

# Représentation focus+contexte de listes hiérarchiques zoomables

Eric Lecolinet – Duc Nguyen

École Nationale Supérieure des Télécommunications (GET/ENST) - CNRS LTCI UMR 5141  
46 rue Barrault, 75013, Paris, France  
[elc@enst.fr](mailto:elc@enst.fr)

## RESUME

Les listes hiérarchiques usuelles ne permettent pas de balayer aisément l'ensemble des données qu'elles représentent, ce qui limite leur performances pour découvrir ou retrouver un élément. Cet article présente une nouvelle technique de visualisation et d'interaction qui tente de remédier à ce problème par l'utilisation conjointe d'une représentation fish eye, d'une forme de zoom sémantique et d'une technique d'interaction s'inspirant des menus gestuels de type *Control menus*. Cette technique permet de parcourir toute l'arborescence en un nombre réduit de gestes facilement exécutables.

**MOTS CLES :** Technique d'interaction, visualisation fish eye, zoom sémantique, listes hiérarchiques.

## ABSTRACT

Ordinary tree widgets do not make it easy to browse all the nodes of the hierarchy, thus reducing the performances for discovering or retrieving an object of interest. This paper presents a new visualization and interaction technique that tries to solve this problem by using a fish eye representation, semantic zooming and an interaction technique that is derived from *Control menus*. This technique makes it possible to browse the entire hierarchy by using a limited amount of mouse gestures.

**CATEGORIES AND SUBJECT DESCRIPTORS:** H.5.2. [Information Interfaces and Presentation]: User Interfaces. I.3.6 [Methodology and Techniques]: Interaction techniques.

**GENERAL TERMS:** Design, Human factors.

**KEYWORDS:** Interaction technique, fish eye visualization, semantic zooming, hierarchical lists.

## INTRODUCTION

L'utilisation de structures de données hiérarchiques est fort commune en informatique. Les hiérarchies permet-

tent de mieux organiser et classifier un ensemble de données dès lors que celui-ci atteint une certaine taille. Les représentations arborescentes sont tout aussi courantes dans les IHMs. En mettant en évidence la structure hiérarchique des données elles permettent de mieux organiser les représentations à l'écran en réduisant le nombre d'éléments présents dans chaque sous partie. Certaines de ces techniques de représentation ont également un deuxième objectif qui est de minimiser l'espace écran nécessaire pour représenter les données. Il en va ainsi des systèmes de menus cascades ou des listes hiérarchiques de type « arbres dépliables ». Seule une sous-partie de l'arborescence est visible à un instant donné dans le cas des menus. Les listes hiérarchiques sont basées sur un principe similaire, sauf qu'elles permettent un contrôle plus fin de la visibilité des objets, plusieurs branches pouvant généralement être ouvertes simultanément.

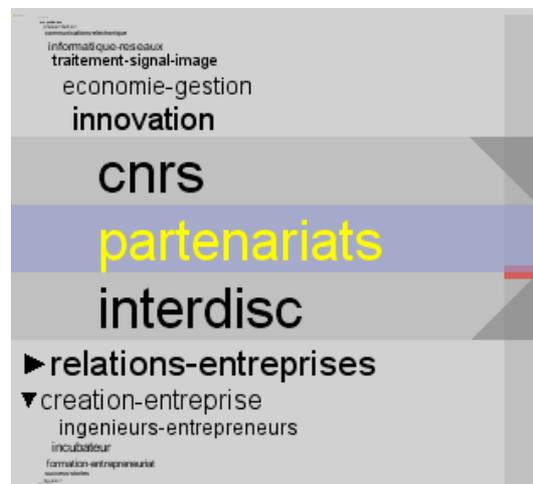


Figure 1 : Aperçu de la méthode proposée.

Ce souci d'économie de l'espace écran a pour conséquence qu'un nombre restreint d'éléments est visible à un instant donné. Ce type de technique impose donc de multiples manipulations interactives pour accéder à l'information souhaitée. Idéalement, pour compenser le fait que les données ne sont pas toutes présentes à l'écran, ces méthodes devraient permettre de balayer l'arbre facilement pour retrouver l'élément souhaité. C'est en partie le cas en ce qui concerne les systèmes de menus. Cependant, leur manipulation devient souvent malaisée et génératrice d'erreurs dès que la taille de

l'arbre est importante. Les arbres dépliables n'offrent généralement pas cette fonctionnalité et nécessitent de déplier complètement l'arbre pour trouver un élément. Une technique d'interaction appropriée est donc essentielle pour permettre une utilisation efficace des représentations arborescentes, surtout lorsque le nombre d'éléments est important.

