

Augmentation de cours et de réunions dans un campus

Eric Lecolinet, Claudie Faure, Isabelle Demeure, Jean-Claude Moissinac, Stuart Pook

École Nationale Supérieure des Télécommunications (GET) – CNRS LTCI UMR 5141

46, rue Barrault – 75634 Paris Cedex 13

elc@enst.fr

RESUME

Cet article présente deux prototypes d'augmentation de cours (ou de conférences) et de réunions travail qui ont été réalisés dans le cadre d'un projet de campus numérique. Les divers aspects qui ont été mis en oeuvre (interaction avec un tableau augmenté, traitement et sauvegarde d'annotations, multimédia, stratégies de communication pour un réseau sans fil, interaction colocalisée, conception centrée utilisateur ...) sont décrits et analysés comparativement aux deux scénarios considérés. Des observations résultant d'évaluations préliminaires sont présentées ainsi qu'une brève comparaison à l'existant.

ABSTRACT

This paper presents two prototypes for augmenting lectures and small group meetings that have been developed in the context of a digital campus. The various aspects that have been developed (augmented whiteboard interaction, annotation management, multimedia, wireless communication strategies, single display groupware, user centred conception...) are described and analyzed relatively to the considered scenarios. Some preliminary observations are presented and compared to other studies.

Categories and Subject Descriptors

D.2.2, D.2.10 [**Software engineering**]: Design, User interfaces, K.3 [**Computers and Education**], K.3 [**Document and Text Processing**] : Document Preparation, Document Capture, C.2 [**Computer-Communication Network**]

Mots clés

nomadisme, classe augmentée, tableau interactif, encre électronique, annotation, stratégies de communication sans fil, interaction multisurface multi-dispositif, single display groupware, nouvelles technologies pour l'enseignement.

Augmentation de cours et de réunions dans un campus

Eric Lecolinet, Claudie Faure, Isabelle Demeure, Jean-Claude Moissinac, Stuart Pook

École Nationale Supérieure des Télécommunications (GET) – CNRS LTCI UMR 5141

46, rue Barrault – 75634 Paris Cedex 13

elc@enst.fr

RESUME

Cet article présente deux prototypes d'augmentation de cours (ou de conférences) et de réunions travail qui ont été réalisés dans le cadre d'un projet de campus numérique. Les divers aspects qui ont été mis en oeuvre (interaction avec un tableau augmenté, traitement et sauvegarde d'annotations, multimédia, stratégies de communication pour un réseau sans fil, interaction colocalisée, conception centrée utilisateur...) sont décrits et analysés comparativement aux deux scénarios considérés. Des observations résultant d'évaluations préliminaires sont présentées ainsi qu'une brève comparaison à l'existant.

ABSTRACT

This paper presents two prototypes for augmenting lectures and small group meetings that have been developed in the context of a digital campus. The various aspects that have been developed (augmented whiteboard interaction, annotation management, multimedia, wireless communication strategies, single display groupware, user centred conception...) are described and analyzed relatively to the considered scenarios. Some preliminary observations are presented and compared to other studies.

Categories and Subject Descriptors

D.2.2, D.2.10 [Software engineering]: Design, User interfaces, K.3 [Computers and Education], K.3 [Document and Text Processing] : Document Preparation, Document Capture, C.2 [Computer-Communication Network]

Mots clés

nomadisme, classe augmentée, tableau interactif, encre électronique, annotation, stratégies de communication sans fil, interaction multisurface multi-dispositif, single display groupware, nouvelles technologies pour l'enseignement.

1. INTRODUCTION

Cet article présente deux prototypes d'augmentation de cours (ou de conférences) et de réunions travail qui ont été réalisés dans le cadre d'un projet de campus numérique. L'objectif général de ce projet, nommé *Campus Mobile*, est d'étudier les technologies et les usages liés à la mobilité et au nomadisme dans le cadre d'un campus universitaire. Outre l'augmentation des cours et réunions, ce projet comprend d'autres composantes telles que l'utilisation de la téléprésence pour l'enseignement à distance, des services personnels mobiles (par exemple le guidage interactif d'une personne dans un campus) et des services mettant en jeu une interaction entre des dispositifs d'affichage virtuels et des terminaux nomades. Tous ces services font largement appel aux technologies de communication sans fil de type WiFi.

La suite de l'article décrit les divers aspects (interaction avec un tableau augmenté, traitement des annotations, multimédia,

stratégies de communication, interaction colocalisée...) mis en oeuvre pour la réalisation d'environnements d'augmentation des cours et des réunions.

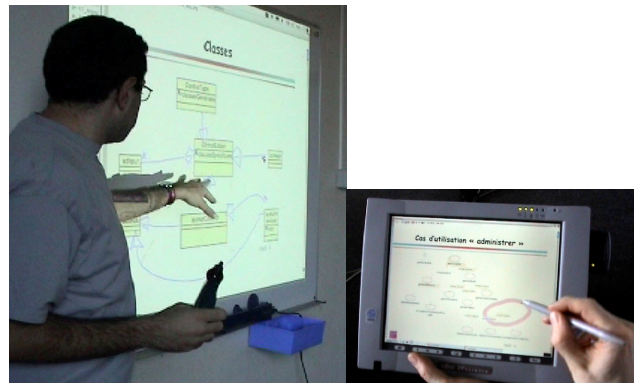


Figure 1: tableau augmenté et dispositif nomade tactile

1.1 Scénarios d'augmentation

Dans le premier scénario étudié, le professeur ou conférencier contrôle sa présentation au moyen d'un tableau augmenté qui permet également de capturer ses annotations. Les transparents et les annotations associées sont envoyées en temps réel aux élèves ou participants via un réseau sans fil. Ces derniers disposent d'un ordinateur portable, d'un tablet PC ou d'un PDA qui leur permet de récupérer ces documents annotés et de rajouter leurs propres annotations. Ce premier scénario est donc essentiellement monodirectionnel, les informations étant envoyées du conférencier vers l'ensemble des participants.

Le second scénario (réunion de groupe) prend le parti inverse : le rôle des participants est totalement symétrique mais ceux-ci sont en plus petit nombre. Le système réalisé permet de partager ou contrôler à distance les ressources d'interaction, tant en entrée qu'en sortie. Ce scénario qui relève en partie du "single display groupware" permet, par exemple, à n'importe quel participant de contrôler le tableau à distance, d'y afficher ses propres documents sans se déplacer, et, sous certaines conditions, à plusieurs participants d'intervenir simultanément sur les mêmes espaces de travail ou les mêmes documents.

Ces deux scénarios peuvent aussi être combinés. On peut, par exemple, considérer une situation de type travaux dirigés où un ou quelques élèves sont temporairement amenés à utiliser des ressources interactives partagées soit pour travailler ensemble, soit pour montrer le résultat de leur travail à l'ensemble de la classe. Enfin d'autres scénarios permettant des interactions hybrides ont également été développés, par exemple pour effectuer des questionnaires interactifs à la volée pendant le cours.

