Naviguer dans des grands arbres avec ControlTree

Caroline Appert[1,2]

[1]LRI - Univ. Paris-Sud
Bât 490 - Orsay, F-91405, France
appert@lri.fr

Jean-Daniel Fekete[2,1]

[2]INRIA
Bât 490 - Orsay, F-91405, France
Jean-Daniel.Fekete@inria.fr

RESUME

Cet article présente ControlTree, une technique originale pour sélectionner un noeud dans une très grande hiérarchie en une interaction continue. ControlTree utilise une représentation noeud-lien qui s'adapte dynamiquement en fonction de la trajectoire du dispositif de pointage afin d'une part d'optimiser l'espace d'affichage et d'autre part de faciliter la sélection d'un noeud. ControlTree utilise un mécanisme de *snapping* et une adaptation de la technique OrthoZoom pour offrir une alternative aux habituelles techniques "pointage puis clic". Nous proposons également une évaluation théorique préliminaire montrant les bénéfices potentiels de ControlTree par rapport au populaire explorateur de fichiers de Microsoft.

MOTS CLES : Navigation, Sélection, Nœud-lien

ABSTRACT

This article presents ControlTree, a relevant technique to select a node is a very large node-link representation using a continuous interaction. The node-link representation is dynamic and driven by the cursor location to optimize screen space and to ease the selections. ControlTree uses a *snapping* mechanism and adapts the OrthoZoom technique to offer an alternative to common "point-and-click" techniques. We also present a preliminary theoretical evaluation showing that ControlTree is a promising technique.

CATEGORIES AND SUBJECT DESCRIPTORS: H.5.2 User Interfaces (H.1.2, I.3.6)

GENERAL TERMS: Design, Human Factors

KEYWORDS: Navigation, Selection, Node-link

INTRODUCTION

Naviguer pour sélectionner un nœud dans une hiérarchie est une tâche fréquente. Tout utilisateur est régulièrement amené à sélectionner un fichier dans son système de fichiers ou à sélectionner une commande dans un grand menu hiérarchique (par exemple, l'application Adobe Illustrator contient un menu intitulé "texte" divisé en trois niveaux hiérarchiques et dont les sous-menus peuvent contenir jusqu'à 150 items). De même, beaucoup de spécialistes manipulent des données organisées hiérarchiquement. Par exemple, un biologiste cherche régulièrement une espèce dans une taxonomie ou un bibliothécaire un livre dans une classification. Sur ces exemples, l'utilisateur cherche à sélectionner le plus rapidement possible un nœud dans une grande hiérarchie dont certaines parties lui sont familières. Les techniques courantes (Microsoft Windows Explorer et les menus hiérarchiques linéaires) privent l'utilisateur de son contexte de travail car elles utilisent une grande partie de la surface de travail et sont donc source d'erreurs et oublis.

ControlTree a pour but d'optimiser le temps nécessaire à la sélection d'un nœud mais également de minimiser la surface d'affichage utilisée afin de préserver la visibilité du contexte de travail. Il s'agit donc d'une part de choisir une bonne représentation et d'autre part d'avoir des interactions de navigation et de sélection efficaces. La représentation nœud-lien est la représentation qui reste la plus familière aux utilisateurs [8] mais elle a toujours été exploitée avec des interactions standards "pointage puis clic" pour naviguer et sélectionner. ControlTree utilise des interactions avancées pour naviguer et sélectionner dans une représentation nœud-lien dynamique (figure 1).

Après une brève revue des travaux antérieurs pour améliorer la visualisation et l'interaction dans des ensembles hiérarchiques, cet article décrit les interactions de ControlTree et présente une évaluation théorique préliminaire montrant les bénéfices de ControlTree par rapport au fameux explorateur de fichiers Microsoft.

TRAVAUX ANTERIEURS

La représentation classique d'un arbre sous forme nœud-lien nécessite une surface d'affichage importante. Elle devient illisible bien avant que le nombre de nœuds n'excède le nombre de pixels. Dès lors, la majorité des solutions à la navigation dans de grands arbres repose sur une technique de navigation dans une interface dont l'affichage est dynamiquement ajusté.

