Représentations Multiples d'une Grande Quantité d'Information

Frédéric Vernier et Laurence Nigay

Laboratoire CLIPS-IMAG,
Equipe IIHM
BP 53 38041 Grenoble, France
{frederic.vernier, laurence.nigay}@imag.fr

RÉSUMÉ

Cet article a trait à la conception et à la mise en œuvre de techniques de visualisation et plus généralement d'interaction dans le cas d'une grande quantité d'informations. Nous proposons d'appliquer une méthode de conception descendante fondée sur une étude de l'utilisateur et de son activité. La variété des cas d'utilisation nous incite à raisonner à un plus haut niveau d'abstraction en considérant des critères d'ergonomie. L'un des critères, la représentation multiple, nous semble crucial en vue de la diversité des utilisateurs et des tâches à accomplir. En présentant notre technique de visualisation à plusieurs facettes, nous énonçons des règles heuristiques de conception d'un ensemble organisé de représentations. Nous examinons ensuite la réalisation logicielle, dernière étape de la méthode.

MOTS CLÉS : Grande quantité d'informations, Représentation multiple, Technique de visualisation et d'interaction.

INTRODUCTION

Les récentes avancées des technologies de la communication ouvrent de nouvelles perspectives. En particulier, les autoroutes de l'information permettent une diffusion instantanée d'une grande quantité d'informations. Mais cette abondance à l'échelle planétaire nourrit en même temps le risque de détruire l'information : cette dernière existe mais elle devient inaccessible. Le problème n'est plus tant la disponibilité de l'information mais la capacité de sélection d'une information qui réponde au besoin. Les travaux récents sur la télévision interactive illustrent parfaitement le problème : une grande quantité d'informations doit être accessible à tout public et tenir compte de la variété des conditions d'usage à domicile. A l'évidence, il convient d'assister les utilisateurs face à cette grande quantité d'informations. Plusieurs approches sont envisageables, comme l'amélioration de l'efficacité des systèmes de recherche d'information afin de réduire la taille de l'espace informationnel. Dans ce papier, notre approche est centrée sur l'utilisabilité et nous visons la conception de formes d'interaction efficaces dans le cas d'une grande quantité d'informations. Nous appliquons une méthode de conception descendante centrée sur l'utilisateur. Pour cela nous identifions des critères d'ergonomie qui servent de guide à la conception.

L'un des critères, la représentation multiple, nous semble important en vue de la diversité des utilisateurs, des situations d'utilisation et des tâches à accomplir. Nous présentons des règles heuristiques pour la mise en œuvre du critère de représentation multiple et les illustrons avec notre technique d'interaction à plusieurs facettes. La réalisation logicielle de ce critère est ensuite expliquée selon le modèle PAC-Amodeus.

METHODE DE CONCEPTION

La tâche de conception d'une interface peut se décomposer en cinq sous-tâches essentielles [3] :

- 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.

Cependant ces étapes de conception sont difficiles à appliquer dans le cas général d'une grande quantité d'informations. En particulier identifier les tâches-utilisateur sans considérer un système particulier semble très ardu si l'on vise à l'exhaustivité. L'analyse commencée lors d'une réunion du groupe de travail FADIVA confirme la difficulté de l'entreprise [12]. Des exemples de tâches sont les suivants :

- trouver un ensemble d'éléments par exploration,
- trouver un ensemble d'éléments à partir de propriétés,
- retrouver un ensemble d'éléments,
- identifier une tendance dans l'ensemble,
- comparer des éléments.

Il est de plus très difficile d'évaluer l'adéquation d'une technique de visualisation à une tâche de l'utilisateur [24]. Une approche consiste alors à raisonner au niveau de l'interaction, et à considérer que quelle que soit la tâche à accomplir, l'utilisateur devra naviguer dans l'espace. Partant de ce constat, nous étudions la navigation pour identifier des critères d'ergonomie que doivent satisfaire les techniques d'interaction à concevoir.

NAVIGATION

Au contraire de l'exploration qui consiste à découvrir un nouvel espace, la navigation implique un objectif à atteindre, comme trouver une information ou un ensemble d'informations. Du point de vue de l'utilisateur, la navigation soulève deux questions [4] :

- Où suis-je ?
- Où est ... ? comment aller à ...? ou encore Existe-t-il ... quelque part dans l'espace ?

Les deux premières questions soulignent l'activité spatiale qu'implique la navigation (les chemins et les lieux). La question "comment aller à ..." identifie outre l'aspect spatial (la direction et le chemin à emprunter) un autre paramètre important : le moyen de locomotion. De plus la dimension temporelle joue un rôle primordial [15]. Le temps et l'espace sont deux ingrédients en intime relation qui caractérisent la navigation. La Figure 1 organise selon deux dimensions les caractéristiques de la navigation.

