

KUP : un modèle pour la présentation opportuniste et multimodale d'informations à des utilisateurs mobiles

Christophe Jacquet

Supélec
91192 Gif-sur-Yvette Cedex
Christophe.Jacquet@supelec.fr

Yacine Bellik

LIMSI-CNRS
BP 133, 91403 Orsay Cedex
Yacine.Bellik@limsi.fr

Yolaine Bourda

Supélec
Yolaine.Bourda@supelec.fr

RESUME

Nous traitons de la conception de systèmes de fourniture multimodale d'informations dans le cadre de l'informatique diffuse. Nous proposons une architecture à agents basée sur un modèle théorique appelé KUP, qui constitue une alternative aux modèles d'architecture classiques en IHM. Ce modèle est accompagné d'un algorithme permettant de choisir la modalité à utiliser pour présenter une information donnée. Le modèle et l'algorithme ont été implémentés sous forme d'une plate-forme baptisée PRIAM. Celle-ci nous a permis de réaliser des expérimentations en laboratoire qui ont montré l'intérêt du système.

MOTS CLES : Informatique diffuse, ubiquitaire, ambiante, multimodalité, mobilité.

ABSTRACT

This paper deals with the design of a multimodal information system in ambient intelligence. Its agent architecture is based on KUP, an alternative to traditional human-computer interaction models. We propose an algorithm for choosing an interaction modality. Then, we introduce PRIAM, a platform that implements this theoretical framework. Finally, we present the results of experiments carried out in pseudo-real scale.

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

GENERAL TERMS: Human Factors, Experimentation

KEYWORDS: Ubiquitous computing, multimodality.

INTRODUCTION

Les usagers des lieux publics éprouvent généralement des difficultés à obtenir les informations dont ils ont besoin, en particulier lors de leurs premières visites. Par exem-

ple, lorsqu'un passager découvre un aéroport, il ignore où se trouve son comptoir d'embarquement. On dispose donc çà et là des dispositifs d'information : écrans, haut-parleurs, kiosques d'information, ou tout simplement des panneaux d'affichage. Cependant, ces points d'information ne sont pas ciblés. Ils sont souvent surchargés d'informations, ce qui rend leur lecture difficile. Pourtant, un utilisateur donné n'est généralement intéressé que par une information unique, qu'il a du mal à retrouver parmi une multitude d'informations superflues.

En réalité, cela ne sert à rien de présenter des informations qui n'intéressent pas l'utilisateur. Nous proposons donc un système ambiant de diffusion d'informations, capable de fournir des informations *personnalisées* aux utilisateurs mobiles.

Ainsi, un dispositif donné ne fournirait que les informations potentiellement intéressantes pour les personnes situées à proximité. Par exemple, des écrans disposés dans un hall d'aéroport n'afficheraient que des informations destinées aux passagers se trouvant devant eux. Le nombre d'informations différentes à afficher serait de fait limité, ce qui améliorerait la lisibilité de l'ensemble.

Si tous les utilisateurs éprouvent des difficultés en environnement inconnu, ces difficultés sont d'autant plus grandes pour des personnes handicapées. En effet, les dispositifs classiques d'information ne prennent généralement pas en compte les handicaps. Par exemple, les écrans ne sont d'aucune utilité pour les aveugles, les haut-parleurs ne sont pas perçus par les sourds, etc.

Pour ces raisons, nous nous intéressons aux présentations *multimodales*. Ainsi, un dispositif n'essaiera de fournir une information à un utilisateur que si sa modalité de sortie peut être perçue par la personne.

De plus, nous souhaitons nous affranchir de toute phase préalable de configuration. Dans [6], nous avons proposé des solutions pour que des dispositifs d'affichage soient capables de *coopérer entre eux* dès qu'ils sont placés quelque part. Dans cet article, nous prolongeons ce fonctionnement sans configuration préalable pour ce qui est de l'adaptation multimodale aux utilisateurs.

Nous commençons par passer en revue des travaux apparentés, puis nous introduisons un nouveau modèle d'architecture pour les systèmes diffus (ou "ambiants"). Nous détaillons la façon dont il est possible de choisir des modalités adaptées aux utilisateurs. Enfin, nous donnons les premiers résultats de notre travail, ainsi que des perspectives pour des travaux à venir.

TRAVAUX CONNEXES

Depuis une dizaine d'années, des travaux ont visé à fournir des informations aux utilisateurs mobiles. En général, les systèmes élaborés ont été construits à partir d'assistants numériques personnels (PDA) : le Cyberguide [9] fournit des informations sur la visite d'un musée ; CoolTown [7] affiche des pages web aux utilisateurs en fonction du lieu où ils se trouvent.

Ces approches ont un gros inconvénient : elles obligent les utilisateurs à porter sur eux un appareil électronique particulier. Même si aujourd'hui la quasi-totalité de la population est équipée de téléphones mobiles, ce n'est pas très pratique de devoir s'arrêter pour consulter son téléphone, surtout si l'on a les mains occupées. Par exemple, dans un aéroport, il est perturbant de devoir s'arrêter, poser ses bagages et sortir un téléphone.

