

Affichages distribués et usage collaboratif

Maxime Collomb

Mountaz Hascoët

LIRMM, UMR 5506 du CNRS. 161 rue Ada, 34 392 Montpellier Cedex 5

{collomb, mountaz}@lirmm.fr

Résumé

Cet article propose une vue d'ensemble des systèmes permettant de gérer des environnements d'affichage distribués. La distribution de l'affichage peut se manifester par l'utilisation de plusieurs affichages mais aussi par le partage d'un affichage entre plusieurs utilisateurs. Dans un premier temps, la notation UDP/C est utilisée. Nous proposons ensuite de la compléter en ajoutant les notions de réplification et de mixage pour mieux expliciter points communs et différences entre les systèmes présentés.

Mots clefs

multi écrans, multi pointeurs, collecticiels, distributed display environment.

1 Introduction

Il est de plus en plus courant pour un utilisateur de disposer de plusieurs surfaces d'affichage. En effet, les cartes graphiques actuelles disposent très souvent de plusieurs sorties et il n'est pas rare de réutiliser un écran provenant d'un ancien ordinateur pour se constituer une configuration disposant d'un affichage distribué.

Il n'est pas rare non plus de disposer de plusieurs machines, le cas le plus courant étant un couple ordinateur portable / ordinateur de bureau.

Enfin, une autre situation courante d'affichage distribué est l'utilisation d'un vidéo projecteur dans une salle de réunion. Cependant, les différentes configurations multi affichages dont on peut disposer amènent des problématiques différentes. En effet, les problèmes techniques sont différents si l'on veut utiliser un écran à plusieurs ou tout seul ou encore si deux écrans sont reliés à une ou deux unités centrales.

Nous allons présenter dans cet article un ensemble représentatif de systèmes mettant en œuvre des environnements d'affichage distribués. L'utilisation de la majeure partie de ces systèmes se traduit par le transfert d'évènements d'entrée ou de flux d'affichage via un réseau (LAN ou WAN). Nous nous intéresserons tout particulièrement à la caractérisation de ces systèmes afin de mettre en évidence les propriétés importantes qui les différencient ou les rapprochent.

2 Notation UDP/C

La notation UDP/C introduite par Lecolinet [1] utilise les abréviations suivantes : **U** pour utilisateur, **D** pour dispositif

d'affichage (ou *display*), **P** pour pointeur et **C** pour unité centrale (ou *computer*).

La notion de dispositif d'affichage nécessite une clarification : dans cette notation, un dispositif d'affichage est assimilé à une carte graphique plutôt qu'à un écran. En effet, le fait qu'il y ait un ou plusieurs écrans en sortie d'une même carte graphique ne fait quasiment aucune différence. La carte graphique se contente de séparer les signaux vidéo et il n'y a pas de changement notable dans l'interface utilisateur.

Il convient également de noter que la notion de pointeur est liée à **D**. Il faut parler de pointeur par dispositif d'affichage. En effet, un pointeur est généralement lié à un écran (et plus particulièrement lorsque l'utilisateur dispose d'un dispositif de pointage direct comme un écran tactile).

Précisons finalement que seule la notion de pointeur est utilisée. Celle-ci renvoie à l'utilisation d'un système de pointage (souris, surface tactile) mais pas à un dispositif de saisie (clavier). On suppose en fait que les 2 sont couplés et que le clavier "suit" la souris.

La notation UDP/C a été initialement utilisée pour la classification des systèmes multi pointeurs mais nous l'utiliserons ici en nous intéressant également aux capacités multi affichages d'un ensemble de systèmes.

3 Les systèmes personnels (1-*.*/*)

3.1 La configuration courante

Le cas classique d'un ordinateur disposant d'1 écran et d'1 souris utilisés par 1 utilisateur correspond à la notation 1-1-1/1. Précisons que cette notation couvre également le cas d'un double affichage (deux écrans connectés sur la même carte graphique).

3.2 Les bureaux à distance

Une variante intéressante est l'utilisation de systèmes qui permettent d'interagir avec des applications qui ne s'exécutent pas sur l'unité centrale C qui contrôle D et P mais sur d'autres machines reliées à C via un réseau. On peut par exemple citer le serveur X-window utilisé dans le monde UNIX, VNC (*Virtual Network Computing*) [2] (figure 1) ou la "connexion au bureau à distance" de MS Windows.