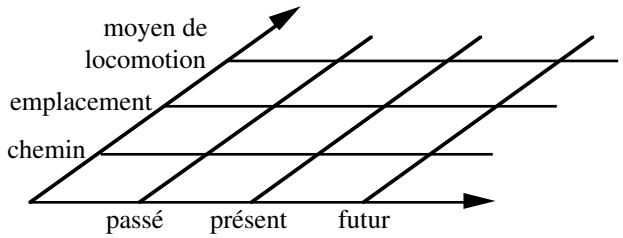


Figure 1 : *Caractéristiques de la navigation.*

Au paragraphe suivant nous exploitons ces caractéristiques pour identifier des critères d'ergonomie, premier pas vers la conception de techniques d'interaction favorisant la navigation.

CRITERES D'ERGONOMIE

Dans [1], nous proposons un guide de conception sous la forme d'un espace organisé de critères d'ergonomie. Nous y distinguons les critères qui contribuent à la robustesse de l'interaction de ceux participant à la souplesse. Dans chacune de ces deux classes, nous avons sélectionné les critères les plus pertinents pour la conception d'une technique d'interaction favorisant la navigation dans une grande quantité d'informations.

Robustesse : les Critères

La robustesse de l'interaction reflète la capacité de l'interface à permettre à l'utilisateur d'atteindre avec succès un objectif visé. Parmi les critères de robustesse, trois d'entre eux semblent cruciaux en regard des caractéristiques de la navigation :

- l'observabilité,
- l'honnêteté,
- la prévisibilité.

L'observabilité est la capacité du système à rendre perceptibles à l'utilisateur les variables d'état internes pertinentes pour la tâche en cours. En nous reposant sur les caractéristiques de la navigation (Figure 1), nous identifions plusieurs variables internes à rendre perceptibles telles que :

- l'ensemble des informations et leurs liens (les emplacements et les chemins),
- la taille de l'espace d'information,
- le temps d'accès à un emplacement qui constitue une méta-information (moyen de locomotion),
- le centre d'intérêt (l'emplacement courant),
- le voisinage du centre d'intérêt ("où suis-je ?").
- les emplacements visités ("où étais-je ?").

Dans [7], l'observabilité de l'ensemble des informations est affinée en deux règles définissant des contraintes que doit vérifier la technique d'interaction :

> R1 : Le nombre de liens partant d'un emplacement doit être "petit" par rapport à la taille de l'espace.
>
> R2 : La distance entre deux emplacements doit être "petite" par rapport à la taille de l'espace.

La deuxième règle (R2) implique que l'utilisateur doit pouvoir atteindre un emplacement en un nombre minime d'actions : en d'autres termes la longueur de la trajectoire d'interaction doit être courte. R2 correspond à notre critère d'"atteignabilité" (en Anglais "reachability") qui contribue à la robustesse de l'interaction [1]. Dans [7] nous trouvons également des règles d'observabilité liées à la navigation telles que :

> R3 : A partir d'un emplacement, l'observabilité des chemins corrects vers un emplacement à atteindre est garant d'une navigation efficace.

Cette règle est actuellement transgressée dans le WWW car la présentation d'un lien (libellé) indique uniquement l'emplacement d'arrivée, qui n'est pas nécessairement celui recherché.

L'honnêteté est un affinement de l'observabilité. Le système est dit honnête si 1) le rendu de l'état observable est conforme à l'état interne et si 2) la forme de ce rendu conduit l'utilisateur à interpréter correctement cet état. L'honnêteté devient un critère fondamental lorsque le rendu de l'espace d'informations exploite une technique déformante comme la vue en œil de poisson [21]. Pour que l'utilisateur puisse naviguer efficacement, sa carte mentale doit être conforme à l'espace des données maintenues par le système.

La prévisibilité est vérifiée lorsque l'utilisateur est capable de prédire le résultat de ses actions. Dans le cas de la navigation dans une grande quantité d'informations, l'utilisateur doit pouvoir prédire l'emplacement qu'il va atteindre suite à une action. La cohérence des moyens de locomotion ou techniques d'interaction en entrée est un facteur qui contribue au critère de prévisibilité.

Souplesse : les Critères

La souplesse de l'interface reflète l'éventail des choix offerts à l'utilisateur. Elle constitue une solution pour

répondre à la diversité des utilisateurs et des tâches. Nous avons également choisi trois critères liés à la souplesse :

- la réutilisabilité des données d'entrée et de sortie,
- l'adaptabilité,
- la représentation multiple.

La réutilisabilité des données d'entrée et de sortie caractérise la capacité du système à permettre à l'utilisateur d'exploiter les sorties comme des entrées et vice versa. Lorsque l'interface vérifie ce critère, elle répond au mode action [11], et présente à l'utilisateur un espace sur lequel il lui est possible d'agir directement. Une telle technique d'interaction offre donc le sentiment d'engagement direct dans l'action sans relais intermédiaire (interface "à la première personne" [13]). Il est important de noter que la manipulation directe ne repose pas nécessairement sur une métaphore du monde réel. Par exemple des moyens de locomotion qui ne correspondent pas à ceux du monde réel sont à concevoir. Dans [5] de telles techniques, qui débordent du cadre de la métaphore du monde réel, sont caractérisées de magiques. Ces dernières n'excluent pas la manipulation directe.