Dans cet article, nous présentons une nouvelle technique de visualisation et d'interaction permettant d'afficher et de manipuler une grande quantité d'éléments organisés sous une forme hiérarchique. Cette technique vise également à minimiser l'espace écran dédié à l'affichage. Elle repose sur l'utilisation conjointe d'une représentation fish eye, d'une forme particulière de zoom sémantique et d'une technique d'interaction s'inspirant des menus gestuels de type *Control menus*. La prochaine section décrit les principes et sources d'inspiration qui sont à la base de cette technique. Elle est suivie par une section précisant certains points concernant son implémentation et la conclusion.

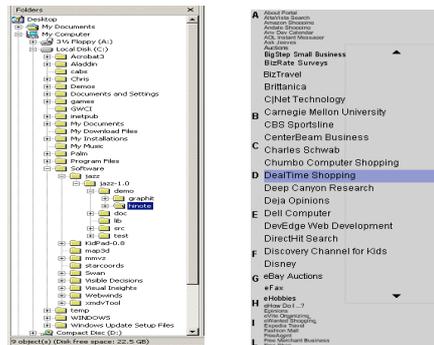


Figure 2 : Liste hiérarchique et FishEye Menu.

## PRINCIPES

### Représentations hiérarchiques et linéaires

Comme expliqué dans l'introduction, les représentations hiérarchiques sont souvent privilégiées lorsqu'il faut afficher un grand nombre de données. Cependant, le fait que tous les objets ne soient pas visibles en permanence dans les représentations arborescentes usuelles impose à l'utilisateur de naviguer fréquemment dans la hiérarchie sauf s'il est suffisamment « expert » pour savoir par avance où se trouve l'objet souhaité. Cette constatation, faite par Bederson dans [2], est l'un des éléments qui l'a conduit à développer une technique non hiérarchique appelée *FishEye Menus* (Figure 2, à droite). Cette technique permet d'afficher un grand nombre d'éléments dans une liste en utilisant le principe du fish eye, la taille des éléments étant inversement proportionnelle à leur éloignement de la zone de focus.

En réalité, l'argument de la visibilité est à prendre avec précaution dans la mesure où, si les éléments restent effectivement *visibles* dans le système qu'il propose (du moins tant que leur nombre reste inférieur au nombre de

pixels disponibles) très peu d'entre-eux sont en fait *lisibles*. Notre conjecture est donc que l'efficacité de cette technique, qui a été montrée expérimentalement, est surtout liée à la combinaison d'une technique d'interaction et de visualisation qui permet de balayer un grand nombre d'éléments en un seul geste. En d'autres termes, même si l'utilisateur ne *voit* pas très bien où se trouve l'élément recherché, il pourra rapidement le retrouver, et ce au moyen d'un geste facile et rapide à exécuter.

Cette idée est à la base de la technique que nous proposons. Celle-ci étend les *FishEye Menus* en permettant à l'utilisateur de naviguer non plus seulement dans une liste mais également dans un arbre. Comme dans le cas des *FishEye Menus*, la non-visibilité (ou non-lisibilité) de la plupart des éléments est compensée par une technique de visualisation et d'interaction qui permet de balayer l'arbre entier au moyen d'un nombre réduit de gestes facilement exécutables. La figure 1 illustre l'utilisation de la technique de fish eye dans notre système (la zone de focus étant approximativement au milieu et représentée avec une couleur légèrement plus sombre). La figure 3 montre un exemple de hiérarchie où deux noeuds sont développés (l'orientation de la flèche avant l'intitulé du noeud et les niveaux d'indentation des intitulés jouent le même rôle que dans les listes hiérarchiques usuelles).