Certes, certains systèmes plus récents, comme le Hello.Wall [12] visent à utiliser de grandes surfaces publiques pour l'affichage d'informations personnelles. Mais pour respecter la vie privée des utilisateurs [14], ces informations ne peuvent pas être diffusées "en clair". Ainsi, le Hello.Wall code-t-il les informations sous forme de motifs lumineux propres à chaque utilisateur. Ceci limite l'intérêt pratique du système, et, de l'aveu même de ses concepteurs, en fait plus un objet artistique qu'une interface réellement utilisable. Pour notre part, nous ne souhaitons pas diffuser d'informations *personnelles*, mais plutôt *effectuer une sélection* parmi les informations disponibles, ce qui limite les problèmes liés à la vie privée des utilisateurs.

Nous avons déjà proposé un modèle et des algorithmes qui permettent d'utiliser différents moniteurs publics pour afficher des informations à de multiples personnes mobiles [6]. Ainsi, nous nous plaçons dans le cadre des recherches sur les environnements d'affichage distribués (DDE, *distributed display environments*) [5]. Cependant, alors qu'habituellement les systèmes de DDE (par exemple [10]) sont basés sur une configuration statique d'écrans, nous avons introduit un modèle dans lequel l'affectation des informations aux écrans évolue de façon complètement dynamique. Le présent article pousse cette idée plus loin, en introduisant une double notion d'*opportunisme* dans la fourniture et la présentation des informations.

En outre, nous souhaitons aller au-delà d'une simple répartition de contenu entre écrans voisins, et notamment, prendre en compte de multiples modalités, ce que

n'aborde pas la problématique des DDE. Le présent article est donc également dédié à la négociation de contenu multimodal entre utilisateurs et dispositifs hétérogènes.

LE MODÈLE KUP

Besoins

Supposons qu'une personne se déplace dans un lieu public, et que des informations la concernant soient diffusées de temps en temps. Au moment de la diffusion d'une information donnée, la personne ne se trouve pas forcément à proximité d'un dispositif de présentation (moniteur, haut-parleur, etc.). En conséquence, cette information doit lui être présentée *plus tard*, lorsqu'elle se trouve en situation de percevoir un tel dispositif. Cela nous conduit à distinguer deux phases découplées :

- dans un premier temps, un item d'information est fourni à une entité informatique, *logique*, associée à l'utilisateur ;
- dans un second temps, cet item est présenté *physiquement* à l'utilisateur via un dispositif et une modalité adaptés (texte sur un écran, synthèse vocale, etc.)

Sources d'information, utilisateurs et dispositifs de présentation

Pour rendre ce découplage possible, nous avons élaboré le modèle KUP, un modèle d'architecture logicielle pour les IHM des systèmes diffus. KUP repose sur trois sortes d'entités *logiques* :

- les sources d'information, chargées de la diffusion des informations. On les note K_ℓ (K pour *knowledge*),
- les entités logiques représentant les utilisateurs, U_ℓ ,
- les entités logiques représentant les dispositifs de présentation, P_ℓ .

Chacune de ces entités logiques correspond à une entité physique associée, respectivement :

- les périmètres spatiaux dans lesquels les informations diffusées sont valides, notés K_φ ,
- les utilisateurs physiques, notés U_φ ,
- les dispositifs de présentation physiques, notés P_φ .

La plupart des modèles d'architecture logicielle pour les IHM (MVC [8], Seeheim [11], ARCH [2] et PAC [3]) sont basés sur des représentations logiques uniquement pour le noyau fonctionnel et l'interface (voir fig. 1). Il n'existe pas de représentation logique de l'utilisateur. À l'opposé, le modèle KUP nous permet de mettre l'entité logique associée à l'utilisateur au centre du système (voir fig. 2) :

- lors de la première phase vue ci-dessus, une source d'informations K_ℓ envoie un item d'information à l'entité utilisateur U_ℓ ;
- lors de la seconde phase, l'entité utilisateur U_ℓ négocie avec une entité de présentation P_ℓ la présentation (concrète) de cet item. La présentation est alors effectuée par un dispositif physique P_φ , à destination de l'utilisateur humain U_φ .

Notons que dans KUP le flot d'information (représenté par les flèches horizontales sur la figure 2) est actuellement

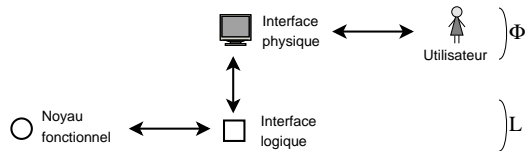


Figure 1 : Dans les modèles d'architecture classiques, l'utilisateur n'a pas de représentation logique. Les lettres Φ et L désignent respectivement les couches physique et logique.

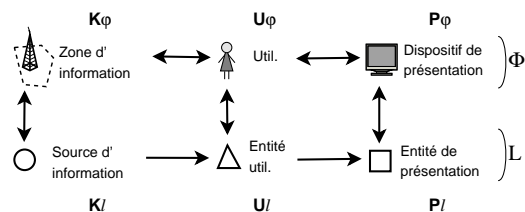


Figure 2 : L'utilisateur se trouve au centre du modèle KUP.

unidirectionnel, alors qu'il est bidirectionnel dans la plupart des modèles classiques. En effet, nous nous sommes intéressés en priorité à la *présentation d'informations* (en sortie). Nous envisageons cependant des extensions dans lesquelles le modèle KUP pourrait être rendu symétrique (entrées/sorties).

Espace perceptuel, espace de rayonnement

Les entités physiques entretiennent entre elles des relations de *perception*. Pour une entité donnée, son *espace de rayonnement* est l'ensemble des positions de l'espace physique desquelles une autre entité peut la *percevoir*. Réciproquement, elle peut percevoir d'autres entités si ces dernières se trouvent en des positions bien particulières. Ces positions constituent son *espace perceptuel*.