Il est également intéressant de citer ici l'architecture INDIGO [3] qui s'appuie sur un ensemble de standards (XML, XSLT, SVG, SOAP, etc.) et qui propose la réalisation d'interfaces distribuées au travers de deux types de composants :



Figure 1: Un bureau MS Windows contenant un bureau MacOS via VNC. MacOS ne s'exécute pas sur la même machine que MS Windows, mais l'affichage du bureau MacOS est reproduit dans une fenêtre de MS Windows.

des serveurs d'objets et des serveurs d'interaction et de rendu. Un serveur d'objets est en fait le noyau fonctionnel d'une application qui expose ses données à un ou plusieurs serveurs d'interaction et de rendu. Les serveurs partagent un modèle conceptuel que le serveur d'interaction et de rendu transforme en un modèle perceptuel. Le serveur d'interaction et de rendu contrôle donc l'interface utilisateur et peut interpréter les interactions de l'utilisateur sur le modèle perceptuel. Ces interactions donnent lieu à des commandes à appliquer au modèle conceptuel qui sont transmises au serveur d'objets.

3.3 Les pointeurs mobiles

Des systèmes comme Synergy [4] ou PointRight [5] permettent d'utiliser un même dispositif de pointage sur plusieurs machines. Les unités centrales doivent être reliées via un réseau, et lorsque le pointeur atteint le bord d'un des écrans, le curseur "saute" sur un autre écran connecté à une autre unité centrale. Ces systèmes correspondent à la notation 1-N-1/N, c'est à dire que 1 utilisateur utilise 1 dispositif de pointage qui peut se déplacer sur N écrans (connectés sur N machines).

De tels systèmes permettent par exemple de contrôler un ordinateur portable et un ordinateur de bureau simplement en utilisant la souris et le clavier de l'ordinateur de bureau. Lorsque le pointeur atteint un bord de l'écran de l'ordinateur de bureau, il ne s'arrête pas mais continue son mouvement et entre sur l'écran du portable. Tout se passe comme si les écrans de l'ordinateur portable et de l'ordinateur de bureau formaient un espace continu. Cependant, l'illusion a ses limites : il n'est pas possible de faire une glisser-déposer pour déplacer une fenêtre ou une icône. Il n'y a pas d'interaction entre les deux espaces de travail (bien que ces Synergy et PointRight gèrent un presse papier partagé qui permet de copier des données sur une machine et de les coller sur une autre).

4 Les logiciels collaboratifs (N-*-*/*)

4.1 Single display groupware (logiciel collaboratif sur un seul écran)

Dans le cadre de cet article, les *Single display groupware* ont la particularité de fonctionner sur une machine unique et

de ne pas nécessiter de transferts réseau. Ces systèmes sont cependant très intéressants car ils permettent la manipulation de plusieurs pointeurs sur un même espace de travail

Dans cette catégorie de systèmes, plusieurs utilisateurs interagissent avec le même dispositif d'affichage mais en utilisant plusieurs dispositifs d'interaction. Ces systèmes sont décrits par la notation N-1-N/1. Un exemple de *Single display groupware* est DiamondSpin [6] (figure 2).



Figure 2: Deux utilisateurs interagissant simultanément avec une table interactive et le logiciel DiamondSpin [6].

Il faut noter que l'utilisation de plusieurs pointeurs simultanément introduit des complications importantes au niveau des interfaces. En effet, que faire si deux personnes naviguent en même temps dans le menu d'une application par exemple ?

4.2 Logiciels collaboratifs synchrones

Cette catégorie de logiciels regroupe les logiciels permettant une collaboration entre des utilisateurs n'étant pas à proximité immédiate les uns des autres. Chaque utilisateur dispose donc de sa propre machine, toutes les unités centrales étant reliées via un réseau. Chaque utilisateur dispose d'une vue sur un espace de travail commun et peut voir les actions des autres utilisateurs.

Lecolinet différencie ici les pointeurs actifs et passifs. En fait, sur un dispositif d'affichage donné, on est susceptible de voir N pointeurs. Un seul de ces pointeurs (celui qui est lié au dispositif d'affichage) sera actif alors que les N-1 autres seront passifs. Il en découle la notation N-N-(1, N-1)/N. Sur un dispositif d'affichage donné, les pointeurs passifs permettent à l'utilisateur du pointeur actif de savoir ce que font les autres et ainsi d'anticiper les conflits pour mieux les éviter.

5 Les interfaces distribuées (*-N-*/*)

Les systèmes présentés jusqu'ici présentent encore des limites. En effet, il est possible d'avoir plusieurs curseurs simultanément sur un même écran, il est possible grâce à une connexion réseau d'utiliser une souris et un clavier sur une autre machine que celle où ils sont connectés. Il est également possible de voir sur son écran personnel des programmes qui s'exécutent sur d'autres machines. Cependant, il serait intéressant de pouvoir faire toutes ces choses dans un même système.

Deux systèmes, encore au stade de recherche, permettent d'intégrer tous ces comportements. Le premier système est UBIT [1] (grâce à l'*Ubit Mouse Server*) qui permet de contrôler plusieurs pointeurs, les différents dispositifs de pointage n'étant pas obligatoirement connectés à la même machine. De plus, l'*Ubit Mouse Server* permet le contrôle d'une application depuis une machine distante ou encore la migration d'une application d'un affichage à un autre.



