

---

# KUP, un modèle pour la présentation multimodale et opportuniste d'informations en situation de mobilité

Christophe Jacquet <sup>\*,\*\*</sup> — Yacine Bellik <sup>\*\*</sup> — Yolaine Bourda <sup>\*</sup>

<sup>\*</sup> *Supélec*  
Plateau de Moulon  
F-91192 Gif-sur-Yvette cedex  
{Christophe.Jacquet, Yolaine.Bourda}@supelec.fr

<sup>\*\*</sup> *LIMSI-CNRS*  
BP 133  
F-91403 Orsay cedex  
Yacine.Bellik@limsi.fr

---

*RÉSUMÉ.* Cet article traite de la conception d'une plate-forme multimodale d'information, dans laquelle des informations sont fournies et présentées à des utilisateurs mobiles de façon opportuniste. Elle repose sur KUP, un modèle d'interaction radicalement différent des modèles classiques en IHM. Les interactions sont basées sur la notion de proximité, que nous définissons formellement à partir d'ensembles de perception et de rayonnement. Nous donnons alors une description détaillée des algorithmes utilisés pour choisir une modalité d'interaction entre un utilisateur et un dispositif interactif.

*ABSTRACT.* This paper deals with the design of a multimodal information platform, in which information items are opportunistically provided and presented to mobile users. It relies on KUP, an interaction model which is radically different from traditional human-computer interaction models. Interactions are proximity-based, so we give formal definitions of related notions, such as perceptive and radiance spaces. Then, we give a detailed description of the algorithms used to choose an interaction modality between a user and a given interaction device.

*MOTS-CLÉS :* informatique ubiquitaire, mobilité, intelligence ambiante.

*KEYWORDS:* ubiquitous computing, mobility, ambient intelligence.

---

## 1. Introduction

Les utilisateurs des lieux publics éprouvent souvent des difficultés à obtenir les informations dont ils ont besoin. Par exemple, lorsqu'un passager arrive dans un aéroport, il ne sait pas *a priori* où il doit se rendre pour prendre son avion. Ces difficultés sont d'autant plus grandes lorsque l'environnement est inconnu. Cependant, elles sont toujours présentes : dans un aéroport, une personne, même habituée, peut difficilement deviner à l'avance où aura lieu son embarquement.

Afin de fournir aux utilisateurs les informations dont ils ont besoin, on installe donc généralement des *points d'information* de place en place. Ce peut être des écrans, des haut-parleurs, des kiosques interactifs, ou simplement des panneaux d'affichage. Cependant, ces points d'information fournissent des renseignements non ciblés, à destination de tout le monde. En conséquence, ils sont le plus souvent surchargés d'informations, ce qui rend leur lecture difficile. Or, une personne donnée est généralement intéressée par une seule information : la rechercher parmi une multitude d'informations inintéressantes est une tâche longue, voire fastidieuse.

Pour améliorer cette situation, et partant du principe qu'il ne sert à rien de présenter une information qui n'intéresse personne, nous nous proposons de concevoir un système *opportuniste* de présentation d'informations, qui ne fournirait des informations qu'aux utilisateurs situés à proximité. Il ne s'agit pas pour autant de fournir des informations *personnelles*, mais plutôt d'opérer une *sélection* parmi toutes les informations, de sorte à ne fournir que les informations *pertinentes* pour les utilisateurs présents. On rejoint ainsi la vision de l'informatique ubiquitaire (Weiser, 1993).

En ce sens, les informations pourraient être fournies par les dispositifs *publics* mentionnés ci-avant. Par exemple, des moniteurs disposés au hasard dans un aéroport pourraient fournir aux passagers situés à proximité des informations sur leur vol ou sur d'autres sujets d'intérêt.

Nous venons de le voir, tous les utilisateurs, quels qu'ils soient, sont confrontés à des difficultés lorsqu'il s'agit d'obtenir des informations et de se diriger dans un environnement inconnu. Cependant, il existe une catégorie de personnes pour lesquelles ces tâches sont particulièrement délicates : les handicapés sensoriels. En effet, les dispositifs d'information ne leur sont pas nécessairement adaptés. Ainsi, un écran d'information ne sera d'aucune utilité à un non-voyant ; de même, un malentendant ne percevra pas les informations diffusées par haut-parleur.

Nous proposons donc de placer la *multimodalité* au centre de notre système : les dispositifs pourront utiliser différentes modalités en sortie. De plus, un dispositif donné ne fournira des informations à un utilisateur qu'à la condition expresse que la modalité de sortie du premier soit compatible avec les modalités d'entrée du second. De cette façon, on éviterait par exemple que l'information destinée à un non-voyant ne soit véhiculée par un moniteur vidéo. Notons que nous ne nous intéressons qu'aux interactions en *sortie*, et pas du tout aux entrées : il s'agit pour nous uniquement de *fournir* de l'information à des utilisateurs.

## 2. Revue de l'existant

### 2.1. Applications contextuelles

Plusieurs systèmes ont déjà été proposés pour fournir des informations contextuelles à des utilisateurs mobiles lors de leurs déplacements. Par exemple, Cool-Town (Kindberg *et al.*, 2001) montre des pages web aux utilisateurs en fonction de leur position. Généralement, ces informations contextuelles sont fournies aux utilisateurs par l'intermédiaire de petits dispositifs portables : par exemple, le Cyberguide (Long *et al.*, 1996), un guide touristique pour les musées, utilisait le Newton d'Apple. Il s'agit le plus souvent de fournir aux utilisateurs des informations sur leur environnement, par exemple, « où se trouve la pizzeria la plus proche ? » (Hull *et al.*, 1997).

Cependant, on peut également utiliser des systèmes publics d'affichage. Par exemple, le Gossip Wall (Streitz *et al.*, 2003) leur assigne plusieurs rôles : fournir des informations d'intérêt général quand personne n'est à proximité, et fournir des informations plus personnelles lorsqu'un utilisateur engage une interaction explicite, ce qui peut soulever des problèmes de respect de la vie privée (Vogel *et al.*, 2004).

### 2.2. Multimodalité

Dans le domaine de la multimodalité, nous utilisons le vocabulaire de Teil *et al.* (2000) : un *mode* correspond à la nature d'un moyen de communication, donc à l'un des sens humains : mode visuel, mode auditif, mode tactile, etc. Une modalité est une forme concrète particulière d'un mode. Par exemple, en ce qui concerne le mode auditif, on peut citer les modalités suivantes : parole, bruit, chant, etc.

Les modalités sont organisées sous forme d'une taxonomie (voir figure 1). Chaque modalité peut posséder des attributs. Par exemple, la modalité *texte visuel* possède les attributs *taille* et *couleur*, qui permettent de caractériser une déclinaison particulière de cette modalité. Une telle déclinaison est appelée *instanciation*.

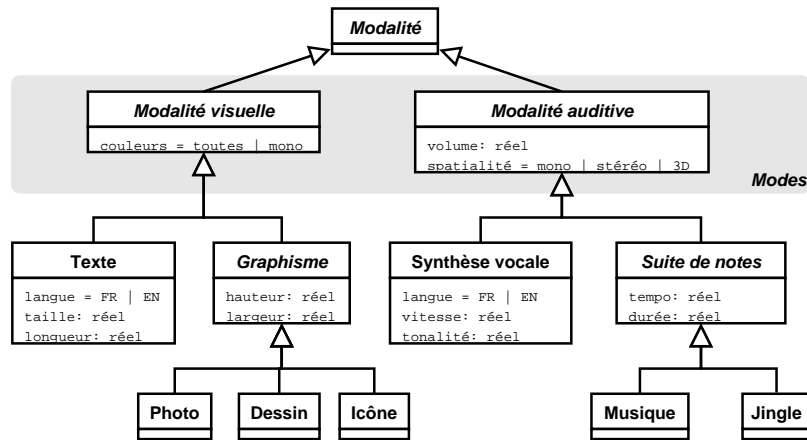
## 3. Le modèle KUP

### 3.1. Modèles d'architecture logicielle classiques en IHM

Depuis 30 ans, plusieurs modèles d'architecture logicielle pour les IHM ont été présentés. Parmi eux, on peut citer MVC pour modèle-vue-contrôleur (Krasner *et al.*, 1988), PAC pour présentation-abstraction-contrôle (Coutaz, 1987) ou ARCH (Bass *et al.*, 1992). Ces architectures mettent en avant au moins deux composantes :

- le *modèle* des informations à présenter. Il représente l'abstraction du problème à traiter, sous forme de données métier. Il est directement lié au *noyau fonctionnel* d'une application donnée (nom donné par ARCH ; PAC l'appelle *abstraction*). Nous noterons **K** cette composante, car c'est elle qui fournit les connaissances (*knowledge*) ;

– une implémentation concrète, voire physique, de l'*interface* avec l'utilisateur et des interactions qui lui sont associées (*vue-contrôleur* pour MVC, *présentation* pour PAC). Nous noterons donc **P** cette entité, car nous nous intéressons uniquement aux interactions en sortie, donc à la *présentation* d'informations.