L'adaptabilité désigne la capacité de l'interface à être adaptable par l'utilisateur. Ce critère est important pour répondre à la variété des utilisateurs et de leurs tâches. Une forme particulière de l'adaptabilité est la représentation multiple de l'espace.

La représentation multiple désigne la capacité de l'interface à rendre perceptible une variable interne selon plusieurs modalités de sortie. Nous l'appliquons ici à la représentation de l'espace des informations suivant des modalités différentes. Pour définir une modalité, nous distinguons le niveau physique (ou matériel) qui fait appel à la perception [17] et le niveau représentationnel qui fait appel à la cognition. Un ensemble de couples <d, s> où d désigne un dispositif physique et s, un système représentationnel, est une façon de caractériser les échanges entre un système et son utilisateur [17]. Dans la suite de l'article, nous utiliserons le terme visualisation qui désigne une modalité de sortie <d=écran, s=fonction de visualisation>. Plusieurs visualisations du même espace d'informations est donc un cas de multimodalité séquentielle en sortie.

Au paragraphe suivant nous présentons des règles heuristiques de conception liées au critère de représentation multiple : la discussion est illustrée par notre technique de visualisation à plusieurs facettes. Le dernier paragraphe intitulé "Réalisation logicielle" concerne la mise en œuvre logicielle de ce critère selon le modèle PAC-Amodeus.

REPRESENTATION MULTIPLE

La représentation multiple d'une grande quantité d'informations vise à mettre en évidence différentes caractéristiques de l'espace pour répondre à la diversité des tâches. Plusieurs présentations de la même information permettent d'effectuer des tâches différentes dans les meilleures conditions[1] [24]. Nous distinguons la représentation multiple de l'espace de celle qualifiée d'"interactive" où l'utilisateur peut filtrer les données à présenter, comme dans le système "Attribute Explorer" [23]. Contrairement à la représentation multiple, la représentation "interactive" n'implique pas la multimodalité puisque l'espace est présenté selon une seule modalité de sortie et seules varient les données présentées. Représentation multiple et représentation "interactive" de l'espace sont deux critères orthogonaux qui visent à faire face à la multiplicité des tâches.

La représentation multiple est un critère largement utilisé dans les interfaces graphiques. Par exemple le logiciel Word de Microsoft intègre trois modes de présentation du texte : mode normal, page et plan. Cependant peu de systèmes satisfont ce critère dans le cas d'un large espace. Ce constat peut s'expliquer par le coût de développement d'une technique de visualisation de grande quantité d'informations. De plus certains systèmes intègrent la représentation multiple à des fins d'apprentissage et cela de façon transitoire. Par exemple le système de gestion de fichier de Windows 95 intègre l'ancienne présentation de DOS (commande "DIR") et de Windows 3.1 (arborescence). Nous étudions ici la représentation multiple d'une grande quantité d'informations afin d'augmenter l'utilisabilité de l'interface. Aussi la représentation multiple ne doit pas augmenter la complexité d'utilisation. Pour cela nous définissons trois règles de conception :

- la facilité de changer de visualisation à tout moment,
- la continuité visuelle lors du passage d'une visualisation à une autre,
- la réutilisabilité des outils de navigation.

Le changement de visualisation définit une tâche articulatoire qui ne modifie pas l'état interne de l'interface. Il convient donc que cette tâche soit facile à accomplir. De plus l'interface doit inciter (principe d'"affordance") l'utilisateur à changer de visualisation selon sa tâche afin d'exploiter au mieux la représentation multiple.

La continuité, qu'elle soit visuelle, auditive ou encore tactile, est liée à la navigation. Il est crucial que l'utilisateur ne soit pas perdu dans l'espace suite à un changement de visualisation. Par exemple dans [9] lorsque l'utilisateur sélectionne un nœud, le carrousel et son ombre s'animent pour amener le nœud sélectionné au premier plan (continuité visuelle). La charge d'évaluation du nouvel état est alors déporté au niveau du processeur sensoriel réduisant ainsi la charge du processeur cognitif. Nous appliquons ici ce principe de continuité lors du passage d'une visualisation à l'autre.

La troisième règle définit des contraintes de cohérence au niveau de l'interaction. Afin d'une part de favoriser l'apprentissage et d'autre part d'augmenter la

[1] Dans le texte original : "Different presentations of the same information best support different tasks".

prévisibilité, les outils de navigation (moyens de locomotion) doivent être cohérents d'une représentation à l'autre.

Au paragraphe suivant nous illustrons le critère de représentation multiple en expliquant notre technique de visualisation à plusieurs facettes. Nous abordons ensuite sa réalisation logicielle en montrant les implications des règles de conception sur l'organisation du logiciel.

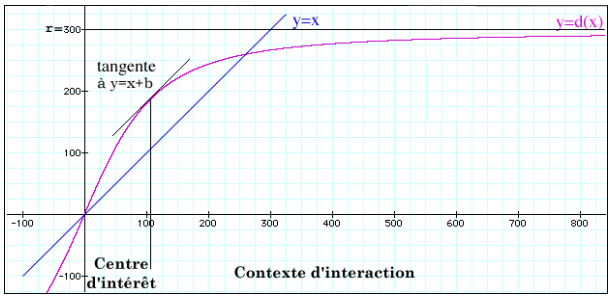
TECHNIQUES DE VISUALISATION

Notre technique de visualisation vise à satisfaire le critère d'observabilité :

- de l'espace ("où j'étais, où je vais"),
- du centre d'intérêt et de son voisinage ("où je suis").