Dans ces définitions, la *perception* s'entend d'un point de vue *sensoriel*. Par exemple, l'espace perceptuel d'un utilisateur voyant contiendra les écrans situés devant lui, à distance de lecture, et les haut-parleurs alentour. Par contre, l'espace perceptuel d'un aveugle situé au même endroit ne contiendra que les haut-parleurs. Ainsi, la perception dépend non seulement de la proximité géométrique, mais également de l'orientation des entités, de leurs capacités sensorielles et des modalités utilisées. Dans la suite, nous utilisons simplement le terme *proximité* pour désigner la *proximité perceptuelle*.

Les relations de proximité proviennent du monde physique, puis elles sont transposées aux entités logiques, qui partagent alors les *mêmes* relations.

Présentation opportuniste d'informations

Les items d'information sont considérés comme élémentaires, et formellement appelés *unités sémantiques*. Ces dernières peuvent transiter sur un réseau, et peuvent s'exprimer selon un certain nombre de modalités.

Parmi les deux phases de *fourniture* puis *présentation* des informations, nous nous intéressons uniquement à la seconde. En effet, la première phase peut être très simple : lorsqu'un utilisateur pénètre dans l'espace perceptuel d'une source d'information, cette dernière fournit si besoin une unité sémantique à l'entité logique U_ℓ .

Pour ce qui est de la phase de présentation, lorsqu'un utilisateur s'approche d'un ou plusieurs dispositifs, le système doit résoudre deux problèmes imbriqués :

1. si plusieurs dispositifs sont disponibles, il faut choisir l'un d'entre eux. Ce point a déjà été traité [6] ;
2. un dispositif de présentation donné doit sélectionner une modalité adaptée à l'utilisateur. Nous nous plaçons dans le cadre de la *multimodalité exclusive* [13] (utilisation d'une seule modalité à la fois), sachant qu'à l'avenir il serait possible d'utiliser plusieurs modalités en complémentarité, redondance ou équivalence [4].

Les deux phases que nous venons de voir rendent le fonctionnement du système doublement opportuniste :

- lors de la *fourniture* des informations : l'entité logique de l'utilisateur reçoit des unités sémantiques au gré des déplacements de l'utilisateur physique ;
- lors de la *présentation* des informations : la présentation des unités sémantiques s'effectue lorsque l'utilisateur rencontre fortuitement un dispositif de présentation.

Architecture

Il aurait été possible de concevoir un système *centralisé*. Cependant, cette approche souffre de plusieurs inconvénients, notamment sa fragilité (si un serveur central est défaillant, cela empêche toutes les entités de fonctionner) et sa rigidité (on ne peut pas déplacer les sources d'information et les dispositifs de présentation à volonté). Au contraire, nous souhaitons pouvoir apporter, déplacer et retirer de nouvelles entités sans qu'il y ait besoin de reconfigurer quoi que ce soit. Le système doit être capable de s'adapter automatiquement.

Nous proposons donc d'implémenter les entités logiques par des agents logiciels : des agents K, U et P, associés respectivement aux entités K_ℓ , U_ℓ et P_ℓ . Les relations de proximité, captées dans le monde physique, sont transposées aux agents.

Nous supposons que les agents peuvent communiquer entre eux via un réseau omniprésent et fiable. Cette supposition est réaliste depuis l'avènement des réseaux mobiles et sans fil. Par ailleurs, les agents sont *réactifs*. Un agent demeure en veille la plupart du temps, et réagit à deux sortes d'événements :

- la réception d'un message sur le réseau, émis par un autre agent ;
- un changement des relations de proximité dans son espace perceptuel (approche ou éloignement d'entités).

Les agents étant réactifs, tous les comportements résultent d'événements dans le monde physique, seul élément proactif du modèle. En pratique, les entités proactives seront en majorité les utilisateurs physiques, dont on suppose qu'ils se trouvent dans une *situation de mobilité*.

Les événements du monde physique sont captés par des dispositifs électroniques, afin de déclencher des comportements dans le monde des agents. Par exemple, la technologie RFID peut être utilisée pour détecter la proximité et construire les espaces perceptuels. Ainsi, les moniteurs d'un aéroport pourraient détecter l'approche des passagers grâce aux tickets de ces derniers, que l'on peut équiper de puces RFID. D'autres techniques sont utilisables, comme la vision par ordinateur ou le Bluetooth.

CHOIX D'UNE MODALITÉ

Taxonomie des modalités

Une *modalité* est une forme concrète de communication basée sur l'un des cinq sens humains [13], par exemple la parole, le texte écrit ou la musique.

Afin de déterminer une modalité utilisable, nous proposons de commencer par construire une taxonomie des modalités. Celle-ci classe les modalités de façon arborescente. Les feuilles représentent des modalités concrètes, tandis que les nœuds internes représentent des modalités abstraites, regroupant plusieurs modalités. La racine est une modalité abstraite qui englobe toutes les autres. Les modalités abstraites du second niveau (visuelle, auditive et tactile) correspondent aux trois *sens* humains actuellement exploités en IHM (visuel, auditif et tactile).