**Figure 1.** Taxonomie partielle des modalités (diagramme de classes UML)

### 3.2. Sources de connaissance, utilisateurs, dispositifs de présentation

Dans le cadre de la conception de notre système mobile et opportuniste, il nous semble judicieux de séparer deux types d'actions, deux phases différentes :

- 1) la *fourniture* d'une information par le noyau fonctionnel à l'utilisateur ;
- 2) la *présentation* de cette information à destination de l'utilisateur.

De cette façon, le système peut *récolter* de façon opportuniste des informations au fur et à mesure de leur découverte, et les mémoriser, *même s'il n'y a pas de dispositif de présentation disponible à ce moment-là*. La présentation se fait plus tard, de façon opportuniste elle aussi, lorsque l'utilisateur se trouve à proximité d'un dispositif de présentation. La décorrélation entre les deux phases est la condition du fonctionnement doublement opportuniste du système.

Pour obtenir cette décorrélation, il ne faut pas que le noyau fonctionnel soit directement relié au dispositif de présentation : il doit exister un intermédiaire entre eux. Sinon, la fourniture et la présentation de l'information seraient forcément liées. Nous introduisons donc une troisième entité, **U**, correspondant à l'utilisateur. Cette entité sera située au centre du modèle afin de permettre le découplage entre la fourniture d'informations par des entités **K** (sources de connaissances), et leur présentation par des entités **P** (dispositifs de présentation). Le nom du modèle est donc *KUP*.

### 3.3. Comparaison par rapport aux modèles existants

Le modèle KUP est doublement original par rapport aux modèles existants :

- il prévoit une représentation logicielle active de l'utilisateur (U), alors qu'elle est absente ou réduite dans les modèles classiques. Cette représentation logicielle va bien au-delà de la simple caractérisation des utilisateurs par un profil ou des préférences ;
- cette entité logicielle représentant l'utilisateur se trouve au centre du modèle, et confère ainsi à l'utilisateur une place prépondérante : toutes les communications au sein du modèle vont désormais transiter par cette entité logicielle *utilisateur*.

Ainsi, le modèle KUP se distingue clairement des modèles classiques (ARCH, Seeheim, PAC, MVC, etc.) du fait que dans ces derniers l'utilisateur est toujours à l'extérieur du système ; il n'y est jamais représenté explicitement, en tant qu'entité active. Dans le modèle KUP, en revanche, l'entité utilisateur U est l'entité centrale.

### 3.4. Notion de proximité

De façon à construire un système opportuniste basé sur les rencontres entre entités, il faut définir formellement les notions de *proximité* correspondantes.

#### 3.4.1. Espace, positions

Les entités physiques (dispositifs de présentation, utilisateurs) sont *localisées* dans un espace. Ce dernier est noté  $\mathcal{E}$  ; il s'agit d'un ensemble de points. Il est possible que certaines dimensions de l'espace  $\mathcal{E}$  correspondent à l'*orientation* des entités.

#### 3.4.2. Espace perceptuel

Informellement, nous souhaitons définir l'*espace perceptuel* d'une entité physique  $e$  comme étant l'ensemble des points de l'espace où  $e$  peut percevoir une autre entité qui s'y trouve. Par exemple, pour une entité utilisateur, l'espace perceptuel pourrait correspondre à son champ de vision. Cependant, cette définition est trop restrictive :

1) les différents sens d'un utilisateur possèdent des caractéristiques de perception différentes. Par exemple, le champ de vision d'un être humain ne correspond pas à sa zone de perception auditive. Ainsi, un écran situé à 2 m derrière un utilisateur ne sera pas perçu par ce dernier, tandis qu'un dispositif de synthèse vocale situé au même endroit le sera ;

2) la perception dépend des *attributs des modalités*. Par exemple, une sonnerie de téléphone mobile émise à cinquante mètres ne sera pas perçue par un être humain, tandis que le son d'une sirène le sera sans problème.

En conséquence, la définition informelle de la notion d'espace perceptuel introduite ci-avant est trop limitée. Elle doit être complétée pour tenir compte des modalités et des instanciations de ces dernières. Pour ce faire, nous introduisons une notion supplémentaire : l'*espace multimodalisé*, ou en abrégé m-espace. Un m-espace est le

produit cartésien de l'espace  $\mathcal{E}$  par l'ensemble des instanciations des modalités utilisables. Par exemple, supposons que les modalités utilisables par une entité sont :

- *sonnerie*, avec un attribut *volume* variant continûment entre 0 et 100 ;
- *texte*, avec un attribut *taille* variant continûment entre 10 et 60 points et un attribut *couleur* pouvant prendre les trois valeurs discrètes *rouge*, *vert* et *bleu*.

Dans ce cas, un élément du m-espace pourrait être le point de coordonnées  $46^{\circ}23'32''$  N,  $1^{\circ}02'56''$  E, avec un texte de taille 23 et de couleur verte.

Cette définition permet de rendre compte des caractéristiques de la perception des utilisateurs humains. Cependant, quel sens peut-on donner à la *perception* d'un dispositif de présentation ? Un dispositif peut être équipé de capteurs multimodaux en entrée, par exemple un microphone, une caméra, un clavier, etc. Dans ce cas, la perception est définie par rapport aux caractéristiques techniques de ces capteurs.

Ces capteurs peuvent utiliser les mêmes modalités que les êtres humains, mais ce n'est pas forcément le cas. Par exemple, on peut supposer qu'un dispositif de présentation dispose d'un équipement électronique lui permettant de détecter les utilisateurs situés à proximité, comme un détecteur d'étiquettes RFID<sup>1</sup>. On peut dire dans ce cas que la méthode de détection représente une *modalité* supplémentaire. Ce n'est pas une modalité au sens classique du terme, car les modalités sont habituellement définies par rapport aux sens des êtres humains (voir section 3.2). Cependant, pour un dispositif électronique, ces moyens de détection jouent exactement le même rôle que les modalités pour les êtres humains. Nous les classifions donc en tant que *modalités*. Par exemple, on peut envisager la modalité *détection par RFID*, qui posséderait un attribut *puissance nécessaire pour effectuer la détection*.

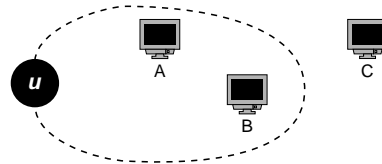
Il est maintenant possible de définir formellement l'*espace perceptuel* d'une entité physique. Il s'agit d'un sous-ensemble d'un m-espace  $\mathcal{M}$ , qui correspond à l'ensemble des *points* perceptibles par l'entité, la perception étant définie comme indiqué ci-avant. Si une entité se déplace, son espace perceptuel sera modifié : dans la plupart des cas, il va *suivre* son entité de façon naturelle. On note  $\mathcal{EP}(e, x)$  l'espace perceptuel d'une entité  $e$  située en  $x \in \mathcal{E}$ . On a  $\mathcal{EP}(e, x) \subset \mathcal{M}$

Par exemple, si on effectue une projection de l'espace perceptuel d'un utilisateur  $u$ , selon la modalité visuelle et à attributs constants (par exemple, à taille de caractères donnée), on obtient un *champ visuel*. Un tel champ visuel est représenté sur la figure 2 : les informations affichées sur les écrans A et B sont perçues par l'utilisateur  $u$ , mais pas celles affichées par C.

Nous avons vu ci-avant que la *perception* d'une entité physique peut être décrite dans un m-espace. Réciproquement, une telle entité peut être vue comme une *source* de contenu multimodal (au sens large) localisée dans l'espace  $\mathcal{E}$ . Au final, une entité est donc localisée dans un m-espace  $\mathcal{M}$ . Comme une entité peut éventuellement dif-

1. Système d'identification par radiofréquences, constitué d'étiquettes passives qu'il est possible d'identifier grâce à un détecteur approprié.

fuser du contenu selon plusieurs modalités, sa localisation n'est pas forcément réduite à un seul point du m-espace ; au contraire, il s'agit d'un *sous-ensemble* du m-espace.