Figure 3: Avec le système I-AM, la fenêtre d'une application s'exécutant sur une machine A peut être affichée en partie sur le dispositif d'affichage de la machine A et en partie sur le dispositif d'affichage d'une machine B, les machines A et B étant reliées via un réseau. Il est bien sûr également possible d'afficher cette fenêtre uniquement sur un des deux dispositifs d'affichage [7].

Le second système est I-AM [7] qui permet lui aussi de contrôler plusieurs dispositifs de pointage agissant sur plusieurs dispositifs d'affichage. Il est également possible de transférer l'affichage d'une application sur un dispositif d'affichage distant (figure 3).

6 Les notions clés : Réplication et mixage

Tous les systèmes présentés permettent à des niveaux différents de répliquer (ou parfois simplement de rediriger) et/ou de mixer les dispositifs de pointages et/ou les dispositifs d'affichage.

6.1 Les dispositifs de pointage

Réplication / redirection. La réplication au niveau des dispositifs de pointage consiste à capturer les mouvements d'un pointeur sur une machine, à les transmettre à une autre machine à travers un réseau, puis à reproduire ces mouvements sur cette autre machine.

Mixage. Le mixage au niveau des dispositifs de pointage consiste à la gestion simultanée de plusieurs pointeurs sur une même surface d'affichage. Ceci introduit d'importants problèmes car il est pratiquement impossible d'utiliser les interfaces actuelles en utilisant plusieurs pointeurs (il est par exemple impossible de dessiner plusieurs formes simultanément dans les logiciels de dessin courants).

En effet, l'utilisation simultanée de plusieurs pointeurs remet en cause les systèmes de fenêtrage actuels qui résident sur la notion d'un focus unique (un seul composant de l'interface est actif à un moment donné).

6.2 Les dispositifs d'affichage

Réplication / redirection. Au niveau des dispositifs d'affichage, la réplication consiste à afficher l'interface d'un programme sur un écran qui n'est pas directement connecté à l'unité centrale qui contrôle l'exécution du programme.

Mixage. Le mixage des dispositifs d'affichage se traduit par la possibilité d'intégrer les interfaces. Intégrer une interface permet de déplacer l'affichage d'un programme d'un dispositif d'affichage à un autre sans que cela n'ait de conséquence sur l'exécution du programme.

6.3 La réplication et le mixage dans les systèmes présentés

Le tableau 1 montre les capacités des différents systèmes présentés dans cet article pour gérer la réplication et le mixage au niveau des dispositifs de pointage et d'affichage.

	Pointage		Affichage		Multi-utilisateurs
	Réplication	Mixage	Réplication	Mixage	
Double-écrans					
X-Window, VNC, Indigo	✓		✓		
Synergy, PointRight	✓				
SDG, DiamondSpin		✓			✓
log. collab. synchrones	✓	✓	✓		✓
UBIT I-AM	✓	✓	✓	✓	

Tableau 1: Gestion de la réplication et du mixage dans les systèmes présentés

On remarque que les systèmes les plus avancés sont UBIT et I-AM. Cependant, ces derniers présentent encore une limitation : ils ne permettent pas une utilisation collaborative.

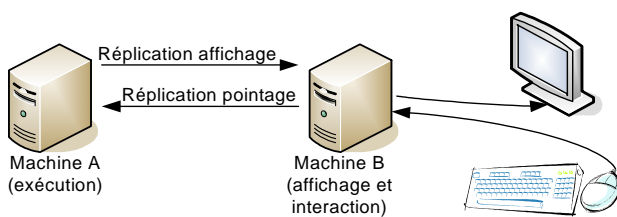
6.4 Granularité

Cependant, définir les notions de réplication et mixage pour caractériser les environnements d'affichage distribués n'est pas suffisant. En effet, les données échangées lors de la réplication, que ce soit au niveau des dispositifs de pointage ou d'affichage, peuvent être de natures très différentes.

Par exemple, prenons le cas d'une application affichant une scène 3D qui s'exécute sur une machine A et s'affiche sur une machine B (figure 4). On peut distinguer deux cas extrêmes au niveau des données échangées entre les machines A et B pour la distribution de l'affichage. Une première possibilité consiste à effectuer le maximum de calculs sur la machine A contrôlant l'exécution du programme. La machine B recevant les données à afficher sous la forme d'un champ de pixels. La machine B ne manipulant qu'un champ de pixels, elle ne sait pas où se trouvent les objets de l'interface (les boutons, par exemple). De ce fait, la machine B ne peut pas traiter les événements issus des dispositifs de pointage et les transmet donc sans interprétation à la machine A. Les transmissions de ce type ne requièrent pas une

machine puissante du côté des dispositifs d'affichage et de pointage (machine B) mais sont exigeantes en ressources réseaux (débit important et temps de latence faible).