Les techniques de visualisation dynamique d'une grande hiérarchie comme Windows Explorer, Hyperbolic Tree Browser [7] ou SpaceTree [8] proposent des visualisations nœud-lien intéressantes mais reposent toutes sur une in-

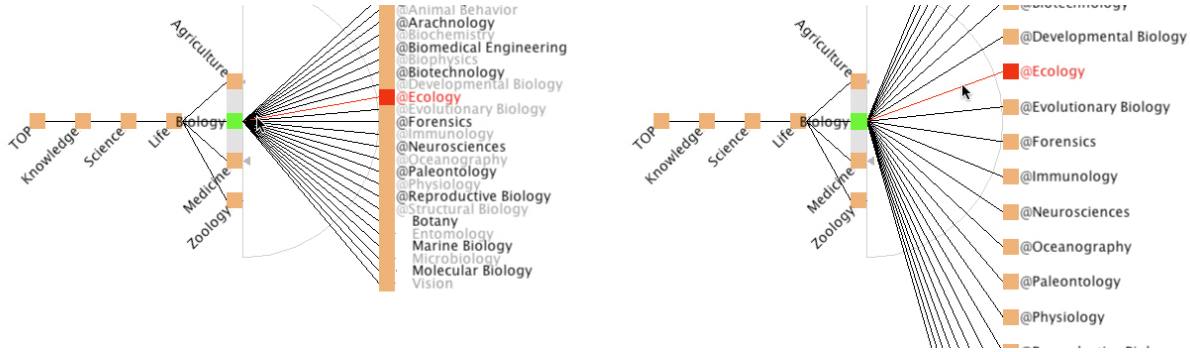

Figure 1 **:** L'interface de ControlTree

teraction classique "pointage puis clic" pour naviguer et sélectionner. Ces techniques ont été conçues dans un but plus large que la sélection d'un nœud et, en particulier, pour des tâches cognitives plus complexes comme la comparaison de plusieurs nœuds, la recherche d'un ancêtre commun, etc. Les évaluations qui comparent ces visualisations [8, 5] ont toutes révélé qu'il n'existe pas de meilleure visualisation pour toutes les tâches.

Au niveau de l'interaction pour les tâches de sélection d'un nœud, seuls les travaux autour des menus hiérarchiques, qui sont bien souvent des hiérarchies relativement petites, ont réellement innové. Les deux travaux les plus innovants sont les *pie menus* [3], qui peuvent être augmentés par un mécanisme de reconnaissance gestuelle pour obtenir des *marking menus* [6], et les menus de crossY [1] qui se déclenchent lors d'un geste de tracé lorsqu'on entre puis qu'on sort de l'item de menu. Les *pie menus* sont des menus circulaires qui permettent d'obtenir un temps de sélection constant quel que soit l'élément sélectionné. Cependant, ces menus ne passent pas à l'échelle : ils sont efficaces lorsque le nombre d'éléments est pair et n'excède pas 8 par niveau. CrossY n'aborde que les listes textuelles et change la structure de la hiérarchie sous-jacente puisqu'il réorganise les éléments d'un même niveau selon une hiérarchie basée sur l'ordre lexicographique afin de pouvoir sélectionner un nœud lorsque la taille de la liste dépasse la taille de l'affichage disponible.

PLACEMENT

ControlTree utilise une représentation nœud-lien horizontale qui évolue continûment au cours de la navigation de l'utilisateur. À partir du nœud courant, ControlTree affiche les fils directs, le chemin et les frères directs afin d'offrir uniquement le contexte nécessaire à la navigation sans surcharger l'espace disponible. L'affichage est continuement mis à jour au cours de la navigation et offre la possibilité à chaque instant de naviguer dans trois zones : la zone des parents (*parent zone*), la zone des frères (*sibling zone*) et la zone des fils (*child zone*).

