

# Des aides transparentes de navigation et un nouveau type de menu pour les interfaces zoomables

Stuart Pook<sup>1,2</sup>

Eric Lecolinet<sup>1</sup>

Guy Vaysseix<sup>2,3</sup>

Emmanuel Barillot<sup>2,3</sup>

<sup>1</sup> École Nationale Supérieure des Télécommunications, CNRS URA 820, 46 rue Barrault, 75013 Paris, France

<sup>2</sup> Infobiogen, 523 place des Terrasses de l'Agora, 91034 Évry cedex, France

<sup>3</sup> Généthon, 1 bis rue de l'Internationale, 91000 Évry, France

stuart@acm.org, elc@enst.fr, vaysseix@infobiogen.fr, manu@infobiogen.fr

## RÉSUMÉ

Le fait que les interfaces zoomables (IZs) fournissent généralement peu de contexte peut parfois rendre leur utilisation difficile. Les utilisateurs ont rapidement tendance à ne plus savoir où ils sont dans l'espace d'information ni ce qu'ils sont en train de regarder. Nous proposons deux aides temporaires et transparentes qui montrent à l'utilisateur sa position dans l'espace d'information. Nous proposons également une représentation hiérarchique qui indique en permanence la structure de l'espace d'information et la position de l'utilisateur dans cette structure.

Les IZs soulèvent également un problème d'interaction. Leurs commandes sont fréquemment utilisées et sur une interface changeante : le modèle WIMP est alors trop limité pour un accès rapide à toutes ces commandes sans l'utilisation de nouvelles techniques d'interaction. Nous présentons un nouveau type de menu, nommé *control menu*, qui facilite l'accès rapide à ces commandes et qui peut être utile dans d'autres types d'applications.

**MOTS CLÉS :** interfaces zoomables, visualisation d'information, transparence, techniques d'interaction, vues focus+contexte, control menus.

## INTRODUCTION

Les interfaces zoomables (IZs) ne sont plus une nouveauté et leurs principes [6] et applications pratiques [2, 7, 9] ont été présentés dans de nombreuses publications. Quand un utilisateur interagit avec une IZ, il voit une vue d'un espace d'information. La vue initiale montre l'espace entier à une échelle permettant de la faire tenir sur l'écran de l'utilisateur. L'utilisateur peut alors zoomer (agrandir) une partie de la vue qu'il trouve intéressante. Les objets graphiques s'agrandissent jusqu'à ce qu'il y ait suffisamment de place sur l'écran pour les remplacer par d'autres objets graphiques qui montrent les données sous-jacentes avec plus de détails. Nous avons employé cette technique, nommée « zoom sémantique », pour visualiser et parcourir la base de données HuGeMap des principales cartes génétiques et physiques du génome humain [15]. Cette IZ a été utilisée pour tester les nouvelles techniques décrites dans cette communication. Pour comprendre les exemples

présentés, il suffit de savoir que la première vue montre 24 chromosomes (figure 1a), que ces chromosomes ont trois cartes génétiques et physiques (figure 1b), et que ces cartes consistent en des marqueurs génétiques positionnés le long d'un axe (figure 1c).

Les IZs sont faites pour présenter un ensemble d'informations aux utilisateurs de ces systèmes. Une des raisons pour lesquelles toutes les potentialités des IZs ne sont pas exploitées est que la vue de l'information présentée aux utilisateurs, le *focus*, ne comporte pas toujours suffisamment de *contexte* pour que les utilisateurs puissent situer ce qu'ils voient dans l'espace d'information. Une fois arrivés dans cette situation les utilisateurs sont désorientés, parfois jusqu'à ne plus comprendre ce qu'ils voient et ne plus savoir où aller : ils sont « perdus dans l'hyperespace ». Nous présentons deux nouveaux outils temporaires que les utilisateurs peuvent faire apparaître et utiliser lorsqu'ils arrivent dans cette situation. Le premier permet aux utilisateurs de situer le focus dans une vue globale de l'espace d'information. Le deuxième leur permet de parcourir en sens inverse le chemin qui les a amenés à se perdre. Nous présentons également une troisième aide de navigation qui est toujours visible et qui facilite la navigation en permanence. Cette aide, une deuxième zone d'affichage dans l'IZ, montre la hiérarchie ou la structure de l'espace d'information et la position de l'utilisateur dans cet espace. Cette zone peut aussi permettre à l'utilisateur de se déplacer dans l'espace d'information.

Bien que les IZs soient des interfaces complexes, l'interaction se fait toujours au moyen de la souris, des menus, et des boutons selon le modèle WIMP traditionnel. Dans une IZ, un utilisateur peut zoomer, dézoomer, faire défiler l'image, créer des lentilles magiques, déplacer des lentilles, retailler des lentilles, zoomer ou faire défiler un portail, etc. Or, les utilisateurs exécutent certaines de ces actions très fréquemment. Un utilisateur zoome jusqu'à ce que l'échelle désirée soit obtenue ou fait défiler l'image jusqu'à ce que l'objet recherché soit trouvé. Les utilisateurs sont également confrontés à des changements fréquents de la représentation, chaque zoom ou dézoom pouvant modifier les objets affichés. Il est donc peu pratique de contrôler une IZ avec des interacteurs sur la vue car ils