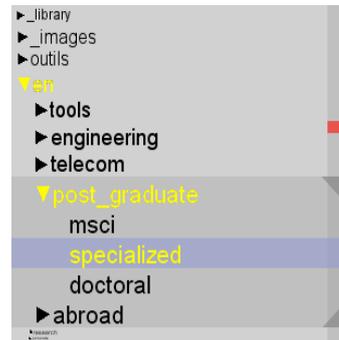


Figure 3 : représentation hiérarchique.

### Représentation hiérarchique et zoom sémantique

Il existe essentiellement deux façons de représenter les arbres. La première classe de techniques met en évidence leur structure en représentant l'articulation des noeuds ou les connexions entre ceux-ci. Les menus et les listes hiérarchiques entrent dans cette catégorie. D'autres techniques représentent également les arcs entre les noeuds (vues hyperboliques, représentations 3D, SpaceTree [6], etc.). Elles sortent du cadre de cette étude dans la mesure où nous souhaitons conserver un type de représentation proche des listes hiérarchiques usuelles. Par ailleurs, elles tendent pour la plupart à nécessiter un espace écran plus important.

La seconde classe se base sur l'imbrication des noeuds, comme l'illustre la technique des Treemaps. Les interfaces zoomables s'appuient sur un principe similaire, le

zoom sémantique pouvant être vu comme une technique qui remplace les noeuds visibles à un certain niveau de zoom par ceux des enfants visibles au niveau de zoom supérieur ou inférieur (suivant le sens du zoom). Cette idée est utilisée dans notre technique pour permettre un passage progressif de l’affichage de l’intitulé d’un noeud à celui de son contenu. Contrairement au cas des listes imbriquées usuelles, le développement d’une branche ne se fait pas de manière « quantique » en cliquant sur un interacteur spécifique (typiquement une flèche ou un signe +), mais continûment en zoomant sur l’élément que l’on souhaite développer. Suivant le principe du zoom sémantique, la taille du noeud concerné augmente jusqu’à ce que celui-ci soit remplacé par son contenu. Une technique de fondu est utilisée pour réduire les discontinuités visuelles. Lorsqu’il augmente en taille, l’intitulé du noeud devient de moins en moins visible tandis que son contenu apparaît progressivement.

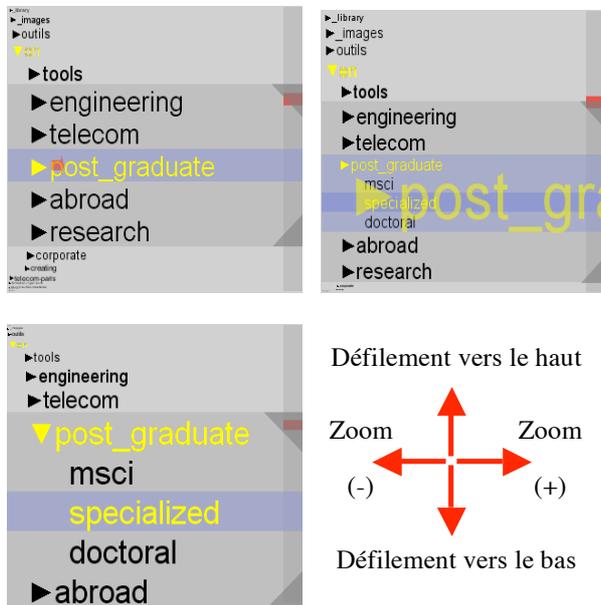


Figure 4 : zoom sémantique et interaction.

La figure 4 illustre cet effet. L’utilisateur zoome sur l’item « *post\_graduate* » (en haut à gauche). La taille de cet élément grossit et son contenu commence à apparaître en transparence (en haut à droite). A la fin de l’opération, l’intitulé initial est entièrement remplacé par le contenu du noeud (en bas à gauche). La même technique est appliquée dans le sens inverse : les branches de l’arborescence sont repliées en effectuant un zoom inverse sur la zone souhaitée.