Les modalités possèdent des *attributs* qui caractérisent les présentations concrètes qui les utilisent. Les attributs peuvent avoir des valeurs discrètes (par exemple, une police de caractères parmi une liste donnée) ou continues (par exemple, la taille d'un texte). Avant de présenter une information selon une modalité, les valeurs de ses attributs doivent être déterminées. Ce processus est appelé *instanciation* [1].

Profils

Nous devons résoudre le problème suivant : un utilisateur souhaite voir présentée une unité sémantique sur un dispositif de présentation donné. Le système doit donc choisir une modalité et l'instancier pour pouvoir réaliser cette présentation. La modalité et son instanciation doivent être compatibles à la fois avec :

- les capacités de l'utilisateur (par exemple, on ne peut pas utiliser une modalité visuelle s'il est aveugle), ou ses préférences (par exemple, s'il préfère le texte aux graphismes, le système doit essayer de prendre en compte ce souhait) ;
- les capacités du dispositif de présentation (par exemple, un écran monochrome ne gère pas la couleur) ;
- les capacités d'expression de l'unité sémantique selon différentes modalités.

S'il y a plusieurs possibilités, le système devra choisir parmi elles la solution *préférée par l'utilisateur*.

Pour résoudre ce problème, on associe un *profil* à l'utilisateur, au dispositif et à l'unité sémantique. Ces profils indiquent quelles modalités peuvent être utilisées et quelles valeurs sont possibles pour les attributs. La solution devra satisfaire *chaque* profil, et se situera donc à l'*intersection* des trois profils.

Nous définissons un profil en tant que pondération de l'arbre des modalités. Un nombre réel compris entre 0 et 1 est associé à chaque nœud de l'arbre. 0 signifie que la modalité ou le sous-arbre correspondant ne peut être utilisé ; 1 signifie qu'il peut être utilisé. Les valeurs intermédiaires indiquent un niveau de préférence. Par exemple, dans le profil d'un non-voyant, le sous-arbre correspondant aux modalités visuelles est pondéré par 0, de façon à ne pouvoir être utilisé. De même, dans le profil d'un moniteur, seul le sous-arbre correspondant aux modalités visuelles possède une pondération non nulle.

Le poids des nœuds détermine le choix d'une modalité. De même, les valeurs possibles pour les attributs sont elles aussi pondérées, ce qui permet de faire le choix d'une instanciation. Plus précisément, chaque valeur possible de chaque attribut reçoit un poids compris entre 0 et 1, avec la même signification que pour les nœuds. Formellement, une *fonction de pondération* est associée à l'attribut. Elle est définie sur les valeurs possibles pour l'attribut, et prend ses valeurs dans l'intervalle $[0, 1]$.

La figure 3 est un exemple partiel de profil, basé sur une taxonomie qui ne contient que deux modalités concrètes. Le profil décrit une personne malvoyante, dont la langue natale est l'Anglais. Les poids des nœuds (pondérations des modalités) sont indiqués dans les ovales noirs. Puisque l'utilisateur a un handicap visuel, mais n'est pas aveugle, le poids des modalités visuelles est faible, mais non nul.

Les fonctions de pondération des attributs sont indiquées dans les rectangles aux coins arrondis. Des fonctions discrètes sont associées aux attributs discrets. Ainsi, on indique des poids pour chaque valeur possible de l'attribut *lang*. Des fonctions continues sont associées aux attributs continus. Par exemple, une fonction associe un poids à chaque vitesse de parole en mots par minute.

Choix d'une modalité

Cette section explique comment les profils peuvent être utilisés pour déterminer quelles sont les modalités adéquates pour effectuer une présentation donnée. La figure 4 donne une vue d'ensemble des différentes étapes.

Pour sélectionner une modalité, le système doit tenir compte de trois profils (ceux de l'utilisateur, du dispositif et de l'unité sémantique). Pour ce faire, nous définissons la notion d'*intersection* des profils.

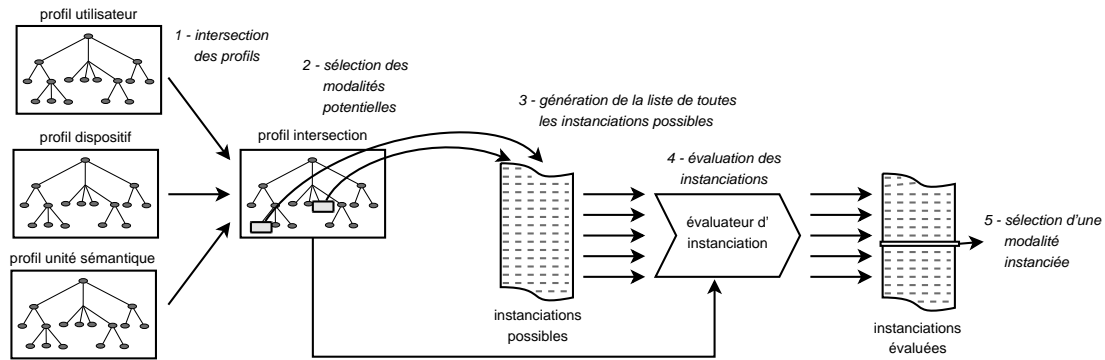


Figure 4 : Vue synoptique de l'algorithme de choix d'une modalité. On calcule d'abord l'intersection des profils, ce qui donne une liste de modalités utilisables. Chaque instanciation possible de ces modalités est alors *évaluée*, de façon à choisir la meilleure.