**Figure 2.** Cas particulier d'espace perceptuel : champ visuel d'un utilisateur  $u$

Formellement, la localisation d'une entité  $e$  est notée  $\ell(e)$ . On a  $\ell(e) \subset \mathcal{M}$ . La localisation d'une entité  $e$  est donc un ensemble de couples  $(x_i, m_i)$ , où  $x_i$  est une position géographique (un élément de  $\mathcal{E}$ ), et  $m_i$  une modalité.

### 3.4.3. Espace de rayonnement

Nous disposons maintenant de toutes les notions nécessaires à la définition de la notion *réciproque* de l'espace perceptuel. L'espace perceptuel caractérise les perceptions d'une entité, donc autrement dit ses *entrées*, c'est-à-dire son comportement en tant que *récepteur multimodal*. Cependant, comme nous venons de le voir, une entité peut également être un *émetteur multimodal*, c'est-à-dire présenter des caractéristiques multimodales en sortie. De même que les caractéristiques en entrée sont décrites par l'espace perceptuel, les caractéristiques en sortie sont décrites par ce que nous appelons l'*espace de rayonnement*.

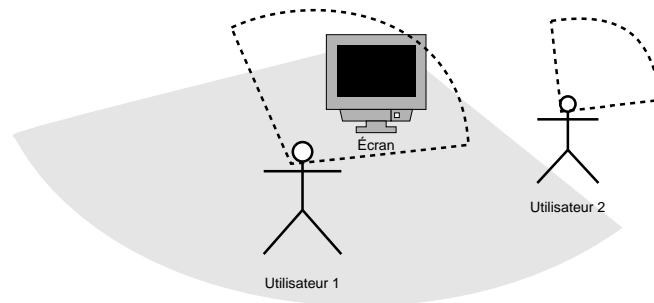
Formellement, on définit l'*espace de rayonnement* d'une entité  $e$  vis-à-vis d'une entité  $d$  comme étant l'ensemble des points  $x$  de l'espace  $\mathcal{E}$  desquels  $d$  peut percevoir  $e$ , c'est-à-dire pour lesquels  $e$  appartient à l'espace perceptuel de  $d$  situé en  $x$  :

$$\mathcal{ER}(e|d) = \{x \in \mathcal{E} \mid \ell(e) \cap \mathcal{EP}(d, x) \neq \emptyset\}$$

L'espace de rayonnement est défini *par rapport à une entité* réceptrice : les espaces perceptuels étant potentiellement différents pour chaque entité, la perception d'un *émetteur* en un point donné de l'espace sera ou non possible selon l'entité considérée. Il est donc impossible de définir un espace de rayonnement dans l'absolu. La figure 3 traite le cas simple où la seule modalité considérée est le texte visuel.

### 3.4.4. Récapitulatif sur les espaces perceptuels et de rayonnement

L'espace perceptuel d'une entité caractérise ses *entrées*, car il détermine la façon dont elle va *percevoir* les autres entités. De même, on peut dire que l'espace de rayonnement caractérise ses *sorties*, car il détermine la façon dont les autres entités vont percevoir l'entité en question. L'espace de rayonnement d'une entité dépend de sa *localisation* au sein d'un m-espace, qui décrit aussi bien sa position géographique que les modalités selon lesquelles elle sait communiquer avec les autres entités.



**Figure 3.** *L'utilisateur 1 peut percevoir l'écran (car il est dans son champ visuel, en pointillés), mais ce n'est pas le cas de l'utilisateur 2. De fait, seul le premier se trouve dans l'espace de rayonnement de l'écran (zone grisée)*

Ces définitions ont introduit une notion de *proximité*. La proximité dépend certes de la position géographique (dans l'espace  $\mathcal{E}$ , ce qui peut inclure non seulement les coordonnées de l'endroit où se trouve une entité, mais également son orientation), mais également des capacités multimodales des entités (localisation dans un m-espace). Ainsi, même si dans la suite nous utilisons parfois (un peu abusivement) les termes *proche* et *proximité* sans plus de qualificatifs, ils doivent être compris au sens de la proximité sensorielle décrite ci-avant.

Par exemple, si un utilisateur aveugle se trouve juste devant un écran, on ne considérera pas que ces deux entités sont proches l'une de l'autre. De même, si un utilisateur voyant se trouve devant un écran, mais lui tourne manifestement le dos, les deux entités ne seront pas non plus considérées comme étant proches.

#### 3.4.5. Cas des sources d'information

Dans ce qui précède, nous avons traité de la localisation et de la perception des entités *physiques* (U et P). Qu'en est-il des entités *abstraites* que sont les sources d'information (K) ? Nous proposons tout simplement d'*étendre* toutes les définitions ci-avant aux sources d'information.

Ainsi, nous considérons que les sources d'information sont localisées, et qu'elles possèdent des espaces perceptuels et de rayonnement. Cependant, nous ajoutons quelques spécificités à ces définitions :

- l'espace de rayonnement d'une source d'information est défini comme étant la zone géographique dans laquelle elle doit diffuser des informations. Autrement dit, son espace de rayonnement vis-à-vis d'un utilisateur  $u$  est la partie de l'espace au sein de laquelle les informations qu'elle détient sont réputées pertinentes pour  $u$  ;
- l'espace perceptuel d'une source d'information est défini comme étant égal à son espace de rayonnement. En effet, une source d'information diffuse des informations à des entités (utilisateurs dans la plupart des cas) qui se situent dans son espace de



rayonnement. La *détection* de ces entités doit donc se produire dans exactement le même périmètre que la diffusion d'informations à leur intention ;

– il paraît raisonnable que l'espace de rayonnement d'une source d'information  $k$  ainsi définie soit le même vis-à-vis de n'importe quelle entité  $u$ . Au lieu de noter son espace de rayonnement  $\mathcal{ER}(i|u)$ , on note alors  $\mathcal{ER}(i)$ . On a toujours  $\mathcal{ER}(i) \subset \mathcal{E}$ .

Notons bien qu'une source d'information (ou de connaissance) représente une *interface* (au sens protocolaire du terme) entre le noyau fonctionnel de l'application et les autres entités du système. *De l'extérieur*, tout se passe *comme si* une source d'information *était* un noyau fonctionnel ou une partie d'un noyau fonctionnel. En réalité toutefois, il est tout à fait possible que le noyau fonctionnel corresponde à un enchaînement complexe de traitements.

Pour une source de connaissances, la diffusion des informations ne fait pas appel à la notion de modalité : une source est seulement sensée diffuser des informations à l'intérieur d'une *zone géographique*. Or, comme pour toute autre entité, la définition de son espace perceptuel et de son espace de rayonnement fait appel à la notion de m-espace. Comment peut-on alors définir un m-espace pour une source d'informations, alors qu'il n'existe pas de notion pertinente de modalité ?

Nous proposons tout simplement d'introduire une unique *pseudomodalité*, notée  $\perp$ . Le m-espace correspondant  $\mathcal{M}$  sera alors défini par :  $\mathcal{M} = \mathcal{E} \times \{\perp\}$ . On définit de cette façon un véritable m-espace, sans *épaisseur* toutefois selon l'axe des modalités. Il est trivialement en bijection avec l'espace géographique  $\mathcal{E}$ .

La figure 4 récapitule le fonctionnement d'une source d'information, par rapport à notre objectif global de fourniture et de présentation opportuniste d'informations. Où qu'elles se trouvent dans le périmètre de l'espace de rayonnement d'une source d'information, les entités utilisateurs reçoivent des informations. En revanche, les dispositifs de présentation utilisés pour effectuer la présentation des informations peuvent varier, en fonction de la position des utilisateurs vis-à-vis desdits dispositifs.

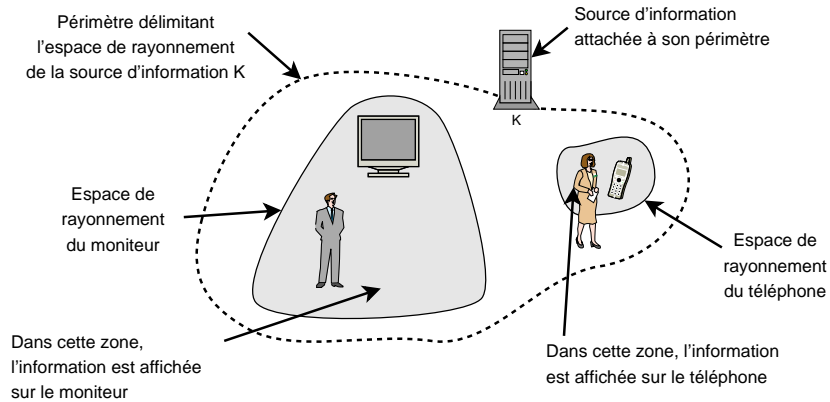
### 3.5. Unités sémantiques

#### 3.5.1. Définitions

Les informations émises par les sources d'information sont appelées *unités sémantiques*, ou en abrégé u.s. Une unité sémantique correspond à une information élémentaire, que l'on peut transmettre sur un réseau, et qui peut s'exprimer dans au moins une modalité. Par exemple, une unité sémantique peut porter l'information correspondant à la porte d'embarquement d'un passager dans un aéroport, aux menus d'un restaurant, ou encore à l'horaire du prochain train allant dans la direction souhaitée.