Outre une meilleure cohérence visuelle, l’utilisation de la transparence a également un intérêt secondaire : elle permet d’« apercevoir » le contenu d’un noeud avant de l’avoir complètement ouvert. Ceci peut être particulièrement utile pour des utilisateurs faisant une utilisation fréquente mais non experte d’un arbre de données (ils ne se rappellent pas exactement où sont les noeuds mais re-

connaissent vite les différentes sous-parties). Enfin, cette technique permet de passer très rapidement de la vue de l’intitulé d’un noeud à celle de ses sous-éléments et ainsi de suite. L’utilisateur peut ainsi, généralement en un seul geste, déplier un noeud, voir si son contenu lui convient, et, selon le cas, continuer à déplier ou au contraire revenir en arrière. Non seulement l’opération de zoom sémantique assure une continuité spatiale qui facilite la compréhension par l’utilisateur de ce qu’il est en train de faire, mais lui évite aussi d’avoir à pointer et cliquer les interacteurs de petite taille servant habituellement à ouvrir ou fermer les branches des listes hiérarchiques.

En conclusion, notre système combine donc deux types de techniques en ce qui concerne les aspects dédiés à la visualisation des données : une représentation fish eye permettant d’afficher un grand nombre d’éléments dans la zone écran allouée combinée à une technique de zoom sémantique pour déplier les noeuds.

### Technique d’interaction

Comme nous l’avons déjà souligné à plusieurs reprises, la fluidité de l’interaction est essentielle pour accéder facilement aux éléments de l’arborescence. La technique d’interaction choisie est inspirée des « menus gestuels » de type *Control Menus* [7]. L’efficacité de ce type de technique, qui permet de sélectionner une commande et de la contrôler continûment en un seul geste a été montrée expérimentalement dans [3]. Dans leur implémentation complète, les Control menus permettent de sélectionner huit opérations disposées en cercle autour du point central autour duquel apparaît le menu. Davantage d’opérations peuvent être contrôlées si le menu dispose de sous-menus cascades. De même que les Marking menus, Les Control menus comportent normalement deux modes. Dans le mode novice, l’utilisateur appuie sur la souris et le menu apparaît au bout d’un bref délai. L’utilisateur bouge alors la souris dans la direction de l’opération souhaitée. Dès qu’il franchit une certaine distance, il sélectionne cette opération qu’il peut alors contrôler interactivement en continuant de déplacer la souris dans 1 ou 2 directions. Dans le mode expert, l’utilisateur fait exactement le même geste, mais plus vite. Dans ce cas, le menu n’apparaît pas et l’interaction devient purement gestuelle. Cette approche permet un apprentissage implicite du mode expert.

Les Control menus ont été par exemple utilisés pour contrôler le niveau de zoom et les déplacements dans le plan d’une interface zoomable [7]. L’opération de zoom était déclenchée par un mouvement horizontal, celle de déplacement par un mouvement initialement vertical (puis bidimensionnel dans le plan une fois l’opération sélectionnée). La même idée a été employée dans [4] pour contrôler le niveau d’échelle et le défilement vertical d’un texte. Dans ce cas, seules quatre opérations (verticales et horizontales mais pas diagonales) étaient utilisées. Enfin, l’utilisation des mouvements dans les direc-

tions orthogonales pour le contrôle continu à été employée d'une manière un peu différente dans [5] et, plus récemment, dans [1] pour des tâches de pointage dans un espace monodimensionnel de grande taille.

L'implémentation mise en oeuvre dans le cas de nos listes hiérarchiques n'utilise que quatre opérations : les déplacements verticaux de la souris contrôlent le choix du point de focus dans la liste tandis que les déplacements horizontaux contrôlent le niveau de zoom sémantique, donc le dépliage de l'arbre. Il est ainsi possible de balayer l'arborescence entière en parcourant une sorte de « chemin sinueux » avec la souris. Ce parcours peut-être effectué en un seul geste si l'arbre n'est pas trop grand et que l'on utilise un dispositif d'interaction indirect comme une souris. Il nécessitera quelques gestes supplémentaires dans le cas contraire (en particulier si le dispositif est direct et de petite taille, comme par exemple l'écran tactile d'un PDA).