Par souci de clarté on se limitera, dans la suite de l'article, à exposer les problèmes rencontrés et solutions proposées pour la réalisation des deux scénarios de base (cours magistral et réunion). Bien que ces situations semblent a priori partager plusieurs points communs elle posent en réalité des problèmes sensiblement différents. Dans le premier cas, les interactions entre les participants sont beaucoup plus limitées mais nécessitent l'envoi en temps quasiment réel (du point de vue utilisateur) d'une quantité non négligeable de données (les documents affichés et les annotations) à un nombre relativement important d'utilisateurs (une cinquantaine ou plus). Du fait des capacités limitées des réseaux sans fil en terme de bande passante, il a été nécessaire de développer un intergiciel spécifique fonctionnant en mode broadcast. Inversement, le second scénario offre des possibilités d'interaction plus puissantes entre les participants mais suppose qu'un nombre limité d'entre eux les utilise simultanément à un moment donné (ce qui constitue une hypothèse réaliste dans une situation de réunion). Dans ce cas les communications se font point à point, soit via un protocole spécialement défini pour le partage de ressources interactives en entrées, soit en utilisant les possibilités d'affichage à distance du protocole X-Window.

Par ailleurs, ces deux scénarios, et plus spécifiquement le premier d'entre eux, nécessitent une interaction efficace avec un tableau augmenté et une gestion appropriée des annotations, tant sur le poste du conférencier que sur ceux des autres participants. Ces deux questions sont partiellement liées, le tableau augmenté devant à la fois permettre de contrôler directement la présentation mais aussi de capturer et d'afficher intelligemment les annotations (celles-ci ne devant pas être affichées de manière statique sur l'espace écran, mais être associées dynamiquement aux pages ou sous parties des documents affichés). Il est de plus nécessaire de sauvegarder les annotations du conférencier et du participant concerné dans un format permettant leur relecture ultérieure afin de pouvoir relire ou "rejouer" la présentation ultérieurement.

Les aspects concernant l'interaction avec le tableau augmenté, la gestion des annotations et les aspects multimédia font l'objet de la section 2. Les stratégies de communication utilisées dans les deux scénarios seront présentées et comparées à la section 3. Nous terminerons par une discussion par rapport à l'état de l'art puis nous conclurons.

2. IHM, ANNOTATION et MULTIMEDIA

2.1 Interaction avec le tableau augmenté

Le tableau augmenté utilisé dans le projet se compose d'un vidéoprojecteur et d'un dispositif de capture MIMIO [24] qui capture les mouvements des stylos sur l'écran. Un pilote spécifique a été réalisé de manière à pouvoir configurer finement ce dispositif et d'en modifier l'usage selon nos besoins.

Un tableau augmenté permet une interaction évidemment plus poussée qu'une simple craie sur un tableau noir. Le conférencier doit non seulement pouvoir écrire et dessiner sur le tableau, mais également surligner du texte, changer de transparent, zoomer ou dézoomer ces derniers, contrôler des applications auxiliaires (par exemple des démonstrations, des vidéos, etc.). Ceci pose plusieurs problèmes en ce qui concerne l'interaction. Tout d'abord, il est nécessaire de gérer efficacement les annotations, et ce au niveau de chaque application où cette fonctionnalité est souhaitable. D'autre part, la richesse d'un tel dispositif peut rendre son utilisation malaisée, surtout dans une situation de cours où l'intervenant ne peut (et ne doit) porter qu'une attention limitée à la manipulation des dispositifs d'interaction.

Nous avons résolu le premier problème en développant un logiciel de présentation qui permet d'afficher les transparents et d'associer et superposer les annotations propres à chacun d'entre eux sous forme d'encre électronique (ce qui signifie que leur tracé est visualisé par le vidéoprojecteur). Cette stratégie permet à la fois d'éviter d'avoir à effacer les annotations chaque fois que l'on change de page (ce qui serait nécessaire si on utilisait de l'encre physique) et de réafficher un transparent déjà annoté. Elle permet de plus de zoomer correctement un transparent comportant des annotations, une possibilité fréquemment utilisée pour faire de la place sur la surface d'affichage afin d'ajouter des annotations supplémentaires. L'utilisation de l'encre électronique a par contre pour inconvénient de provoquer un problème d'occlusion, l'ombre de l'utilisateur masquant partiellement le tableau. Ce problème est particulièrement gênant si le vidéoprojecteur est en position basse (par exemple s'il est posé sur une table). Il est donc préférable de disposer d'une salle pré-équipée possédant un vidéoprojecteur fixé au plafond (ou encore d'utiliser un système de rétro-projection, comme dans d'autres études, mais ce type d'équipement est rarement disponible en pratique).

Concernant le problème plus général de l'interaction avec un tableau augmenté, deux types de techniques sont habituellement utilisées. La première consiste à utiliser le tableau comme un espace de bureau usuel (l'interaction se faisant via les habituels boutons, décorations de fenêtres, etc.). Cette solution n'est pas très bien adaptée du fait de la taille et de la position du tableau et des applications qui s'y trouvent représentées. D'une part certains widgets peuvent être difficiles à atteindre (trop hauts ou trop bas). D'autre part, ce type d'interaction risque de nécessiter d'incessants déplacements du conférencier (pour atteindre le bon widget chaque fois qu'il veut changer de mode, etc.)

Inversement, certains dispositifs (comme par exemple le MIMIO ou le SMARTBOARD) proposent un arsenal d'outils matériels spécialisés (quatre stylos de couleurs différentes et une double gomme sur le MIMIO). Cette approche « naïve » (dans la mesure où elle tend à reproduire le réel tel quel) s'avère en fait assez peu commode à l'usage. Tout d'abord, nous avons pu constater au cours de quelques expérimentations en situation de cours, que le besoin premier des conférenciers n'est pas de dessiner avec quatre couleurs différentes mais de changer facilement de mode (en pratique : dessiner, surligner ou sélectionner/contrôler). Nous avons commencé par « détourner » les stylos du MIMIO de leurs fonctions originelles en attribuant un mode à chacun des stylos (l'un pour dessiner, l'autre pour surligner, etc.). Cette fonctionnalité, non prévue par les concepteurs du MIMIO a été rendue possible grâce à la réalisation du pilote spécifique évoqué plus haut. Cependant cette solution s'est elle aussi avérée insatisfaisante. Il est en effet peu aisé de manipuler cinq dispositifs différents lorsque l'on s'adresse à un auditoire. D'une part, l'utilisateur doit fréquemment rechercher où il a posé le « bon » outil, d'autre part les confusions d'outils sont fréquentes, à moins que leurs formes ne soient très différenciées (typiquement, la gomme et un stylo).

