

Mémoire Spatiale & Gestion de Fenêtres

Guillaume Faure

Université Paris Sud 11 & CNRS, INRIA
LRI, Orsay, France
guillaume.faure@lri.fr

RESUME

Lors de l'usage d'un système informatique dont l'interface utilisateur est un environnement de bureau virtuel, l'utilisateur doit gérer un nombre toujours croissant de fenêtres (documents, applications). Pourtant, les outils qui lui sont fournis pour exécuter cette tâche complexe ont peu évolués depuis les 30 dernières années. Afin de réduire la charge cognitive assumée par l'utilisateur lors de son interaction avec le système nous proposons d'utiliser le *Contexte de Travail* : un ensemble d'informations assimilé par l'utilisateur, mais n'ayant pas trait à sa tâche principale. Nous nous intéressons particulièrement à la Mémoire Spatiale qui regroupe les informations relatives au placement des objets à l'écran. Nous présentons ici des techniques d'interaction utilisant le *Contexte de Travail* à leur avantage, ainsi qu'un cas d'étude du *Contexte Spatial*.

MOTS CLES : Contexte, tâches, Bureau Virtuel, glisser-déposer

ABSTRACT

When using a computer following the Desktop metaphor, the user manages an ever increasing number of documents and windows. Unfortunately, the available tools to achieve this task have barely changed for the last 30 years. In order to reduce the user's cognitive load when using the system, we propose to take advantage of the Working Context: information known to the user, but that does not belong to the main task. We mainly focus on the Spatial Context which contains the location of onscreen objects. We present some interaction techniques taking into account the Working Context and a case study of the Spatial Context.

CATEGORIES AND SUBJECT DESCRIPTORS: H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces, Theory and methods

GENERAL TERMS: Human factors

KEYWORDS: Context, task, Desktop metaphor, drag-and-drop

Introduction

À l'heure actuelle, la majorité des interfaces utilisateur dérivent du Bureau Virtuel. Cette famille d'interfaces, basée sur la métaphore du bureau, est apparue avec le Xerox Alto. Dès le début des années 80, son adoption par l'informatique grand publique le place en standard de fait. Ce type d'environnement adopte le paradigme dit Windows, Icons, Menus and Pointing (WIMP). Les fenêtres, redimensionnables et pouvant se chevaucher, permettent l'exécution concurrente de plusieurs programmes. Ces fenêtres constituent un espace d'affichage propre à chaque application ou document ouvert.

Lors de l'avènement des environnements de bureaux virtuels, la puissance des machines et les possibilités réduites offertes par les logiciels limitaient le nombre de fenêtres affichées simultanément. Cependant, la puissance de calcul disponible a augmenté, les usages de l'outil informatique se sont diversifiés et l'espace d'affichage disponible s'est accru. Les utilisateurs sont de ce fait plus à même de mener plusieurs tâches de front sur un même système et à conserver plus de fenêtres affichées simultanément. Ces évolutions ont entraîné la modification progressive des habitudes de gestion de fenêtres. Il semble être, par exemple, de moins en moins d'utilisateurs dimensionnant systématiquement leurs fenêtres de façon à couvrir entièrement l'espace disponible [7]. Enfin, les applications elles-mêmes se sont enrichies introduisant de nouveaux moyens de gérer de multiples documents en leurs sein même (onglets, vues séparées, espaces de docking...) ¹. L'ensemble de ces facteurs fait de la navigation dans l'interface utilisateur une tâche complexe et coûteuse en temps.

Lors de l'usage d'un système informatique l'utilisateur dispose, en plus des connaissances requises pour mener à bien sa tâche principale, d'un ensemble d'informations n'ayant pas trait à cette dernière. Cependant ces informations peuvent être importantes lors de la manipulation d'objets nécessaires à la réalisation de cette tâche. Nous appelons ici cet ensemble d'informations le *Contexte de Travail*. Nous pressentons que réutiliser ces informations doit pouvoir aider l'utilisateur dans la gestion de son espace de travail et rendre son interaction avec le système plus satisfaisante.

¹voir par exemple <http://dubroy.com/blog/2009/01/29/my-talk-at-mozilla/>.