Disposer les étiquettes des nœuds pour une représentation nœud-lien est un problème difficile. La figure 2 montre l'affichage que nous avons choisi. L'arbre est affiché hor-

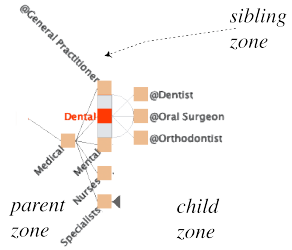

Figure 2 **:** Nœud-lien dynamique de ControlTree

izontalement et les nœuds d'un même niveau sont alignés verticalement pour améliorer la lisibilité des relations. Cet affichage permet d'afficher horizontalement les étiquettes des fils directs du nœud courant (sur la figure 2, le nœud courant est *Dental*) et d'offrir ainsi une très bonne lisibilité pour ceux-ci. Cependant, afficher des étiquettes horizontales pour les frères et les nœuds du chemin sans occlusion demande d'utiliser des algorithmes de placement avancés et, bien souvent, l'affichage produit occupe beaucoup de place à l'écran. Pour résoudre ce problème, ControlTree affiche les étiquettes des nœuds des zones des parents et des frères avec un angle de 45 degrés. Cet affichage a également la propriété de ne pas obstruer la zone d'interaction la plus courante : celle qui va du nœud courant à un de ces fils. En effet, alors que les étiquettes des fils sont alignées à gauche, les étiquettes des parents et frères sont alignées à droite.

LES SELECTIONS FACILES SONT TRIVIALES

ControlTree prend en compte la position relative du curseur par rapport au nœud courant pendant l'interaction pour faciliter l'acquisition du prochain nœud. En effet, le nœud courant est entouré de zones actives qui déclenchent différentes actions (zones grisées de la figure 3). L'action dépend de la position du curseur par rapport à ces zones :

1. Dans la zone des parents, si l'utilisateur va au-delà du seuil vertical t_{parent}, le nœud parent est sélectionné par un mécanisme d'attraction (*snapping* : le nœud se déplace pour venir sous le curseur) ; sinon le nœud parent est mis en surbrillance.

2. Dans la zone des frères, si l'utilisateur va au-delà du seuil horizontal $t_{sibling}$ du haut (resp. bas), le frère du haut (resp. bas) est sélectionné par un mécanisme de *snapping* ; sinon le frère du haut (resp. bas) est mis en surbrillance.

3. Dans la zone des fils, si l'utilisateur va au-delà du seuil semi-circulaire t_{child}, le fils le plus proche est sélectionné par un mécanisme de *snapping* ; sinon le fils le plus proche est mis en surbrillance.

La *distance à un fils* c est la distance entre le curseur et la ligne qui représente le lien du nœud courant vers c. Le *fils le plus proche* est donc celui qui minimise cette distance par rapport à la position courante du curseur.

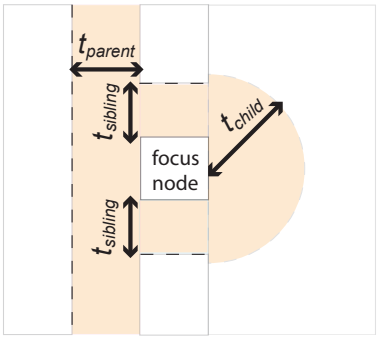

Figure 3 **:** Zones d'interaction autour du nœud courant

L'affichage horizontal de ControlTree peut entraîner des distances élevées entre le nœud courant et les fils qui sont aux extrémités. Le mécanisme de *snapping* permet de résoudre ce problème en "attirant" le nœud sous le curseur au lieu que le curseur aille au nœud. Cependant, le *snapping* n'est approprié que si la probabilité de faire une mauvaise anticipation est faible (c'est-à-dire lorsque la difficulté de sélection est assez faible) afin d'éviter les erreurs. La prochaine section décrit l'adaptation de notre technique OrthoZoom [2] afin de contrôler l'affichage et que les sélections restent dans un intervalle de difficulté raisonnable.

LES SELECTIONS DIFFICILES SONT POSSIBLES

ControlTree prend en compte la trajectoire du curseur pour faciliter l'acquisition d'un nœud dans les cas difficiles. En effet, il n'existe pas de valeur unique pour t_{child} qui facilite la sélection dans tous les cas : la précision angulaire requise pour sélectionner un nœud augmente avec le degré de son nœud parent. Pour contourner ce problème, la figure 4 illustre deux options envisageables : (i) augmenter l'espacement entre les nœuds d'un même niveau ou (ii) augmenter la valeur du seuil t_{child}. Prises indépendamment, ces solutions sont assez simplistes puisque dans les deux cas, l'espace d'affichage disponible est une limite rapidement atteinte : pour (i), il y a plus d'espace consommé verticalement (de plus, certaines sélections sont plus faciles que d'autres) et pour (ii), il y a plus d'espace consommé horizontalement.