Nous avons donc finalement adopté une technique hybride qui combine l'utilisation de deux outils bien différenciés (une gomme et un *seul* stylo) et d'une palette mobile transparente. L'effet du stylo dépend du mode choisi dans la palette tandis que nous avons utilisé une particularité de la gomme du MIMIO pour lui attribuer une double fonctionnalité. Cet outil comprend en effet deux capteurs correspondant normalement à une petite et une grande gomme. L'effet d'un de ces deux capteurs a été modifié de tel

sorte qu'il déplace la palette jusqu'à l'endroit pointé sur le tableau. Le conférencier peut ainsi réaliser toutes les actions souhaitées en utilisant uniquement deux outils (un dans chaque main) et sans avoir jamais à se déplacer.

Cette stratégie, qui constitue est une forme d'interaction bi-manuelle simplifiée [4] (la main non dominante servant à déplacer la palette, la main dominante à effectuer l'action de dessin, la sélection, etc.) semble être particulièrement efficace en situation d'usage. Nos évaluations informelles ont montré que tous les utilisateurs, qu'ils soient professeurs ou étudiants, utilisaient cette solution spontanément mais ne se servaient que très rarement des stylos « modaux » supplémentaires lorsque ceux-ci étaient également à leur disposition. La Figure 2 montre un exemple typique de ce type d'utilisation on l'enseignante conserve les outils en main tout en présentant le contenu de ses transparents oralement.

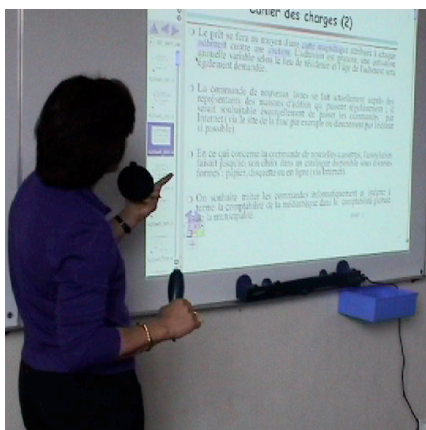


Figure 2: gomme et stylo tangibles

2.2 Traitement des documents et annotations

Le système développé manipule deux types de données : les documents de présentation (en pratique, une suite de transparents) et les annotations capturées via le tableau augmenté ou ajoutées par l'étudiant. Des formats standard ont été choisis pour assurer la sauvegarde, la transmission et la lecture durant le cours ainsi que pour la relecture ultérieure des documents.

Les transparents peuvent être écrits à la main ou sous forme électronique, typiquement sous forme d'une suite de pages HTML, d'une présentation PowerPoint ou d'un document PDF. Ils sont convertis préalablement au cours en un format interne compatible avec le logiciel de présentation. Notre prototype actuel accepte des formats externes tels que HTML ou PowerPoint et les convertit en un ensemble de pages XHTML et d'images GIF ou JPEG. Ce transcodage qui intègre le texte et les images constitue la représentation qui est envoyée aux machines des participants pendant la présentation. Cette représentation peut être utilisée pour relire le cours ultérieurement soit au moyen du même logiciel de présentation soit en utilisant un navigateur Web standard.

Cet ensemble de transparents est enrichi des annotations effectuées par le professeur et l'étudiant. Ces annotations sont constituées de marques graphiques auxquelles sont associées des indications temporelles. Chaque marque graphique comprend les données capturées entre le moment où le conférencier appuie et relâche le stylo sur l'écran augmenté. Pour des raisons d'efficacité ce signal est filtré afin de ne conserver qu'un nombre de points

raisonnable et pour supprimer les points aberrants dus à des erreurs de détection du dispositif de capture.

Plusieurs projets existants utilisent un format ad hoc pour sauvegarder les annotations [2, 14, 23] ce qui impose l'utilisation d'un logiciel de relecture spécifique. Cette contrainte est plus gênante dans le cas de la relecture du document que pour son affichage pendant le cours et la capture des annotations sur le tableau augmenté. En effet, l'utilisation d'un tableau augmenté impose de toute façon l'installation de logiciels et matériels appropriés. Elle se fait forcément dans un cadre particulier, par une personne initiée à ces technologies, dont le principal besoin est d'avoir des logiciels qui facilitent sa tâche de présentation. A contrario, il est souhaitable de pouvoir relire et rejouer le document final sous une plateforme quelconque sans avoir à utiliser des logiciels trop spécifiques. Nous nous sommes donc intéressé à deux formats de codage proposés par le W3C : *Inkml* [15], une grammaire XML destinée au codage des annotations et *SVG* [33] un standard bien connu pour la représentation de données graphiques vectorielles.

Nous avons finalement choisi SVG, ce format possédant les caractéristiques adéquates pour représenter nos annotations. Inkml aurait permis une représentation plus élaborée mais comporte de double inconvénient d'être peu répandu et d'être (par définition) limité au traitement des annotations. A l'inverse, il existe aujourd'hui de nombreux logiciels ou modules d'extensions (en particulier pour les navigateurs Web) capables d'afficher du SVG. Ces logiciels sont de plus capables de restituer l'intégralité de la présentation en prenant également en charge l'affichage du texte et des images (et même l'audio avec l'extension Adobe de SVG 1.1 et sa future normalisation dans SVG 1.2)

Le document SVG généré pendant la conférence permet de la rejouer ultérieurement en assurant la correspondance temporelle entre les transparents, les annotations, et la capture audio du cours (lorsque celle-ci est également effectuée). Enfin, le fait que SVG soit une grammaire XML de représentation de données vectorielles autorise des possibilités étendues d'adaptation des représentations à l'écran, les fichiers générés pouvant être facilement transformés par les outils XML standard.

La transcription du cours est donc constituée d'un document multimédia composée de texte, images, graphique, et, éventuellement, audio. D'autres projets ont étudié la représentation et l'exploitation de transcription audio (ou vidéo) de cours (par exemple [1, 6]). Nous avons privilégié la flexibilité et la possibilité de modifications a posteriori via l'utilisation de grammaires XML. Le consortium *IMS* propose des schémas XML pour la gestion de diverses ressources liées à l'éducation (et en particulier les cours). Les représentations de haut niveau utilisées pour le stockage des transcriptions de cours respecte le schéma IMS approprié ce qui permet diverses fonctionnalités d'accès de haut niveau aux transcriptions (par exemple, pour effectuer des recherches dans une base de cours). Cependant, un format multimédia est nécessaire pour l'enregistrement et la relecture des cours. Certains projets utilisent des formats spécifiques [1], SMIL [32, 38] ou MPEG-4 [12]. Nous avons finalement préféré SVG car c'est le seul format XML capable de gérer les images, les annotations et l'audio.