Ces objectifs sont communs à de nombreuses techniques de visualisation telles que la vue en œil de poisson [21, 6], le mur en perspective [15], ou encore le carrousel [20].

Notre technique de visualisation repose sur une fonction de déformation de l'espace présentée à la figure 2 [10]. Cette fonction d constitue le point de départ de notre conception.



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

$$d^{-1}(x) = x.\frac{z}{\sqrt{r^2-x^2}+(z-o)}$$

Figure 2 : *Fonction de déformation.*

Dans les équations de la Figure 2, x désigne la distance en pixels que l'on cherche à déformer, tandis que r, o et z sont les paramètres de la déformation.

r désigne la valeur limite de la fonction. Par exemple, à la Figure 2, la fonction de déformation d(x) est comparée à l'identité avec r fixé à 300 pixels. r fixe donc la surface de visualisation à l'écran : ainsi dans la visualisation sphérique de la Figure 3-a, r désigne le rayon de la sphère. o et z sont les paramètres qui déterminent la puissance de l'effet de "loupe". A la Figure 2, o et z ont respectivement la valeur 150 et 112 pixels.

A partir de la fonction de déformation, d(x), nous calculons la fonction inverse $d^{-1}(x)$ (Figure 2). Lorsque l'utilisateur interagit avec l'espace des informations, la fonction inverse permet de calculer la position des clicks souris dans l'espace non déformé. Nous utilisons aussi cette fonction inverse pour définir de nouvelles visualisations de l'espace comme celle de la Figure 4-b.

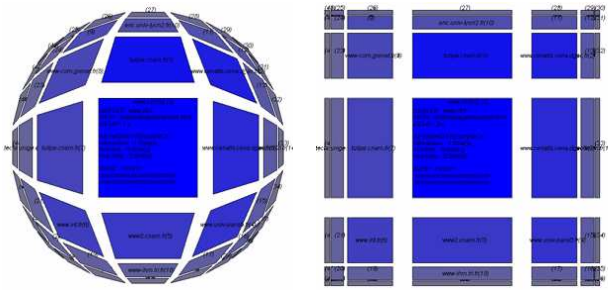
Nous expliquons d'abord comment appliquer la fonction d pour définir plusieurs visualisations de l'espace puis introduisons de nouvelles fonctions de déformation obtenues par composition de fonctions.

Calcul de la Visualisation à partir de la Fonction de Déformation

Comme le montre la Figure 3, la fonction de déformation peut être appliquée aux coordonnées polaires (α,di) (Figure 3a) des emplacements de l'espace initial ou encore aux coordonnées cartésiennes (x,y) (Figure 3b).

Dans le cas de la déformation polaire, l'angle est conservé et seule la distance di est transformée. Intuitivement cette visualisation (Figure 3-a, coordonnées polaires) nous semble mieux satisfaire le critère d'honnêteté en vue de l'analogie avec une demi-sphère. Elle s'apparente à une métaphore du monde réel, comme lorsque l'on fait tourner un globe.

La déformation cartésienne quant à elle conserve l'alignement en x et en y de l'espace initial. Ainsi si l'espace initial contient des lignes ou des colonnes, celles-ci seront perçues par l'utilisateur (perception par groupements d'objets selon la théorie de psychologie ICS [16]).

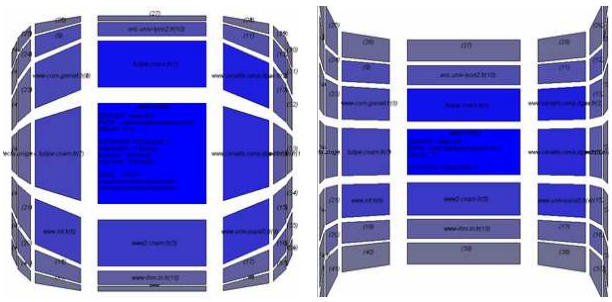


(a) $(\alpha,di) \rightarrow (\alpha,d(di))$ (b) $(x,y) \rightarrow (d(x),d(y))$

Figure 3 : *Déformations polaire et cartésienne.*

De nouvelles visualisations sont obtenues en combinant les deux approches précédentes. Par exemple la Figure 4-a représente l'espace déformé obtenu en appliquant une déformation cartésienne sur l'axe des abscisses et une projection de la déformation polaire sur l'axe des ordonnées. On obtient alors une visualisation qui offre les avantages des deux approches : métaphore du monde réel tout en respectant l'alignement des colonnes de l'espace de départ.