Travaux Antérieurs

Quelques travaux se sont déjà penchées sur l'utilisation du *Contexte de Travail* pour faciliter l'usage des environnements de bureaux virtuels. Un exemple classique est l'introduction de la notion "d'activité". Cette dernière se caractérise par les liens existant, dans le but d'achever une tâche, entre des outils, des documents et des personnes [13]. Le premier système qui proposa de regrouper les fenêtres en activités, afin d'explicitier ces liens, fût Rooms [6]. Plus récemment, des systèmes comme Giornata [13] ou WindowScape [12] se focalisent sur la souplesse d'interaction requise pour la gestion des activités et ne se limitent plus au simple regroupement de fenêtres. D'autres, comme GroupBar [8], s'efforcent de conserver le plus possible les habitudes des utilisateurs en augmentant une interaction classique du changement de fenêtre. Finalement, certains systèmes à l'instar de ScalableFabric [10] et WindowScape [12] utilisent le *Contexte Spatial* d'une activité (disposition des fenêtres composant l'activité) afin l'identifier celle-ci lors des changements d'activité. La réification de ces liens entre éléments formant une activité a par exemple pour avantage de diminuer le coût d'un changement d'activité.

Une autre composante du *Contexte de Travail* est le *Contexte Spatial*. Il fait appel à deux types de mémoire. La première est la mémoire spatiale à long terme. Cette mémoire est forgée par l'habitude. Il est maintenant bien établi dans la communauté IHM que la modification de la disposition des objets graphiques au cours du temps dégrade les performances [5]. Sans la stabilité des éléments graphiques dans l'espace, l'utilisateur est dans l'incapacité de créer des automatismes. C'est également cette mémoire qui permet aux utilisateurs de retrouver facilement leurs documents dans des gestionnaires de fichiers spatiaux comme le Finder ou des environnements virtuels 3D [9]. La seconde forme de mémoire est la mémoire spatiale à court terme. Elle est, par exemple, construite et utilisée lors de recherches visuelles. Cette mémoire, volatile et limitée, enregistre les positions relatives entre éléments [11]. Conserver la disposition des éléments lorsqu'une recherche visuelle est à prévoir peut améliorer les performances en permettant une automatisation plus aisée des actions répétitives.

Le but de ma thèse est de mieux comprendre le *Contexte de Travail* dans le cadre des gestionnaires de fenêtres et des environnements de bureaux virtuels. Je souhaite tout particulièrement explorer le *Contexte Spatial*. À ma connaissance, bien que la mémoire à long terme soit convenablement exploitée par les chercheurs en IHM, la mémoire spatiale à court terme reste encore sous-exploitée.

Techniques d'interaction

J'ai tout d'abord conçu, en collaboration avec d'autres chercheurs, plusieurs techniques d'interaction dans le cadre des environnements de bureaux virtuels [4]. Ces

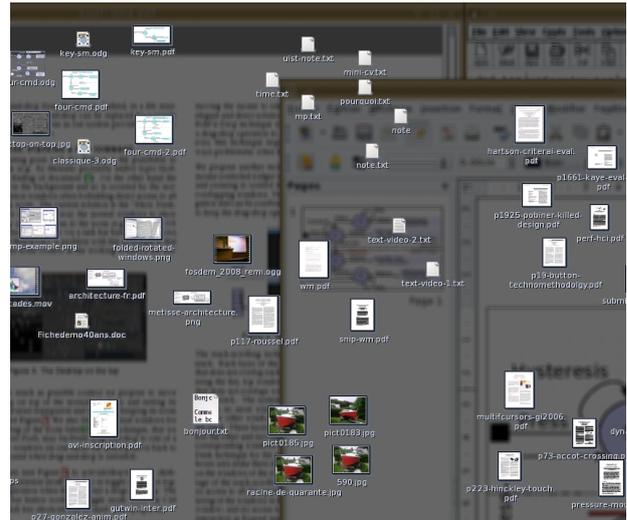


Figure 1 : Desk-Pop : le Bureau est placé au dessus des fenêtres. Le fond du Bureau est rendu semi-transparent et légèrement flouté. L'utilisateur peut alors voir à la fois le contenu du Bureau et les fenêtres que ce dernier recouvre.

techniques sont implémentées dans le système Metisse [1] et prolongent des travaux initiés dans [2].

