



HAL
open science

Redirection d'applications existantes et nouvelles interactions pour des usages collaboratifs co-localisés sur une table interactive

Guillaume Besacier, Frédéric Vernier, Olivier Chapuis, Nicolas Roussel

► To cite this version:

Guillaume Besacier, Frédéric Vernier, Olivier Chapuis, Nicolas Roussel. Redirection d'applications existantes et nouvelles interactions pour des usages collaboratifs co-localisés sur une table interactive. IHM '07: Proceedings of the 19th International Conference of the Association Francophone d'Interaction Homme-Machine, Nov 2007, Paris, France. pp.271-274, 10.1145/1541436.1541491 . hal-00539224

HAL Id: hal-00539224

<https://hal.science/hal-00539224>

Submitted on 24 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Redirection d'applications existantes et nouvelles interactions pour des usages collaboratifs co-localisés sur une table interactive

Guillaume Besacier^{1,2} Frédéric Vernier¹
guillaume.besacier|frederic.vernier@limsi.fr

Olivier Chapuis^{2,3} Nicolas Roussel^{2,3}
chapuis|roussel@lri.fr

¹ LIMSI-CNRS
BP 133
91403 Orsay Cedex, France

² LRI – Univ. Paris-Sud & CNRS
Bâtiment 490, Université Paris-Sud
91405 Orsay Cedex, France

³ INRIA
91405 Orsay Cedex, France

RESUME

Cet article décrit un module logiciel et un ensemble de techniques d'interaction permettant l'utilisation à plusieurs d'applications existantes non modifiées sur une table interactive. Nous décrivons tout d'abord le module logiciel que nous avons développé pour permettre l'interaction avec des applications X Window sur une table, à travers le système Metisse et la boîte à outils Diamond-Spin. Nous présentons ensuite plusieurs problèmes spécifiques à la collaboration co-localisée sur une table interactive et proposons enfin plusieurs techniques d'interaction pouvant être spécifiquement mises en oeuvre pour permettre l'utilisation collaborative de ces applications initialement mono-utilisateur et mono-pointeur.

MOTS CLES : Table interactive, X Window, boîte à outils.

ABSTRACT

This paper describes a software component and several interaction techniques enabling existing non-modified desktop applications to be used by several users on a tabletop system. We describe our software implementation, design to redirect X Window applications to tabletop using the Metisse server and the DiamondSpin toolkit. We then present some co-located collaborative work problems, and suggest several interaction techniques that can be used to allow collaborative use of these single-user single-pointer applications.

CATEGORIES AND SUBJECT DESCRIPTORS: H.5.2 [User Interfaces]: Graphical user interfaces (GUI), Interaction styles, Windowing systems.

GENERAL TERMS: Design

KEYWORDS: Tabletop, X Window, toolkit.

INTRODUCTION

Les nouvelles technologies et la collaboration distante occupent une place de plus en plus importante dans le monde du travail et de la communication. Pourtant, les réunions en face-à-face restent extrêmement présentes. Des alternatives numériques existent sans rencontrer le succès auquel on pourrait s'attendre. Ainsi, les tables interactives, ou tabletop (par référence au desktop et au laptop), sont des ordinateurs se présentant sous la forme d'une surface horizontale, tenant lieu de périphérique d'entrée et de sortie, partagée par plusieurs utilisateurs. Le surcoût de prix ne peut pas complètement expliquer cet absence de succès et de nombreux travaux de recherche en IHM font le pari que le verrou à lever se situe entre des systèmes mal adaptés à ce nouvel usage et des utilisateurs qui ne peuvent pas se permettre de remettre complètement en question un des outils principaux de leur vie professionnelle.

Les interfaces tabletop font l'objet de nombreuses recherches, avec, par exemple, la conférence *IEEE International Workshop on Horizontal Interactive Human-Computer Systems* [8], et un numéro spécial de *Computer Graphics and Applications* sur les systèmes tabletop [7].

Les systèmes de tables interactives introduisent de nouveaux défis dans la conception d'interactions et d'applications. Nous pouvons entre autre citer :

- plusieurs utilisateurs partagent la même surface et peuvent interagir ensemble ou en parallèle ;
- ils sont placés tout autour de la table, et ont donc des points de vues différents sur les documents ;
- la quantité de documents numériques disposés sur la table peut être très importante, et inclure des documents personnels à chaque utilisateur.

Nous relevons plusieurs difficultés auxquelles font faces les études sur les tables interactives. Outre l'absence du dispositif parfait (table trop petites, ombres dues aux projecteurs, latence des systèmes de vision, etc.), nous notons que la plupart des travaux de recherche testent de nouvelles techniques d'interaction pour tabletop sur des documents statiques (par exemple : trier des photographies [5], organiser des objets dans l'espace [12]) ou se servent d'icônes abstraites [1]. L'utilisation de documents statiques, tels que des photographies ou des icônes, permet sans doute de tester un grand nombre de techniques d'interaction mais ne préfigure certainement pas l'utilisation réelle de ce type de système.