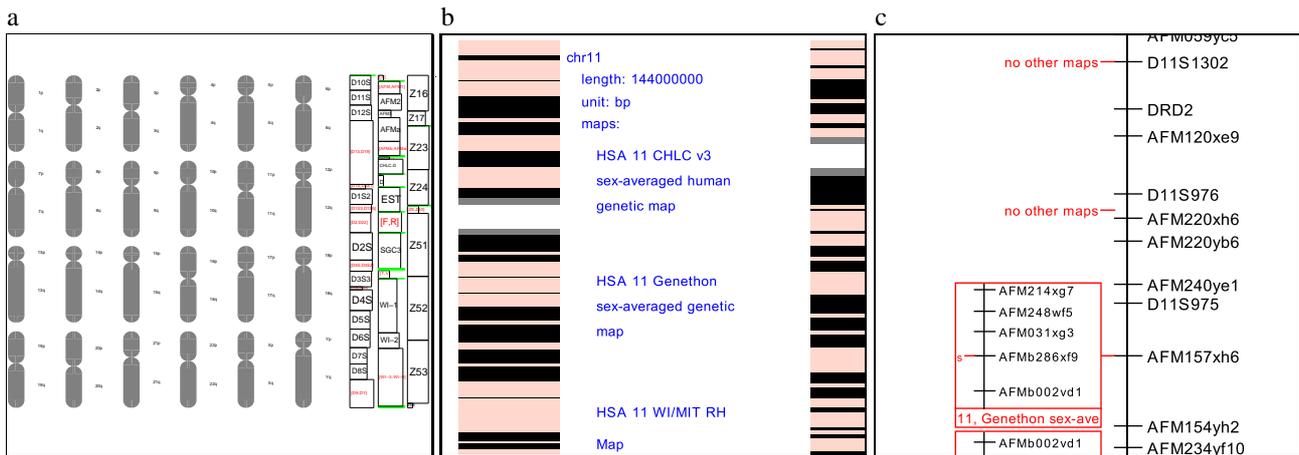


Figure 1 : (a) vue initiale ; (b) zoom sur le chromosome 11, et on voit maintenant ses trois cartes ; (c) les marqueurs d'une carte

changent de position très souvent. Les utilisateurs ont besoin de moyens simples et uniformes pour interagir avec ces interfaces complexes. Nous proposons donc un nouveau type de menu, appelé *control menu*, qui permet un accès rapide aux nombreuses opérations présentes dans une IZ. Dans la continuité de la sélection l'utilisateur peut contrôler l'opération avec les mouvements de la souris dans deux dimensions. Un seul interacteur est donc suffisant pour la réalisation d'une opération complexe.

Une IZ implémentant, en Java et C++, les nouvelles techniques présentées dans cette communication peut être testée à l'URL <http://www.infobiogen.fr/services/zomit/>.

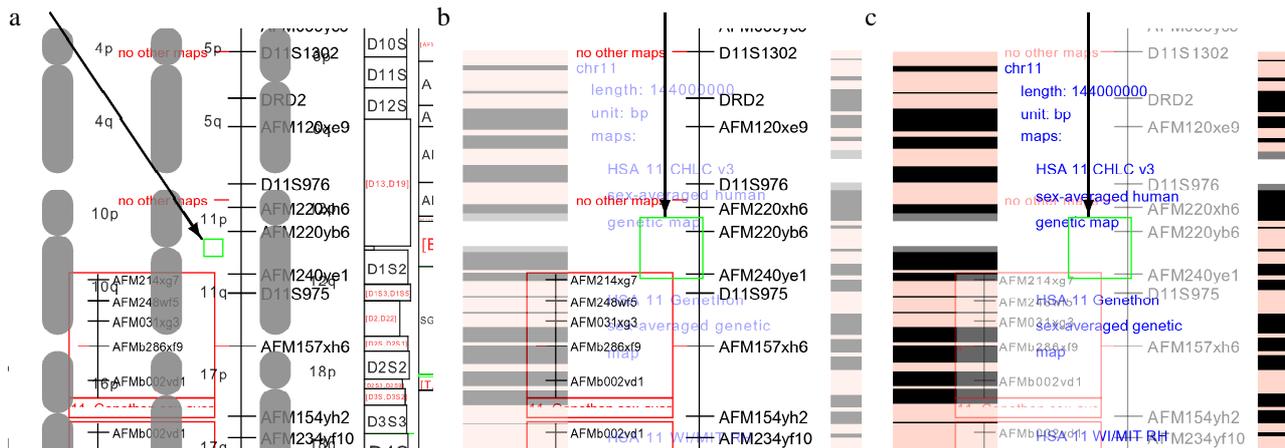
## COUCHE DE CONTEXTE

Avec une IZ, un utilisateur ne peut voir qu'une vue à la fois : le focus. Le contexte du focus a été perdu. D'autres types d'interfaces proposées en visualisation de l'information, tentent d'intégrer le contexte et le focus dans une vue unique. Par exemple, les vues *fish-eye* [5] intègrent le contexte et le focus en affichant une partie de l'information qui entoure le focus. Chaque information est affichée ou non en fonction de son niveau d'intérêt et de sa distance par rapport au focus (plus un objet est loin du focus plus son niveau d'intérêt doit être élevé pour qu'il soit affiché). Avec le Document Lens [16] les utilisateurs voient une zone de focus d'une ou plusieurs pages d'un document. Les pages situées dans la zone de focus sont lisibles tandis que les autres pages (situées dans la zone de « contexte ») restent visibles mais sont trop petites pour être lisibles. Elles servent à aider les utilisateurs à situer le focus dans l'ensemble du document. Les surfaces pliables en trois dimensions [3] et les vues hyperboliques [13] montrent toujours l'ensemble de l'espace d'information. Dans ces représentations, l'espace consacré à chaque objet est inversement proportionnel à la distance de cet objet par rapport au focus. Toutes ces techniques déforment la représentation graphique de l'espace d'information en éliminant certains objets ou en modifiant leur taille ou leur position jusqu'à, parfois, les rendre méconnaissables.