Le premier outil proposé est un mécanisme d'historique pour les actions dites de "copier-ou-déplacer". Les dernières actions de "copier-coller", de "glisser-déposer" et les dernières sélections sont enregistrées par le système. Nous avons pris un soin tout particulier à concevoir la partie interaction de cet outil. Du fait de leur utilisation fréquente certaines interactions, comme le "copier-coller", entrent dans les habitudes de l'utilisateur. C'est pourquoi nous avons étendu, lorsque cela était applicable, l'interaction classique de chaque technique pour exploiter l'historique correspondant. De plus, ces techniques ne perturbent que faiblement la tâche principale puisqu'elles ne requièrent pas de navigation complexe dans l'interface utilisateur. L'ensemble de l'interaction prend place localement, au point d'attention de l'utilisateur.

Le second outil, Desk-Pop (Figure 1) est un croisement entre une visualisation et une technique d'interaction servant à donner à l'utilisateur l'accès à son Bureau². Afin de tirer parti de la mémoire spatiale de l'utilisateur, le placement des fenêtres reste inchangé. Nous plaçons le Bureau au dessus de celles-ci et appliquons une série d'effets graphiques (semi-transparence + flou) afin que le contenu du Bureau et les fenêtres qu'il recouvre soient visibles simultanément.

La dernière interaction que nous avons conçue, *Stack-Leafing*, permet la navigation entre fenêtres, Bureau inclus. Encore une fois, plutôt que de déplacer les fenêtres ou de leur appliquer une transformation, nous conservons leur disposition et ne modifions que leur empilement. Le principe de fonctionnement de *Stack-Leafing* est présenté

²Ici, le Bureau est défini comme l'espace se trouvant à l'arrière-plan et contenant des icônes.

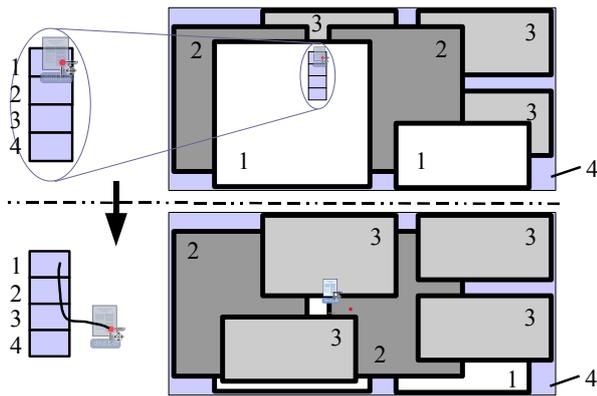


Figure 2 : Stack-leaving. : les fenêtres ne se chevauchant pas sont regroupées en une couche. Les couches ainsi créées sont elles-mêmes regroupées en une pile. Le Bureau constitue une couche à part entière se trouvant au dessous de la pile. Un widget d'interaction (à gauche) permet de naviguer entre les couches. Quitter ce widget préserve la dernière couche sélectionnée.

à la Figure 2. Nous faisons l'hypothèse que la conservation de ces informations spatiales permet une navigation plus aisée et une recherche plus rapide de la fenêtre d'intérêt.

Pendant la création de ces outils, nous nous sommes attachés à suivre deux lignes directrices : *i*) Nos outils sont conçus pour fonctionner sans configuration ou planification préalable. Nous cherchons à diminuer la charge assumée par l'utilisateur, non à le surcharger. *ii*) Nos interactions sont conçues pour s'intégrer le plus naturellement possible dans le flot de travail de l'utilisateur et conservent le *Contexte Spatial* autant que faire se peut. Ces deux points devraient permettre une plus grande efficacité et un confort accru pour l'utilisateur.

La Mémoire Spatiale : Un Cas d'Étude

Si le *Contexte Spatial* est un concept souvent évoqué au sein de la recherche en IHM, des techniques d'interaction comme *Exposé* d'Apple ou la technique *Show Desktop* utilisée couramment dans des systèmes comme Microsoft Windows invalident la connaissance que l'utilisateur a de l'organisation des fenêtres. Les transitions entre *Contextes Spatiaux* s'effectuent différemment selon la technique utilisée. Là où *Show Desktop* propose une restauration brutale des fenêtres, *Exposé* emploie une animation. Au contraire, la technique *Desk-Pop* permet de conserver l'organisation des fenêtres et préserve donc une partie du *Contexte Spatial*.