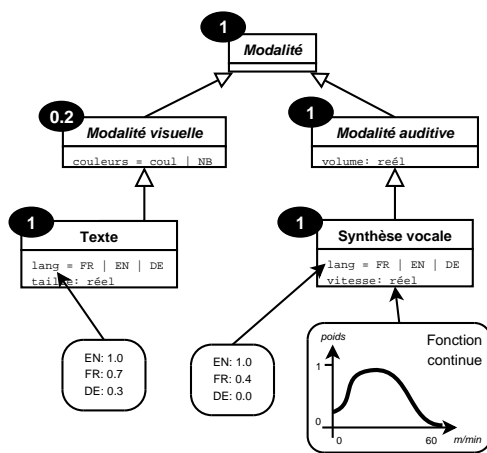


Figure 3 : Profil partiel (pour ne pas alourdir le schéma, certaines fonctions ne sont pas indiquées).

L'intersection de n profils p_1, \dots, p_n est un profil (i.e. un arbre de modalités pondéré) dans lequel les poids sont ainsi définis :

- le poids d'un nœud est le produit des n poids du même nœud dans les profils p_1, \dots, p_n ;
- la fonction de pondération d'un attribut est le produit des n fonctions de pondération du même attribut dans les profils p_1, \dots, p_n .

Nous appelons cette opération *intersection* vue sa sémantique naturelle. En effet, un nœud est pondéré par 0 dans le profil résultant si et seulement si l'un des profils p_1, \dots, p_n est lui-même pondéré par 0. Le profil résultant est noté p_{\cap} . Il indique quelles modalités peuvent être utilisées pour présenter une unité sémantique donnée à un utilisateur donné, sur un dispositif donné. Il permet également de déterminer les valeurs des attributs pour la modalité choisie (instanciation, voir ci-dessous).

Il s'agit maintenant de choisir une modalité concrète à partir de p_{\cap} , i.e. une feuille de l'arbre. Pour ce faire, le système *évalue* chaque feuille. L'évaluation d'une feuille est un nombre réel qui rend compte des poids attribués à ce nœud et à tous ces ancêtres dans l'arbre. Ainsi, si

un nœud interne a un poids nul, cela signifie que le sous-arbre correspondant est dans tous les cas inutilisable, et donc l'évaluation de toutes ses feuilles doit valoir 0. Nous pourrions donc définir l'évaluation d'une feuille comme étant le *produit* des poids de ses ancêtres. Cependant, dans ce cas, les feuilles situées profond dans l'arbre auraient par nature plus de chances d'être créditées d'une faible évaluation que les feuilles peu profondes.

Pour éviter ce problème, nous *normalisons* ce produit par rapport au nombre d'ancêtres. Nous définissons donc l'évaluation d'une modalité concrète comme étant la *moyenne géométrique* des poids de tous ses ancêtres (y compris lui-même). À partir de là, nous pouvons facilement décider d'utiliser la modalité concrète qui dispose de l'évaluation la plus élevée.

La figure 5 illustre l'intersection des profils et l'évaluation des modalités sur un exemple simple. Dans ce cas, le système utiliserait la modalité dont l'évaluation vaut 0.65.

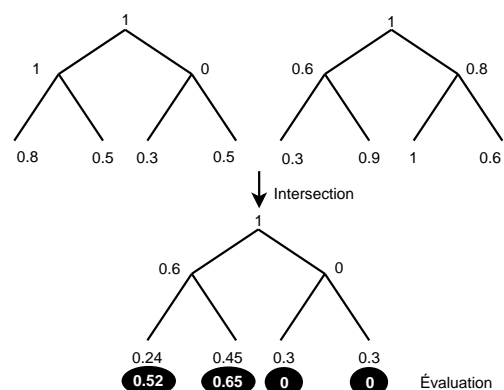


Figure 5 : Intersection et évaluation.

Instanciation de la modalité choisie

Une fois qu'une modalité est sélectionnée, il s'agit de l'*instancier*, c'est-à-dire de déterminer les valeurs de ses attributs. C'est ici qu'interviennent les fonctions de pondération de p_{\cap} . Il faut réaliser un *compromis global* entre les capacités et préférences de tous les utilisateurs

situés à proximité, de toutes les unités sémantiques à présenter, et du dispositif de présentation.

Par exemple, supposons que deux utilisateurs veulent chacun qu'une unité sémantique soit affichée sur un écran, sous forme d'une ligne de texte. Tous les deux désirent que la taille des caractères soit la plus grande possible, mais malheureusement, la surface disponible à l'écran est limitée. Ceci impose des contraintes sur les tailles de police utilisables, et donc il faut trouver un compromis.

Nous supposons qu'il y a un certain nombre d'unités sémantiques présentes sur un dispositif donné, pour un total de n attributs, dont les domaines sont notés D_1, \dots, D_n . Nous appelons *espace des combinaisons d'attributs* l'ensemble de toutes les combinaisons de valeurs possibles. Cet espace est noté $\Omega = D_1 \times D_2 \times \dots \times D_n$ (produit cartésien).

Certains des éléments de cet ensemble ne sont pas compatibles avec les contraintes intrinsèques du dispositif de présentation. Nous définissons donc $\tilde{\Omega}$ comme étant le sous ensemble de Ω dont les éléments sont compatibles avec ces contraintes. Alors, la "meilleure" combinaison d'attributs est un élément de $\tilde{\Omega}$. Informellement, la "meilleure" solution est celle qui satisfait au mieux le plus d'utilisateurs. Voyons comment ceci peut se formaliser.