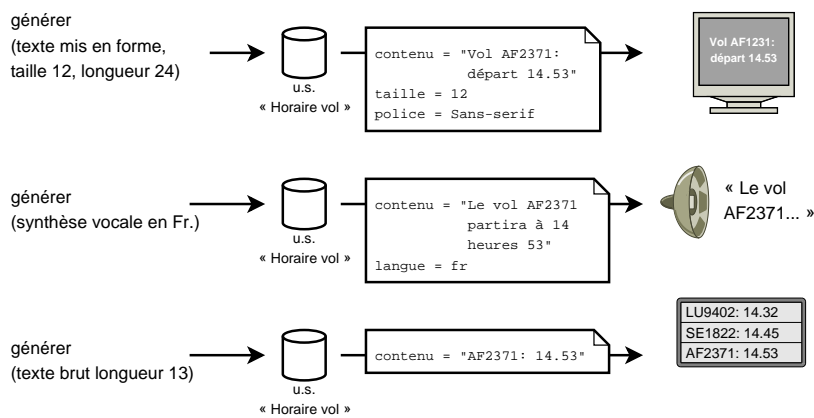
Les u.s. ont pour vocation d'être exprimées sur un dispositif de présentation, et selon une modalité donnée. Il est donc nécessaire de leur associer un contenu concret dans la modalité en question. Cependant, la génération *automatique* de contenu fait l'objet de recherches propres (Zock *et al.*, 2002, par exemple) : nous ne détaillons

donc pas ces processus. Pour nous, la génération de contenu concret est vue comme une *boîte noire* dont on spécifie l'entrée (modalité concrète instanciée), et dont on utilise la sortie (contenu concret).



**Figure 4.** Les utilisateurs reçoivent des informations dans la totalité de l'espace de rayonnement de la source d'information K, mais les dispositifs de présentation peuvent varier

La figure 5 résume le processus de génération de contenu concret par une unité sémantique donnée, selon différentes modalités.



**Figure 5.** Génération de contenu concret. Une même u.s. génère des contenus concrets différents pour chaque modalité instanciée

### 3.5.2. Utilisation des unités sémantiques

Les unités sémantiques entrent en jeu dans le système au cours de deux phases :

- dans un premier temps, une source d'information (K) fournit une u.s. à un utilisateur (U) lorsque ce dernier entre dans son espace de rayonnement. Nous ne traitons pas de la détermination des u.s. pertinentes en fonction des besoins des utilisateurs, qui est un problème à part (modélisation des buts). Nous supposons donc qu'une source d'information sait quelle u.s. fournir à quel utilisateur, sans donner plus de détails ;
- dans un deuxième temps, cette u.s. est présentée à l'utilisateur (U) par un dispositif de présentation (P), lorsque l'utilisateur pénètre dans l'espace de rayonnement de ce dernier. Il est alors nécessaire de choisir et d'instancier une modalité : cette opération est détaillée au cours de la section 4.

## 3.6. Architecture à agents

### 3.6.1. Introduction

Voyons maintenant comment le modèle abstrait KUP peut être concrétisé sous la forme d'une architecture informatique. Bien entendu, nous pourrions envisager de construire une solution basée sur une architecture *centralisée*. Ainsi, toutes les entités correspondraient à des processus (ou plus généralement à des segments de code) s'exécutant sur un système informatique central. Cependant, nous pensons qu'une telle méthode présente un certain nombre d'inconvénients, principalement :

- sa *fragilité* : une panne des serveurs centraux paralyse le système en entier ;
- et sa *rigidité* : il est impossible de déplacer les dispositifs de présentation à volonté, du moins sans travail de reconfiguration.

A l'inverse, nous souhaitons donner au personnel des lieux dans lesquels notre système sera déployé la possibilité de déplacer les dispositifs de présentation, d'en apporter de nouveaux si un événement particulier survient, etc., tout ceci sans être obligé de configurer quoi que ce soit. Les dispositifs de présentation doivent être capables de s'adapter aux changements, sans intervention humaine.

Nous proposons donc de bâtir une architecture logicielle décentralisée, basée sur la notion d'*agent*. Ainsi, chacun des trois types d'entités évoqués précédemment correspondra à un type d'agents :

- les *agents-utilisateurs* (U) : ils correspondent aux utilisateurs ; ils connaissent leurs préférences ;
- les *agents-informateurs* (K) : ils correspondent aux sources d'information ; ils fournissent des informations aux agents-utilisateurs ;
- les *agents présentateurs* (P) : ils correspondent aux dispositifs de présentation ; ils sont capables d'évaluer le coût de la présentation d'une u.s. sur un dispositif, et d'effectuer cette présentation.

Ainsi, le monde des agents constitue le *miroir* du monde réel, du moins en ce qui concerne nos trois types d'entités d'intérêt.

### 3.6.2. *Caractéristiques des agents*

#### 3.6.2.1. Communication

Nous supposons que tous les agents peuvent communiquer entre eux. Cette supposition nous semble réaliste : les communications peuvent passer par des réseaux sans fils de type WiFi, qui sont maintenant monnaie courante. De plus, de tels réseaux sont largement déployés dans certains bâtiments. Lorsque des réseaux de type WiFi ne sont pas disponibles, il est possible d'avoir recours à des réseaux de téléphonie mobile, dont les zones de couverture sont beaucoup plus étendues.

#### 3.6.2.2. Perception, proximité

Des notions de perception et de rayonnement ont été définies pour les entités physiques et abstraites (voir section 3.4). Ces relations sont alors *projetées* dans le monde des agents. Ainsi, si par exemple une entité utilisateur *a* perçoit une entité dispositif de présentation *b*, alors la même relation existera entre les agents associés : nous dirons par abus de langage que l'agent *a* perçoit l'agent *b*. De façon homonyme, nous dirons également que l'agent *a* est *proche* de l'agent *b*.

Mais comment le système informatique accédera-t-il à cette notion de perception ? Par exemple, comment pourra-t-on savoir en pratique que l'utilisateur *a* perçoit le dispositif *b* ? Comme nous l'avons déjà évoqué, il faudra effectuer des *mesures*. Par exemple, la proximité peut être détectée par des lecteurs d'étiquettes RFID, ou encore des systèmes Bluetooth ou à infrarouges (Harter *et al.*, 1994).

#### 3.6.2.3. Réactivité

Les agents sont *réactifs* : ils restent en sommeil la plupart du temps, et réagissent lorsque des événements particuliers se produisent. En pratique, un agent donné *a* peut réagir à trois sortes d'événements :

- 1) un autre agent *b* vient de s'approcher de *a* ;
- 2) un agent *b*, auparavant proche de *a*, vient de s'éloigner ;
- 3) *a* vient de recevoir un message par le réseau, en provenance d'un agent *c* quelconque, qui n'est pas nécessairement proche de *a*.

Ainsi, si les agents se trouvaient seuls dans le système informatique, *il ne se passerait jamais rien*. Les agents ont des comportements réactifs lorsque se déplacent les entités qu'ils représentent. Cela signifie que toute la proactivité du système est assurée par les utilisateurs humains : ce sont ces derniers qui vont se déplacer (ou déplacer des dispositifs de présentation !) et de là, déclencher des cascades de réactions dans le système. Au final, on ne construit pas réellement un *monde d'agents*, mais plutôt une *agentification du monde réel*.

On rejoint ainsi la vision de l'intelligence ambiante (Ducatel *et al.*, 2001), dans laquelle des systèmes informatiques sont *discrètement* à l'écoute des actions des êtres humains, et peuvent ainsi intervenir de façon opportune et non intrusive.

A la lumière de cette organisation sous forme d'agents, il est possible de détailler un peu plus les étapes de la procédure de *fourniture et présentation* d'informations. Les étapes sont alors les suivantes :

- 1) un agent utilisateur entre dans l'espace de rayonnement d'un agent informateur ;
- 2) *éventuellement*, et à un moment donné, non spécifié, l'agent informateur envoie une unité sémantique à l'agent utilisateur ;
- 3) celui-ci va chercher à effectuer la présentation de l'u.s. pour l'utilisateur (réel, humain, physique). Il prend donc contact avec chaque agent présentateur dont le dispositif se trouve dans l'espace perceptuel de l'utilisateur, et lui demande d'évaluer la présentation de l'u.s. Si possible, il choisit l'un des agents présentateurs et lui confie la présentation de l'u.s. Nous allons maintenant détailler cette dernière étape.

