en œuvre des résultats de mon étude

L'étude dont les résultats sont présentés au chapitre 2 consistait dans une première partie à recenser et comparer dans un espace de classification les techniques d'interaction envisageables. Les critères de classification sont des critères d'ergonomie qui participent à l'amélioration de l'utilisabilité d'un système interactif. A partir de ce recensement, mon travail a consisté à proposer une technique d'interaction qui vise à vérifier les critères d'ergonomie qui me semblent les plus pertinents dans le cas d'une tâche de recherche d'information sur le WWW.

La continuité de ce travail théorique consiste donc à mettre en oeuvre la technique d'interaction proposée au sein d'une interface pour des moteurs de recherche du WWW.

3.2. Démarche de travail

Pour appréhender et conduire le projet de réalisation, j'ai appliqué les étapes de conception d'un système interactif présentées à la figure 10.

3.2.1. Les phases de conception de l'interface

Je rappelle ici les étapes de conception d'une interface [Coutaz 90]. La tâche de conception peut se décomposer en cinq sous-tâches essentielles : identification de l'utilisateur, définition des tâches et des objets spécifiques au domaine d'application, définition des objets et fonctions informatiques, définition de la présentation de l'interface, et évaluation expérimentale. Ce schéma est une vue simplifiée du processus de conception. En pratique, les sous-tâches sont chacune soumises à des révisions accompagnées de nombreux retours arrière.

3.2.1.1. Identification de l'utilisateur

Dans le cas du WWW, nous ne pouvons pas faire d'hypothèses sur les utilisateurs. Celui-ci peut être novice, occasionnel ou expert.

3.2.1.2. Définition des tâches et des objets spécifiques au domaine

La tâche consiste à rechercher une information. Les objets du domaine sont donc vagues pour l'utilisateur. Il s'agit d'un ensemble d'informations.

3.2.1.3. Définition des objets et fonctions informatiques

Le noyau fonctionnel du système à concevoir est déjà mis en oeuvre et définit donc un ensemble d'objets et fonctions informatiques. L'interface peut cependant augmenter les capacités

fonctionnelles du noyau fonctionnel. Pour les objets informatiques de notre interface, nous avons choisi ceux du noyau fonctionnel : une requête, des réponses sous forme de liens hypertexte et des pages WWW. L'objectif de mon projet est de mettre en oeuvre une technique d'interaction qui manipule de tels objets informatiques.

Par rapport à un cycle de développement plus classique issu du génie logiciel, il est classique de considérer que la phase de définition des besoins englobe les trois premières étapes ci-dessus de la méthode de conception. La phase de Spécification externe correspond à la définition de la présentation de l'interface. Ceci fera l'objet du paragraphe 3.4. La phase d'évaluation expérimentale n'est pas encore faite : je décris donc cette phase comme une perspective à mon projet dans le paragraphe 4.2.

3.2.2. Les phases de conception logicielle de l'interface

En ce qui concerne la conception logicielle, j'ai appliqué un modèle d'architecture logicielle PAC-Amodeus [Nigay 94] qui constitue la phase de spécification globale du système. L'architecture de mon système est décrite au paragraphe 3.5.2. Cette phase de conception globale est suivie d'une phase de spécification détaillée de chacune des parties du système : serveur et client. Les résultats de cette phase sont décrits au paragraphe 3.6.

3.2.3. Plan : les résultats de chaque étape de conception

Je rappelle d'abord les critères d'ergonomie que j'ai identifiés lors de l'étude. Ceux-ci constituent un plan qualité simplifié qui identifie les critères que le système doit vérifier. Outre ces critères d'ergonomie, des critères de génie logiciel doivent être respectés. En particulier la portabilité du code produit par rapport au noyau fonctionnel. Ceci a été défini dans le sujet de stage.

Dans le paragraphe 3.4.2.2, je présente l'interface que j'ai conçue. Ceci constitue les spécifications externes de mon système. Cette interface se justifie par les critères d'ergonomie introduits au paragraphe précédent.

Après avoir conçu l'interface, il convient d'étudier la réalisation logicielle de mon système. Au paragraphe 3.5.2., je présente l'architecture logicielle de celui-ci. Il s'agit de spécifications globales. Le paragraphe 3.6. présente les spécifications détaillées où je décris en détail les services de chacune des parties du système.

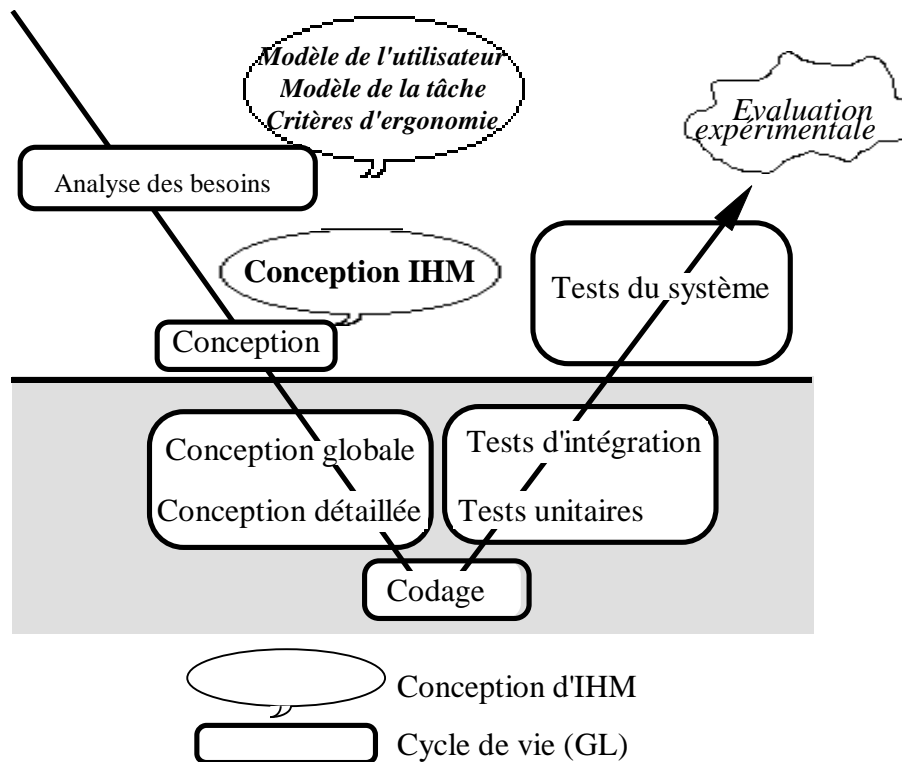


Figure 10 : les étapes de conception et de réalisation logicielle d'un système interactif

3.3. Les critères de qualité

3.3.1. Les critères de génie logiciel

Les critères de génie logiciel permettent de planifier et de qualifier la réalisation d'une application informatique. Parmi ces nombreux critères certains sont plus importants pour notre système tels que :

- la portabilité du code par rapport au noyau fonctionnel,
- la modifiabilité du code en vue d'une mise au point itérative par évaluation expérimentale,
- l'utilisabilité du système que nous cherchons à améliorer par rapport aux systèmes existants.

3.3.2. Les critères d'ergonomie

Je rappelle brièvement les critères que j'ai retenus aux paragraphes 2.3.2 et 2.3.3.

- Les critères de robustesse
 1. *L'observabilité*
 2. *L'honnêteté*
 3. *La prévisibilité*
- Les critères de souplesse
 1. La réutilisabilité
 2. L'adaptabilité
 3. La multiplicité

3.4. Les spécifications externes du système

3.4.1. Les techniques d'interaction

Les techniques d'interaction sont des techniques informatiques de visualisation de données qui en plus tiennent compte de l'interaction avec l'utilisateur. Ces techniques tel que le « Drag Mag » [Ware 95] ou les vues en œil de poisson [Sarkar 92] permettent de visualiser un grand nombre d'informations.