3. INTERACTION ET COMMUNICATION ENTRE PARTICIPANTS

Comme expliqué dans l'introduction, nous nous sommes intéressés à des situations qui mettent en jeu des interactions entre plusieurs participants. Dans le premier scénario les interactions sont monodirectionnelles, du poste du conférencier vers ceux de *tous* les autres participants. Dans le second scénario, les interactions sont multidirectionnelles, mais n'impliquent qu'un petit nombre de participants à un instant donné. Ce second scénario, qui permet de partager ou de contrôler à distance les ressources d'interaction constitue une forme de travail collaboratif. La suite de la section décrit ces deux cas de figure plus en détail.

3.1 Interaction 1-vers-N

Ce schéma contraste avec le schéma 1-vers-1 largement utilisé dans le modèle client-serveur, qui est basé sur l'envoi de requêtes par des utilisateurs de services (ou "clients") à un serveur qui fournit la réponse correspondant au service demandé en retour.

Le schéma de communication 1-vers-N (avec $N \gg 1$) est plus naturellement supporté par les technologies de type broadcast (ou multicast). Cette technologie est donc à la base de l'intergiciel qui a été spécifiquement développé dans le cadre du projet pour effectuer l'envoi des transparents et des annotations vers tous les participants. L'utilisation de réseaux de communications sans fil constitue une autre contrainte importante ainsi que celle de terminaux pouvant avoir une capacité limitée (surtout dans le cas d'utilisation de PDAs). Les caractéristiques prises en compte pour un support efficace des communications sans fil (de type 802.11b dans le cas présent) concernent la bande passante limitée et les possibilités de déconnexions fréquentes. Nous avons également tenu compte de la mémoire et de la vitesse de CPU limitées des terminaux de type PDA (des *HP Ipaq* sous Linux embarqué dans le cadre de nos expérimentations). Par contre nous n'avons pas traité le problème de l'autonomie de la batterie, lequel est souvent critique lorsque l'on utilise des dispositifs mobiles, surtout lorsque la communication est sans fil.

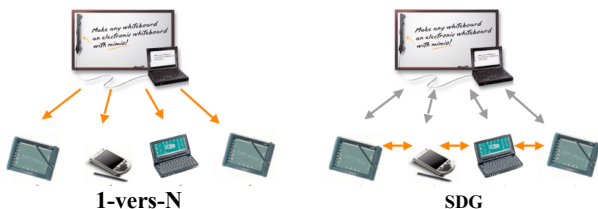


Figure 3: schémas de communication

Dans les solutions middleware traditionnelles, on trouve souvent des mécanismes de communication basés sur un couplage fort tels que l'invocation à distance de méthodes (comme dans CORBA et Java RMI). Ces mécanismes nécessitent de connaître l'adresse du ou des destinataires avant d'envoyer le message ce qui rend ces solutions difficiles à utiliser lorsque ces destinataires changent ou si leur nombre varie. Ce type de communication est de plus synchrone. Par opposition, les intergiciels de type MOM (Message-Oriented Middleware) permettent un couplage faible entre les programmes envoyant et recevant les messages ainsi qu'un haut niveau d'anonymie. Ces intergiciels permettent en outre des communications asynchrones, qui conviennent mieux à des environnements mobiles et sans fils.

JMS de Sun est devenu une interface de référence pour ce type d'intergiciels. Plusieurs projets, comme par exemple *Pronto* [39], se sont intéressés au développement de systèmes JMS pour les environnements mobiles et sans fil. Pronto est d'autant plus intéressant pour les applications qui nous concernent qu'il utilise du multicast comme mécanisme de transport. Il fournit une transmission efficace des messages d'un émetteur vers plusieurs souscripteurs et permet d'accroître leur nombre sans augmentation du trafic réseau. *Wireless JMS* [18] constitue un autre exemple de ce type de système dont le développement a été en partie inspiré par le projet Campus Mobile. Parallèlement nous avons également développé un système robuste de communication 1-vers-N en mode broadcast destiné à répondre aux besoins spécifiques de notre projet (et aux contraintes énoncées plus haut). Ce choix de développer un système spécifique (nommé *Campcast*) a été motivé par le fait qu'il était nécessaire de pouvoir contrôler complètement le protocole de broadcast pour obtenir les propriétés souhaitées et limiter autant que possible l'utilisation mémoire. Ce protocole et son implémentation sont décrits dans la section suivante.

3.1.1 Implémentation du protocole *Campcast*

Ce protocole est implémenté au moyen de deux programmes. Le premier (*CampcastS*) s'exécute sur la machine où les transparents sont affichés et reçoit ses commandes du logiciel de présentation utilisé par le conférencier. *CampcastS* envoie les messages appropriés en mode broadcast au second programme (*CampcastC*) qui s'exécute sur la machine de chaque destinataire. De son côté, *CampcastC* stocke les transparents et les annotations et effectue des demandes de retransmission lorsque les données sont incomplètes (une caractéristique importante de l'envoi en mode broadcast étant que les paquets de données envoyés peuvent n'être jamais reçus, ou bien être reçus dans le désordre). Ce programme permet enfin d'informer le logiciel de présentation utilisé par le destinataire que le transparent souhaité est disponible et prêt à être affiché. Les échanges entre ces deux programmes de communication et les logiciels de présentation (le logiciel "*Campus*" dans cet exemple) sont illustrés à la Figure 4.

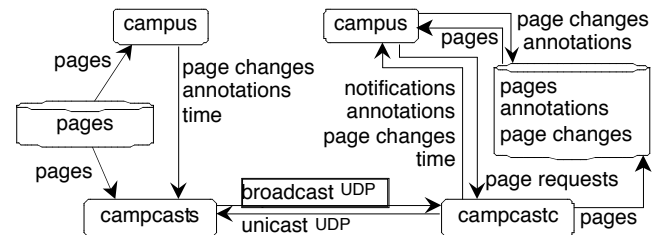


Figure 4: échange d'informations en mode broadcast

Au moins une fois par seconde, *CampcastS* envoie un message de contrôle contenant le nom du transparent courant, le numéro de la dernière annotation ainsi qu'une valeur indiquant si ce programme est prêt à accepter d'éventuelles demandes de retransmission. Ceci permet d'assurer la robustesse du protocole en cas de pertes de paquets, ce qui arrive fréquemment en mode broadcast sur un réseau sans fil. De plus, les transparents sont découpés en paquets (afin que seuls les paquets perdus soient retransmis) qui tiennent dans un paquet UDP (car si un fragment d'un paquet UDP est perdu, le paquet entier doit être renvoyé). Enfin les transparents sont envoyés un par un lorsque cela est possible, et toujours avec un transparent d'avance par rapport à ce qui est affiché à l'écran afin d'éviter les latences quand on change de transparent. Les

annotations sont envoyées au fur et à mesure de leur capture sur le tableau augmenté. Leur numérotation permet de déterminer si des annotations ont été perdues, auquel cas une demande de retransmission est automatiquement faite par *CampcastC*.