Contrairement à ces techniques qui déforment la représentation, nous proposons une *couche de contexte* qui combine le focus et son contexte sans déformation. La couche de contexte est temporaire et affichée uniquement quand l'utilisateur le désire. Pendant son utilisation elle se superpose en transparence à la vue principale. Des études [8] montrent que l'affichage de couches transparentes superposées est bien accepté par les utilisateurs. L'utilisateur fait déplacer les couches transparentes sur la vue principale. Ce mouvement aide l'utilisateur à distinguer les objets se trouvant dans la couche de contexte de ceux qui sont dans la vue principale. De plus, l'affichage de cette couche est temporaire afin de ne pas surcharger l'écran quand l'utilisateur n'a pas besoin de voir le contexte. Elle disparaît automatiquement dès que l'utilisateur relâche le bouton de la souris à la fin du geste qui a provoqué son affichage. L'interaction se fait au moyen d'une technique adaptée qui sera détaillée dans la section « Interaction dans les interfaces zoomables ».

La couche de contexte est centrée sur la vue du focus. La position et la taille de la vue du focus sont indiquées par un rectangle sur la couche de contexte. Cette dernière peut être contrôlée dans deux directions orthogonales : l'échelle (c'est-à-dire le niveau de zoom sémantique) du contexte et le niveau relatif de transparence des deux vues. Le réglage de l'échelle permet d'afficher, comme couche de contexte, une vue à n'importe quelle échelle située entre celle de la vue initiale et celle de la vue du focus. L'utilisateur peut donc faire varier l'échelle du contexte afin de situer la zone de focus dans un contexte adéquat.

La figure 1a montre la vue initiale dans notre IZ et la figure 1c une vue que l'utilisateur peut voir après avoir navigué quelque temps dans l'espace d'information. Dans cet exemple, cette deuxième vue ne contient pas d'élément permettant à l'utilisateur de savoir sur la carte de quel chromosome il se trouve. Dans cette situation l'utilisateur peut alors afficher la couche de contexte ce qui donne la figure 2a. Celle-ci est une superposition de la figure 1c (le focus) sur la vue initiale représentée figure 1a (le con-



**Figure 2 :** (a) le focus et le contexte combinés (détail) ; (b) quelle carte est-ce? ; (c) concentrer sur le contexte.

texte). La position du focus est indiquée sur la couche de contexte par un rectangle situé au centre (une flèche met en évidence ce rectangle sur la figure 2a). Sur l'écran ce rectangle est vert afin d'être très visible. Dans la figure 2a ce rectangle est sous le mot « 11p » et l'utilisateur sait donc que le focus montre une partie du chromosome 11. Dans la figure 2b, l'utilisateur a zoomé la couche de contexte (mais pas la vue de focus) ce qui donne maintenant comme contexte une vue où les noms des cartes génétiques sont visibles (figure 1b). L'utilisateur peut voir le rectangle (indiqué par une flèche) qui rappelle la position du focus sur le texte « HSA 11 CHLC v3 sex-averaged human genetic map ». L'utilisateur sait donc que le focus montre actuellement cette carte génétique. L'utilisateur a pu voir le contexte à deux échelles différentes de manière à pouvoir situer le focus dans deux contextes différents.

Notre système permet d'autre part à l'utilisateur de se concentrer sur l'une des deux vues en changeant la transparence relative de ces deux vues. La transparence peut varier continuellement d'un état où seule la vue de focus (vue principale) est visible jusqu'à l'état opposé où seule la couche de contexte est visible. Ainsi par exemple, la figure 2c est similaire à la figure 2b sauf que l'utilisateur est maintenant en train de se concentrer sur le contexte et a quasiment fait disparaître le focus. Le rectangle qui rappelle la position du focus est toujours visible et l'utilisateur peut voir plus clairement que le focus montre actuellement la carte « ... CHLC ... ».

### COUCHE HISTORIQUE

La couche de contexte décrite ci-dessus permet à l'utilisateur de trouver une réponse à la question « où suis-je? ». Une autre question importante est « comment suis-je arrivé ici? ». Nous proposons une nouvelle vue appelée *couche historique* qui permet de se déplacer interactivement entre la vue initiale et la vue courante en suivant le chemin emprunté par l'utilisateur. De même que pour la couche de contexte, la couche historique est affichée en transparence et de manière temporaire afin de ne pas surcharger l'écran. Elle disparaît dès que l'utilisateur relâche

le bouton de la souris à la fin du geste qui a provoqué son affichage.

Le parcours de l'utilisateur dans l'espace d'information peut être considéré comme une succession de vues de l'espace entre la vue initiale et la dernière vue courante (en haut de la figure 3). L'utilisateur fait varier interactivement la couche historique qui se superpose à la vue initiale (en bas de la figure 3). L'utilisateur peut ainsi « revenir dans le temps » interactivement en comparant l'évolution des vues courantes successives avec la vue initiale. Cette comparaison se fait directement par superposition grâce à la transparence. Elle est également facilitée par le fait que les tailles et positions de la couche historique et de la dernière vue courante sont rappelées à l'écran par deux rectangles de couleurs différentes (indiqués figure 3 par des lignes en pointillé).

De même que pour la couche de contexte il est possible de contrôler interactivement la transparence relative de deux couches. Par contre, les rectangles rappelant les positions des vues sont toujours affichés de la même manière et ne sont pas affectés par le niveau de transparence.

### HIÉRARCHIES ET NAVIGATION RAPIDE

Les IZs sont souvent utilisées pour visualiser des données organisées hiérarchiquement. De ce fait les objets de l'espace d'information qui ne sont pas accessibles à partir d'objets visibles dans la vue initiale sont parfois difficiles à trouver, l'utilisateur ne sachant pas nécessairement à quel endroit zoomer pour les trouver. La technique des zones critiques [11], par exemple, propose de créer les indications montrant où zoomer pour trouver des objets autrement invisibles, mais ce problème est loin d'être complètement résolu.