Une autre approche consiste à appliquer la fonction de déformation et sa réciproque. Bien que la fonction réciproque aille à l'encontre de l'observabilité du centre d'intérêt (effet de loupe inverse), sa combinaison avec la fonction de déformation offre l'avantage de rendre plus observable le voisinage du centre d'intérêt. L'exemple de la Figure 4-b, est obtenu en combinant d (cartésien) sur l'axe des abscisses et une fonction dérivée de d^{-1} (polaire) sur l'axe des ordonnées. Alors que le centre d'intérêt est présenté de façon moins précise qu'à la Figure 4-a, son voisinage est plus observable.



(a) Déformations cartésienne en x, et polaire en y.

(b) Déformations cartésienne en x, et polaire inverse en y.

Figure 4 : *Déformations différentes sur chaque axe.*

Composition de Fonctions de Déformation

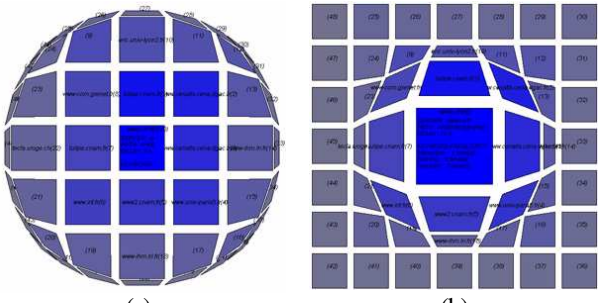
A partir de deux fonctions de déformation d et d', nous définissons une nouvelle fonction, d_{nouv}, en les composant selon une fonction caractéristique c dont le domaine de valeurs est [0,1] :

$$d_{nouv} = c(x)d(x) + (1-c(x))d'(x)$$

Pour éviter une rupture visuelle (fonction de déformation discontinue), deux solutions sont possibles :

- c est continue.
- c est discontinue (valeurs 0 ou 1) et passe de 0 à 1 au niveau de l'abscisse, V, du point d'intersection des deux fonctions d et d'. La nouvelle fonction se définit alors comme suit :

$$d_{nouv} = \text{si } (x < V) \text{ alors } d(x) \text{ sinon } d'(x)$$



(a) Sphère tronquée

(b) Colline

Figure 5 : *Déformations composées.*

Nous composons deux fonctions de déformation pour obtenir les avantages des deux visualisations. Par exemple à la Figure 5-a, la sphère tronquée est obtenue en composant la fonction identité (Id) et la fonction d de la Figure 2 :

$$d_{5a} = \text{si } (x < V) \text{ alors } Id(x) \text{ sinon } d(x)$$

La visualisation en "colline" de la Figure 5-b est la fonction duale de la fonction précédente :

$$d_{5b} = \text{si } (x < V) \text{ alors } d(x) \text{ sinon } Id(x)$$

Nous avons montré comment définir plusieurs visualisations d'un même espace à partir d'une fonction de déformation ou de la composition de deux fonctions. La validation de ces visualisations ne pourra se faire que par expérimentation avec des utilisateurs. Le paragraphe suivant présente notre système VITESSE qui intègre ces visualisations.

REALISATION LOGICIELLE : LE SYSTEME VITESSE

Présentation de VITESSE

Dans le système VITESSE présenté à la Figure 6, nous avons exploité la technique de visualisation à plusieurs facettes présentée ci-dessus pour représenter les réponses fournies par un moteur de recherche du WWW. Les réponses obtenues peuvent définir un espace informationnel de grande taille.

Les moteurs de recherche comme Lycos [14], AltaVista [2] ou QuiQuoiOu [19] affichent les réponses à une requête par ordre de pertinence sous la forme de listes, chaque liste contenant dix réponses. Une réponse est constituée d'un lien vers une page et de quelques informations sur cette page.

Dans VITESSE, l'ensemble des réponses est visualisé selon la technique présentée au paragraphe précédent. Une réponse est représentée par un quadrilatère. La position et la couleur du quadrilatère traduit la pertinence de la réponse. En effet les réponses sont positionnées en escargot, en commençant par la réponse la plus pertinente placée au centre. Comme le montre la Figure 6, une vue par avion non déformée de l'ensemble des réponses est initialement proposée à l'utilisateur. A tout instant l'utilisateur peut changer de visualisation grâce à la palette d'icônes à gauche de l'espace des réponses. Chaque icône correspond à une visualisation exposée au paragraphe précédent telle que la sphère, la sphère tronquée, le cylindre ou encore la colline. Le changement de visualisation est progressif pour garantir la continuité visuelle. Il est accompagné d'un son continu pour souligner la phase de transition (redondance entre l'animation et le son). En dessous de l'espace des réponses se trouvent les outils de navigation (déplacement à droite, à gauche, vers le haut, vers le bas

et re-positionnement de l'espace au centre) : ils sont communs à toutes les visualisations (règle de conception).