Dans la suite de ce qui a été fait ci-dessus, nous définissons la *fonction d'évaluation* d'une modalité concrète comme étant la moyenne géométrique des fonctions d'évaluation des attributs de la modalité concrète et de ses ancêtres. S'il y a p attributs, de domaines d_1, \dots, d_p , de fonctions de pondération f_1, \dots, f_p , la fonction d'évaluation de la modalité concrète, notée e , est définie sur $d_1 \times d_2 \times \dots \times d_p$ par :

$$e(x_1, x_2, \dots, x_p) = \sqrt[p]{f_1(x_1) \times f_2(x_2) \times \dots \times f_p(x_p)}$$

Ainsi, chaque utilisateur intéressé par une ou plusieurs unités sémantiques donne lieu à une fonction d'évaluation. Supposons qu'il y a q fonctions d'évaluation, e_1, \dots, e_q . Prenons l'une d'entre elles, notée e_i . e_i est a priori définie sur un sous-ensemble Ω , donc on peut la prolonger (par projection) sur $\tilde{\Omega}$ ou $\tilde{\Omega}$. Cette extension est notée \tilde{e}_i .

On peut alors associer un vecteur à q composants à chaque élément $\omega \in \tilde{\Omega}$, composé des q valeurs $\tilde{e}_1(\omega), \dots, \tilde{e}_q(\omega)$ classées par ordre croissant. Ce vecteur est appelé *évaluation* de ω , et est noté $e(\omega)$. Pour une combinaison donnée de valeurs d'attributs, $e(\omega)$ est la liste des évaluations de la combinaison, commençant par celle attribuée par l'utilisateur le moins satisfait.

Nous voulons donner satisfaction au plus grand nombre, et éviter que quiconque soit négligé. Pour ces raisons, nous décidons de choisir la combinaison d'attributs dont la pire évaluation (donc la première composante du vecteur $e(\omega)$) est maximale. En cas d'égalité sur les premières

composantes, cela revient à choisir la combinaison ω pour laquelle $e(\omega)$ est le plus grand selon l'ordre lexicographique.

Exemple — supposons qu'un dispositif doit présenter trois unités sémantiques pour trois utilisateurs A, B and C . Le système doit déterminer les valeurs de cinq attributs, étant données les évaluations fournies par les utilisateurs. Le problème se formalise ainsi :

| ω – Valeurs | e_A | e_B | e_C | $e(\omega)$ – Éval. |
|--------------------|-------|-------|-------|---------------------|
| (fr, 4, de, 6, 7) | 0.7 | 0.8 | 0.6 | (0.6, 0.7, 0.8) |
| (it, 2, en, 9, 1) | 0.9 | 0.3 | 0.7 | (0.3, 0.7, 0.9) |
| (en, 2, de, 3, 5) | 0.8 | 0.7 | 0.9 | (0.7, 0.8, 0.9) |
| (es, 8, fr, 1, 3) | 0.6 | 0.9 | 0.5 | (0.5, 0.6, 0.9) |
| (de, 3, es, 7, 5) | 0.2 | 0.4 | 0.95 | (0.2, 0.4, 0.95) |

La première colonne contient les combinaisons des attributs. Les trois colonnes suivantes contiennent les évaluations des utilisateurs correspondantes, et la dernière colonne le vecteur d'évaluation global, composé des valeurs des trois colonnes précédentes classées par ordre croissant. La solution choisie est la troisième (grisée), car elle maximise le niveau de satisfaction de l'utilisateur le moins bien loti (toutes les évaluations valent au moins 0.7 dans cette solution).

IMPLÉMENTATION ET ÉVALUATION

Afin de réaliser des expérimentations, nous avons implémenté une plate-forme qui met en œuvre les concepts et mécanismes décrits ci-dessus. Elle est appelée PRIAM, pour PRésentation d'Informations dans l'AMbiant. Cette implémentation, réalisée en Java, permet aux entités de communiquer entre elles de façon transparente grâce au mécanisme des RMI (Remote Method Invocation).

Les évaluations ont pour objectif de démontrer l'intérêt des systèmes de présentation d'informations dynamiques pour des utilisateurs mobiles. Elles mettent en jeu des écrans (simulés par des écrans d'ordinateur). La proximité entre les écrans et les utilisateurs a été captée par détection de badges à infrarouges. D'autres techniques auraient pu être utilisées, comme les RFID, mais les infrarouges ont l'avantage de permettre non seulement la détection de la proximité, mais également de l'orientation des personnes. Ainsi, une personne qui tournait le dos à un écran n'est pas détectée, ce qui correspond bien à la notion de *proximité perceptuelle*.

Affichage dynamique pour la recherche d'informations

Nous avons effectué une évaluation de l'impact de l'affichage dynamique d'informations en termes de temps de recherche d'un item d'information. Les sujets (au nombre de 16) devaient retrouver un item particulier parmi une liste d'items similaires. Nous leur avons proposé deux tâches différentes : retrouver une note obtenue à un examen, et retrouver les détails d'un vol dans un

aéroport. Nous avons mesuré le temps de recherche pour chaque utilisateur, en fonction du nombre d'utilisateurs rassemblés autour du dispositif d'affichage. Nous avons fait varier ce nombre entre 1 et 8, ce qui correspond au nombre maximum d'utilisateurs pouvant se trouver à proximité immédiate d'un même dispositif.