Les techniques présentées dans la section précédente aident l'utilisateur à comprendre l'espace d'information de la vue initiale jusqu'à la vue courante. Cependant, elles n'informent pas l'utilisateur sur ce qui se trouve dans d'autres régions ni, surtout, ce qui peut être trouvé en zoomant davantage. D'autre part, dans les IZs classiques, l'utili-

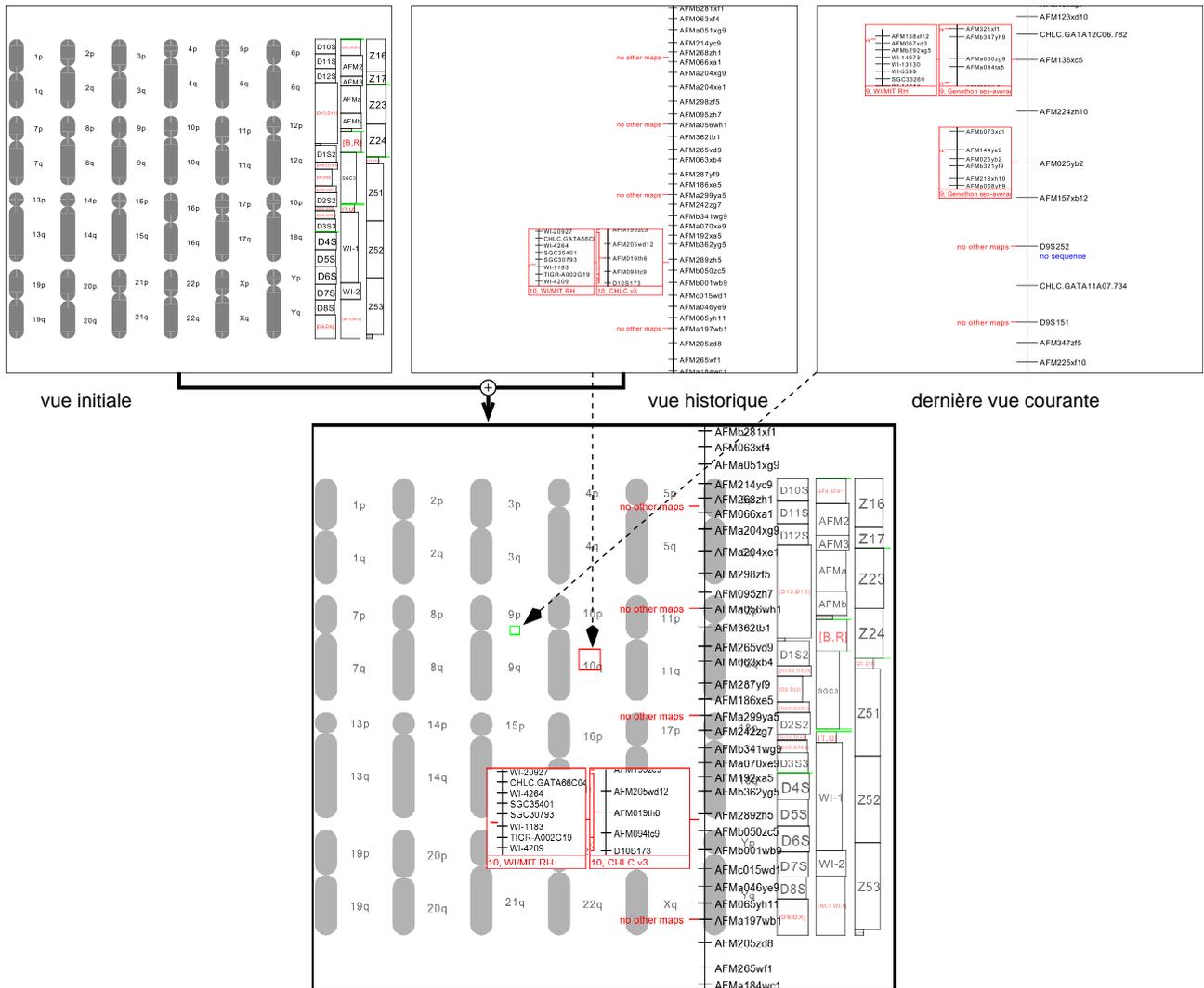


Figure 3 : superposition de la couche historique et de la vue initiale

sateur ne peut pas non plus utiliser l'organisation hiérarchique des données pour naviguer plus efficacement dans l'espace d'information : l'utilisateur regardant les détails d'un objet n'a pas la possibilité de dézoomer automatiquement pour voir l'objet en entier et ne peut pas non plus utiliser la hiérarchie pour se déplacer d'un sous-objet vers un autre sous-objet du même type.

Les IZs ne donnent qu'une vue qui est une tranche horizontale de l'espace 3D d'information (en considérant que la dimension verticale est celle du zoom). Nous proposons une seconde vue orthogonale à la première, qui est une tranche verticale aplatie de l'espace d'information. Cette seconde vue (figure 4) affiche les noms des objets qui sont situés au dessus de la vue courante dans la hiérarchie. Comme les objets ont également des types (un objet peut être un chromosome, une carte, une séquence, etc) la hiérarchie entière des types d'objets peut être montrée dans la seconde vue si l'espace d'information est suffisamment régulier. Sinon, seulement une partie de la hiérarchie des types centrée sur la position de l'utilisateur peut

être montrée. Cette seconde vue indique à l'utilisateur la structure de l'espace d'information, sa position courante, quelles sont les autres informations disponibles, leur disposition spatiale, et comment les trouver.

La figure 4 montre une vue dans l'IZ (à gauche) et la vue de la hiérarchie (à droite). Dans cet espace régulier, les chromosomes sont visibles sur la vue initiale. Les chromosomes possèdent deux « bras », des données et des cartes. Cette structure est affichée dans la vue de la hiérarchie dès que l'IZ est lancée. La position de l'utilisateur dans la structure est indiquée en gris (magenta sur l'écran) ; dans cet exemple l'utilisateur est en train de regarder la carte « Généthon » sur le chromosome 9. Cette vue indique également que si l'utilisateur continue de zoomer sur la carte il trouvera les marqueurs puis leur séquences.