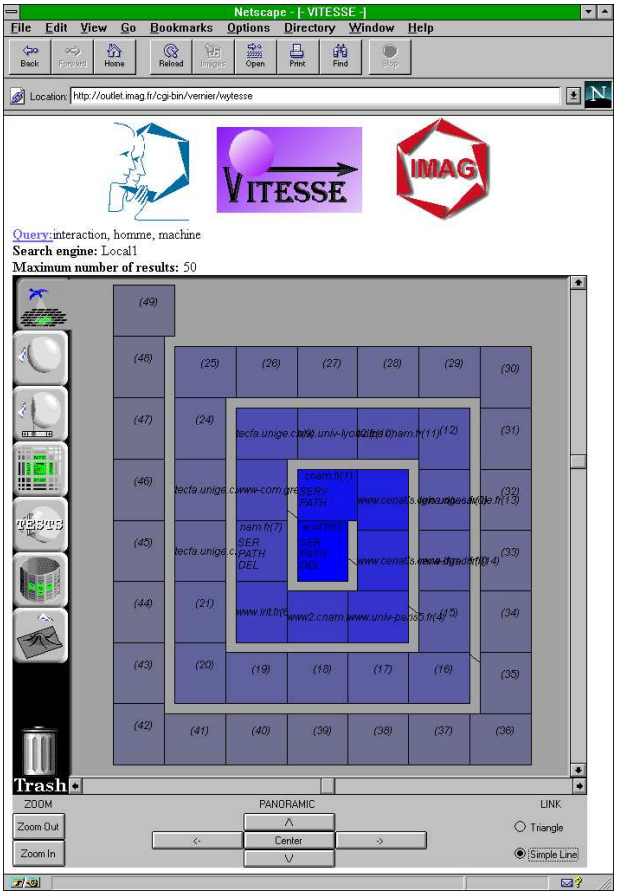


Figure 6 : *Le système VITESSE.*

Architecture Logicielle

Nous avons appliqué le modèle d'architecture PAC-Amodeus pour réaliser le système VITESSE, développé en langage JAVA. Une description détaillée de PAC-Amodeus peut être consultée dans [17]. Comme le montre la Figure 7, PAC-Amodeus préconise cinq composants logiciels : le Contrôleur de Dialogue (CD) composant principal est organisé en une hiérarchie d'agents PAC.

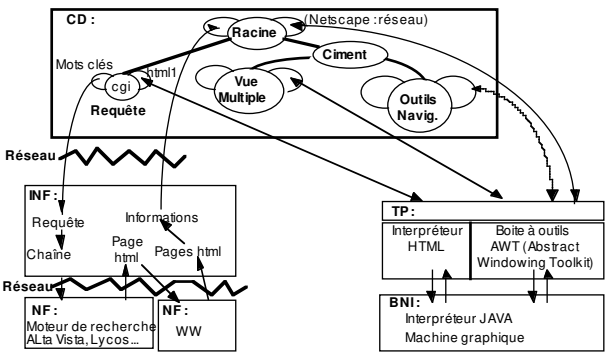


Figure 7 : *Architecture globale du système VITESSE.*

Architecture Globale

La Figure 7 présente l'architecture du système VITESSE.

Le Noyau fonctionnel (NF) maintient les concepts du domaine applicatif : le moteur de recherche. C'est le rôle de l'Interface avec le Noyau Fonctionnel (INF) de communiquer avec le NF par des envois de requête. L'INF sert d'adaptateur et permet de rendre le Contrôleur de Dialogue indépendant du moteur de recherche utilisé. De plus l'INF peut transformer les informations reçues du NF : par exemple l'INF vérifie la validité des réponses (liens inexistants) et calcule le nombre d'occurrences des mots de la requête pour chaque réponse.

A l'opposé de l'arche, le composant bas niveau d'interaction (BNI) désigne la plate-forme d'accueil logicielle et matérielle. Il contient en particulier l'interpréteur JAVA. Aucun code supplémentaire n'a été nécessaire pour ce composant. Le composant technique de présentation (TP) définit les règles de correspondance entre objets de présentation et objets d'interaction. Il comprend en particulier la boîte à outils AWT. Ce composant vise à rendre le contrôleur de dialogue indépendant des outils de développement.

Le Contrôleur de Dialogue, clé de voûte du système, prend en charge l'enchaînement des tâches et gère chaque fil de dialogue grâce à une hiérarchie d'agents PAC. Notre technique de visualisation à plusieurs facettes est donc réalisée dans le CD sous la forme d'une hiérarchie d'agents. Avant de détailler cette hiérarchie, il est important de noter que chaque agent est indépendant du Noyau Fonctionnel et des outils de développement. Ainsi le code de notre technique de visualisation est réutilisable et ne dépend pas ni d'un moteur de recherche, ni d'une boîte à outils.

Réalisation de la Technique de Visualisation

Pour définir la hiérarchie d'agents PAC, nous appliquons les règles heuristiques de mise en œuvre présentées dans [17]. Nous appliquons la règle "vue multiple" qui permet de réaliser le critère de représentation multiple. Cette règle s'énonce comme suit :

- chaque visualisation est gérée par un agent PAC, "Visualisation",
- un agent père "Vue Multiple" assure la cohérence entre les vues.

Nous obtenons la hiérarchie de la Figure 8. L'abstraction de l'agent "Vue Multiple" contient l'espace des informations non déformé (les coordonnées des quatre sommets des quadrilatères placés en escargot sans déformation). Sa présentation définit la palette qui permet de changer de visualisation. Son contrôle maintient le type de visualisation courante sélectionnée.