La gestion des transparents est légèrement différente car elle implique une interaction avec le logiciel d'affichage du destinataire. Plus précisément *CampcastC* alerte le logiciel d'affichage quand le transparent courant a changé. Ce logiciel peut alors afficher ce transparent, sauf s'il n'a pas encore été complètement reçu. Dans ce cas il envoie à *CampcastC* une demande de retransmission des parties manquantes. Les demandes de retransmissions sont effectuées par unicast UDP contrairement aux envois de données vers les destinataires, qui sont toujours faits en mode broadcast, y compris pour les retransmissions. Ceci permet de réduire le nombre de retransmissions en évitant que le même paquet soit envoyé successivement à plusieurs destinataires qui ne l'auraient pas reçu.

3.2 Interaction multiple colocalisée

Bien que les vidéoprojecteurs soient aujourd'hui couramment utilisés pendant les cours et les réunions, il est généralement uniquement possible à la personne dont l'ordinateur est physiquement connecté au vidéoprojecteur de contrôler l'écran et d'afficher des documents. La prise de contrôle de cet écran par une autre personne nécessite une laborieuse opération de déconnexion et reconnexion de câbles, pouvant d'ailleurs souvent conduire à des situations de blocage informatique. De plus il n'est pas possible que plusieurs personnes interviennent simultanément (ou quasi-simultanément : chacun son tour mais en échangeant les rôles sans délai) tant en entrée (contrôle interactif à distance) qu'en sortie (affichage de documents provenant de plusieurs machines).

Il est pourtant facile d'imaginer des scénarios où ce type de services serait fort utile : par exemple pour permettre aux participants d'une réunion de montrer successivement leurs documents sans se déplacer ni rebrancher leurs machines, ou d'afficher et de contrôler simultanément des documents qui se complètent. En situation de cours ou de travaux dirigés, cette possibilité pourrait permettre à un ou plusieurs élèves de montrer à tous le résultat d'un exercice, ou encore à plusieurs élèves de travailler ensemble sur les documents affichés sur l'écran partagé. Une extension de ce scénario consiste enfin à permettre de partager et de répliquer les espaces de travail ou les documents sur un nombre quelconque de dispositifs d'affichage (par exemple les écrans des PCs de plusieurs participants et pas seulement l'écran partagé).

Ce type de services peut sembler naturel, surtout avec la popularisation des réseaux sans-fil et l'utilisation généralisée des portables et dispositifs de vidéoprojection en situation de réunion. Malheureusement, les systèmes de fenêtrage et les toolkits graphiques actuels n'ayant pas été conçus pour permettre l'interaction multi-utilisateurs, la mise en oeuvre de ce type de systèmes pose de nombreux problèmes. Par exemple, tous les systèmes de multi-fenêtrage actuels sont faits pour gérer un unique pointeur souris. Lorsqu'une machine dispose de plusieurs souris, les événements qui en proviennent sont tous fusionnés pour contrôler le seul pointeur visible à l'écran. Les boîtes à outils graphiques standard ne sont pas non plus capables de distinguer plusieurs flux d'événements en entrée. Ainsi, même si plusieurs pointeurs étaient disponibles et leurs flux d'événements logiquement séparés, ceux-ci seraient in fine fusionnés au niveau

applicatif. Le même problème se pose en ce qui concerne les sorties, aucun toolkit graphique standard n'étant capable d'afficher simplement des interfaces graphiques simultanément sur plusieurs machines distantes.

Plusieurs systèmes ont été proposés pour résoudre certains de ces problèmes, soit pour permettre à plusieurs participants de contrôler à distance et à tour de rôle un ou plusieurs dispositifs interactifs d'affichage (PC munis de vidéoprojecteur ou d'écrans augmentés) [5,17,21,34], soit pour permettre l'interaction simultanée de plusieurs utilisateurs sur un écran partagé (une forme de collaboration appelé « single display groupware ») [26,31,36]. D'autres systèmes s'intéressent à la distribution ou à la migration de l'affichage des applications ou des environnements de travail [3,19,29,37].

Dans la suite cette section, nous analysons plus en détails les types de services nécessaires à ce que nous regroupons sous l'appellation « interaction multiple colocalisée », c'est-à-dire les formes de travail collaboratif où tous les participants sont situés en un même lieu et peuvent voir et contrôler le ou les mêmes espaces interactifs. Nous présentons ensuite une solution complète qui remplit l'ensemble des critères souhaités.

3.2.1 Services requis pour l'interaction colocalisée

La base des services requis est de permettre de **multiplexer les entrées** (événements souris et clavier) provenant de plusieurs machines sources sur une machine cible distante (typiquement, celle qui contrôle l'écran partagé). Plusieurs cas de figures peuvent être distingués :

- *Le contrôle à distance simple* : un seul utilisateur peut contrôler à distance la machine cible à un instant donné. Le contrôle par plusieurs utilisateurs n'est possible que si ceux-ci assument à tour de rôle.
- *Le contrôle multiple (ou single display groupware)* : plusieurs utilisateurs peuvent contrôler simultanément des pointeurs distincts (ou pseudo-pointeurs) sur la machine cible. Deux cas doivent être considérés suivant que ces utilisateurs interagissent avec la même application ou des applications distinctes sur le même espace de travail.
- *Le contrôle à distance multiple* : une extension du cas précédent où le contrôle se fait à distance (mais toujours dans le même lieu)
- *Le contrôle à distance simple ou multiple sur plusieurs cibles* : une extension des cas précédents où il est possible de contrôler des dispositifs d'affichages appartenant à plusieurs machines cibles

Le contrôle à distance simple (ou « à tour de rôle ») est un service disponible dans divers systèmes de recherche [5,17] ou des utilitaires comme *Synergy* [34] ou *VNC* [29]. Le contrôle multiple pose quant à lui des problèmes bien plus complexes pour les raisons exposées plus haut (pas de support de base des pointeurs multiples). Ce service est donc rarement disponible ou implémenté de manière spécifique au niveau applicatif [11,31]. Dans ce dernier cas, l'application doit elle-même gérer les sources d'entrées des pseudo-pointeurs, leur affichage à l'écran ainsi que les communications à distance. Les autres applications ne recevront pas les événements associés à ces pseudo-pointeurs (qu'ils soient locaux ou provenant de machines distantes) et ceux-ci ne seront pas visibles sur ces applications.

Une extension intéressante des système d'interaction colocalisée est de permettre de **multiplexer les sorties** graphiques sur des machines distantes. Deux cas se présentent en pratique: l'affichage à distance d'une application sur une autre machine que celle où elle s'exécute (avec parfois des possibilités de migration dynamique) et la distribution (ou la réplication graphique) d'une application sur plusieurs machines.