Comme il est expliqué au chapitre précédent, je préconise l'emploi de deux techniques d'interaction pour représenter les données en provenance des moteurs de recherche du WWW (aussi appelé Web) :

- une vue en œil de poisson (« **fisheye view** »),
- une lentille magique (« **magic lense** »).

La combinaison de ces deux techniques devrait permettre de représenter l'information très dense et très volumineuse des moteurs de recherche. En effet les moteurs de recherche rendent de très nombreux pointeurs (parfois plusieurs milliers) et chaque pointeur correspond à une page au format HTML qui est souvent très riche (textes, sons, images, animations,). La vue en œil de poisson aura pour but de tordre l'espace où sont affichés les pointeurs afin de pouvoir interagir directement avec l'ensemble de **toutes** les données. La lentille magique permettra de filtrer les différents médias qui rentrent dans la composition d'une page au format HTML. Cette dernière permettra de faire évoluer l'interface avec les médias toujours plus nombreux qui sont aujourd'hui reconnus par les navigateurs de la toile d'araignée mondiale.

3.4.2. L'interface

3.4.2.1. L'interface à réaliser

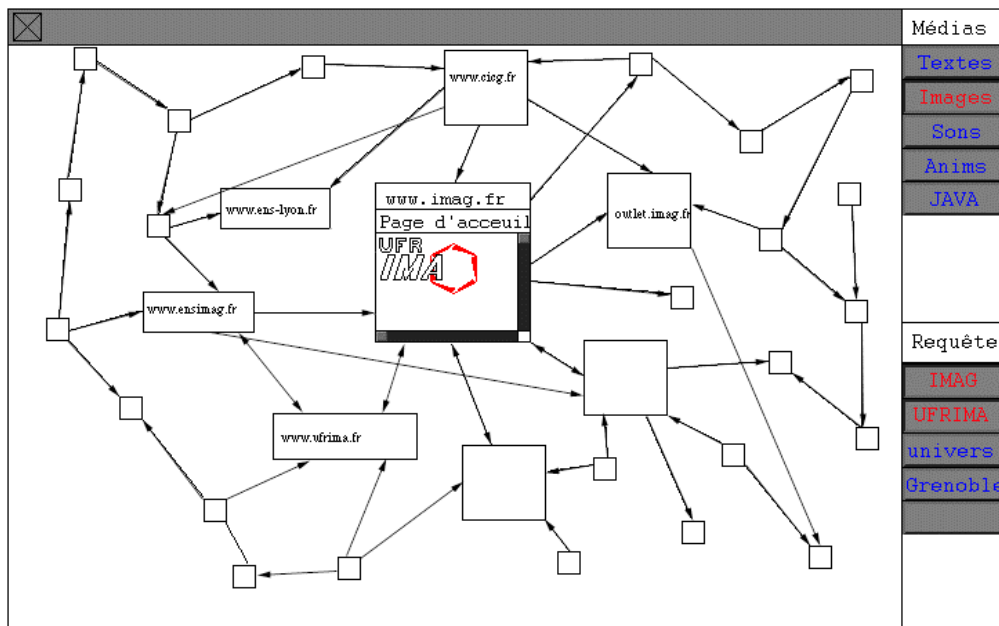


Figure 11 : Fenêtre des résultats

La figure 11 représente la page des résultats d'une requête. Elle est composée d'une zone de travail (à gauche) et d'une zone de contrôle (boutons de droite).

- Dans la zone de travail est affiché l'ensemble des réponses selon une vue en œil de poisson. Cette technique a pour but de représenter le contexte et le focus de l'utilisateur. Elle permet donc de garder une vue globale de l'espace d'information pendant le traitement d'une partie de ces mêmes informations.
- Dans la zone de contrôle se trouve une palette d'objets qui permettent de paramétrer la technique d'interaction. Dans l'exemple de la figure 11 j'ai choisi de pouvoir paramétrer les médias représentés et les mots-clés qui influencent la requête.

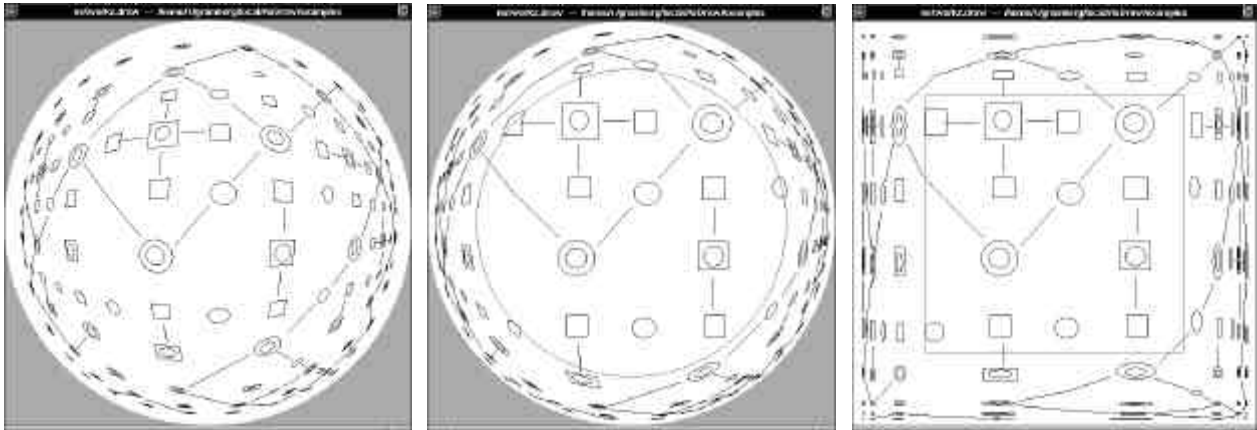


Figure 12 : trois autres vues en œil de poisson

La vue en œil de poisson de la figure 11 modifie l'emplacement et la taille des états du graphe. Elle permet de garder des états rectangulaires dans lesquels il est plus facile d'insérer du texte ou des images. C'est l'une des vues possible. La figure 12 représente trois autres vues qui ont été proposées par [Volkmar 95] lors de la conférence CHI (Computer Human Interaction) de 1995. Elle sont différentes de la première par leurs techniques de déformations de l'espace.

La première des vues de la figure 12 projette les données sur une sphère. Elle me semble plus intéressante car elle repose sur une métaphore du monde réel : celle du globe que l'on fait tourner pour voir les pays. L'utilisateur la compare facilement à une sphère. Si un élément est plus petit qu'un autre, l'utilisateur comprend que la cause en est son éloignement du centre d'intérêt et non sa faible importance. Il est alors naturel d'essayer de rapprocher cet élément si celui-ci paraît intéressant. Toutefois, la vue déforme les rectangles et il est très difficile d'y inscrire un intitulé. De plus il ne s'agit pas d'une vraie sphère. En effet une sphère possède une surface finie et lorsqu'on la fait tourner, le même motif revient périodiquement. Dans cette vue en œil de poisson, les bords de la sphère représentent l'infini et il est possible de visualiser ainsi un très grand nombre d'informations.

La seconde vue de la figure 12 est un raffinement de la première. En tronquant cette sphère sur le devant nous proposons une surface plane à l'utilisateur. Cela permet d'obtenir un focus plus facile à lire (écriture, visualisation d'une figure géométrique,...) mais cela peut provoquer une discontinuité visuelle des informations, lors du défilement.