Afin d'appréhender l'impact sur le pointage à la souris de ces modifications de *Contexte Spatial* et des transitions les accompagnant, nous avons mené une expérimentation [3]. Le protocole expérimental s'assure tout d'abord de la mémorisation de la position de la cible principale de la part du sujet. Ce dernier doit ensuite effectuer une tâche de pointage. Durant cette phase, la cible peut se comporter de trois façons afin de modéliser le comportement des

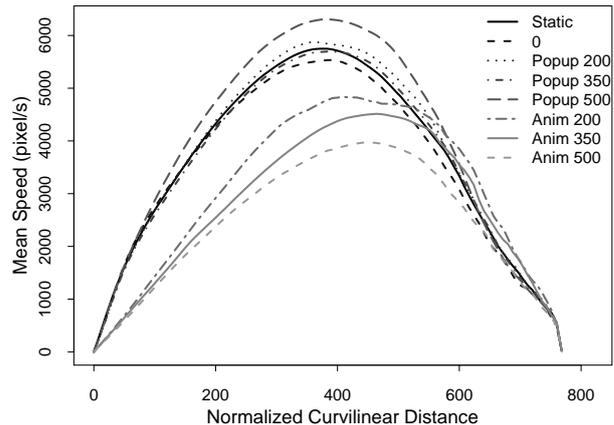


Figure 3 : Courbes des vitesses / distance parcourue (amplitude de 768 pixels). Les courbes représentant les délais 0 et Statique et "Popup" présentent des profils de vitesse similaires. Les conditions d'animation, pour leurs part, ont un pic de vitesse moindre plus vers la fin du mouvement.

techniques discutées ci-dessus : *i*) disparaître puis reprendre sa position par une translation d'une certaine durée : le **Délai** (Exposé d'Apple) ; *ii*) disparaître puis surgir à son emplacement initial après un certain **Délai** (Show Desktop). *iii*) rester affichée à sa position (Desk-Pop).

L'analyse des résultats montre que pour des délais allant jusqu'à 200 ms le temps de pointage n'est pas significativement influencé par le comportement de la cible. Cependant, la modification du *Contexte Spatial* entraîne une augmentation significative du nombre d'erreurs. Nous avons également étudié les courbes de vitesse par rapport à la distance parcourue au cours de la phase de pointage (Figure 3). Dans le cas des cibles surgissantes, les profils sont similaires à ceux d'un pointage classique : une cloche dont la pointe de vitesse se situe aux environs de la moitié du mouvement. Au contraire, dans le cas d'un pointage sur une cible animée, les profils sont plus aplatis et le pic de vitesse se déporte vers la fin du mouvement. Cela indique que malgré des temps proches, le pointage sur des cibles surgissantes et sur des cibles animées diffère dans leur exécution.

Cette expérimentation suggère qu'une technique telle que *Desk-Pop* engendrerait moins d'erreurs de pointage, la rendant moins frustrante pour l'utilisateur. Nous avons également constaté qu'à partir de 350 ms, la transition entre deux *Contextes Spatiaux* affecte les performances de pointage. Cependant, pour une technique comme *Exposé* ces 350 ms sont probablement nécessaires à l'utilisateur pour appréhender l'animation. Une interaction préservant le *Contexte Spatial*, comme le fait *Desk-pop*, peut être avantagée par l'absence de transition.

Conclusion : Discussions et Perspectives

Mon travail de thèse a commencé par l'étude des environnements de bureaux virtuels et de leurs limitations. Cette analyse m'a permis de découvrir un certain nombre de fac-

teurs regroupés ici sous l'appellation de *Contexte de Travail*. Après un examen des outils de gestion de fenêtres actuels, il m'est apparu qu'une meilleure prise en compte de ce *Contexte de Travail* pouvait alléger la charge cognitive de l'utilisateur. En particulier, la réutilisation et simplification de ce *Contexte de Travail* sont deux notions que je voudrais approfondir.

Je me suis ensuite focalisé sur la partie spatiale du Contexte de Travail. Ce *Contexte Spatial*, bien que connu de la recherche en IHM me semble peu exploité à l'heure actuelle dans le cadre de la gestion de fenêtres. L'être humain a développé la faculté de mémoriser facilement l'emplacement d'objets dans le monde réel. Il est probable qu'il en soit de même pour les objets constituant un environnement de bureau virtuel. J'ai donc conçu des techniques de navigation tirant parti de cette mémoire spatiale dans le but d'accroître le confort de l'utilisateur, faciliter la recherche et garder l'utilisateur en contexte.