La boîte à outils DiamondSpin [10] permet de construire des interfaces pour les tables interactives, en se basant sur les composants d'interface utilisateur classiques (tel que les fenêtres, les boutons, ou les barres de menus) augmentés d'interactions spécifiquement conçues pour les tables interactives (comme la possibilité de tourner les fenêtres, d'attacher un clavier virtuel à une zone de texte, ou de déplacer un menu surgissant). Toutefois, elle nécessite le développement d'applications spécifiques. Nos essais ont souvent suscité un intérêt des utilisateurs pour la possibilité d'avoir d'abord leurs documents standard avant de réfléchir à la création de nouvelles formes de supports au travail de réunion. Or il n'est pas raisonnable d'envisager la réécriture d'une suite bureautique complète, d'un navigateur web, ou des applications métiers spécifiques à un groupe donné d'utilisateur.

Le but de cet article est de montrer comment rediriger l'interface d'une application existante non modifiée vers le système de table interactive. Nous présenterons ensuite des interactions destinées à la collaboration co-localisée sur une table, et la façon dont nous pouvons les mettre en œuvre plus facilement et les tester de manière plus approfondie, grâce à notre technique.

Nous utilisons les logiciels DiamondSpin [10] et Metisse [3], ainsi qu'une table multitouche DiamondTouch [4].

IMPLÉMENTATION D'UN CLIENT METISSE

Metisse est basé sur X Window, le système couramment utilisé pour gérer l'interaction homme-machine par l'écran, la souris et le clavier sous Linux et les diverses variantes d'UNIX. Au lieu de faire le lien entre les applications et l'écran, le clavier, et la souris, le serveur Metisse, développé pour le système de fenêtrage du même nom [3], stocke les fenêtres à afficher sous forme d'images numériques, et simule l'écran et le clavier. Le serveur Metisse est un serveur X modifié pour dessiner les fenêtres hors écran, rendre leurs images accessibles à un compositeur et rediriger les événements souris/clavier envoyés par ce compositeur vers les applications (figure 1). Les applications n'ont pas besoin d'être modifiées ou recompilées, et ne savent pas qu'elles ont été redirigées.

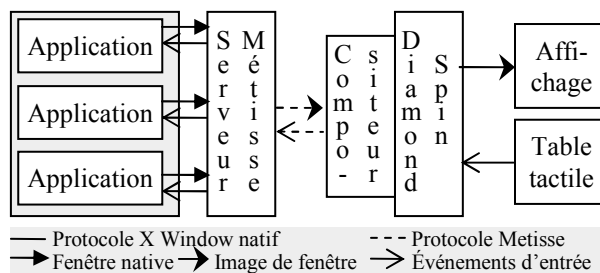


Figure 1 : Architecture de la redirection d'applications.

Affichage

Le serveur X de Metisse rend accessible les images des fenêtres par une connexion réseau. Nous avons donc implémenté un module client pour DiamondSpin [10], une boîte à outils open-source pour table interactive. Du point de vue du serveur Metisse, notre module est un compositeur qui va récupérer les images des fenêtres et générer des événements clavier et souris. Le protocole de communication entre le serveur Metisse et le compositeur est dérivé du protocole RFB : les mises à jour des fenêtres (on ne pas seulement du Frame Buffer) sont envoyées sous la forme d'images rectangulaires correspondants à la plus petite zone englobant la modification. Ces images sont récupérées et stockées par DiamondSpin dans des textures OpenGL, qui les positionne ensuite de manière appropriée par rapport aux utilisateurs (orientées dans le sens de lecture de l'utilisateur qui manipule la fenêtre, ...) et les affiche sur la table.



Figure 2 : Plusieurs applications existantes (Firefox, Nautilus et OpenOffice) ouvertes par deux utilisateurs. Le document OpenOffice est partagé.

Nous remplaçons à la volée quelques éléments d'interface, comme les barres de titre des fenêtres. En effet, les décorations conçus pour les environnements de bureau ne nous semblent pas adaptés aux tables interactives. Par exemple, nous utilisons des barres de titre plus grosses, pour une manipulation avec le doigt plus facile, et situées en bas des fenêtres, pour éviter que la mains de

l'utilisateur ne cache le contenu quand il manipule la fenêtre par sa barre de titre (voir figure 2).

Les événements clavier et souris

Dans un environnement classique, l'élément essentiel de la gestion du clavier et de la souris est le focus. L'ensemble du système, incluant toutes les applications ouvertes, est mono utilisateur. A un instant donné, un seul élément de l'interface reçoit les événements du clavier. D'autre part, certains éléments du système sont modaux, soit au niveau d'une application (par exemple, une boîte de dialogue modale), soit au niveau de l'ensemble du système (par exemple, un menu surgissant).