Enfin, la dernière vue est très spéciale. Elle permet de garder des éléments rectangulaires et de conserver le parallélisme dans la géométrie des informations. Elle occupe un espace

rectangulaire et permet donc une occupation maximale de la fenêtre. Toutefois elle ne repose pas sur une métaphore du monde réel. Elle n'incite donc pas à être manipulée. J'ai qualifiée cette vue de "bi-cylindrique", car elle donne l'impression d'un cylindre vertical lorsque l'utilisateur déplace la souris de haut en bas et d'un cylindre horizontal quand il bouge le pointeur verticalement.

3.4.2.2. L'interface réalisée

L'interface que j'ai réalisée n'intègre pas la lentille magique qui était prévue par faute de temps. Nous avons décidé de tester en priorité la vue en œil de poisson. L'interface réalisée se compose d'une page d'accueil présentée à la figure 13a, et d'une page de résultats sous différentes formes présentées aux figures 13b à 13g.

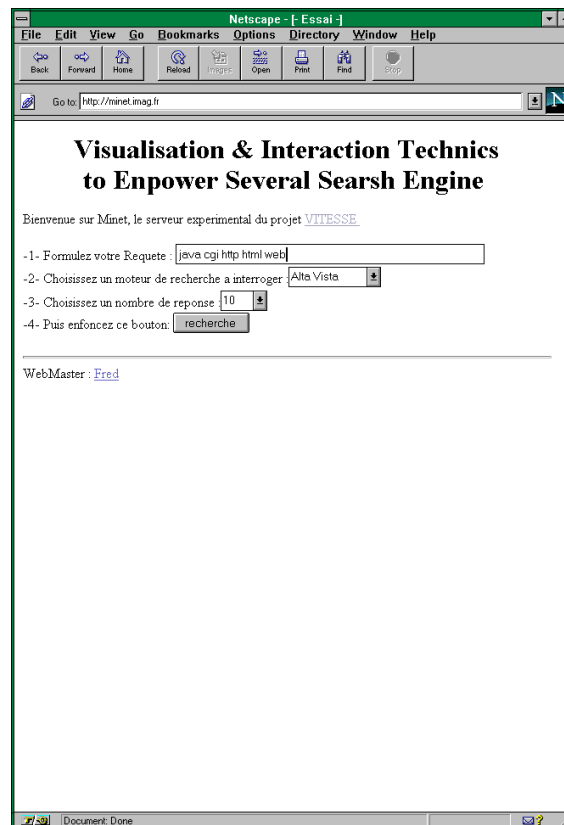


Figure 13a : Page d'accueil de mon système

La page d'accueil permet de remplir les champs nécessaires au fonctionnement du système. L'URL de cette page est : <http://minet.imag.fr> mais cette adresse n'est pas définitive. Outre les mots-clés, l'utilisateur peut saisir le nom du moteur de recherche qu'il souhaite utiliser ainsi que le nombre maximum de données que le système ira récupérer. Lorsque toutes les entrées du système sont valides, l'utilisateur lance la recherche en appuyant sur le bouton « Recherche ».

Les pages de réponses (figures 13b à 13g) ont toutes la même interface (en dehors bien sûr de la technique d'interaction). En haut une ligne d'objets est affichée permettant de configurer la technique d'interaction. La technique d'interaction est au centre, et en dessous se trouvent d'autres outils pour la configuration.

- Dans la partie haute, à gauche, se trouvent quatre cases à cocher qui permettent de passer d'une vue à une autre. A droite de ces cases, se situe un bouton à 2 états permettant de visualiser les liens sous forme de segment ou de triangle. Le triangle étant large vers l'état d'où part le lien et se rétrécit vers l'état d'arrivée. Enfin, à l'extrême droite sur cette ligne se trouve un bouton simple permettant de revenir à la requête. Ce bouton est donc redondant avec le bouton « Back » commun à tous les navigateurs du WWW mais se situe plus près de la zone de travail de l'utilisateur.
- La technique d'interaction affiche les états et les transitions du graphe des URL. Il est possible de déplacer le graphe par manipulation directe avec la souris et de le faire glisser. Au centre de la technique se situe le focus d'interaction qui est coloré en vert (gris clair sur les copies d'écran).
- En bas se trouvent deux boutons à gauche et un champ de texte et un ascenseur à droite. Le premier bouton sert à recentrer la technique d'interaction sur le focus de départ et le second bouton sert à charger la page-réponse correspondante au focus de l'utilisateur. Le champ de texte permet de saisir un nombre pour modifier l'écartement de deux états du graphe et l'ascenseur permet de faire varier la taille des états entre 0 et l'écartement de ces derniers.