Le premier cas permet à un ou plusieurs participants d'afficher leurs propres documents sur l'écran partage sans avoir à recopier ces documents ni à effectuer aucun branchement (ou d'utiliser un terminal, comme un PDA, qui ne dispose généralement pas de sortie VGA). Il permet enfin d'afficher simultanément des documents provenant d'espaces de travail différents sur le même écran (et de les faire apparaître ou disparaître dynamiquement et facilement).

Le second cas est encore plus intéressant dans la mesure où l'application peut s'afficher à la fois sur la machine source et la machine cible, ce qui permet de la contrôler à distance (soit pour éviter de se déplacer, soit parce que cette machine dispose de dispositifs d'interaction ou de logiciels spécifiques, par exemple un écran tactile et un logiciel d'affichage permettant l'annotation). Plus généralement, cette propriété peut permettre le travail collaboratif synchrone de plusieurs utilisateurs, chacun voyant l'application affichée sur sa propre machine.

X-Window ou des extensions comme *RDP* (pour Windows) offrent des fonctionnalités d'affichage à distance des applications. Cependant ces systèmes n'ont pas été initialement conçus pour le travail collaboratif et un important travail d'encapsulation est nécessaire pour pouvoir utiliser ces possibilités aisément dans le cadre qui nous intéresse. La distribution ou la réplication graphique posent des problèmes plus complexes et sont ainsi rarement disponibles. *VNC* [29] fournit un service de ce type mais avec diverses limitations : plusieurs utilisateurs ne peuvent pas réellement interagir simultanément (en pratique le contrôle du pointeur actif est séquentiel) et ce type de réplication graphique (par copie de blocs de pixels) ne permet pas d'adapter finement la représentation graphique au dispositif cible. Enfin, certaines boîtes à outils pour le travail collaboratif (par exemple [30]) permettent de créer des applications interactives distribuées qui ne reposent pas sur la réplication graphique. Ce type de système est plutôt destiné au cas du travail collaboratif à distance qui pose des problèmes différents qui sortent du cadre d'étude de cet article (où les participants sont par définition co-localisés).

3.2.2 Système développé

Le système développé pour *Campus Mobile* supporte tous les types de multiplexage en entrée et en sortie qui ont été présentés dans la section précédente.

Le **multiplexage en entrée** est géré par un ensemble de serveur d'interaction appelés *serveurs UMS*. Chaque serveur offre les services requis pour le *contrôle à distance multiple* de la machine sur laquelle il s'exécute. Connectés entre eux via un réseau, ces serveurs permettent de définir une topologie d'écrans qui permet de mettre en commun l'ensemble des ressources interactives.

Un serveur UMS peut gérer plusieurs pseudo-pointeurs simultanément (en plus du pointeur natif). Chaque pseudo-pointeur correspond à un *flux* logique d'interaction et est affiché par le serveur sous forme d'une petite fenêtre dépourvue de décoration qui contient l'image d'un pointeur. Ces flux peuvent être dynamiquement connectés à une ou plusieurs *sources*

d'événements. Une source d'événements peut être locale ou distante. Dans le premier cas elle contient les événements émis par un dispositif d'interaction non standard, local à la machine qui héberge le serveur (par exemple une souris ou un MIMIO). Le terme "non standard" signifie ici que ce dispositif n'est pas géré par le système de fenêtrage de la machine hôte et qu'il ne permet donc pas de contrôler le pointeur natif. Dans le second cas, la source d'événements provient d'une machine distante et peut être associés, selon les cas, à son pointeur et clavier natif ou aux pseudo-pointeurs gérés un serveur UMS distant. Le transport des événements entre machines se fait via une connexion TCP/IP à l'aide du "*protocole UMS*", conçu pour minimiser la quantité de données échangées avec les serveurs.

Cette architecture client/serveur présente l'avantage de gérer toutes les ressources d'interaction non standard globalement pour un ordinateur donné. Ceci évite non seulement d'avoir à gérer ces services aux niveau applicatif mais permet surtout de les rendre accessibles à toutes les applications et d'en permettre le partage. Ainsi les pseudo-pointeurs s'afficheront au dessus de n'importe quelle application et permettront de la contrôler de la même manière qu'en utilisant le pointeur natif (sauf dans certains cas critiques). Ceci a également pour conséquence importante qu'il n'est pas nécessaire de réécrire ou d'adapter les applications existantes tant que l'on se borne à utiliser les pointeurs multiples pour contrôler des applications distinctes (ou la même application, mais à tour de rôle).

Le cas où plusieurs pointeurs sont utilisés *simultanément* pour contrôler la même application est plus complexe. Par défaut, les événements sont mélangés, les toolkits graphiques usuels n'étant pas à même de distinguer les flux événementiels, comme expliqué précédemment. Cette séparation est cependant possible à condition de modifier l'application, ou le toolkit graphique, de manière de les rendre compatibles avec le protocole UMS. Ce dernier permet en effet aux applications de déterminer le flux auquel les événements appartiennent (les flux associés aux pseudo-pointeurs étant alors envoyés aux applications soit par une connexion socket spécifique soit via un codage particulier des événements X). Cette propriété n'a été pour l'instant testée qu'avec la librairie *Ubit* [22, 37], un nouveau toolkit graphique développé parallèlement à ce projet. Ce toolkit possède entre autres, la capacité de pouvoir totalement séparer la gestion de plusieurs flux d'interaction, ce qui permet à plusieurs utilisateurs d'interagir simultanément sur plusieurs widgets d'une même application sans interférence. Une telle propriété n'est certainement pas totalement vérifiée avec les toolkits graphiques usuels. Il est cependant vraisemblable qu'ils acceptent des formes limitées mais néanmoins intéressantes d'interaction multi-pointeurs.

En ce qui concerne le **multiplexage des sorties**, nous avons là encore utilisé les propriétés du toolkit *Ubit*. Celui-ci permet en effet la réplication graphique d'applications ou de parties d'applications sur des machines distantes. Ce mécanisme est dynamique (les parties répliquées peuvent apparaître ou disparaître à tout moment) et permet également à plusieurs utilisateurs d'interagir simultanément sur les parties répliquées (chaque réplication correspondant à un flux événementiel différent).

Le principal avantage de la réplication graphique est qu'elle permet de créer des applications collaboratives très simplement. Les applications interactives distribuées sont généralement difficiles à concevoir et implémenter car elle nécessitent des

mécanismes de synchronisation complexes et qui ne tolèrent aucune latence. La réplication graphique offre une solution triviale dans la mesure où il n'y a pas de données à synchroniser (à part les représentations graphiques) puisque toutes les données sont gérées par une seule application. Cette solution a par contre pour inconvénient de nécessiter une bande passante relativement importante. Elle ne peut donc convenir que si le nombre de répliqués est limité et si le réseau possède les caractéristiques adéquates. En pratique ces contraintes posent rarement problème dans le cadre qui nous intéresse, même en utilisant un réseau sans fil : il en effet peu probable que plusieurs participants modifient en même temps plusieurs documents affichés sur plusieurs écrans.