Sur une table interactive, en revanche, plusieurs utilisateurs interagissent en même temps avec le système. Une gestion spécifique du focus est nécessaire pour rendre ceci transparent aux applications redirigées. Cette gestion peut également découpler le focus du clavier de celui de la souris.

Ainsi, pour le clavier, nous utilisons des claviers virtuels (projetés sur la table), chacun associé à une fenêtre donnée. Il n'y a donc pas un focus clavier, ni un focus clavier par utilisateur, mais un focus clavier par fenêtre. Le mécanisme existant de l'application pour déterminer le composant (zone de texte, bouton, etc.) actif peut donc être conservé. L'utilisateur peut afficher ou masquer le clavier de chaque fenêtre.

Pour la gestion du focus de la souris, au contraire, le mécanisme de redirection actuel ne permet de solution satisfaisante. Du point de vue de l'hôte des applications (figure 1, le rectangle gris), il n'y a qu'un seul curseur : les applications redirigées pensent se partager une unique souris entre elles toutes. Quand l'utilisateur touche brièvement la table (l'équivalent d'un clic), le curseur, du point de vue l'application, se déplace à la vitesse maximale vers la zone touchée, et effectue un clic. Le curseur se déplace ensuite à la prochaine zone touchée, éventuellement par un autre utilisateur.

Ce mécanisme interdit toute opération de type glisser-déposer ainsi que tout verrouillage lié au focus de la souris, tel que les menus surgissants (qui disparaissent dès que le système détecte un clic ailleurs que sur le menu).

Nous limitons l'impact de ce problème en utilisant un serveur Metisse par application. Ainsi, chaque application a, de son point de vue, sa propre souris, et son propre écran. Deux utilisateurs peuvent travailler en parallèle sur deux applications différentes. Toutefois, cette méthode limite sévèrement l'intérêt d'une table interactive en tant que système collaboratif. Non seulement les utilisateurs ne peuvent pas collaborer sur le même document, mais leurs documents respectifs ne peuvent pas in-

teragir ou s'échanger des données (par exemple, chaque document à son propre presse-papier de copier-coller).

APPLICATIONS AUX TABLES INTERACTIVES : LA COLLABORATION CO-LOCALISÉE

Les tables interactives sont utilisées pour des applications collaboratives entre les utilisateurs co-localisés. Cette forme de collaboration diffère des travaux habituels sur les collecticiels car une large part des problèmes classiques sont résolus par des protocoles sociaux mis en place par les utilisateurs [6]. Le partage de la même surface physique de travail permet aux participants d'avoir connaissance des actions des autres utilisateurs, mais également de leurs intentions. Par exemple, il est rare que deux participants essaient de déplacer la même fenêtre au même moment car l'un voit l'autre tendre le bras vers la fenêtre avant qu'il ne commence à la déplacer.

Lors de la création de nouvelles interactions, la difficulté principale posée par notre système est que nous travaillons avec des images numériques des fenêtres. Ainsi, la copie de fenêtres utilisée ci-dessous n'est qu'un affichage en double de la même image.

Télépointeur

Le télépointeur permet à un utilisateur de mettre en évidence une zone d'un document partagé sur toutes les copies du document, à partir d'une action sur sa copie du document (la copie de documents est décrite dans [2]). Il s'agit de l'exact équivalent des télépointeurs utilisés dans le domaine des collecticiels du type vues multiples sur un affichage unique. Ici, nous affichons toutes les vues au même endroit, pour des utilisateurs co-localisés. Nous cherchons donc à résoudre, non pas un problème de distance, mais un problème de différence d'orientation.

Dans le cas d'un document papier imprimé en plusieurs exemplaires, l'utilisateur devrait d'abord repérer la zone à mettre en évidence sur sa copie, puis la repérer sur la copie de l'autre participant (qu'il voit à l'envers), et enfin se pencher sur la table de réunion pour pointer la zone. Notre système lui permet de se restreindre à la première étape (repérer la zone sur sa copie numérique du document). Nous tirons parti de la synchronisation entre les copies pour exprimer simplement cette zone en terme de pixels, et nous l'affichons en surimpression de la texture de la fenêtre, par exemple sous la forme d'une zone plus claire (métaphore du spot de lumière).

Annotation collaborative

L'annotation à main levée est très répandue sur les tables interactives. Dans le cas de nos documents d'applications non modifiées, il se heurte au problème de l'opacité des processus de l'application pour le système de table interactive. En effet, l'annotation ne peut se faire qu'au travers d'une vue, par l'intermédiaire des fenêtres redirigées. La vue peut changer (par exemple, la fenêtre peut être équipée d'une barre de défilement), et il faut alors

reconstruire le lien visuel entre les annotations et le contenu des documents.