Chaque agent "Visualisation" contient dans son contrôle la fonction de déformation. L'abstraction maintient l'espace déformé selon la fonction. Les données ne sont pas dupliquées, seules les positions des emplacements sont stockées (c.-à-d. les positions des quatre sommets des quadrilatères). La présentation gère la visualisation. A un instant donné, un seul agent "Visualisation" est créé. Au changement de visualisation, un agent éphémère "Animation" est créé dynamiquement pour gérer le passage d'une visualisation à l'autre. Cet agent "Animation" reçoit de son agent père "Vue Multiple" la déformation courante (d1) et la déformation à atteindre (d2). Après n affichages (n=10 dans VITESSE), cet agent possède la même présentation que l'agent "Visualisation 2" correspondant à la visualisation à atteindre. L'agent "Animation" est alors détruit et un agent "Visualisation 2" est créé.

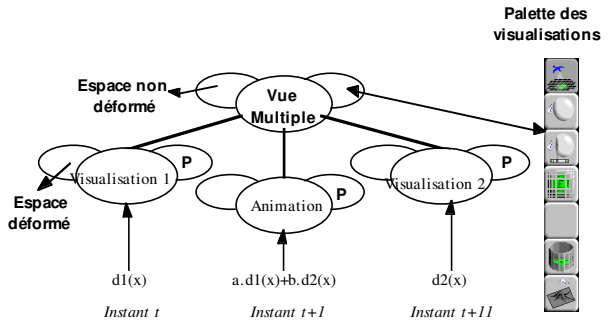


Figure 8 : *Première étape de construction de la hiérarchie PAC.*

Nous appliquons ensuite la règle "ciment" pour gérer les outils de navigation. Cette règle préconise un agent père "ciment" pour répercuter les actions reçues par un agent fils à un autre agent fils. Nous obtenons la hiérarchie de la Figure 9. L'agent "Outils" reçoit par sa présentation les événements de l'utilisateur. Ceux-ci sont envoyés à l'agent "Vue Multiple" par l'intermédiaire de l'agent "ciment".

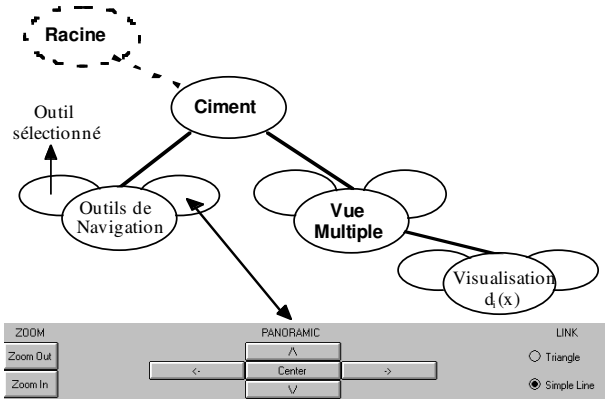


Figure 9 : *Hiérarchie d'agents du contrôleur de dialogue.*

La décomposition en une hiérarchie d'agents PAC permet une forte modularité du code. En particulier chaque visualisation est gérée par un agent distinct : ceci permet de facilement rajouter une visualisation sans remettre en cause le code qui réalise les autres visualisations. De plus la décomposition logicielle nous permet de plus facilement vérifier le respect des trois règles de conception énoncées au paragraphe "Représentation multiple". Pour vérifier la première règle "facilité de changer de visualisation" le concepteur doit porter son attention sur la facette présentation de l'agent "Vue Multiple". Dans VITESSE, ce module définit les boutons contenant les pictogrammes pour chaque visualisation. Le respect de la deuxième règle "continuité lors du passage d'une visualisation à une autre" est directement vérifiable par la présence de l'agent "Animation". Enfin la règle "réutilisabilité des outils de navigation" est aussi directement vérifiable par la présence d'un seul agent "Outils" pour toutes les visualisations.

CONCLUSION

En appliquant une méthode de conception descendante, nous avons étudié les aspects de conception de l'interface et ceux de réalisation logicielle de systèmes traitant de grands espaces d'informations. Dans cet article nous avons d'abord décrit une liste de critères d'ergonomie pour guider la conception. Le choix de ces critères repose sur les caractéristiques de la navigation dans un espace. Nous avons ensuite développé l'un des critères, la représentation multiple de l'espace des informations. Après avoir énoncé trois règles de conception à appliquer dans le cas de la représentation multiple, nous avons montré comment les vérifier en phase de réalisation logicielle selon le modèle PAC-Amodeus. Nous avons illustré les étapes de conception avec notre système VITESSE qui offre une technique de visualisation et plus généralement d'interaction à plusieurs facettes.

L'étape suivante consiste à valider expérimentalement ces résultats. Les critères d'ergonomie identifiés serviront de guide à l'évaluation. Nous espérons aussi identifier comment les utilisateurs perçoivent et utilisent les visualisations définies dans VITESSE. Cette expérimentation se fera sur le WWW, en capturant les actions de l'utilisateur grâce au protocole de communication CGI.