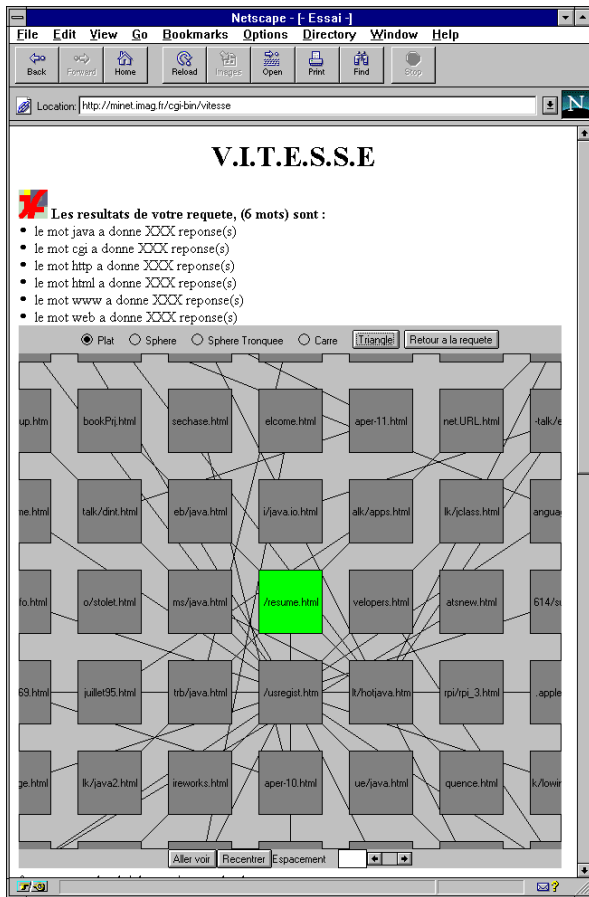


Figure 13b : vue à plat avec les liens en segment

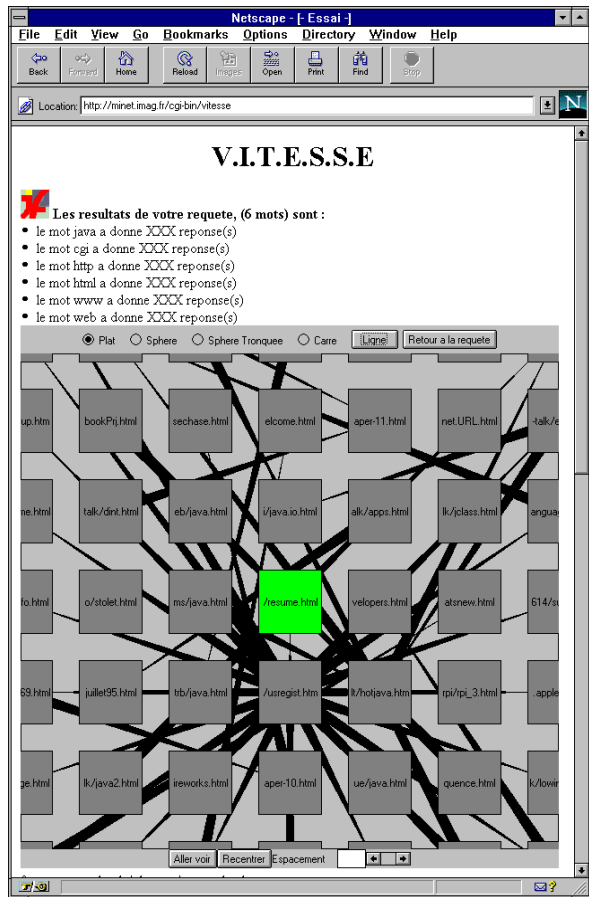


Figure 13c : vue à plat avec les liens en triangle

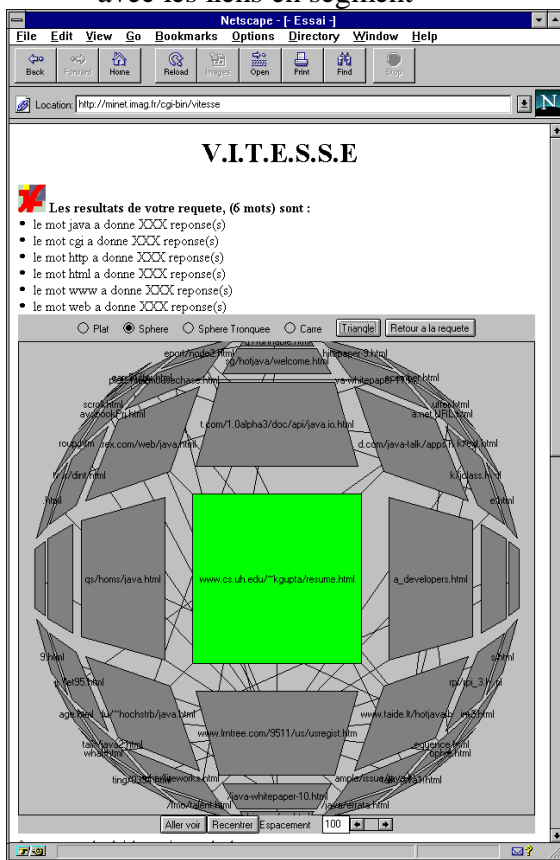


Figure 13d : vue en œil de poisson

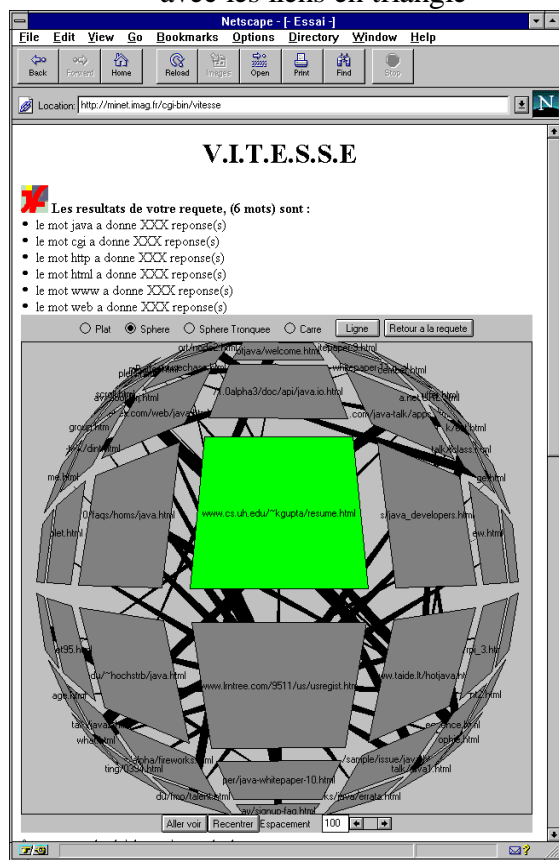


Figure 13e : vue en œil de poisson

avec les liens en segment

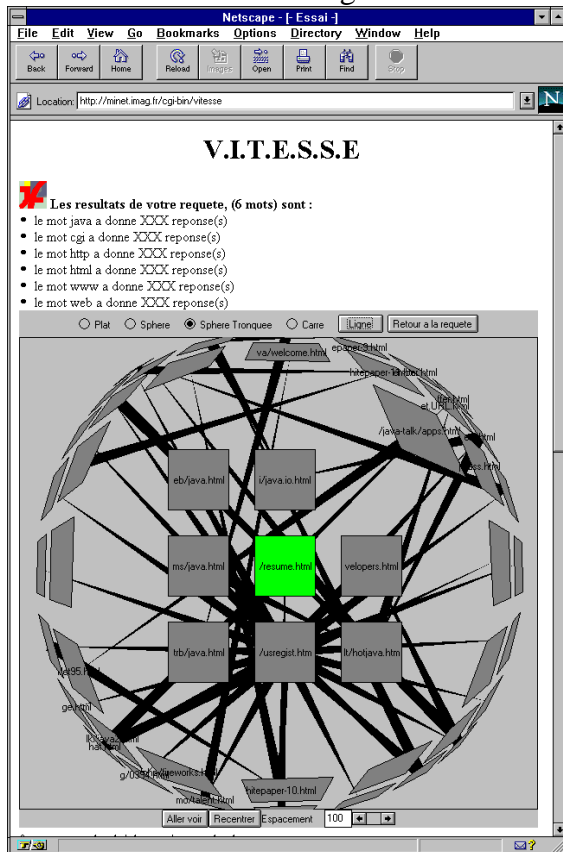


Figure 13f : vue en œil de poisson tronquée avec les liens en triangle

avec les liens en triangle

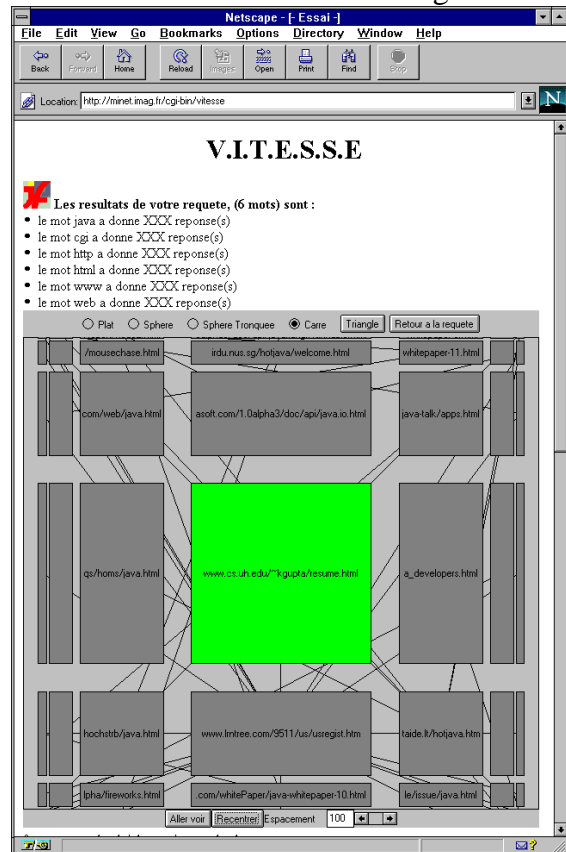


Figure 13g : vue bicylindrique avec les liens en segment

3.5. Réalisation logicielle : spécification globale

Dans ce paragraphe, je détaille ma réalisation logicielle. Celle-ci vérifie le modèle d'architecture PAC-Amodeus. Aussi, au paragraphe 3.5.1. j'explique le modèle PAC-Amodeus puis au paragraphe 3.5.2. son application à ma réalisation.