#### 4. Algorithme pour le choix de modalité

##### 4.1. Modélisation de l'utilisateur et de l'environnement

Souchon *et al.* (2002) définissent le contexte comme étant *l'environnement complet dans lequel une tâche est effectuée*. Cet environnement est représenté par trois modèles : modèle de l'utilisateur, modèle de la plate-forme (*i.e.* des équipements informatiques), et modèle de l'environnement (conditions physiques<sup>2</sup>, lieu, environnement social et organisationnel).

Le profil sensoriel de l'utilisateur caractérise les capacités ou incapacités physiques de l'utilisateur (vue, ouïe), ainsi que ses compétences (langues parlées et lues). Ces caractéristiques permettent ou interdisent l'utilisation de certaines modalités. Ce profil est relativement peu dynamique. Il peut évoluer au cours du temps (vieillesse des organes, maladies ou au contraire guérisons, apprentissage de nouvelles langues, etc.), mais ces évolutions sont relativement lentes. Au niveau de l'échelle de temps de l'utilisation de notre système, on peut le considérer comme étant statique.

Ce profil sensoriel doit également permettre à l'utilisateur d'exprimer ses préférences : lorsqu'il est possible d'utiliser plusieurs modalités pour présenter une u.s., il est normal d'utiliser celle que l'utilisateur préfère. Ces préférences peuvent concerner un mode, une modalité, ou même un attribut. Par exemple, on peut imaginer qu'un locuteur francophone de naissance, mais qui comprend l'Anglais, préfère écouter des messages en Français plutôt qu'en Anglais.

De la même façon qu'un profil sensoriel est associé à chaque utilisateur pour modéliser ses capacités et préférences d'interaction, les caractéristiques d'interaction des

---

2. Exemples : luminosité, pression, etc.

dispositifs de présentation doivent elles aussi être précisées. Par exemple, il est important de formaliser le fait qu'un écran sait utiliser la modalité textuelle, avec une taille de texte pouvant prendre certaines valeurs ; qu'un haut-parleur sait utiliser la modalité auditive avec certains volumes sonores, etc. Ceci permet donc d'établir un profil pour chaque dispositif de présentation, chargé de recenser ses capacités multimodales en sortie.

Le profil d'un dispositif de présentation peut comprendre des caractéristiques statiques aussi bien que dynamiques :

- par exemple, un écran ne prend en charge que des modalités visuelles, un écran alphanumérique ne prend pas en charge les modalités graphiques : ce sont des caractéristiques *statiques*.

- par exemple, un téléphone dont la charge de la batterie passe en-dessous d'un seuil critique peut désactiver sa fonction vibreur : le fait que cette modalité soit utilisable ou non est donc une caractéristique *dynamique*.

L'environnement, tel que défini plus haut, nous intéresse dans la mesure où il peut perturber les interactions entre les dispositifs et les utilisateurs. Il s'agit donc de modéliser ces perturbations, de façon à les contourner. Par exemple, dans un environnement bruyant, un système de haut-parleur devra augmenter son volume sonore de façon à couvrir le bruit ambiant. Nous pourrions donc introduire un profil de l'environnement, qui contiendrait des contraintes supplémentaires introduites par ses perturbations.

Cependant, une telle organisation laisserait supposer qu'il existe quelque part une entité *environnement* chargée de maintenir à jour ce profil. Le profil d'environnement serait alors utilisé lors de chaque présentation d'information. Cela nous paraît relativement compliqué, et nous préférons proposer une solution plus simple : intégrer le profil de l'environnement aux caractéristiques dynamiques des profils des dispositifs de présentation. Par exemple, s'il y a du bruit, le profil d'un dispositif haut-parleur indiquera simplement que le niveau sonore doit être supérieur à une valeur donnée.

Cette solution présente l'avantage de la simplicité, et de plus, elle nous semble correspondre à une implémentation assez naturelle. En effet, il paraît raisonnable qu'un dispositif de présentation se charge lui-même de vérifier les conditions dans lesquelles ses messages pourront être diffusés. Par exemple, un haut-parleur disposera d'un microphone chargé de vérifier que le bruit ambiant n'est pas trop élevé ; un écran disposera d'un capteur de luminosité afin de s'assurer que le contraste par rapport à la lumière ambiante est bon, etc. Dans ces conditions, il est donc naturel que les contraintes dynamiques imposées par l'environnement soient représentées au sein du profil du *dispositif*.

Enfin, dans la définition d'une unité sémantique (voir section 3.5), nous avons indiqué qu'une telle u.s. est capable de produire un contenu concret dans un certain nombre de modalités. Il est donc nécessaire de pouvoir indiquer quelles sont ces modalités *utilisables* par les unités sémantiques. Pour cette raison, nous décidons d'attribuer un profil à chaque unité sémantique. Ce profil est statique. Il indique simplement

quelles sont les modalités selon lesquelles l'u.s. sait générer un contenu concret. En résumé, il existe donc trois sortes de profils différents :

- profil d'un utilisateur : il décrit les capacités et préférences d'un utilisateur donné. Il possède une composante dynamique, mais qui varie lentement ;
- profil d'un dispositif de présentation : il décrit les capacités du dispositif, en relation avec les contraintes extérieures imposées par l'environnement. Les pannes de dispositifs ainsi que la variation des conditions de l'environnement imposent à ce profil une grande dynamique ;
- profil d'une unité sémantique : ce profil statique décrit les modalités selon lesquelles l'u.s. sait générer un contenu concret.

Lors de la présentation d'une u.s. donnée, pour un utilisateur donné, sur un dispositif de présentation donné, il est donc nécessaire de *mettre en relation* ces trois profils. Les modalités et valeurs d'attributs utilisables sont celles qui sont *communes* à ces trois profils. Autrement dit, les solutions se trouvent à l'*intersection* de ces profils.

Les sections suivantes se fixent donc comme objectif de définir formellement la notion de *profil*, ainsi que la notion d'*intersection de profils*. Une fois ces notions posées, il est possible de donner l'algorithme proprement dit.

#### 4.2. Arbres taxonomiques

Avant de passer à l'expression des profils, nous devons définir la notion préliminaire d'*arbre taxonomique*. Un arbre taxonomique permet de définir formellement une taxonomie des modalités telle que celle de la figure 1.

Un attribut est un couple  $\langle n, E \rangle$  où  $n$  est une chaîne de caractères (on note  $n \in \mathbb{S}$ ) et  $E$  un ensemble quelconque. On dit que cet attribut, de nom  $n$ , est à valeurs sur  $E$ . On dit aussi que  $E$  est le *domaine* de l'attribut. Nous avons vu précédemment que l'on envisage généralement deux grands cas : soit  $E$  est un ensemble fini, soit  $E$  est un intervalle de  $\mathbb{R}$ . Cependant, il est bien entendu possible d'envisager d'autres situations. On note  $\mathcal{A}$  l'ensemble des attributs.

On appelle *attribut valué* associé à l'attribut  $a = \langle n, E \rangle$  un couple  $\langle n, f \rangle$  où  $f$  est une fonction de  $E$  dans  $[0, 1]$ . L'ensemble des attributs valués est noté  $\mathcal{AV}$ .

On note  $\mathcal{AT}$  l'ensemble des *arbres taxonomiques*. Les éléments de  $\mathcal{AT}$  sont définis comme suit :

Si  $S = \{s_1, s_2, \dots, s_m\} \subset \mathcal{AT}$  (i.e. si les  $s_i$  sont des arbres taxonomiques, étant entendu que l'ensemble  $S$  peut être vide), si  $n \in \mathbb{S}$ , si  $A = \{a_1, a_2, \dots, a_p\} \subset \mathcal{A}$  (i.e. si les  $a_i$  sont des attributs) alors  $\langle n, A, F \rangle$  est un arbre taxonomique. La racine de cet arbre taxonomique a pour nom  $n$ . Ses attributs sont les  $a_i$  ; ses arbres fils sont les  $s_i$ . Si  $S = \emptyset$  on dit que l'on a affaire à une *feuille*.

### 4.3. Arbres de pondération

#### 4.3.1. Introduction

Passons maintenant plus directement à l'expression des profils, et introduisons la notion d'*arbre de pondération*. Le principe d'un arbre de pondération est simple : il s'agit d'ajouter des pondérations à un arbre taxonomique, qui permettront d'exprimer les capacités, préférences et contraintes des utilisateurs, des dispositifs et des u.s.