La technique d'« Excentric Labeling » [4] permet d'identifier des objets sur l'écran. Cette technique étiquette, avec des bulles d'aide dans la vue principale, les objets se trouvant autour du curseur. Nous proposons une autre technique, non intrusive, pour identifier les objets autour du

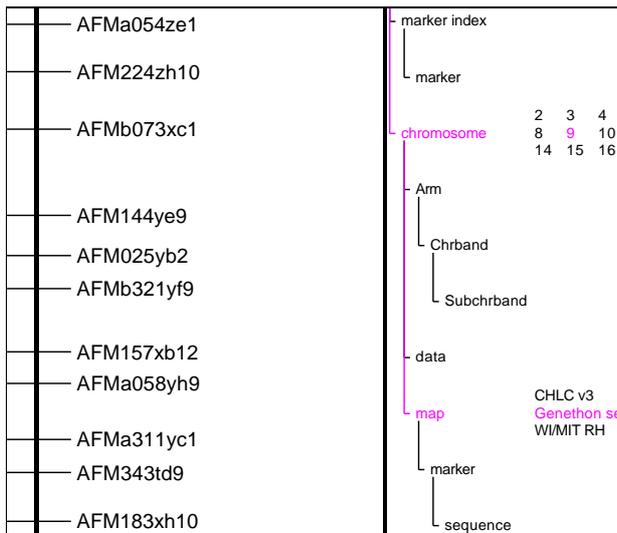


Figure 4: hiérarchie

curseur. Quand l'utilisateur déplace le curseur sur la vue principale, la vue de la hiérarchie est mise à jour avec le type et le nom de l'objet sous le curseur. Si le curseur quitte la vue principale ou s'il n'est pas sur un objet, la vue de la hiérarchie indique le plus bas niveau dans la hiérarchie auquel tous les objets visibles appartiennent. Cette technique d'étiquetage présente des similarités avec les bulles d'aide dans la mesure où l'utilisateur n'a pas besoin de demander l'information. Un avantage de notre technique est qu'elle n'empiète pas sur la vue principale.

Cette technique offre également un moyen efficace à l'utilisateur pour se déplacer car celui-ci peut se servir de la vue de la hiérarchie pour naviguer et pour aller directement des objets visibles aux objets liés mais qui ne sont pas dans la vue courante. Si l'utilisateur clique sur le mot « chromosome » dans la figure 4, l'IZ dézoomera suffisamment pour montrer le chromosome 9 en entier. L'utilisateur pourrait également cliquer sur les mots « CHLC v3 » (le nom de la carte à côté de la carte « Génethon » sur le chromosome 9) afin de se déplacer vers la même région sur la carte « CHLC v3 ».

### Évaluation

Pour faire une évaluation des techniques de visualisation proposées dans cette communication nous avons créé une version modifiée de notre IZ sans la vue de la hiérarchie. Huit sujets, des volontaires parmi nos collègues d'Infobiogen, ont été formés sur notre IZ. Pendant l'évaluation on leur a demandé de répondre à 22 questions de type choix multiple. La séance d'entraînement leur a montré comment répondre à ce genre de questions avec et sans la vue hiérarchique. Les sujets ont été divisés en 4 groupes, et ont traité 11 questions avec aide de la hiérarchie, et 11 sans: 2 groupes ont commencé par répondre à 11 questions avec aide de la hiérarchie, les 2 autres groupes ont commencé sans cette aide. Aussi chaque question a été traitée avec l'aide de la hiérarchie par 2 groupes de sujets, et sans par les deux autres groupes.

Pour chaque sujet nous avons calculé le temps pris pour répondre aux 11 questions sans la vue hiérarchique divisé par le temps pris pour répondre aux 11 autres questions avec la vue hiérarchique. Une valeur supérieure à 1 voulait donc dire que la vue hiérarchique aidait dans l'exécution des tâches. La moyenne était de 1,58 avec un écart type de 0,54. Ce grand écart type venait du fait que certains utilisateurs ne connaissaient pas la structure de l'espace d'information avant cette étude. Ces utilisateurs trouvaient que la séance d'entraînement n'était pas suffisante et donc que la seconde partie des questions était plus facile. En général, cependant, les sujets étaient plus rapides avec la vue hiérarchique et les commentaires étaient positifs. Les autres techniques présentées dans cette communication sont en cours d'évaluation.

### INTERACTION DANS LES INTERFACES ZOOMABLES

Des menus standards tels que pull-down, pop-up et marking menus [12] permettent aux utilisateurs de choisir des opérations à effectuer. Les pop-up et marking menus sont contextuels: ils sont activés à un endroit précis dans l'interface. L'interface peut alors adapter le contenu du menu à cette position et associer l'action choisie grâce au menu à cette même position.

Les pop-up menus et marking menus ne permettent pas de contrôler l'opération choisie de façon continue (par exemple pour effectuer un défilement). Ils ne permettent pas non plus aux utilisateurs de fournir des paramètres pour l'opération. Par exemple, une opération telle que le changement de la taille de police dans un traitement de texte nécessite souvent une boîte de dialogue pour fournir la nouvelle taille. Les utilisateurs doivent utiliser le menu et ensuite se concentrer sur un deuxième interactif, typiquement une boîte de dialogue. Une fois la nouvelle taille entrée, la boîte de dialogue disparaît et l'utilisateur doit à nouveau changer de contexte.

L'opération de déplacement dans une IZ nécessite soit un bouton de souris spécifique (ou un bouton de souris et un modificateur) afin que l'utilisateur puisse faire glisser l'image, soit deux barres de défilement. On ne peut se déplacer avec un menu standard sauf avec des commandes peu ergonomiques telles que « déplacer un peu vers la gauche ». De même, zoomer est une autre opération qu'il est malaisé de faire avec un menu. Les utilisateurs veulent zoomer jusqu'à ce que la bonne échelle ait été trouvée. Avec un menu standard l'utilisateur ne peut zoomer que par étapes prédéfinies et par utilisation répétitive du menu.