D'une part, dans des expériences témoins, il s'agissait de retrouver une note parmi 450 sur des listes papier, ou bien un vol parmi 20 sur un écran statique. D'autre part, en utilisant un système dynamique, un panneau d'affichage affichait uniquement les informations pertinentes pour les personnes situées à proximité (i.e. 1 à 8 items).

Cette expérience a montré que la recherche d'informations est bien plus rapide lorsque les listes sont dynamiques :

- pour la recherche de notes (voir fig. 6), les gains en temps s'échelonnent de 51 % à 83 % (selon le nombre d'utilisateurs), avec une moyenne à 72 % ;
- pour la recherche de vols (voir fig. 7), les gains en temps s'échelonnent de 32 % à 75 % (selon le nombre d'utilisateurs), avec une moyenne à 52 %.

Cependant, les utilisateurs sont généralement perturbés par les items qui apparaissent et disparaissent dynamiquement, ce qui provoque des ré-affichages à chaque fois, de façon à maintenir les listes classées par ordre alphabétique ou chronologique. Ce problème pourrait être résolu, soit en prévoyant des transitions douces entre affichages successifs (i.e. lors de l'ajout ou de la suppression d'items), soit en ajoutant les nouveaux items en fin de liste.

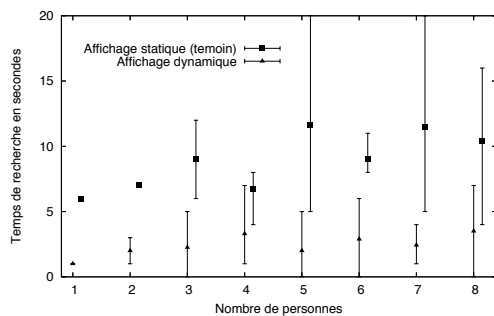


Figure 6 : Temps de recherche d'une note en fonction du nombre de personnes présentes. Les lignes verticales représentent les écarts-types, et les points les valeurs moyennes.

Amélioration des déplacements dans une gare

Dans une seconde expérience, nous avons *ajouté* des informations dynamiques à des écrans d'affichage initialement statiques, comme ceux situés dans les passages souterrains des gares. Dans ces passages, il y a généralement un écran près des escaliers qui mènent à chaque quai : il affiche les horaires des trains au départ de ce quai. Cependant, lorsqu'un passager doit effectuer une correspondance, il ne sait pas a priori dans quelle direction se diriger dans le passage souterrain, donc en moyenne une fois sur deux, il est obligé de revenir sur ses pas après

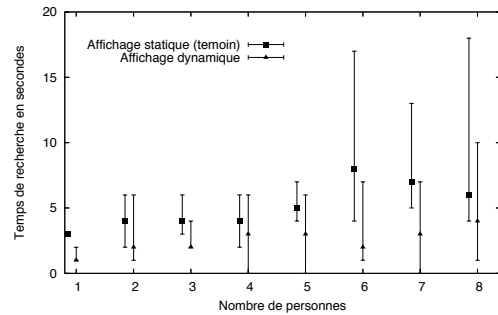


Figure 7 : Temps de recherche d'un vol d'avion en fonction du nombre de personnes présentes.

avoir parcouru la totalité du souterrain dans la mauvaise direction.

Nous avons affiché des informations personnalisées sur *n'importe-quel* écran dont s'approche un utilisateur. Ces informations comportent le numéro de quai, ainsi qu'une flèche indiquant la direction à prendre. Cela ne *remplace* pas les affichages statiques habituels, mais les *complète*. Selon notre hypothèse, ces informations supplémentaires pourraient aider les utilisateurs à trouver leur quai.

Nous avons reproduit un passage souterrain dans un couloir du laboratoire. Cinq écrans d'ordinateur simulaient les écrans d'un passage souterrain. Les personnes partaient de positions aléatoires dans le couloir, et devaient trouver un "quai" donné. Ils connaissaient uniquement leur ville de destination, et pas leur numéro de quai. Nous avons compté le nombre de déplacements élémentaires des utilisateurs (n_u), et nous l'avons comparé avec le nombre *optimal* de déplacements (n_o). Nous appelons *longueur relative* des déplacements le rapport $\frac{n_u}{n_o}$.

Lorsque nous avons fourni uniquement des informations statiques aux sujets, ces derniers ont souvent fait des erreurs, ce qui a causé des déplacements superflus (voir tableau 1). Par contre, lorsque nous leur avons fourni des informations dynamiques personnalisées, les sujets ont systématiquement suivi des chemins optimaux (longueur relative égale à 1). Ces résultats ont été confirmés même lorsque plusieurs utilisateurs, qui devaient se rendre sur des quais différents, se trouvaient en même temps dans le couloir. De plus, les utilisateurs ont semblé apprécier le système, et n'ont paru ni perturbés ni distraits.

| Sujet | n_u | n_o | Longueur relative |
|---------|-------|-------|-------------------|
| a | 7 | 4 | 1.75 |
| b | 3 | 3 | 1.00 |
| c | 9 | 2 | 4.50 |
| Moyenne | — | — | 2.42 |

Table 1 : Longueurs relatives lorsque l'on ne diffuse que des informations statiques.