Nous proposons d'instrumenter les composants qui servent couramment à effectuer de tels changements sur la vue, en utilisant les technologies d'accessibilité. Ces technologies permettent à des applications tierces (habituellement des applications proposant un moyen alternatif d'utiliser un ordinateur aux personnes souffrants d'handicaps visuels) d'énumérer et de contrôler la hiérarchie des composants d'interface utilisateur. Elles sont mises en œuvre dans Metisse, et sont par exemple utilisées par Façades [11].

Ainsi, nous pourrions exprimer un changement de la vue par une transformation affine de l'image de la fenêtre (translation, zoom, etc.). Il nous suffira alors d'appliquer la même transformation aux annotations concernées.

CONCLUSION

Dans cet article, nous avons présenté une technique pour intégrer des applications existantes dans un système de table interactive. L'utilisation d'applications existantes nous permet d'être au plus près des usages des utilisateurs, et de leur proposer de nouveaux modes d'interactions adaptées à leurs tâches.

Nous avons implémenté cette technique sous la forme d'un module client pour le serveur Metisse, que nous avons intégré à la boîte à outils open-source DiamondSpin. Ce module est écrit en Java et l'affichage est effectué avec OpenGL. L'ensemble du module est constitué d'environ 2000 lignes de code, et le rafraîchissement de l'écran s'opère à la même vitesse que celui des applications redirigées, sur un ordinateur PC standard.

Enfin, Nous avons proposé deux interactions, permettant à plusieurs utilisateurs co-localisés autour d'une table interactive d'utiliser une application existante de manière collaborative : un système de télépointeur, et l'annotation des documents au travers de leurs vues.

Dans un avenir proche, l'évolution des technologies pour les ordinateurs de bureaux devrait nous permettre de résoudre le problème du focus de la souris. Ainsi Microsoft développe Multipoint [9], un système permettant de connecter plusieurs souris à un ordinateur PC, chaque souris correspondant à un pointeur individuel. Cette technologie incorporera un système de gestion du focus adapté à un environnement multi pointeurs que nous pourrions réutiliser. Le support de plusieurs curseurs système fera également parti du serveur X de Metisse grâce au projet MPX (the Multi-Pointer X Server, <http://wearables.unisa.edu.au/mpx/>).

Nous pourrions ensuite expérimenter l'utilisabilité des applications conçues pour les ordinateurs de bureau sur

des tables interactives. Surtout, cela fournira un cadre générique pour la création de situations de travail concrètes, répondant aux besoins professionnels des utilisateurs, pour l'expérimentation de nouvelles interactions à destination des tables interactives.

REMERCIEMENTS

Nous tenons à remercier Michel Beaudouin-Lafon et Gaëtan Rey pour leur soutien et leurs commentaires. Le projet DigiTable est financé par le RNTL. Nous remercions la région Île-de-France pour son soutien financier.

BIBLIOGRAPHIE

1. Agarawala, A., Balakrishnan, R. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. *Proc. CHI 2006*, pp. 1283-1292.
2. Besacier, G., Rey, G., Najm, M., Buisine, S., Vernier, F. Paper Metaphor for Tabletop interaction design. To appear in *Proc. of HCI International 2007*, 22-27 July 2007, Beijing, P.R. China.
3. Chapuis, O., Roussel, N. Metisse is not a 3D desktop! *Proc. UIST'05*, pp. 13-22.
4. Dietz, P., Leigh, D. DiamondTouch: A Multi-User Touch Technology. *Proc. UIST 2001*, pp. 219-226.
5. Hinrichs, U., Carpendale, S., Scott, S. D., Pattison ; E. Interface Currents: Supporting Fluent Collaboration on Tabletop Displays. *Proc. of Smart Graphics 2005*, pp. 185-197.
6. Huang, E. M., Mynatt, E. D. Semi-public displays for small, co-located groups. *Proc. of CHI 2003*, pp. 49-56.
7. *IEEE Computer Graphics and Application*. Interacting with Digital Tabletops. Volume 26, number 5, september/october 2006.
8. Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (Tabletop 2006), Janvier 2006, Adélaïde, Australie, ISBN 0-7695-2494-X
9. Pawar, U., Toyama, K., Pal, J. Multiple Mice for Computers in Education in Developing Countries. *International Conference on Information Technologies and Development*, May 2006.
10. Shen, C., Vernier, F., Forlines, C., Ringel, M. DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction. *Proc. CHI 2004*, pp. 167-174.
11. Stuerzlinger, W., Chapuis, O., Phillips, D., Roussel, N. User interface façades: towards fully adaptable user interfaces. *Proc. UIST 2006*, pp. 309-318.
12. Wu, M., Balakrishnan, R. Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. *Proc. UIST03*, pp. 193-202.