La réalisation logicielle de l'interface soulève de nombreux problèmes d'architecture qu'il m'a semblé intéressant d'aborder. La source des problèmes est principalement la plate-forme de développement choisie.

Le modèle d'architecture logicielle que j'ai appliqué est le modèle PAC-Amodeus [Nigay 94]. PAC-Amodeus reprend les composants du modèle ARCH [UIMS 92] dont il organise le composant Contrôleur de Dialogue en une hiérarchie d'agents PAC [Coutaz 90].

3.5.1. Les composants du modèle d'architecture PAC-Amodeus

L'utilisateur et le Noyau Fonctionnel produisent et consomment des informations par l'intermédiaire du Contrôleur de Dialogue, le composant clé de voûte de l'architecture.

Le Noyau Fonctionnel maintient les concepts propres au domaine applicatif : les requêtes, les réponses et les algorithmes de recherche.

Le composant Interface du Noyau Fonctionnel a un rôle de traducteur. Ce composant sert donc d'interface conçue pour absorber les changements entre ses voisins directs. Par exemple, des changements dans le noyau fonctionnel provoqueraient uniquement des modifications du composant Interface du Noyau Fonctionnel.

Le Contrôleur de Dialogue est organisé en une hiérarchie d'agents PAC qui forme un pont entre l'Interface du Noyau Fonctionnel et les Techniques de Présentation. Avant de détailler la hiérarchie d'agents nécessaire à notre interface, nous devons préciser les responsabilités de ce composant : enchaînement des tâches, transformation de formalisme et mécanisme de correspondance.

- Une tâche correspond à un objectif de l'utilisateur et définit un ou plusieurs fils de dialogue, reflète sa structuration. L'enchaînement entre les fils n'est pas fixe et dépend du plan opportuniste de l'utilisateur. Cette observation conduit à exploiter l'approche multi-agent en associant un ensemble d'agents à un fil de dialogue.
- Par fonction, l'Interface du Noyau Fonctionnel et les Techniques de Présentation reposent sur des formalismes distincts, le premier dirigé par des considérations opératoires, le second par des techniques de présentation comme celles offertes par une boîte à outils. Les informations en transit dans le Contrôleur de Dialogue sont donc transformées.
- Les changements d'état dans l'Interface du Noyau Fonctionnel doivent être reflétés dans la présentation et vice-versa. Aussi, le Contrôleur de Dialogue maintient des liens entre les objets conceptuels et les objets de présentation.

Le composant Technique de Présentation définit les règles de correspondance entre objets de présentation et objets d'interaction. En sens inverse, ce composant abstrait les informations spécifiées par l'utilisateur dans le formalisme du Contrôleur de Dialogue.

Le composant Bas Niveau d'Interaction désigne la plate-forme d'accueil logicielle et matérielle. Ce niveau regroupe les services d'acquisition, d'estampille et de répartition des événements.

3.5.2. Architecture logicielle de mon interface selon PAC-Amodeus

La figure 14 présente l'architecture logicielle de ma réalisation. Je choisis de découper le système en deux parties : un serveur (partie grisée de la figure 14) et un client (Résultat, Graphe et Palette). Cette décomposition m'a amené à instancier le modèle PAC-Amodeus de la façon suivante :

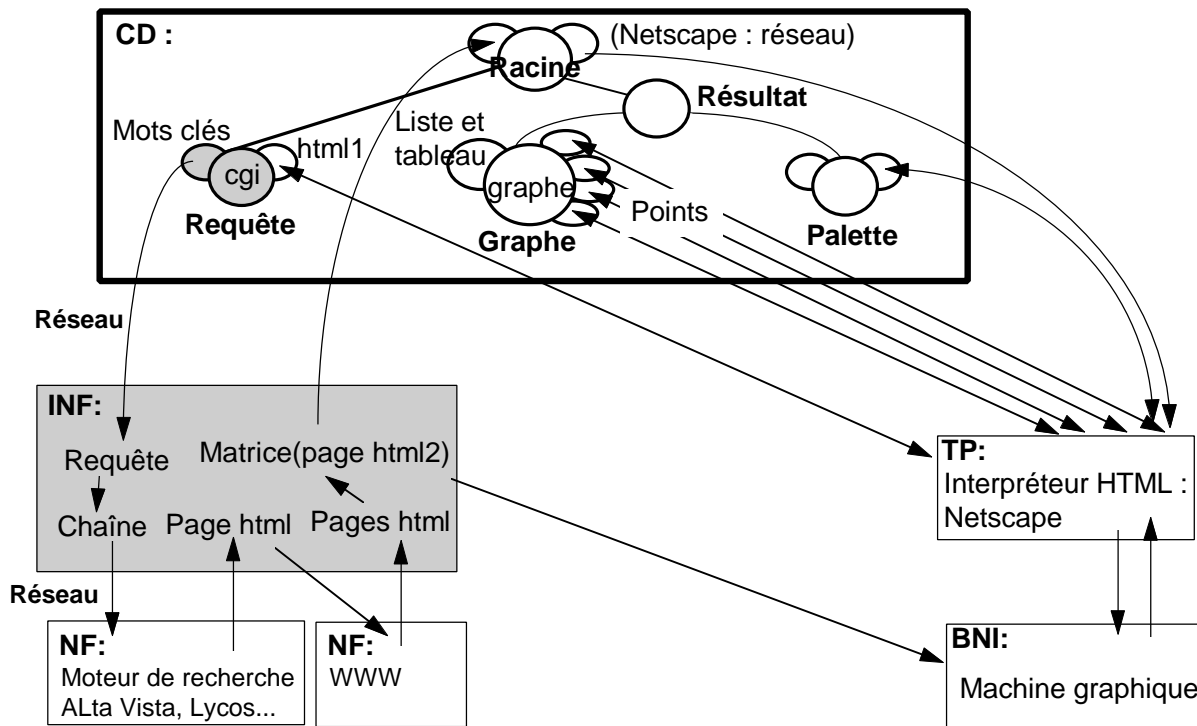


Figure 14 : Architecture logicielle de ma réalisation.

J'ai rencontré de nombreux problèmes dans l'application du modèle PAC-Amodeus pour la conception logicielle de ma solution. Ces problèmes sont principalement dus à la plate-forme logicielle Java, CGI et HTML.

Je considère deux Noyaux Fonctionnels (NF) l'un constitué d'un moteur de recherche et l'autre des pages HTML d'internet. Le moteur de recherche peut être changé par l'utilisateur : AltaVista, lycos, Youpi etc.

L'Interface avec le Noyau Fonctionnel (INF) a un rôle de traducteur. Il est développé en Java et se situe sur la machine du serveur. Les requêtes spécifiées par l'utilisateur sont traduites en une chaîne de caractères transmises au moteur de recherche. Le résultat est une page HTML qui contient la liste des références à des pages HTML. Les pages correspondantes sont retrouvées et une matrice de liens entre pages est construite. L'INF localise les appels à un moteur de recherche dans un composant et rend le reste du code développé indépendant du moteur de recherche (portabilité vers un autre moteur de recherche).