#### « Control menu »

Nous proposons un nouveau type de pop-up menu, appelé *control menu*, qui peut intégrer jusqu'à deux barres de défilement ou « spin-boxes » [14]. Avec ce menu les utilisateurs peuvent choisir des opérations et les contrôler d'un seul geste.

Un control menu fonctionne un peu comme un marking menu. L'utilisateur novice enfonce le bouton de la sou-

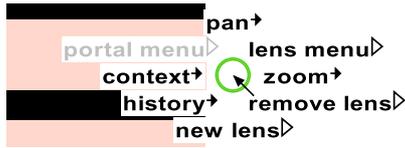


Figure 5 : le control menu dans notre IZ

ris, attend 0,3 secondes jusqu'à ce que le menu soit affiché centré sur la position du curseur, puis déplace le curseur dans la direction de l'opération désirée (figure 5). Le menu disparaît et l'opération commence dès que le curseur a été déplacé de la *distance d'activation* depuis le centre du menu. (Nous avons empiriquement choisi une distance d'activation de cinq fois le radius du cercle au centre du menu.) L'opération se termine quand l'utilisateur relâche le bouton de la souris. Les utilisateurs experts font le même geste que les novices mais sans effectuer la pause qui fait afficher le menu. Ainsi, les utilisateurs experts ne sont pas distraits par l'apparition du menu et les utilisateurs novices apprennent progressivement le geste expert.

Ces control menus sont particulièrement utiles dans les IZ à cause des nombreuses opérations contrôlées avec la souris dans ces systèmes. Cependant, ils pourraient également être efficaces dans les interfaces plus traditionnelles comme par exemple les traitements de texte.

#### Une barre de défilement

Une entrée dans un control menu peut être utilisée pour modifier le niveau de zoom dans une IZ. Cette opération illustre l'intégration d'un menu et d'une barre de défilement dans un seul interacteur. La figure 6 montre les

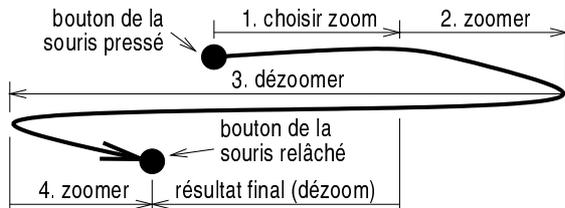


Figure 6 : zoomer avec un control menu

mouvements de la souris pendant l'utilisation d'un control menu pour choisir et contrôler une opération de zoom ou de dézoom. L'utilisateur enfonce le bouton de la souris et déplace celle-ci de la distance d'activation (le mouvement numéro 1 dans la figure 6) vers la droite (l'opération « zoom » étant sur la droite du control menu représenté figure 5). Ceci amorce l'opération de zoom et le curseur change à ce moment sur l'écran. À partir de ce moment, les mouvements de la souris vers la droite (mouvements 2 et 4 dans la figure 6) zooment la vue et les mouvements vers la gauche (mouvement 3) la dézooment. Le contrôle par retour d'information est immédiat : la vue change dès que l'utilisateur bouge la souris. L'utilisateur relâche le bouton de la souris lorsque l'échelle voulue (un dézoom dans la figure 6) a été obtenue.

Pendant l'opération de zoom, l'utilisateur peut annuler le zoom en déplaçant, sur une grande distance, la souris dans la direction orthogonale (vers le haut ou vers le bas dans le cas de l'exemple). Il peut alors confirmer l'annulation en relâchant le bouton de la souris ou annuler l'annulation en retournant la souris vers sa position verticale initiale. Dans ce cas, le bouton de la souris est toujours enfoncé et l'utilisateur peut continuer son zoom. Seules les opérations où une des deux directions de déplacement de la souris (verticale ou horizontale) n'est pas utilisée peuvent être annulées de cette manière.

#### Deux barres de défilement

Un control menu peut aussi être utilisé pour effectuer des défilements bi-directionnels. Il remplace alors deux barres de défilement. L'opération de défilement est sélectionnée en enfonçant le bouton de la souris et en déplaçant la souris vers le haut (« pan » sur la figure 5 en haut du menu). La vue suit le curseur pendant l'opération.

Un control menu est bien adapté pour contrôler deux paramètres dits intégraux : deux paramètres sont intégraux [10] quand leurs attributs se combinent dans la pensée de l'utilisateur en un seul attribut composé. Par exemple, les coordonnées  $x$  et  $y$  d'un objet sont intégrales car les utilisateurs les combinent et les considèrent comme la position de l'objet. Un déplacement diagonal de la souris a alors une signification simple : déplacer l'objet sur la diagonale. Par contre, la taille et la couleur d'un objet ne sont pas deux paramètres intégraux. Si un control menu était utilisé pour contrôler de tels paramètres simultanément, un déplacement diagonal de la souris n'aurait pas de signification évidente.

#### Sélectionner du texte

Un control menu peut être utilisé dans un traitement de texte traditionnel où une des opérations les plus courantes est de sélectionner du texte. Dans une telle application, l'opération « sélectionner » est placée à la droite du control menu (c'est-à-dire en remplaçant l'opération « zoom » dans la figure 5). Dans la figure 7a l'utilisateur a indiqué

- (a)  His tender heir might bear his memory  
But thou contracted to thine own bright eyes,  
Feed'st thy light's flame with self-substantial fuel,
- (b)  His tender heir might bear his memory:  
But thou contracted to thine own bright eyes,  
Feed'st thy light's flame with self-substantial fuel,
- (c)  His tender heir might bear his memory:  
But thou contracted to thine own bright eyes,  
Feed'st thy light's flame with self-substantial fuel,

Figure 7 : sélectionner du texte avec un control menu

le début du texte devant être sélectionné en enfonçant le bouton de la souris quand le curseur était entre les mots « might » et « bear ». Il déplace ensuite le curseur de la distance d'activation vers la droite. À ce moment l'opération commence et le texte situé entre la position initiale du curseur (quand le bouton de la souris a été enfoncé) et sa position actuelle est sélectionné (figure 7b). Tout en

gardant le bouton de la souris enfoncé, l'utilisateur peut ensuite déplacer le curseur vers le bas ou la droite afin de sélectionner davantage de texte (figure 7c), vers la gauche afin de sélectionner moins de texte ou vers le haut pour sélectionner le texte entre la nouvelle position du curseur et la fin du mot « might ». L'opération se termine et le texte est sélectionné quand l'utilisateur relâche le bouton.