Un autre inconvénient de cette approche est qu'elle semble rendre difficile l'adaptation des représentations graphiques aux caractéristiques des surfaces d'affichages (une propriété appelée *plasticité* [3]) : il est par exemple souhaitable qu'un même document n'ait pas la même taille sur un grand écran et sur un PDA. Ce problème est résolu par le toolkit Ubit, celui-ci permettant de configurer indépendamment chaque représentation (et même de représenter de manière complètement différente les mêmes données sur chaque écran).

L'implémentation de la réplication dans le toolkit Ubit repose actuellement sur le protocole de communication de X-Window (qui permet l'affichage à distance d'applications, mais ne fournit aucun support de haut niveau pour la réplication). Cette dépendance impose l'utilisation d'un serveur X, quelque soit la plateforme utilisée (Unix, MacOSX ou Windows). Il serait cependant envisageable de supprimer cette contrainte en utilisant d'autres protocoles tels que ceux de RDP ou VNC.

En conclusion la réplication graphique semble une solution appropriée pour le SDG dans la mesure où elle permet d'obtenir une grande flexibilité et des performances adéquates sans pour autant nécessiter un effort logiciel important.

4. PROJETS SIMILAIRES ET DISCUSSION

Un objectif maintenant prioritaire dans l'éducation est l'usage des nouvelles technologies de l'information et de la communication pour améliorer l'enseignement et l'acquisition des connaissances. Les scénarios prototypes définis au début du projet décrit dans cet article résultent d'enquêtes auprès d'utilisateurs potentiels et d'observations de terrain. Nous avons choisi de concevoir un système qui intègre plusieurs technologies et qui couvre des situations différentes (salle de cours ou réunions) contrairement à d'autres systèmes plus spécialisés.

En réponse aux envies d'usages des étudiants, le système peut rejouer le cours. Cette demande était explicite dans les scénarios que les étudiants ont construit quand on leur a demandé d'anticiper la future université idéale. Néanmoins, ils affirment que la présence dans une salle de cours semble le meilleur moyen pour acquérir des connaissances. Les enseignants peuvent profiter de l'interaction directe avec l'auditoire et des questions qui leur sont posés pour ajuster leur présentation. La capture des événements qui se produisent pendant le cours et l'interaction entre participants jouent un rôle important quand il s'agit d'améliorer l'enseignement.

Les étudiants disent, qu'après plusieurs jours, ils rencontrent des problèmes pour comprendre leurs notes et même les documents projetés en cours (si ils sont disponibles). Le contenu d'un cours est en grande partie exprimé par des événements fugitifs, comme les explications orales ou les notes écrites au tableau.

Classroom2000 [1] fut un des premier projet sur la capture et l'intégration des différents média utilisés en cours (diapositives, notes, son, vidéo, sites Web). La vidéo du cours donne un fort sentiment de présence mais ce média est très gourmand en espace mémoire, de plus la présentation conjointe des diapositives et de la vidéo lors de la révision du cours disperse l'attention de l'étudiant. Pour ces raisons, nous avons choisi de ne pas prendre le cours en vidéo mais d'enregistrer l'exposé oral et des notes ajoutées au tableau afin d'assurer un accès permanent à ces événements fugitifs.

L'utilisation d'un tableau interactif implique de définir une interaction spécifique. Dans *Interactive Presentation Manager* [16], l'interaction avec une grande surface verticale est très semblable à celle que les ordinateurs de bureau proposent aux utilisateurs. Pour l'utilisateur, qui n'aime pas changer ses habitudes, la forte analogie entre l'utilisation des craies et feutres et l'interaction avec un tableau augmenté est attrayante [1, 13]. Néanmoins, ne viser que l'analogie revient à ne pas exploiter le contexte numérique offert par ces tableaux pour enrichir l'interaction. Des menus contextuels et des commandes gestuelles sont bien adaptés aux tableaux interactifs. Dans [13] les tracés manuscrits de mots ou de formules mathématiques sont reconnus lors des cours de physique et de mathématiques. Dans de grand amphithéâtres, le haut du tableau est souvent trop éloigné de l'enseignant pour être atteint. Dans ce cas, un ordinateur à stylo est utilisé comme une télécommande pour envoyer des informations sur le tableau [25]. Ce type de dispositif permet aussi à l'utilisateur de garder le contact visuel avec les étudiants et de se mouvoir librement dans la classe [2, 26]. Les solutions logicielles développées dans le cadre de notre projet permettent ce type d'interaction (et d'une manière très générale puisque plusieurs personnes peuvent éventuellement prendre le contrôle en entrée ou en sortie de l'écran comme expliqué à la section 3). La Figure 1 (à droite) illustre ce type de situation.

Un enseignant debout, seul face à un groupe dense d'étudiants n'est pas une situation qui facilite les relations interpersonnelles. Une vision proche des diapositives et des annotations de l'enseignant, par opposition à la vision distante d'un tableau public, peut aider chaque étudiant à ressentir que le cours d'adresse vraiment à lui et non à un groupe dont il est un membre parmi plusieurs. Les étudiants sont équipés de dispositifs personnels sur lesquels s'affiche le contenu du tableau public [10, 23, 26, 32]. L'ajout (ou l'attachement) de notes personnelles aux diapositives aide l'étudiant à personnaliser les supports de cours [7, 19, 35]. Ces notes restent liées aux diapositives après l'intégration des différents média du cours [20]. Les étudiants peuvent échanger leurs notes pendant le cours [14, 25]. L'interaction dans la classe peut être augmentée par des questions et réponses médiatisées par ces dispositifs personnels [14, 25, 26, 28].

5. CONCLUSION ET PERSPECTIVES

Un aspect caractéristique du projet Campus Mobile est sa multidisciplinarité. Il en a en effet nécessité de mettre en oeuvre et de coordonner des technologies appartenant à des domaines de recherche variés (l'interaction homme-machine, les réseaux sans-fil, les intergiciels, la gestion de documents et d'annotations, les collecticiels, etc.). Cette approche diffère de la plupart des autres projets réalisés ou en cours dans des domaines comparables, ces derniers étant souvent centrés sur des aspects plus spécifiques.

Un autre point important a été d'essayer d'associer les futurs utilisateurs à divers stades de conception et de réalisation du système. Nous avons ainsi commencé par établir un certain nombre de scénarios d'usage en interrogeant des élèves et des professeurs, ce qui a permis d'établir un certain nombre de points convergents qui sont à la base des systèmes réalisés. Des tests informels de certaines parties ont également permis de mettre en évidence certains problèmes parfois un peu négligés, comme par exemple les problèmes d'interaction avec un tableau augmenté. Nous avons également procédé à quelques évaluations qualitatives en situation de cours, qui seront complétées et affinées prochainement.