Enfin la multimodalité dans VITESSE se traduit actuellement

- par plusieurs visualisations qui utilisent le même dispositif physique, l'écran,
- par l'usage redondant d'une modalité sonore et d'une animation graphique lors du passage d'une visualisation à une autre.

Nous envisageons d'augmenter l'interface de VITESSE en utilisant des modalités sonores, en particulier pour les points de référence marqués par

l'utilisateur, qui fonctionneront comme des balises sonores.

BIBLIOGRAPHIE

1. Abowd G., Coutaz J. et Nigay L. Engineering for Human-Computer Interaction. In Proceedings of IFIP TC2/WG2.7 Working Conference on Engineering for Human-Computer Interaction (August 10-14 1992, Ellivuori, Finland), pp. 113-128.

2. AltaVista. http://www.altavista.digital.com/

3. Coutaz J. Interface Homme-Ordinateur : Conception et Réalisation. Dunod Publication (1990).

4. Dieberger A. Navigation in Textual Virtual Environments using a City Metaphor. PhD thesis. http://www.lcc.gatech.edu/faculty/dieberger/ (1994).

5. Dieberger A. On Magic Features in (Spatial) Metaphors SigLink Newsletter, Vol.4, No.3, (December 1995). http://www.lcc.gatech.edu/faculty/dieberger/magic_features.html.

6. Furnas G., Generalised Fisheye Views. In Proceedings of CHI'86 Human Factors in Computing Systems. (1986), pp. 16-23.

7. Furnas G. Effective View Navigation. In Proceedings of CHI'97 (March 22-27 1997, Atlanta, Georgia), pp. 367-374.

8. Gram C. et Cockton G. Design Principles for Interactive Software. Chapman & Hall (1984), pp. 133-187.

9. Hemmje M., LyberWorld : A 3D Graphical User Interface for Fulltext Retrieval. In Proceedings of CHI'95 Conference companion (May 7-11 1995, Denver, Colorado), pp. 407-408.

10. Hovestadt V., Gramberg O. et Deussen O. Hyperbolic user interface for computer aided architectural design. In Proceedings of CHI'95 (May 7-11 1995, Denver, Colorado), pp. 304-305.

11. Hutchins E., Hollan J. et Norman D. Direct Manipulation Interfaces. User Centered System Design, New Perspectives on Computer Interaction, édité par D. A. Norman, S.W. Draper, Hillsdale, New Jersey : Lawrence Erlbaum Associates, (1986), pp. 87-124.

12. Ioannidis Y. The FADIVA Benchmark - Version 0*. ftp://ftp.dis.uniroma1.it/pub/santucci/in/FADIVA/Ioannidis.ps.

13. Laurel B. Interface as Mimesis. User Centered System Design, New Perspectives on Computer Interaction, édité par D. A. Norman, S.W. Draper, Hillsdale, New Jersey : Lawrence Erlbaum Associates, (1986), pp. 67-85.

14. Lycos http://www-fr.lycos.com/

15. Mackinlay J., Robertson G. et Card S. The Perspective Wall : Detail and Context Smoothly Integrated. In Proceedings of CHI'91 (April 27-May 2 1991, New Orleans, Louisiana), pp. 173-179.

16. May J.,Scott S. et Barnard P. A Psychological **G**uide 4th edition http://www.shef.ac.uk/~pc1jm/guide.html.

17. Nigay L. Conception et modélisation logicielles des systèmes interactifs. Thèse de l'Université Joseph Fourier (1994).

18. Preece J. Human-Computer Interaction. Addison-Wesley (1994).

19. QuiQuoiOu http://www.wanadoo.fr/bin/frame.cgi?service=quiquoiou

20. Robertson G., Mackinlay J. et Card S. Cone Trees : Animated 3D Visualisations of Hierarchical Information. In Proceedings of CHI'91 (April 27-May 2 1991, New Orleans, Louisiana), pp. 189-194.

21. Sarkar M. et Brown M. Graphical fisheye views of graphs. In Proceedings of CHI'92 (May 3-7 1992, Monterey, California), pp. 83-91.

22. Stone M., Fishkin K. et Bier E. The Movable Filter as a User Interface Tool. In Proceedings of CHI'94 (April 24-28 1994, Boston, Massachusetts), pp. 306-312.

23. Tweedie L., Spence B., Williams D. et Bhogal R. The Attribute Explorer. In CHI'94 Video Proceedings and Conference Companion, (April 24-28 1994, Boston, Massachusetts), pp. 435-436.

24. Tweedie L. Characterizing Interactive Externalizations. In Proceedings of CHI'97 (March 22-27 1997, Atlanta, Georgia), pp. 375-382.

25. VITESSE, Visualisation & Interaction Techniques to Enhance Superscalar Searsh Engine. http://iihm.imag.fr/vernier/Vitesse.html.

26. Ware C. et Lewis M. The DragMad Image Magnifier. In Proceedings of CHI'95 Conference Companion (May 7-11 1995, Denver, Colorado), pp. 407-408.