La seconde possibilité consiste à un échange de données de plus haut niveau entre les machines A et B. La machine A transmet à la machine B une description de la scène 3D. Et c'est la machine B qui effectue le rendu de la scène. Dans ces conditions, la machine B est en mesure d'interpréter certains événements issus des dispositifs de pointage sans en référer à la machine A. Par exemple, pour naviguer dans la scène 3D (zoom, panoramique, etc.), la machine B peut effectuer elle-même le déplacement de la caméra. Il peut cependant être nécessaire de notifier la machine A de ce déplacement de caméra. Apparaissent alors les questions de synchronisation entre les données de la machine A et de la machine B, questions récurrentes dans le cadre des environnements d'affichage distribués.



Granularité des données échangées pour la réplication niveau affichage

I-AM, INDIGO : synchronisation des modèles
X-Window, UBIT : commandes d'affichage
VNC : champs de pixels

Granularité des données échangées pour la réplication niveau pointage

INDIGO : commandes (événements interprétés)
I-AM, VNC, PointRight, Synergy : événements abstraits
X-Window, UBIT : événements systèmes

Figure 4: Différences de granularité de la réplication au niveau de l'affichage et du pointage pour les systèmes présentés.

La figure 4 présente les différentes granularités des systèmes présentés précédemment dans cet article, au niveau de l'affichage et du pointage. Ainsi, au niveau de l'affichage, le protocole de VNC transmet des informations de très bas niveau puisqu'il s'agit de pixels. X-Window et UBIT (qui est basé sur X-Window) transmettent quand à eux des commandes d'affichage (exemple : le tracé d'un rectangle se traduit par le transfert de la commande `XDrawRectangle`, commande qui est exécutée par la machine B). Et enfin, I-AM et Indigo procèdent à une synchronisation des modèles d'interface entre les différentes machines. Par conséquent, la machine B contrôle les éléments de l'interface et leur affichage.

Au niveau des événements de pointage, le client et le serveur X-Window s'échangent des événements systèmes de bas niveau (tous du même type `XEvent`), tandis que des systèmes comme I-AM, VNC, PointRight ou Synergy pro-

cèdent à une abstraction des événements avant de procéder à leur réplication. Ceci est dû à la nature multi plateforme de ces systèmes (des événements issus d'un système MS Windows nécessitent une interprétation pour être répliqués sur un système MacOS, par exemple). Et enfin, Indigo gère l'interaction du côté de la machine B et seules les commandes interprétées sont transmises à la machine A.

7 Conclusion

On dispose de plus en plus souvent de plusieurs affichages dans notre environnement de travail. Certains systèmes simplifient l'utilisation de ces affichages simultanément. Il existe également des systèmes permettant d'utiliser une surface d'affichage de manière collaborative.

Cependant, les environnements d'affichage distribués permettant une utilisation collaborative sont confrontés au frein que sont les interfaces graphiques logicielles. En effet, dans l'état actuel des choses, elles représentent un verrou au développement d'applications collaboratives car elles ne permettent que très rarement d'être utilisées par plusieurs utilisateurs simultanément.

Les environnements d'affichages distribués et le travail collaboratif font souvent l'objet de domaines de recherche dissociés. Nous avons présenté dans cet article un ensemble représentatif de systèmes se rapportant à ces deux domaines et nous avons proposé des propriétés qualitatives permettant de caractériser ces systèmes. Nous espérons ainsi clarifier des domaines qui sont de plus vastes.

Références

- [1] Eric Lecolinet. Multiple pointers : a study and an implementation. Dans *IHM 2003*, pages 134–141, New York, NY, USA, 2003. ACM Press.
- [2] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, et Andy Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1) :33–38, 1998.
- [3] Renaud Blanch, Michel Beaudouin-Lafon, Stéphane Conversy, Yannick Jestin, Thomas Baudel, et Yun Peng Zhao. Indigo : une architecture pour la conception d'applications graphiques interactives distribuées. Dans *IHM'05*. ACM Press, 2005.
- [4] Chris Schoeneman. Synergy. <http://synergy2.sourceforge.net/>.
- [5] Brad Johanson, Greg Hutchins, Terry Winograd, et Maureen Stone. Pointright : experience with flexible input redirection in interactive workspaces. Dans *UIST '02*, pages 227–234, New York, NY, USA, 2002. ACM Press.
- [6] Chia Shen, Frédéric D. Vernier, Clifton Forlines, et Meredith Ringel. Diamondspin : an extensible toolkit for around-the-table interaction. Dans *CHI '04*, pages 167–174, New York, NY, USA, 2004. ACM Press.
- [7] Christophe Lachenal. *Modèle et infrastructure logicielle pour l'interaction multi-instrument multisurface*. Thèse de doctorat, 2004.