Une pondération est un nombre réel compris entre 0 (inclus) et 1 (inclus également). Une pondération peut être située à deux endroits différents :

- sur un nœud : la pondération s'applique alors au sous-arbre ayant ce nœud pour racine. Une pondération à 1 signifie que les modalités du sous-arbre sont acceptées, voire souhaitées, tandis qu'une pondération à 0 signifie que les modalités correspondantes sont refusées, ou non prises en charge. Les valeurs intermédiaires permettent de nuancer ces deux extrêmes, et ainsi d'exprimer des *niveaux de préférence* ;

- sur un attribut : on donne alors une fonction définie sur l'ensemble des valeurs possibles de cet attribut, et à valeurs dans  $[0; 1]$ . Cette fonction indique la pondération accordée à chaque valeur possible de l'attribut. La signification des pondérations est la même que précédemment.

Un *profil* (tel qu'évoqué précédemment) est défini comme étant un arbre de pondération dont la racine correspond à la racine de la taxonomie des modalités. La figure 6 donne un exemple de profil partiel. Il pourrait correspondre à un utilisateur malvoyant, qui préférerait donc largement les modes auditifs aux modes visuels : les pondérations correspondantes sont indiquées en blanc sur fond noir, à proximité des nœuds. Les fonctions de pondération sont indiquées pour quelques attributs.

#### 4.3.2. Définition formelle

Soit un arbre taxonomique donné  $T$ . On note  $\mathcal{AP}_T$  l'ensemble de des arbres de pondération associés à  $T$ . Les éléments de  $\mathcal{AP}_T$  sont du type  $\varphi(T)$  où  $\varphi$  est une *pondératrice* sur  $T$ , définie de la façon suivante.

Une *pondératrice*  $\varphi$  est une fonction définie sur l'ensemble des sous-arbres de  $T$  (qui est un sous-ensemble de  $\mathcal{AT}$ ), à valeurs dans  $\mathbb{S} \times [0, 1] \times \wp(\mathcal{AV}) \times \wp(\mathcal{AP}_T)$ . Elle est de la forme suivante :

$$\varphi(\langle n, \{ \langle n_1, E_1 \rangle, \dots, \langle n_m, E_m \rangle \}, \{ s_1, \dots, s_p \} \rangle) = \langle n, x, \{ \langle n_1, f_1 \rangle, \dots, \langle n_m, f_m \rangle \}, \{ \varphi(s_1), \dots, \varphi(s_p) \} \rangle$$

où  $x \in [0, 1]$ , et pour tout  $i$  compris entre 1 et  $m$ ,  $s_i \in \mathcal{F}(E_i, [0, 1])$

En d'autres termes, une pondératrice  $\varphi$  associe une pondération  $x \in [0, 1]$  à chaque nœud de  $T$ , et une fonction à valeurs dans  $[0, 1]$  (appelée *fonction de pondération*) à chaque attribut des nœuds de  $T$ .



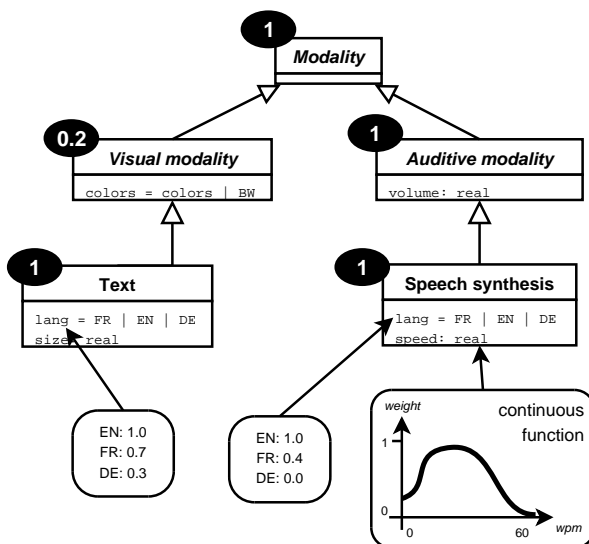


Figure 6. Exemple de profil partiel (arbre de pondération)

Plus formellement, on dit que  $f$  est une *fonction de pondération sur  $E$*  si  $f$  est une fonction définie sur  $E$ , et à valeurs dans  $[0; 1]$ . On note  $\mathcal{FP}_E$  l'ensemble des fonctions de pondération sur  $E$ .

#### 4.3.3. Intersection d'arbres de pondération

Nous disposons maintenant du formalisme nécessaire pour définir l'*intersection* des arbres de pondération (et donc des profils), telle qu'elle a été évoquée à la fin de la section 4.1. Nous proposons tout simplement d'effectuer le produit des poids et des fonctions de pondération qui se correspondent. De cette façon, les pondérations défavorables se retrouveront sur l'intersection.

Formellement, soit  $T$  une taxonomie fixée, et  $A_1$  et  $A_2$  deux éléments de  $\mathcal{AP}_T$ . Il existe donc une pondératrice  $\varphi_1$  telle que  $A_1 = \varphi_1(T)$ , et une pondératrice  $\varphi_2$  telle que  $A_2 = \varphi_2(T)$ .

On définit alors une nouvelle pondératrice  $\psi$  sur l'ensemble des sous-arbres (au sens large) de  $T$  de la façon suivante :

Soit un nœud de  $T$  :  $N = \langle n, \{\langle n_1, E_1 \rangle, \dots, \langle n_m, E_m \rangle\}, \{s_1, \dots, s_p\} \rangle$ ,  
 sa pondération dans  $A_1$  :  $\varphi_1(N) = \langle n, x_1, \{\langle n_1, f_{1,1} \rangle, \dots, \langle n_m, f_{m,1} \rangle\}, S_1 \rangle$ ,  
 et sa pondération dans  $A_2$  :  $\varphi_2(N) = \langle n, x_2, \{\langle n_1, f_{1,2} \rangle, \dots, \langle n_m, f_{m,2} \rangle\}, S_2 \rangle$ .

Alors on pose :

$$\psi(N) = \langle n, x_1 \cdot x_2, \{\langle n_1, f_{1,1} \cdot f_{1,2} \rangle, \dots, \langle n_m, f_{m,1} \cdot f_{m,2} \rangle\}, \{\psi(s_1), \dots, \psi(s_p)\} \rangle$$

$B = \psi(T)$  est donc un arbre de pondération sur  $T$ . On dit alors que  $c$ 'est l'*intersection* des arbres de pondération  $A_1$  et  $A_2$ . On note  $B = A_1 \sqcap A_2$ .

#### 4.4. Evaluation d'un arbre de pondération

Voici où nous en sommes dans le cheminement global :

1) nous avons vu que la prise en compte des diverses contraintes (dispositifs de présentation, environnement, utilisateur, unités sémantiques) conduisait à l'obtention de trois arbres de pondération ;

2) il est alors possible de réaliser l'intersection de ces trois arbres (au sens de la loi  $\sqcap$ ), afin d'obtenir un nouvel arbre de pondération : nous venons de voir les règles de calcul correspondantes. Cet arbre représente les modalités et les valeurs d'attributs compatibles avec chacun des intervenants ;

3) il reste alors à utiliser cet arbre afin de déterminer *in fine* quelle est la modalité à utiliser. La notion d'*évaluation* permet de répondre à cette question.

Les modalités concrètes sont les feuilles des arbres (de taxonomie aussi bien que de pondération). On va donc donner une évaluation (un poids) à chaque feuille de l'arbre de pondération, et choisir la feuille qui aura le meilleur poids.

Schématiquement, pour chaque feuille on va suivre le chemin qui mène de la racine de l'arbre à cette feuille, et on va calculer un poids qui dépendra des pondérations de nœuds et d'attributs rencontrés sur le chemin. On veut :

- que les feuilles situées sur un sous-arbre qui possède un nœud parent dont la pondération est 0 aient un poids de 0 (caractère absorbant de 0), et qu'une pondération à 1 n'influe pas négativement le résultat (caractère neutre de 1) : on a donc envie de faire le *produit* des pondérations le long des chemins de l'arbre ;

- que la longueur du chemin n'influe pas sur le calcul : on peut alors songer à faire non pas le produit, mais la moyenne géométrique des pondérations. En effet, les produits ayant plus de facteurs sur les chemins les plus longs, le poids correspondant aurait d'autant plus de chance d'être faible. On constaterait donc *statistiquement* un déséquilibre de poids entre les chemins les plus longs et les chemins les plus courts ;

- que les attributs entrent en jeu de façon semblable : on songe donc à faire la moyenne géométrique des pondérations des attributs ;

- que toutes les pondérations d'attributs réunies aient autant de poids que toutes les pondérations de nœuds réunies : on fera donc le produit des deux moyennes géométriques considérées.