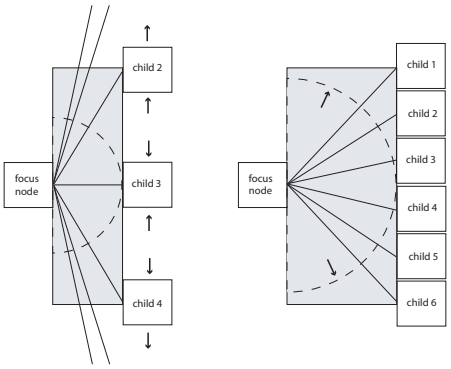

Figure 4 **:** Stratégies simplistes pour faciliter une sélection

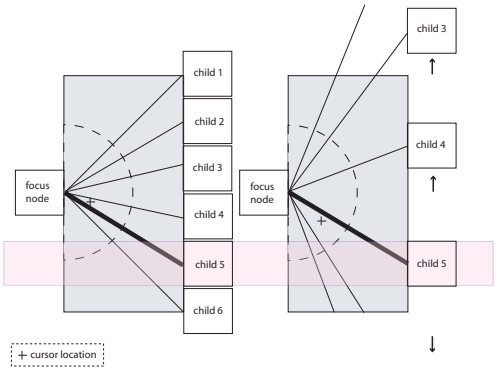

Figure 5 **:** Stratégie de ControlTree pour faciliter une sélection

ControlTree propose une solution plus satisfaisante en adaptant notre technique OrthoZoom aux représentations nœud-lien. OrthoZoom est une technique de pointage multi-échelle 1D qui utilise un périphérique de pointage 2D standard. Elle permet de contrôler le facteur de zoom selon la dimension orthogonale (l'abscisse de la souris par exemple) à celle utilisée pour la navigation (l'ordonnée de la souris par exemple). ControlTree de son côté utilise la distance entre le curseur et le nœud courant pour contrôler l'espacement entre les nœuds et la direction pour choisir les nœuds qui restent dans la surface d'affichage. Cette approche résout les deux problèmes exposés précédemment en contrôlant dynamiquement le nœud-cible potentiel et la précision angulaire autour de ce nœud en fonction du mouvement de l'utilisateur. À chaque mouvement du curseur, ControlTree cherche le nœud le plus proche c et ajuste l'espacement autour de c tout en préservant la position de c. c repousse progressivement ses frères pour être plus facilement accessible (figure 5). Ainsi, lors de la navigation, l'utilisateur garde le regard sur le prochain nœud à atteindre et suit la ligne droite qui l'y mène.

Bien que ControlTree minimise l'affichage occupé, l'utilisateur peut atteindre les bords du dispositif d'affichage. Il peut alors déplacer la totalité de l'arbre par un drag (l'arbre s'actualisant continûment en fonction de

sa nouvelle position). Il peut aussi utiliser la roulette de la souris pour faire défiler les fils du nœud courant.

ALGORITHME

Bien que cette adaptation permette d'atteindre plus facilement un nœud même pour de grands degrés, la difficulté augmente avec le degré du nœud si la valeur de t_{child} est identique pour tous les nœuds. En effet, t_{child} définit l'aire de contrôle disponible pour contrôler l'espacement entre les nœuds et influe donc sur l'utilisabilité. Par exemple, l'utilisateur peut être déstabilisé si, dans le cas d'un nœud ayant un grand degré, un déplacement d'un pixel fait passer le nombre de nœuds visibles de 20 à 5. De plus, si le rapport entre le degré et la taille de l'aire devient très grand, les nœuds peuvent sortir de la surface d'affichage avant que l'utilisateur n'ait eu une chance de les atteindre. Pour résoudre ce problème, l'algorithme de ControlTree ajuste la valeur de t_{child} en fonction du degré du nœud courant afin de fournir une difficulté bornée pour tous les nœuds de l'arbre. L'algorithme de ControlTree dépend de l'espacement minimal et maximal entre les nœuds, S_{min} et S_{max}.