Les opérations avec des paramètres (des changements de taille de police par exemple) sont décomposées en deux étapes. Le texte à modifier est sélectionné comme décrit ci-dessus puis le menu est de nouveau utilisé pour exécuter l'opération de changement de taille. Cette opération a un paramètre, la nouvelle taille, qui peut être choisi pendant l'opération en déplaçant le curseur vers le haut ou vers le bas.

Une opération telle que « couper » peut être fournie de la même manière (deux utilisations du menu) ou via une seule entrée dans le menu qui combine les deux opérations en sélectionnant le texte puis en l'effaçant à la fin de l'opération. Des commandes simples peuvent alors être exécutées en minimisant l'usage des menus.

### Boutons

Les control menus peuvent contenir des commandes simples qui n'ont pas de paramètre. Les commandes qui peuvent être annulées sont exécutées dès que le curseur a été déplacé de la distance d'activation depuis l'endroit où le bouton de la souris a été enfoncé. Puisque l'utilisateur maintient le bouton de la souris enfoncé, un mouvement du curseur dans l'autre direction annule l'opération. L'utilisateur peut alors annuler l'annulation en déplaçant à nouveau le curseur dans l'autre sens. L'opération ne devient définitive que lorsque le bouton de la souris est relâché.

### Sous-menus

Un control menu peut avoir des sous-menus. L'utilisateur enfonce le bouton de la souris et déplace le curseur dans la direction de l'entrée dans le control menu correspondant au sous-menu désiré. Une entrée dans le sous-menu est ensuite sélectionnée de la même manière.

### Control menus versus marking menus

Avec un marking menu la distance parcourue par le curseur n'a pas d'importance. Seule la forme du mouvement est importante et elle est analysée une fois que le bouton de la souris est relâché (ou quand l'utilisateur arrête de bouger le curseur afin d'ouvrir un sous-menu). Avec un control menu la distance parcourue par le curseur est informante. La position du curseur est constamment analysée et dès que la distance d'activation de la souris a été atteinte, l'opération commence.

### Control menus, contexte et historique

Un control menu est utilisé pour contrôler les couches de contexte et d'historique dans notre IZ. Ainsi, la couche de contexte a deux paramètres : l'échelle et le niveau de transparence. L'échelle est contrôlée avec des mouvements horizontaux de la souris et la transparence avec des mouvements verticaux. Le contrôle de la couche historique

est similaire : la position sur le chemin de l'utilisateur est contrôlée avec des mouvements horizontaux et la transparence avec des mouvements verticaux. Ces paramètres ne sont pas intégraux et nous examinons actuellement si cela crée des problèmes pour les utilisateurs. Comme expliqué précédemment, ces couches n'existent que pendant le geste du control menu ; dès que l'utilisateur relâche le bouton de la souris, la couche disparaît.

### Analyse des propriétés des control menus

L'analyse des propriétés des control menus, via le modèle d'interaction de Beaudouin-Lafon [1], permet de positionner ce nouvel interacteur dans l'espace du modèle.

Un control menu est contextuel : il agit soit sur l'objet soit sur la position se trouvant sous le curseur. Il ne nécessite pas non plus de se déplacer vers une barre de menu. L'écart spatial (la distance sur l'écran entre le menu et l'objet sur lequel il agit) est donc nul. Lors du zoom et du défilement, l'écart temporel (le temps entre le mouvement de la souris et la réaction de l'objet manipulé) est également nul car l'interface réagit immédiatement quand l'utilisateur déplace la souris. Le degré d'indirection (qui combine ces deux écarts) est donc faible.

Le ratio entre le nombre de degrés de liberté fourni par notre interacteur et le nombre de degrés de liberté de la souris (le degré d'intégration) est de 2/2 car, pour toutes les interactions dans notre IZ, les deux degrés de liberté de la souris sont utilisés.

Le degré de compatibilité (la similarité entre l'action physique de l'utilisateur et la réponse de l'objet sur lequel il agit) est élevé pendant un défilement car les mouvements de la souris sont directement reflétés par les mouvements de la vue. Ce degré de compatibilité est moins élevé pendant un zoom du fait de la possibilité d'annulation. En effet, tandis que les mouvements horizontaux sont en relation directe avec les mouvements de la souris, l'annulation requiert des mouvements verticaux, ce qui peut paraître moins évident à l'utilisateur.

### IMPLÉMENTATION

Les idées présentées dans cette communication ont été réalisées et testées dans un système de type client serveur conçu pour être utilisé sur Internet (figure 8). Le client est

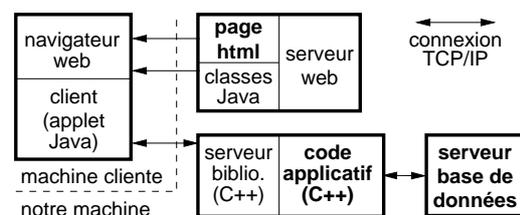


Figure 8 : implémentation client/serveur

une applet Java qui communique avec un serveur écrit en C++ [15]. Le client envoie la position de l'utilisateur dans l'espace d'information au serveur et le serveur répond en

envoyant les objets à afficher à cette position. Le client stocke ces objets et peut donc répondre rapidement aux changements d'échelle ou de position qui redemandent l'affichage de ces objets ; seul l'affichage des nouveaux objets est ralenti par la latence de la connexion avec le serveur. Le serveur s'exécute sur la même machine que la base de données afin de pouvoir lire rapidement les nombreuses informations requises pour construire les objets graphiques à envoyer au client. Le client et la bibliothèque du serveur avec lequel il communique sont génériques ; il n'est donc pas nécessaire de les modifier pour visualiser un autre type de données. Seul le code qui lit la base de données et celui qui appelle la bibliothèque, en texte gras dans la figure 8, doivent être modifiés pour une application différente.