Le Contrôleur de Dialogue (CD) est localisé sur la machine du serveur et sur la machine du client. Le CD est développé en Java (machine client) et en CGI (machine serveur). Il est constitué d'une hiérarchie d'agents. L'agent "Requête" gère la saisie d'une requête. La partie Présentation de l'agent "Requête" est une page HTML. Son Contrôle et son Abstraction sont écrits en CGI. L'agent "Palette", développé en Java, propose un ensemble de boutons à l'écran pour choisir la présentation du graphe. Cet agent gère le mode courant d'affichage. L'agent principal est l'agent "Graphe", écrit en Java. Son Abstraction maintient la matrice représentant le graphe à afficher. Le Contrôle calcule et gère une représentation intermédiaire du graphe avant affichage. Enfin les facettes Présentation correspondent aux quatre techniques d'interaction que j'ai présentées au paragraphe 4.2.2. L'agent "Résultat" est un agent de contrôle sans Abstraction et sans Présentation. Il maintient les liens entre les agents "Graphe" et "Palette". Enfin, l'agent "Racine" gère la communication entre les agents "Résultat" et "Requête".

Le composant Technique de Présentation (TP) correspond à l'interpréteur HTML. Il est localisé sur la machine client. Aucun code n'a été développé pour ce composant.

Le composant Bas Niveau d'Interaction (BNI) dénote la plate-forme logicielle et matérielle. Il englobe la machine graphique. Aucun code n'a été développé pour ce composant.

Le point particulier de cette architecture est la distribution des composants du modèle PAC-Amodeus sur des machines différentes. De plus l'agent "Requête" du Contrôleur de Dialogue est distribué sur deux machines distantes. Mon projet permet donc d'identifier un cas d'instanciation de PAC-Amodeus.

3.6. Réalisation logicielle : spécification détaillée

J'ai réalisé les deux parties du système client-serveur. Elle gère l'interface et les événements utilisateur. Lorsqu'elle en a besoin, elle contacte la partie serveur pour recueillir des informations. La partie serveur se trouve à distance sur une machine dédiée à ce type de tâche.

3.6.1. La partie client

La partie client est celle qui se trouve sur la plate-forme de l'utilisateur final. J'ai choisi le langage JAVA pour réaliser cette partie. En effet JAVA peut se compiler en « applets » et s'insérer dans une page HTML. Je vérifie ainsi le critère de réutilisabilité des données en entrée puisqu'il est très simple d'exécuter du code JAVA avec des données en entrée depuis une page HTML. Cette partie du code maintient la ou les techniques d'interaction. Pour cela, il est nécessaire de tenir compte des exigences de l'utilisateur (configuration du système) et d'établir des connections au serveur pour recevoir les informations demandées par l'utilisateur.

3.6.1.1. Les outils mathématiques pour les vues oeil de poisson

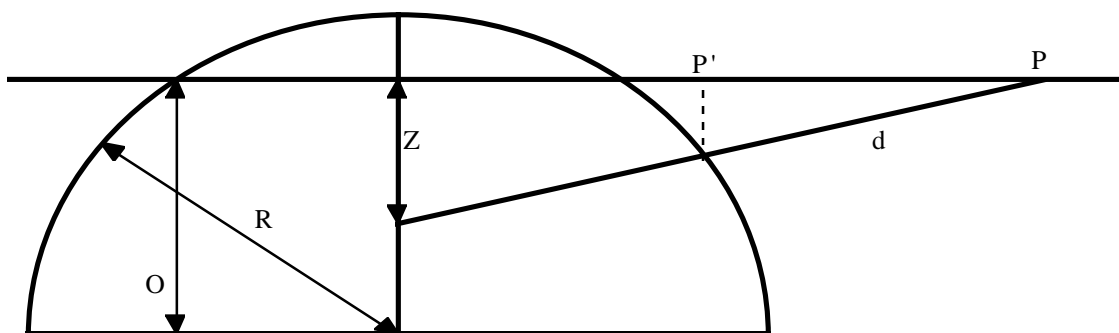


Figure 15 : Transformation du point P en P'

Une vue en œil de poisson est une projection d'un plan sur une sphère. Dans la figure 15, le point P est projeté sur la sphère. Comme l'écran physique n'est pas sphérique, nous devons projeter à nouveau sur le plan. Sur la figure 15, ce dernier point se nomme P'. Grâce aux formules classiques de géométrie, nous pouvons calculer la fonction de transformation de l'abscisse x de P vers l'abscisse x' de P' :

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

Cette formule ne doit pas être calculé à chaque nouvel appel à la fonction de transformation. Certains calculs se repètent :

- $\sqrt{r^2 - (z - o)^2 + z^2(z - o)^2}$
- $z(z - o)$
- z^2

Ces valeurs ne sont donc calculées qu'une seule fois.

Le schéma de la figure 15 et la formule proviennent de [Volkmar 95]. Le schéma est très explicatif et vous pouvez voir comment remplacer la sphère par une autre figure géométrique.

Pour obtenir la formule, il faut calculer l'intersection de la droite « d » et du demi-cercle. Une fois que l'on connaît l'abscisse de ce point il suffit de la projeter sur le plan. Malheureusement l'article mentionné n'indique pas la fonction inverse or celle-ci est nécessaire pour l'interaction puisqu'il va falloir transformer les clics de la souris par la fonction inverse pour savoir où l'utilisateur a cliqué dans la sphère. J'ai donc calculé la formule inverse :

$$x' = \frac{z \times x}{\left| z + \sqrt{r^2 + x^2} - o \right|}$$

Cependant la fonction de l'article n'étant pas bijective, cette fonction inverse ne s'applique que sur l'ensemble d'arrivée de la première fonction. Dans le cas de la vue en œil de poisson, seul les clics à l'intérieur du cercle sont cohérents. De plus les clics sur le bord de la sphère sont un cas limite et peuvent être considérés comme des points très éloignés. Il faut donc tenir compte de cette restriction lors de l'implémentation de la fonction inverse.

3.6.1.2. Les communications en JAVA

Netscape et tous les autres butineurs ne permettent de se connecter qu'avec le site o? a été téléchargé l' « applet » JAVA. Cette restriction impose donc à l'utilisateur de charger son applet sur le serveur qui traitera sa requête. Cette restriction a l'avantage de permettre au serveur d'évoluer puisqu'il est possible de modifier l' « applet » JAVA qui est forcément stocké sur le serveur. Les deux parties du système (client et serveur) peuvent donc évoluer simultanément sans se soucier des mises à jour de la partie cliente. JAVA possède en plus une notion de sockets et de connection à Internet. Il propose même pour cela des objets prédéfinis. Ce langage est donc idéal pour mon système.

3.6.1.3. Les modules mis en œuvre

En JAVA chaque module est représenté par un objet. Le module principal hérite de l'objet « applet » et peut ainsi être contenu dans une page HTML. Dans le code ce module s'appelle « GraphBrowser ». Il fait appelle à deux sous-modules essentiels :

- Un module de configuration contenant les 3 composantes du modèle PAC :

1. Un module de présentation divisé en deux parties (la barre de configuration au dessus de la technique d'interaction et celle en dessous).
 2. Un module abstraction qui gère l'état courant de la configuration
 3. Un module de contrôle qui fait le lien entre les deux premiers modules et qui fait interface avec le reste du code.
- Un module de technique d'interaction lui aussi basé sur l'architecture PAC :

3.6.2. La partie serveur

La partie serveur correspond à un C.G.I. (Common Gateway Interface). Ce type de programme est très souvent écrit en langage C ou en langage de script. Notre système se connecte par le réseau à de nombreux sites et traite des fichiers de texte de tailles importantes. Pour ces raisons, j'ai choisi le langage C qui possède des bibliothèques bien documentées pour ces deux tâches.