Ainsi donc, si sur un chemin, et pour un choix donné d'attributs, on a :

- $p$  attributs valués, notés  $a_1, \dots, a_p$ , de fonctions de pondération  $f_1, \dots, f_p$  ;
- $q$  nœuds de pondérations  $x_1, \dots, x_q$ .

On donnera à ce chemin (et donc à la feuille correspondante) un poids du type :

$$w = \sqrt[p]{f_1 \times \dots \times f_p} \times \sqrt[q]{x_1 \times \dots \times x_q} = (f_1 \times \dots \times f_p)^{\frac{1}{p}} \times (x_1 \times \dots \times x_q)^{\frac{1}{q}}$$

Ceci est très schématique : s'il est effectivement possible d'effectuer le produit des pondérations des nœuds (ce sont des réels compris entre 0 et 1), il n'est en revanche pas possible d'effectuer aussi simplement le produit des fonctions de pondération des attributs. En effet, ces fonctions de pondération sont potentiellement définies sur des ensembles complètement différents. Leur produit reste à définir.

Soient donc deux fonctions de pondération  $f$  et  $g$  sur des ensembles respectifs  $F$  et  $G$  :  $f \in \mathcal{FP}_F$  et  $g \in \mathcal{FP}_G$ . On définit alors la fonction  $h$  suivante :

$$h : (y, z) \in F \times G \mapsto f(y) \cdot g(z) \in [0; 1]$$

On note  $h = f * g$ . On définit ainsi une loi  $*$  (*produit*) sur les fonctions de pondération.  $f * g$  est une fonction de pondération sur l'ensemble  $F \times G$ .

Ainsi, de façon précise, le poids donné à la feuille évoquée plus haut est une fonction de pondération ainsi définie :

$$w(y_1, y_2, \dots, y_p) = (x_1 \times x_2 \times \dots \times x_q)^{\frac{1}{q}} \cdot (f_1(y_1) \cdot f_2(y_2) \cdot \dots \cdot f_p(y_p))^{\frac{1}{p}}$$

Le poids  $w$  est donc une fonction sur  $E_1 \times E_2 \times \dots \times E_p$ . Ses valeurs sont dans l'intervalle  $[0; 1]$ . Plus le poids est élevé (*i.e.* proche de 1), plus l'instance de modalité correspondante est jugée *intéressante*.

La figure 7 résume les processus d'intersection et d'évaluation d'arbres de pondération. Pour des raisons de simplicité, on n'a pas représenté les attributs. En pratique, ceux-ci recevraient des fonctions de pondération, et donc au final, les poids des feuilles seraient fonction des attributs.

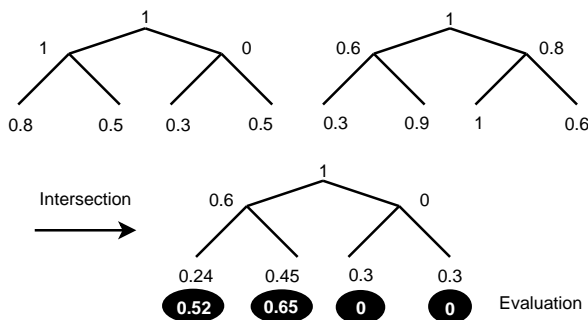


Figure 7. Intersection et évaluation d'arbres de pondération

#### 4.5. *Instanciation*

Grâce à la méthode décrite ci-avant, on obtient donc un ensemble de feuilles (*i.e.* de modalités) *évaluées*. L'évaluation d'une modalité donnée est une fonction définie sur l'ensemble de ses attributs.

D'emblée, il se peut que toutes ces évaluations soient nulles, ce qui est le cas si les profils sont incompatibles (par exemple, utilisateur non-voyant qui s'approche d'un écran). Bien entendu, nous nous intéressons au cas où certaines de ces évaluations sont non nulles. Plus particulièrement, nous allons supposer dans la suite qu'une seule évaluation est non nulle : ceci ne réduit pas la généralité du raisonnement, car si un dispositif physique donné présente plusieurs modalités (par exemple un ensemble écran + haut-parleur), il est toujours possible de le scinder en plusieurs dispositifs logiques ne disposant chacun que d'une seule modalité. Notons alors qu'il est possible d'interpréter la procédure d'intersection ainsi : il s'agit de déterminer si la modalité du dispositif de présentation est compatible à la fois avec l'utilisateur et l'unité sémantique.

Cette réflexion en amène une autre : *l'instanciation s'effectue du point de vue du dispositif*. En effet, c'est le dispositif qui va effectuer une présentation de l'unité sémantique en utilisant les valeurs d'attributs issues de l'instanciation. Ces valeurs doivent donc tenir compte de *toutes* les unités sémantiques présentées par le dispositif. Par exemple, sur un écran qui présenterait trois unités sémantiques, il n'est pas concevable que l'une d'entre elles occupe 90 % de la place disponible, et laisse seulement 10 % de l'espace aux deux autres, les rendant par là même illisibles.

Il est donc nécessaire de réaliser un compromis entre les attributs de toutes les u.s. présentées sur le dispositif, en fonction des desiderata des utilisateurs de chacune d'entre elles. Nous donnons ci-après une méthode pour d'obtention de compromis.

##### 4.5.1. *Espace des combinaisons possibles d'attributs*

Soit donc un dispositif donné, qui permet la présentation d'unités sémantiques selon *une* modalité donnée. Cette modalité est caractérisée par un certain nombre d'attributs, dont les domaines de variation sont notés  $E_1, \dots, E_p$ . On note  $\Omega = E_1 \times \dots \times E_p$ . Ces attributs sont tous les attributs présents dans l'arbre taxonomique sur le chemin menant de la racine à la feuille correspondant à la modalité considérée.

Sur ce dispositif, on souhaite présenter  $n$  unités sémantiques  $u_1, \dots, u_n$ . Pour chacune de ces  $n$  u.s., il faut donc déterminer une *instanciation* de la modalité, c'est-à-dire une valeur pour chacun des  $p$  attributs. Autrement dit, pour chaque u.s., on doit choisir un élément  $\omega \in \Omega$ . Au final, cela revient à choisir  $n \cdot p$  valeurs ( $p$  valeurs d'attributs pour chacune des  $n$  u.s.), c'est-à-dire un élément  $\pi$  de  $\Pi = \Omega^n$ .

Nous supposons que le dispositif présente des contraintes, notamment des contraintes de place. On introduit donc une *fonction de contraintes*,  $z$ , telle que  $z(\pi)$  vaut 1 si et seulement s'il est possible de présenter les u.s. sur le dispositif avec le choix d'attributs  $\pi$ .

Par exemple, sur un écran, il n'est pas possible d'affecter une surface arbitraire à chaque unité sémantique, car la somme des surfaces individuellement affectées à chaque u.s. doit être inférieure à la surface de l'écran  $S$ . Dans ce cas, la fonction  $z$  vaudrait 1 si la surface occupée par l'ensemble des u.s. était inférieure à  $S$ , et 0 sinon.

On introduit l'ensemble  $\tilde{\Pi}$  des éléments de  $\Pi$  qui satisfont les contraintes du dispositif :  $\tilde{\Pi} = \{\pi \in \Pi \mid z(\pi) = 1\}$ .  $\tilde{\Pi} \subset \Pi$ .

#### 4.5.2. *Evaluation des combinaisons d'attributs*

On cherche alors à *évaluer* chacun des éléments de  $\tilde{\Pi}$ , vis-à-vis des préférences des utilisateurs. On suppose qu'il y a  $m$  utilisateurs. Il n'y a aucune raison pour que  $m$  soit égal à  $n$ , car un utilisateur peut être intéressé par plusieurs u.s., et réciproquement, plusieurs utilisateurs peuvent être intéressés par la même u.s. On suppose que l'utilisateur numéro  $i$  est intéressé par  $k$  u.s., d'indices  $j_1$  à  $j_k$ . A un choix  $\pi$  des attributs, et à une u.s. d'indice  $j_\ell$  ( $\ell \in [1, k]$ ) l'utilisateur numéro  $i$  associe donc une évaluation notée  $e_{i,\ell}(\pi)$ . Bien entendu, le nombre de ces évaluations dépend de l'utilisateur, mais chacun en fournit au moins une<sup>3</sup>. Pour l'ensemble des utilisateurs, on obtient donc au total  $q$  évaluations, avec  $q \geq m$ .