La suite du projet portera sur l'amélioration des outils existants, en particulier en ce qui concerne l'IHM, la découverte de services et les aspects liés au travail collaboratif.

6. REFERENCES

- [1] Abowd G.D. *Classroom 2000: An experiment with the instrumentation of a living educational environment*. IBM Systems Journal, Vol. 38, No. 4, 1999.
- [2] Anderson R., Anderson R., Simon B., Wolfman S. A., VanDeGrift T., Yasuhara K. *Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses*. Proc. SIGCSE'04, ACM, Norfolk, USA, 2004.
- [3] Balme L, Demeure A, Barralon N, Coutaz J, Calvary G. *Ethylene: a Software Architecture Reference Model for Distributed, Movable, Plastic User Interfaces*. EUSAI'04 pp. 291-302
- [4] Bier E.A., Stone M, Pier K., Buxton W., DeRose T.D., *Toolglass and Magic Lenses: The See-Through Interface*, Proc. ACM- SIGGRAPH, pp. 73-80, 1993.
- [5] Booth K.S. et al., *The "Mighty Mouse" Multi-Screen Collaboration Tool*, Proc UIST 2002, ACM, 209-212.
- [6] Brotherton J.A., *Enriching Everyday Experiences through the Automated Capture and Access of Live Experiences: eClass*. Georgia Tech, Ph.D. Thesis, Dec. 2001.
- [7] *Campus Mobile*: www.infres.enst.fr/~elc/campmob/
- [8] Chung G., Dewan P. Flexible Support for Application-Sharing Architecture. *Proc ESCSW*, 2001, Kluwer, 99-118.
- [9] B. David, F. Mercier, R. Chalon, O. Delotte, J.P. Berthet, *Amphi Interactif: état de l'art et projet DRIM-AP*, Premières Journées Francophones: Mobilité et Ubiquité, ACM 2004.
- [10] Denoue L., Singh G., Das A., *Taking Notes on PDAs with Shared Text Input*. EDMEDIA, Lugano, Switzerland, 2004.
- [11] Dragicevic P, Fekete J-D. Input Device Selection and Interaction Configuration with ICON, Proc. IHM-HCI 2001, Springer-Verlag, 543-548.
- [12] Envivio, *the 4Forum product*: www.envivio.com
- [13] Friedland G., Knipping L., Rojas R., *E-Chalk - Technical description*. Freie Universität Berlin, Institut für Informatik, Technical Report B-02-11, 2002, www.echalk.de
- [14] Iles A., Glaser D., Kam M., Canny J., *Learning via Distributed Dialogue: Livenotes and Handels Wireless*. Proc. of CSCS, Boulder CO, Lawrence Erlbaum, pp. 408-417, 2002.
- [15] *Ink Markup Language*: www.w3.org/2002/mmi/ink
- [16] IPM, *Interactive Presentation Manager*. Tutorial For Windows, Numonics Corporation, 2002. www.touchboards.com/numonics/ipm500TM.html
- [17] Johanson B., Hutchins G., Winograd T., Stone M., *PointRight: Experience with Flexible Input Redirection in Interactive Workspaces*, Proc. UIST, ACM, 227-234, 2002.
- [18] Kaddour M., Pautet L., *A Middleware for Supporting Disconnections & Multi-Network Access in Mobile Environments*. Proc. of 2nd Conference on Pervasive Comp. USA, 2004.
- [19] Lachenal C., *Modèle et infrastructure logicielle pour l'interaction multi-instruments multisurfaces*, thèse de l'Université Joseph Fourier – Grenoble 1, 2004.
- [20] Landay J.A., Davis R.C., *Making sharing pervasive: Ubiquitous computing for shared note taking*. IBM Systems Journal, Vol. 38, N. 4, pp. 531-550, 1999.
- [21] E. Lecolinet, *Pointeurs multiples: étude et implémentation*, Actes Journées IHM'03. pp. 134-141. Caen. ACM Press 2003.
- [22] Lecolinet E., *A molecular architecture for creating advanced interfaces*, ACM UIST, pp. 135-144. 2003.
- [23] Lienhard J., Lauer T., *Multi-Layer recording as a New Concept of Combining Lecture Recording and Students' Handwritten Notes*. Proc. Multimedia'02, ACM, 2002.
- [24] MIMIO: www.mimio.com
- [25] Mühlhäuser M., Trompler C., *Digital Lecture Halls Keep Teachers in the Mood and Learners in the Loop*. Proc. of E-Learn, AACE ed., Montreal, Canada. pp. 714-721, 2002.
- [26] Myers, B., *Using Handhelds and PCs Together*. Communications of the ACM, Vol. 44, No. 11, pp. 34-41, 2001.
- [27] Mynatt E. D., Igarashi T., Edwards W. K., Lamarca A. *Designing an Augmented Writing Surface*. IEEE Computer Graphics and Applications, pp. 55-61, 2000.
- [28] Ratto, M., Shapiro, B. R., Griswold, W. G., Truong, T. M., Griswold W.G., *The ActiveClass Project*. Computer Support for Collaborative Learning, Kluwer, pp. 477-486, 2003.
- [29] Richardson T, Stafford-Fraser Q., Wood K.R., Hopper A, *Virtual Network Computing*. IEEE Trans. on Internet Computing, 2(1), pp. 33-38, 1998.
- [30] Roseman M., Greenberg S., *Building Real Time Groupware with GroupKit, a Groupware Toolkit*. ACM ToCHI, 3(1), pp. 66-106, 1996.
- [31] Stewart J., Bederson B.B, Druin A., *Single display groupware: A model for co-present collaboration*. Proc. CHI, ACM. pp. 286-293, 1999.
- [32] *Synchronized Multimedia* : www.w3.org/AudioVideo/
- [33] *Scalable Vector Graphics, XML Graphics for the Web*, W3C recommendation: www.w3.org/Graphics/SVG/
- [34] Synergy <http://synergy2.sourceforge.net/>
- [35] Truong K.N., Abowd G.D., Brotherton J.A., *Personalizing the Capture of Public Experiences*. CHI Letters 1(1), ACM, pp. 121-130, 1999.
- [36] Tse E, Greenberg S. *SDGToolkit: A Toolkit for Rapidly Prototyping Single Display Groupware*. Proc. CSCW 02.
- [37] Ubit GUI toolkit: www.enst.fr/elc/ubit/
- [38] Yang C.-C., Yang Y.-Z., Lin K.-Y., *A SMIL-based Lesson Recording System for Distance Learning*. Proc. of Conf. on Distributed Multimedia Systems, pp. 486-489, 2001.
- [39] Yoneki E., *Pronto: Mobile Gateway With Publish-Subscribe Paradigm Over Wireless Network*. DS Online: (dsonline.computer.org), Expert Authored Articles, May 2003.

Columns on Last Page Should Be Made As Close As Possible to Equal Length