3.6.2.1. Les communications sur le réseau

Un premier programme de traitement récupère et traite l'information en provenance du moteur de recherche. Je récupère ainsi les états de mon graphe. Nous pouvons ensuite imaginer que le serveur se connecte aux adresses URL retrouvées par le moteur de recherche afin de compléter les informations : liens hypertextes qui peuvent exister entre les nombreuses pages de réponses. Ce programme peut ainsi effectuer un traitement supplémentaire sur les données recueillies (filtre de langue, vérification de la présence des mots-clés, élimination des serveurs erronés ou trop lents,...). Ce code correspond au composant Interface du Noyau Fonctionnel (INF) décrit au chapitre précédent. Le code de ce traitement n'est pas encore réalisé. J'utilise simplement un générateur aléatoire pour simuler les liens hypertextes.

Un tel programme ralentit le système à cause de la lenteur du réseau. Toutefois si le serveur est une machine spécialement conçue pour ce type de travail (une machine multitâche connectée à Internet par de nombreuses liaisons) l'utilisateur qui se connecterait par modem gagnerait à faire sous-traiter ce travail par le serveur.

De plus un algorithme incrémental de création du graphe permettrait à l'utilisateur de mieux contrôler le processus (l'arrêter, le suspendre, modifier les mots-clés car les premières réponses semblent inadaptées, etc).

C'est le principe du retour d'information le plus immédiat possible. De plus, il est possible de concevoir une parallélisation d'un tel processus surtout entre machines distantes. Par exemple une machine localisée aux USA pourrait se charger de construire le graphe des serveurs de la toile d'araignée américaine, car elle y accéderait plus rapidement, puis transmettrait le résultat partiel à l'utilisateur en Europe.

Enfin ce code doit retourner une page HTML de présentation des résultats qui contiendra entre autres les ancres suivantes : `<APPLET> ... </APPLET>`. L'utilisateur charge alors l'application JAVA sur son système et peut visualiser les réponses.

3.6.2.2. Le traitement des fichiers ASCII

Les communications sur le port 80 que nous venons d'exposer reposent sur des transferts de fichiers ASCII contenant des caractères. Toute l'information contenue dans ces fichiers n'est pas

pertinente pour mon système. Un traitement consiste donc à rechercher et extraire l'information pertinente de ces fichiers. Pour cela j'ai utilisé un traitement séquentiel des fichiers. Il était possible d'introduire un analyseur lexical et un analyseur syntaxique du langage HTML mais c'était une solution trop lourde pour le traitement simple que nous avons à réaliser. En effet le langage HTML est composé d'ancres qui se trouvent entre les signes « < » et « > ». Par exemple le tag définit l'insertion d'une image à cet endroit de la page du WWW. Il est donc possible d'utiliser la grammaire de ce langage pour traiter les données.

Malheureusement, les pages HTML ne répondent pas toutes aux mêmes exigences. Si certaines suivent scrupuleusement la grammaire du langage, la majorité des pages HTML sont écrites en HTML approximatif. La faute en incombe aux butineurs qui sont suffisamment tolérants pour visualiser ces pages mal écrites. De plus, peu d'auteur en HTML utilisent les logiciels adéquats qui permettent d'écrire du HTML grammaticalement correcte.

J'ai donc décidé de ne pas utiliser de grammaire HTML mais de rechercher séquentiellement des repères (chaînes de caractères qui identifient un passage particulier) du texte à traiter. Cette méthode ne me garantit pas de pouvoir utiliser tous les moteurs de recherche mais elle s'est avérée suffisante pour les cinq moteurs de recherche les plus utilisés. Cette méthode exploite un fichier contenant les repères pour chacun des moteurs de recherche que je souhaite utiliser.

3.6.2.3. Les modules mis en œuvre

- le programme « VITESSE », le programme principal dont le sigle signifie : Visualization and Interaction Techniques to Empower Several Search Engine qui peut se traduire en français par techniques d'interaction et de visualisation pour améliorer les moteurs de recherche,
- l'en-tête « Global », définissant les structures de données et les variables communes aux autres modules du serveur,
- le module « KeyWords », gestion des mots-clés (stockage, fonctions d'ajout, suppression, etc),
- le module « StringFilter », fonctions d'analyse de chaînes de caractères adaptées au langage HTML et au moteur de recherche *,
- le module « WebConnect », fonctions de connexion, manipulation des sockets UNIX adaptées au moteur de recherche *.

* Adapté au moteur de recherche signifie que les fonctions du module utilisent les données contenues dans un fichier appelé SE.list qui contient une liste de moteurs de recherche ainsi que leurs caractéristiques.

3.6.3. Le couple client-serveur

Le modèle client serveur permet d'avoir des traitements sur la machine de l'utilisateur et sur le serveur. Macintosh offre depuis peu un logiciel appelé Project X [PROJECTX 96] qui tourne seul sur la machine client. Mais déjà une nouvelle version est sortie où lorsque le client demande des informations sur certains sites d'Internet, le logiciel se connecte à ce serveur pour recevoir des informations complémentaires. Le modèle client-serveur offre avantages et inconvénients.

3.6.3.1. Avantages

Les principaux avantages identifiés de ce modèle sont :

- la décharge d'une partie du travail et des données de la machine client vers le serveur,
- l'inutilité de mise à jour des données (le moteur de recherche en est responsable),
- la possibilité pour le client de se connecter à plusieurs serveurs concurrents,
- la liberté du client qui n'a plus à se préoccuper des évolutions des moteurs de recherche (adresse, mode d'emploi,...) puisqu'il suffit qu'une personne se charge de maintenir à jour le fichier SE.list,
- la possibilité de recevoir des informations de façon continue (bien que cela ne soit pas implémenté dans notre système, JAVA permet de manipuler des données qui arrivent au fur et à mesure contrairement aux pages HTML classiques),
- la possibilité de facturation du service (grâce à un porte-monnaie électronique).

3.6.3.2. Inconvénients

Parmi les inconvénients de ce modèle, je retiendrai les quatre points suivants :

- la lenteur de l'accès aux données par Internet par rapport à l'accès au disque dur,
- la possibilité de se faire espionner par le serveur,
- l'augmentation du risque de panne due au rajout d'un intermédiaire entre la source d'information et l'utilisateur,
- la suppression des publicités qui font vivre les moteurs de recherche. Cela risque de poser un problème aux organismes qui distribuent gratuitement leur moteur de recherche à des fins publicitaires.

Chapitre 4

Conclusion et perspectives

4.1. Bilan du travail réalisé

Mon étude a consisté à étudier et comparer les techniques d'interaction pour une grande quantité d'information. Pour une réalisation logicielle ultérieure, j'ai étudié les moteurs de recherche du WWW qui rendent une grande quantité d'information en réponse à une requête. Les moteurs de recherche du WWW constituent mon domaine applicatif. Mon approche a été centrée sur l'utilisateur. Après avoir étudié les problèmes d'utilisation des moteurs de recherche du WWW, j'ai choisi un ensemble de critères d'ergonomie non satisfaits qui me semble important. Ces critères concernent le cas d'une grande quantité d'information et ne sont pas dépendants de mon domaine applicatif. J'ai ensuite étudié les techniques d'interaction existantes dans la perspective des critères choisis. J'ai ensuite proposé une solution d'interface et donc de technique d'interaction qui vise à vérifier les critères d'ergonomie identifiées lors de mon étude bibliographique. Après avoir conçu l'interface, j'ai développé le système. Le système est développé et les objectifs du stage sont donc atteints.

4.1.1. Les méthodes et modèles appliqués

Cette étude et réalisation m'a permis d'appliquer des modèles et outils relevant du domaine des IHM.