## IMPLEMENTATION

### Zones de focus et de contexte

Notre représentation comporte trois zones : une zone de focus et deux zones de contexte (figure 5). La zone de focus contient les éléments intéressants choisis par les utilisateurs à un moment donné. Tous les éléments contenus dans cette zone ont la même taille verticale. Les tailles des éléments dans les zones de contexte varient en fonction de la distance entre ces éléments et la zone de focus (plus un élément est plus proche de cette zone, plus sa taille est grande). Ces tailles sont calculées par une fonction de déformation qui a la forme suivante :

$$F(x) = TF * \left( \frac{x}{NC} \right)^a \text{ avec : } a = \frac{TF * NC}{TF + T} - 1$$

où NC est le nombre d'éléments dans les zones de contexte haute et basse, T leur taille totale en pixels, TF la taille d'un élément dans la zone de focus (qui est fixée a priori) et X l'indice de l'élément dans la zone de contexte. De plus, le système impose que les tailles des éléments dans les zones de contexte restent toujours inférieures à celles des éléments dans la zone de focus.

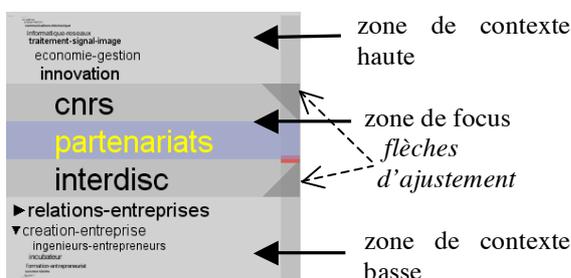


Figure 5 : Les trois zones du menu.

Le nombre d'éléments situés dans la zone de focus peut être ajusté en cliquant sur les flèches d'ajustement si-

tuées à droite de la représentation (figure 5). Ce nombre est toujours impair afin d'assurer une symétrie de présentation. Le chemin sélectionné est affiché en haut de la fenêtre pour indiquer comment on peut atteindre l'élément courant à partir de l'origine et afin d'éviter de se perdre dans l'espace d'information (Figure 6). De plus, tous les noeuds traversés sont affichés en jaune ce qui permet également de mettre en évidence le chemin sélectionné ainsi que les relations de parenté entre les noeuds concernés.

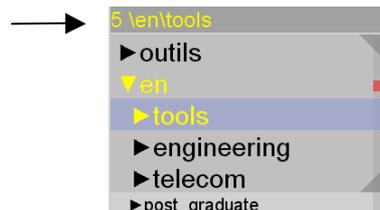


Figure 6 : Affichage du chemin sélectionné

## CONCLUSION ET PERSPECTIVES

Les applications envisagées concernent plus particulièrement les cas où l'espace écran est limité mais où il est nécessaire d'accéder rapidement à un nombre important d'éléments. C'est pourquoi cette technique semble bien adaptée aux dispositifs de petite taille à écran tactile comme les PDAs. Elle pourrait aussi constituer une bonne alternative aux menus que l'on trouve fréquemment dans les sites Web, afin de faciliter l'exploration de menus souvent touffus tout en minimisant l'espace écran requis (une condition indispensable pour éviter d'encombrer inutilement la page d'accueil. Un menu de ce type a d'ailleurs été implémenté pour représenter la hiérarchie des pages du site Web de l'ENST [8]. Enfin, les propriétés de cette nouvelle technique doivent être analysées plus finement et faire l'objet d'une comparaison quantitative avec les techniques classiques.

## BIBLIOGRAPHIE

1. Appert, C, Fekete JD. OrthoZoom Scroller: 1D Multi-Scale Navigation, Proc. CHI 2006 (à paraître)
2. Bederson, B. B., Fisheye Menus, Proc. ACM UIST 2000, pp. 217-226, ACM Press.
3. Guimbretiere F., Martin A., Winograd T. Benefits of Merging Command Selection and Direct Manipulation, ACM Trans. on Computer-Human Interaction (TOCHI) Vol. 12, Issue 3, pp. 460 – 476, 2005.
4. Lecolinet, A molecular architecture for creating advanced interfaces, CHI Letters. pp. 135-144. ACM Press 2003.
5. Masui T., LensBar – Visualization for Browsing and Filtering Large Lists of Data. Proc. Infoviz 1998.
6. Plaisant C., Grosjean J., Bederson B.B, SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Proc. InfoViz 2002.
7. S. Pook, E. Lecolinet, G. Vaysseix, E. Barillot Control Menus: Execution and Control in a Single Interactor.. Proc. CHI 2000, pp. 263-264. ACM Press, 2000.
8. Video : <http://www.enst.fr/~elc/FishTree>