La transparence des couches est simulée dans l'applet Java au moyen de fonctions *xor* pour des raisons de performance. La transparence est par contre effective quand le client tourne comme une application Java.

## CONCLUSION ET PERSPECTIVES

Les deux aides utilisant les couches présentées dans cette communication permettent aux utilisateurs de s'orienter s'ils se perdent dans les grands espaces d'information disponibles dans les IZs. La superposition de couches transparentes permet d'intégrer le focus et le contexte dans la même vue. Ces techniques seront prochainement évaluées plus en détail afin de mieux comprendre leur usage par les utilisateurs.

Les hiérarchies présentées fournissent un nouveau moyen de combiner le focus et le contexte dans les IZs. Elles aident les utilisateurs à mieux se localiser dans l'espace d'information et les renseignent sur les informations disponibles. L'évaluation a montré l'utilité de cette technique de représentation. Les évaluations à venir vont examiner comment les utilisateurs se servent des hiérarchies et des couches quand ces deux aides sont disponibles.

Nous proposons également un nouveau type de menu afin de faciliter la maîtrise de ces IZs complexes. Nous suggérons que l'utilisation de ces control menus ne se limite pas aux IZs mais qu'ils seraient utiles pour d'autres types d'interfaces homme-machine. Les investigations à venir vont, entre autres, examiner s'il est malaisé pour l'utilisateur d'être obligé, pour faire défiler la vue dans une direction quelconque, de bouger initialement la souris vers le haut pour déclencher l'opération. Nous essayerons également d'étendre l'usage des ces menus à des interactions variées, telle que par exemple le changement de la police du texte, ce qui nécessitera de faire apparaître une représentation visuelle adéquate pour guider l'utilisateur au cours de l'interaction.

Nous espérons que la combinaison de ces nouvelles aides à la navigation et d'une nouvelle façon d'interagir avec les IZs pourra offrir aux utilisateurs un meilleur contrôle sur une meilleure interface.

## REMERCIEMENTS

Ce travail a bénéficié du soutien de l'Union européenne (contrat BIO4-CT96-0346) et du CNET (contrat 97 754 21).

## REFERENCES

1. Beaudouin-Lafon, M. Instrumental interaction: An interaction model for designing post-WIMP user interfaces. In *CHI 2000* (The Hague, The Netherlands, Apr. 2000), ACM, 447–453.
2. Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *J. Vis. Lang. Comput.* 7 (Mar. 1996), 3–32.
3. Carpendale, M. S. T., Cowperthwaite, D. J., and Fracchia, F. D. 3-dimensional pliable surfaces: For the effective presentation of visual information. In *UIST '95* (Pittsburgh PA, USA, Nov. 1995), ACM, 217–226.
4. Fekete, J.-D., and Plaisant, C. Excentric labeling: Dynamic neighborhood labeling for data visualization. In *CHI '99* (Pittsburgh PA, USA, May 1999), ACM, 512–519.
5. Furnas, G. W. Generalized fisheye views. In *CHI '86* (Boston MA, USA, Apr. 1986), ACM, 16–23.
6. Furnas, G. W., and Bederson, B. B. Space-scale diagrams: understanding multiscale interfaces. In *CHI '95* (Denver CO, USA, 1995), ACM Press, 234–241.
7. Furnas, G. W., and Zhang, X. MuSE: A multi-scale editor. In *UIST '98* (San Francisco CA, USA, Nov. 1998), ACM, 107–116.
8. Harrison, B. L., Kurtenbach, G., and Vicente, K. J. An experimental evaluation of transparent user interface tools and information content. In *UIST '95* (Pittsburgh PA, USA, Nov. 1995), ACM, 81–90.
9. Hightower, R. R., Ring, L. T., Helfman, J. I., Bederson, B. B., and Hollan, J. D. PadPrints: Graphical multiscale web histories. In *UIST '98* (San Francisco CA, USA, Nov. 1998), ACM, 121–122.
10. Jacob, R. J. K., and Sibert, L. E. The perceptual structure of multidimensional input device selection. In *CHI '92* (Monterey CA, USA, May 1992), ACM, 211–218.
11. Jul, S., and Furnas, G. W. Critical zones in desert fog: Aids to multiscale navigation. In *UIST '98* (San Francisco CA, USA, Nov. 1998), ACM, 97–106.
12. Kurtenbach, G., and Buxton, W. User learning and performance with marking menus. In *CHI '94* (Boston MA, USA, Apr. 1994), ACM, 258–264.
13. Lamping, J., and Rao, R. The hyperbolic browser: A focus+context technique for visualizing large hierarchies. *J. Vis. Lang. Comput.* 7, 1 (Mar. 1996), 33–35.
14. Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. Control menus: execution and control in a single interactor. In *CHI 2000 Extended Abstracts* (The Hague, The Netherlands, Apr. 2000), ACM, 263–264.
15. Pook, S., Vaysseix, G., and Barillot, E. Zomit: biological data visualization and browsing. *Bioinformatics* 14, 9 (Nov. 1998), 807–814.
16. Robertson, G. G., and Mackinlay, J. D. The document lens. In *UIST '93* (Atlanta GA, USA, Nov. 1993), ACM, 101–108.