Lorsqu'un nœud n de degré $\deg_n$ devient le nœud courant, t_{child} doit vérifier la condition (1) pour garantir que chaque nœud puisse être atteint :

$$t_{child} = x \times d \text{ avec } r^x \leq \deg_n \qquad (1)$$

Donc, pour déterminer le placement des fils de n qui est situé en (x_n, y_n) dans une surface d'affichage de dimension $w \times h$, ControlTree calcule la plus petite valeur de x qui vérifie (1) puis t_{child} qui est donc une fonction logarithmique de $\deg_n$. ControlTree peut alors afficher les fils de n en fonction de t_{child} et de la surface disponible.

1. Calculer la valeur minimale de t_{child} qui vérifie (1) ;
2. Calculer la hauteur maximale disponible : $h_{max} = \max(y_n, (h - c_y))$
3. Calculer l'espace courant s entre les fils de n : $s = \begin{cases} h_{max}/\deg_n & \text{si } h_{max}/\deg_n \geq S_{min} \\ S_{min} & \text{sinon} \end{cases}$

Une fois ce placement initial fait, lorsque le curseur est à une distance d de n, l'espacement entre les nœuds vaut : $s + D \times \Delta$ avec $\Delta = (S_{max} - s)/t_{child}$

CONTROLTREE vs. EXPLORER

Dans cette section, nous comparons théoriquement ControlTree par rapport à Windows Explorer (figure 6), qui est la technique la plus utilisée pour naviguer dans des arbres. ControlTree ne peut être analysé avec le modèle KLM [4] car la difficulté de sélection à chaque niveau est une fonction complexe du degré du niveau. Cependant, nous pouvons comparer les distances parcourues par le curseur dans les deux cas.

Supposons que $\bar{n}$ soit le degré moyen d'un nœud dans un arbre. En utilisant ControlTree, suivre un chemin de longueur d (c.à.d. d niveaux hiérarchiques) requiert de parcourir une distance $d \times \log(\bar{n})$ dans tous les cas. Avec Microsoft Windows Explorer, la distance est $d \times \bar{n}$ dans le cas où seulement les nœuds du chemin sont ouverts, par exemple le nœud "United Kingdom" dans la partie gauche de la Figure 6. Cependant, elle est de $\bar{n}^d$ dans le cas où tous les nœuds y compris ceux qui n'appartiennent pas au chemin sont ouverts. Les distances parcourues sont donc plus petites avec ControlTree qu'avec Explorer.

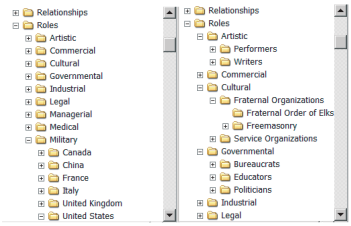

Figure 6 : Microsoft Windows Explorer

CONCLUSION

ControlTree est une interface originale pour naviguer et sélectionner des nœuds dans de grandes hiérarchies qui évoluent dynamiquement en fonction de la navigation. Les analyses préliminaires montrent les gains potentiels de cette technique en termes de distances parcourues. Nous projetons d'évaluer les performances de façon plus approfondie en menant une expérimentation contrôlée.

BIBLIOGRAPHIE

1. G. Apitz and F. Guimbretière. CrossY: a crossing-based drawing application. *Proc. ACM UIST '04*, pages 3–12, 2004.
2. C. Appert and J. Fekete. Orthozoom scroller: 1d multi-scale navigation. In *Proc. ACM CHI '06*, pages 21–30, New York, NY, USA, 2006. ACM Press.
3. J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. *An empirical comparison of pie vs. linear menus*. ACM Press New York, NY, USA, 1988.
4. S. Card, T. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Commun. ACM*, 23(7):396–410, 1980.
5. A. Kobsa. User Experiments with Tree Visualization Systems. *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 9–16, 2004.
6. G. Kurtenbach, A. Sellen, and W. Buxton. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction*, 8(1):1–23, 1993.
7. J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. ACM CHI '95*, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
8. C. Plaisant, J. Grosjean, and B. Bederson. SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. *Proc. IEEE INFOVIS 2002*, pages 57–64, 2002.