J'envisage plusieurs axes pour la suite de mes travaux de thèse. Je vais tout d'abord continuer à travailler sur l'aspect spatial du *Contexte de Travail*. La technique *Stack-Leafing* reste à évaluée. Il est important d'en comparer les performances par rapport à d'autres techniques de navigation. Je veux également terminer la caractérisation de la Mémoire Spatiale, afin de comprendre les aspects importants à prendre en compte pour des environnements de bureaux virtuels. Pour cela, il me faut identifier les informations enregistrées par l'utilisateur. Par exemple, l'utilisateur conserve-t-il en mémoire la profondeur à laquelle se trouve une fenêtre ? Parallèlement, la représentation de la profondeur peut-elle être enrichie pour en permettre une meilleure appréciation par l'utilisateur ?

La mise en parallèle et l'étude des notions d'onglets et de Bureaux Virtuels Multiples me paraissent intéressantes dans le cadre de la navigation dans l'interface. Bien que toutes les deux servent à regrouper des contenus, les onglets permettent de concentrer l'information, alors que les Bureaux Virtuels Multiples permettent de l'aérer. Cependant, ces outils sont séparés et n'ont, la plus part du temps, qu'une relation ténue avec le système de fenêtrage. Les onglets, par exemple, sont le plus souvent gérés directement par les applications, chacune d'elles proposant ses propres mécanismes. Ceci accroît le problème de la navigation déjà évoqué précédemment.

Le *Contexte de Travail* en général et le *Contexte Spatial* en particulier, bien que connus des chercheurs en IHM, n'ont peut-être pas encore exprimés tout leur potentiel. J'espère qu'une exploration plus en profondeur suivie d'une exploitation raisonnée de ces outils pourra augmenter le confort d'utilisation de l'outil informatique.

BIBLIOGRAPHIE

1. Chapuis, O., and Roussel, N. Metisse is not a 3d desktop! In *Proceedings of UIST '05*, pages 13–22,

2005. ACM.

2. Chapuis, O., and Roussel, N. Copy-and-paste between overlapping windows. In *Proceedings of CHI '07*, pages 201–210, 2007. ACM.
3. Faure, G., Chapuis, O., and Beaudouin-Lafon, M. Acquisition of animated and pop-up targets. In *Proceedings of INTERACT '09*, pages 372–385, 2009. Springer Verlag and IFIP.
4. Faure, G., Chapuis, O., and Roussel, N. Power tools for copying and moving: Useful stuff for your desktop. In *Proceedings of CHI '09*, pages 1675–1678, 2009. ACM.
5. Findlater, L., Moffatt, K., McGrenere, J., and Dawson, J. Ephemeral adaptation: the use of gradual onset to improve menu selection performance. In *Proceedings of CHI '09*, pages 1655–1664, 2009. ACM.
6. Henderson, Jr., D. A., and Card, S. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Trans. Graph.*, 5(3):211–243, 1986.
7. Hutchings, D. R., and Stasko, J. Revisiting display space management: understanding current practice to inform next-generation design. In *Proceedings of GI '04*, pages 127–134, 2004. Canadian Human-Computer Communications Society.
8. Patrick, G. S., Baudisch, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D., and Andrews, D. Groupbar: The taskbar evolved. In *Proceedings of OZCHI 2003*, pages 34–43, 2003.
9. Robertson, G., Czerwinski, M., Larson, K., Robbins, D. C., Thiel, D., and van Dantzich, M. Data mountain: using spatial memory for document management. In *Proceedings of UIST '98*, pages 153–162, 1998. ACM.
10. Robertson, G., Horvitz, E., Czerwinski, M., Baudisch, P., Hutchings, D. R., Meyers, B., Robbins, D., and Smith, G. Scalable fabric: flexible task management. In *Proceedings of AVI '04*, pages 85–89, 2004. ACM.
11. Shen, Y. J., and Jiang, Y. V. Interrupted visual searches reveal volatile search memory. *Journal of Experimental Psychology: Human Perception & Performance*, 32(5):1208–1220, 2006.
12. Tashman, C. Windowscape: a task oriented window manager. In *Proceedings of UIST '06*, pages 77–80, 2006. ACM.
13. Volda, S., Mynatt, E. D., and Edwards, W. K. Reframing the desktop interface around the activities of knowledge work. In *Proceedings of UIST '08*, pages 211–220, 2008. ACM.