Tout d'abord, j'ai appliqué une méthode de conception centrée sur l'utilisateur qui m'a permis de prendre en compte les critères d'ergonomie identifiés lors de l'étude. L'interface conçue vise à vérifier ces critères.

Pour la réalisation logicielle j'ai appliqué un modèle d'architecture logicielle PAC-Amodeus. L'application du modèle n'a pas été immédiate et a permis d'identifier des problèmes intéressants. PAC-Amodeus a été appliqué dans de nombreux systèmes interactifs mais c'est la première fois que le modèle est appliqué dans le cas de la plate-forme logicielle choisie : Java et C.G.I. Une architecture logicielle se devant d'organiser tout le code du système, l'interpréteur HTML et l'interface de Netscape ont dû être intégrés à l'architecture.

4.1.2. Les vues en oeil de poisson

L'utilisation du système au sein de l'équipe m'a permis de constater qu'aucune des trois vues en œil de poisson n'est préférée. Elles me semblent donc être toutes nécessaires. De plus la vue sans déformation que j'ai développée initialement pour m'aider dans la mise au point semble aussi être utilisable. Je recommande donc de garder aussi cette représentation. De plus, elle me permettra aussi à vérifier lors de tests en vraie grandeur si les utilisateurs utilisent les vues en oeil de poisson.

Il s'agit ici de résultats très préliminaires et seule une évaluation expérimentale à grande échelle me permettra de conclure sur la nécessité des vues multiples. Les vues qui ont été réalisées peuvent être comparées à d'autres techniques d'interaction [Rao 96] tel que le « Perspective Sheet » ou bien le « Document Lens ». Les critères d'ergonomie sont certes vérifiés mais peut-être pas nécessaires.

4.1.3. L'interface

L'interface elle-même m'a amené à des constatations intéressantes.

En effet la présence de boutons permet entre autre de différencier l'« applet » JAVA d'une simple image dans la page WWW. Ces boutons incitent donc l'utilisateur à essayer de manipuler les données et pas simplement à sélectionner celles qui sont présentes lors du chargement.

La vue en oeil de poisson en forme de sphère est sans doute celle qui incite le plus à être manipulé. C'est donc celle qu'il convient de présenter par défaut à l'utilisateur.

D'autres composants viendront s'ajouter à l'interface pour l'évaluation expérimentale. J'envisage une zone à l'écran pour configurer la vue en oeil de poisson et ainsi connaître les préférences de l'utilisateur. Je propose de rajouter ces suppléments en-dessous de la zone de travail : l'utilisateur doit pouvoir utiliser les données sans avoir à faire coulisser l'ascenseur de la fenêtre vers le bas. L'objectif est bien sûr de réduire le nombre d'actions de l'utilisateur dans sa tâche principale, rechercher de l'information.

4.1.4. L'utilisabilité générale

L'utilisabilité du système paraît correcte notamment grâce à la réutilisabilité des données en entrée et en sortie. Elle permet d'utiliser le système avec un minimum de contraintes. Mais pour évaluer l'utilisabilité d'un système, il faut procéder à une incontournable évaluation expérimentale.

4.2. Perspectives

La phase suivante consiste à finaliser ma réalisation logicielle et à évaluer expérimentalement ma solution.

- Les évolutions du système s'organisent selon trois axes :
 1. ajouter la lentille magique par dessus la vue en oeil de poisson,
 2. étoffer l'information avec des renseignements tels que la date, le titre de la page, les liens entre les pages ou la langue dans laquelle est écrite la page,
 3. améliorer la rapidité du système en utilisant un serveur plus performant, en lançant le système avant que toutes les informations aient été récoltées ou en utilisant un moteur de recherche qui puisse renvoyer plus de dix réponses à la fois.

- Mon application est déjà accessible par le WWW ce qui me permet d'envisager une grande quantité d'évaluateurs. Je prévois d'utiliser le CGI pour capturer les actions de l'utilisateur et de les étudier en détail pour orienter les futures évolutions de ce logiciel. L'évaluation

expérimentale via le réseau pose de nouveaux problèmes, car l'utilisateur et son contexte d'utilisation ne sont pas accessibles. Par exemple un arrêt de l'activité au milieu d'une expérimentation (détecté grâce à la capture des actions) sera difficile à interpréter : s'agit-il d'un temps de réflexion ou d'une pause café ? De plus il me semble important d'expliquer aux utilisateurs qu'ils sont épiés et surveillés à distance. Enfin, il faudra tenir compte de la diversité des utilisateurs et pour cela diffuser en dehors du campus l'adresse WWW d'expérimentation.

Références bibliographiques

- [ALTAVISTA] : <http://www.altavista.digital.com/>
- [Coutaz 90] : Joëlle Coutaz
Interfaces homme-machine :conception et réalisation
DUNOD informatique
- [ECILA] : <http://ecila.ceic.com>
- [Nigay 94] : Laurence Nigay
Conception et Modélisation logicielles des systèmes interactifs
Thèse de l'Université Joseph Fourier
- [Rao 96] : Ramana Rao
Quand l'information parle à nos yeux
Article tiré de « La Recherche » mensuel de Mars 1996 (p. 66-73)
- [Foley 95] : Sougata Mukherjea and James D. Foley
Showing the context of nodes in the World Wide Web
Article tiré de CHI'95 Conference companion de 1995 (p. 326-327)
- [PROJECTX 96] : Logiciel ProjectX d'Apple téléchargeable sur internet à
<http://www.atg.apple.com/go/projectx/Download/Download.html>
- [UIMS 92] : The UIMS Workshop Tool Developers : A Metamodel for the
Runtime Architecture of an Interactive System,
SIGCHI Bulletin, 24, 1 (p. 32-37)
- [Sarkar 92] : Manojit Sarkar and Marc H. Brown
Graphical fisheye views of graphs
Article tiré du CHI'92 Conference proceedings de 1992 (p. 83-91)
- [Unger 92] : J. Larson and C. Unger
Engineering for Human-Computer Interaction
Livre édité chez North Holland I.F.I.P (423 pages)
- [Volkmar 95] : Volkmar Hovestadt, Oliver Gramberg et Oliver Deussen
Hyperbolic user interface for computer aided architectural design
Article tiré du CHI'95 Conference companion de 1995 (p. 304-305)
- [Ware 95] : Colin Ware and Marlon Lewis
The DragMad Image Magnifier
Article tiré du CHI'95 Conference companion de 1995 (p. 407-408)
- [YAHOO] : <http://www.yahoo.com>

Glossaire

Abréviations

C.G.I.	: Common Gateway Interface
C.H.I.	: Computer-Human Interaction. (voir IHM)
C.L.I.P.S.	: Communication Langagière et Interaction Personne-Système
G.I.F.	: Graphic Interchange Format
H.C.I.	: Human-Computer Interaction (voir IHM)
H.T.M.L.	: HyperText Markup Language
I.H.M.	: Interaction Homme-Machine (en anglais “HCI” ou “CHI”)
I.I.H.M.	: Ingénierie de l’interaction homme-machine
I.M.A.G.	: Informatique et Mathématiques appliquées de Grenoble
S.R.I.	: Systèmes de Recherche d’Informations
W.W.W.	: World Wide Web (toile d’araignée mondiale)

Marques commerciales

- Sun et JAVA sont des marques déposées ou des marques commerciales de Sun microsystem.
- Microsoft et Internet Explorer sont des marques déposées ou des marques commerciales de Microsoft.
- Netscape est une marque déposée de Netscape Communications Corporation.
- AltaVista est un sigle déposé de Digital Equipment Corporation.
G.I.F. est un sigle déposé de CompuServe Incorporated.