Les évaluations ont montré l'intérêt des systèmes d'affichage dynamique des informations pour des utilisateurs mobiles. Ainsi, ces systèmes réduisent grandement le temps de recherche des informations dans une liste. De plus, fournir des informations supplémentaires et personnalisées sur la direction à suivre conduit à une réduction drastique des déplacements inutiles.

CONCLUSIONS ET PERSPECTIVES

Nous avons présenté un modèle et un algorithme pour la conception de systèmes multimodaux de présentation d'informations. Ce travail est utilisé pour fournir des informations à des utilisateurs mobiles. Ainsi, des informations personnalisées peuvent être fournies à des utilisateurs, *via* des dispositifs publics de présentation. Nous avons réalisé des expérimentations dans des conditions semi-réelles, ce qui nous pousse à formuler les perspectives suivantes.

Sur un écran donné, il serait intéressant de *classer* les unités sémantiques selon des critères pertinents. Par exemple, un *niveau de priorité* pourrait être affecté à chaque unité. Cela permettrait aux informations de haute priorité (par exemple, l'horaire d'un vol dont l'embarquement est sur le point de se terminer) d'apparaître en premier. De même, il pourrait y avoir des priorités entre utilisateurs. Ainsi, les informations concernant les handicapés, ou les abonnés, pourraient être jugées plus prioritaires.

Dans cet article, la notion de proximité utilisée est *binnaire* : des entités sont soit proches, soit éloignées. En réalité, on pourrait introduire des *degrés de proximité*, ou même une distance. Ces paramètres pourraient être utilisés lors de l'instanciation. Par exemple, les écrans pourraient afficher en plus gros caractères les textes destinés à des personnes éloignées.

Nos expériences se sont déroulées en environnement simulé (une pièce et un couloir d'un laboratoire). Il sera nécessaire de réaliser des expériences grandeur nature, par exemple dans un aéroport ou une gare. Leur rôle ne sera pas de tester ou valider les algorithmes, car cela a déjà été fait, mais plutôt :

- d'évaluer l'*utilisabilité* globale des systèmes, et notamment les réactions des utilisateurs face à des interfaces hautement dynamiques ;
- d'étudier l'impact sociologique du système ;
- de tester l'*utilisabilité* de notre plate-forme PRIAM, du point de vue du concepteur d'applications.

Ce travail s'est uniquement concentré sur le traitement des interactions, et n'a pas traité les problèmes liés au respect de la vie privée des personnes. Or ceux-ci risquent d'être particulièrement prégnants lors de l'utilisation d'un système tel que le nôtre. En effet, si un utilisateur se trouve seul devant un écran, alors seules ses informations sont affichées, par exemple la destination de son train. Une personne mal intentionnée et dissimulée à proximité pourrait alors surprendre ces informations privées.

Afin d'évaluer l'impact de ces problèmes et de proposer des solutions, des études aussi bien sociologiques qu'ergonomiques seront nécessaires.

BIBLIOGRAPHIE

1. André, E. The Generation of Multimedia Presentations. In Dale, R., Moisl, H., and Somers, H., editors, *A Handbook of Natural Language Processing*, pages 305–327. 2000.
2. Bass, L., Faneuf, R., Little, R., Mayer, N., Pellegrino, B., Reed, S., Seacord, R., Sheppard, S., and Szczur, M. R. A Metamodel for the Runtime Architecture of an Interactive System. *SIGCHI Bulletin*, 24(1):32–37, 1992.
3. Coutaz, J. PAC, an Object-Oriented Model for Dialog Design. In Bullinger, H.-J., and Shackel, B., editors, *INTERACT'87*, pages 431–436. North-Holland, 1987.
4. Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., and Richard, Y. M. Four easy pieces for assessing the usability of multimodal interaction: the CARE properties. In *INTERACT'95*, pages 115–120, 1995.
5. Hutchings, D., Stasko, J., and Czerwinski, M. Distributed display environments. *Interactions*, 12(6):50–53, 2005.
6. Jacquet, C., Bellik, Y., and Bourda, Y. PRIAM : affichage dynamique d'informations par des écrans coopérants en environnement mobile. In *Actes d'Ubimob 2006, conférence francophone ubiquité et mobilité*, pages 33–40, Sept. 2006.
7. Kindberg, T., and Barton, J. A Web-based nomadic computing system. *Computer Networks*, 35(4):443–456, 2001.
8. Krasner, G. E., and Pope, S. T. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object Oriented Programming*, 1(3):26–49, 1988.
9. Long, S., Kooper, R., Abowd, G. D., and Atkeson, C. G. Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. In *Mobile Computing and Networking*, pages 97–107, 1996.
10. Mansoux, B., Nigay, L., and Troccaz, J. The Mini-Screen: an Innovative Device for Computer Assisted Surgery Systems. *Studies in Health Technology and Informatics*, 111:314–320, 2005.
11. Pfaff, G. E., editor. *User Interface Management Systems: Proceedings of the Seeheim Workshop*. Springer, 1985.
12. Streitz, N. A., Röcker, C., Prante, T., Stenzel, R., and van Alphen, D. Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments. In *HCI International*, June 2003.
13. Teil, D., and Bellik, Y. Multimodal Interaction Interface using Voice and Gesture. In Taylor, M. M., Néel, F., and Bouwhuis, D. G., editors, *The Structure of Multimodal Dialogue II*, chapter 19, pages 349–366. 2000.
14. Vogel, D., and Balakrishnan, R. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *UIST '04*, pages 137–146. ACM Press, 2004.