On définit alors une fonction d'évaluation  $e$  sur  $\tilde{\Pi}$  comme suit : pour un élément  $\pi$ , on obtient exactement  $q$  évaluations d'utilisateurs. On les classe par ordre croissant, et on en forme ainsi un  $q$ -uplet de la forme  $\langle e_1, e_2, \dots, e_q \rangle$  avec  $e_1 \leq e_2 \leq \dots \leq e_q$ . Ce  $q$ -uplet est noté  $e(\pi)$ . Il s'agit de l'*évaluation* de la combinaison d'attributs  $\pi$ .

#### 4.5.3. *Choix d'une combinaison d'attributs*

Il s'agit maintenant de déterminer quelle est la *meilleure* combinaison d'attributs. Comment la définir ? Le but du système est de donner un confort maximal à un maximum d'utilisateurs. Bien entendu, dès l'instant où il y a plusieurs utilisateurs, le niveau de préférence atteint ne pourra pas être le même pour tout le monde : certains seront mieux lotis que d'autres. Il s'agit donc de trouver un compromis entre les désirs des uns et des autres.

Il nous semble donc important d'assurer un niveau de préférence *minimum* pour tout le monde. Si on s'assure que le niveau de préférence *atteint par l'utilisateur le moins bien loti* dépasse une valeur minimale, alors on est sûr que personne ne peut être sacrifié au profit des autres. Cette propriété est intéressante, car notre but est de fournir un système qui soit utile à *tous* ses utilisateurs, pas un système qui en privilégie certains au détriment des autres.

Les  $e(\pi)$  étant des  $q$ -uplets de préférences classées par ordre croissant, ce qui précède nous pousse à choisir les  $q$ -uplets dont le *premier élément* est maximal (*i.e.* pour lesquels la préférence la moins bonne est maximale). De la sorte, on est assuré que toutes les préférences vaudront *au moins* la valeur de ce premier élément. Si *plusieurs*

3. Si un utilisateur ne fournissait pas d'évaluation, cela voudrait dire qu'il n'est intéressé par aucune u.s. de l'écran, donc il n'aurait rien à faire dans la liste des utilisateurs considérés.

$\pi$  sont évaluées en des  $q$ -uplets de même premier élément, on les départagera selon leur deuxième élément, et ainsi de suite. Ceci revient à choisir la combinaison  $\pi$  d'attributs telle que  $e(\pi)$  soit maximal *au sens de l'ordre lexicographique*.

Par exemple, supposons qu'un dispositif doit présenter trois u.s. pour trois utilisateurs  $A$ ,  $B$  et  $C$ . Le système doit déterminer les valeurs de cinq attributs, étant données les évaluations de trois utilisateurs. Dans le tableau 1, la première colonne contient les combinaisons de valeurs d'attributs. Les trois suivantes contiennent les évaluations d'utilisateurs correspondantes, et la dernière le vecteur d'évaluation global, composé des valeurs des trois précédentes classées par ordre croissant. La solution choisie est la troisième, car elle maximise le taux de satisfaction de l'utilisateur le moins satisfait. Ainsi, toutes les évaluations valent au moins 0,7.

$\omega$ – Valeurs	$e_A$	$e_B$	$e_C$	$e(\omega)$ – Evaluation
(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)

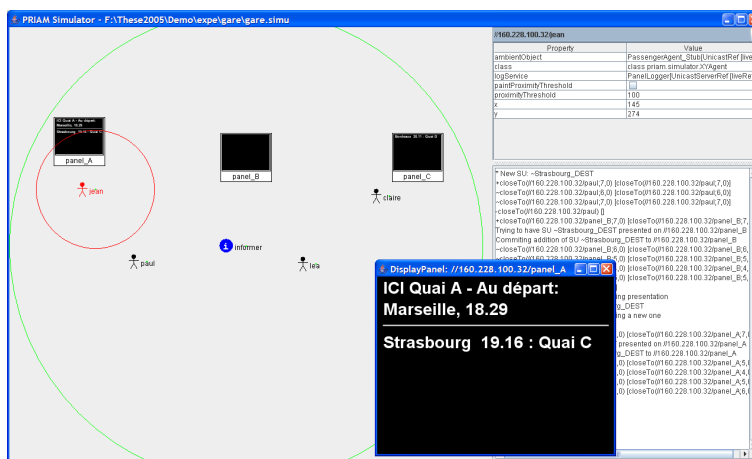
**Tableau 1.** Choix d'une combinaison de valeurs d'attributs

## 5. Implémentation

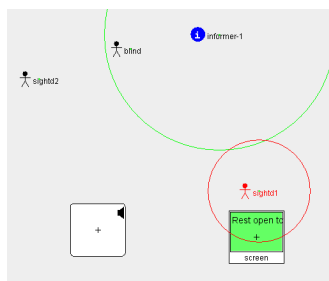
Nous avons implémenté ces algorithmes sous forme d'une plate-forme appelée PRIAM, pour présentation d'informations dans l'ambient. Cette plate-forme est construite selon une architecture à agents. Les agents sont implémentés en Java, ce qui leur permet de fonctionner sur des environnements matériels divers. De plus, ils peuvent facilement s'échanger des messages par le réseau à travers le mécanisme des RMI (*remote method invocation*) intégré à Java.

Comme la mise en place d'une expérimentation grandeur nature est relativement lourde et coûteuse en temps, nous avons réalisé un simulateur qui permet de tester tous les composants finaux d'une application, sans pour autant devoir les déployer en conditions réelles dès les premiers essais (voir figure 8).

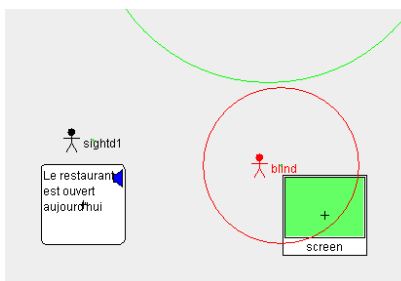
Ce simulateur nous a permis de vérifier le bon fonctionnement des algorithmes. Par exemple, les figures 8b et 8c mettent en jeu deux utilisateurs (un voyant et un non-voyant), ainsi qu'un écran et un dispositif sonore. Lorsque l'utilisateur voyant se présente, l'écran et le système sonore peuvent lui donner une information (b). Par contre, lorsque le non-voyant s'approche, l'écran n'affiche rien car sa modalité de sortie n'est pas compatible avec le profil du non-voyant (c). Tous les aspects des algorithmes peuvent être testés de cette manière, avec le nombre nécessaire de dispositifs de présentation, d'utilisateurs et de sources d'information.



(a)



(b)

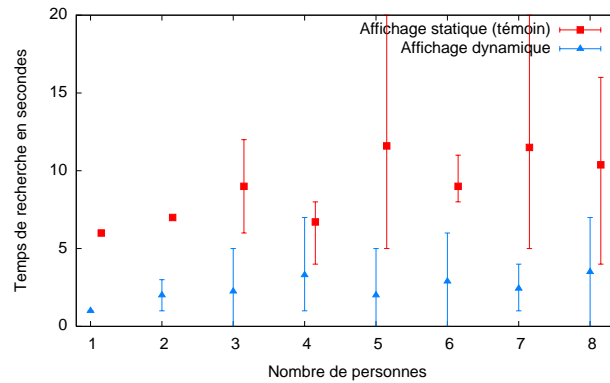


(c)

**Figure 8.** Le simulateur de la plate-forme PRIAM

Nous avons également réalisé une expérimentation en conditions quasi réelles. Nous avons demandé à des sujets de retrouver une note d'examen dans une liste, statique ou dynamique. Les listes statiques comportaient 450 noms et notes (expérience témoin). Le système dynamique n'affichait lui que les informations relatives aux utilisateurs situés à proximité d'un écran. Nous avons alors mesuré le temps de recherche des notes : les résultats sont donnés sur la figure 9.

Dans cette expérience, la recherche des informations a été au moins 50 % plus rapide avec le système dynamique, ce qui montre l'intérêt de notre système. Pour mieux évaluer ses apports en situation réelle, il sera cependant nécessaire de mettre en œuvre une expérimentation grandeur, par exemple dans une gare ou un aéroport.



**Figure 9.** Temps de recherche des notes en fonction du nombre de personnes. Les points représentent les valeurs moyennes, les barres les écarts types

## 6. Conclusions et perspectives

Nous avons présenté un modèle conceptuel, ainsi que des algorithmes qui permettent la réalisation d'un système de présentation multimodale d'informations. Nous avons rapidement introduit notre implémentation de ce modèle, la plate-forme PRIAM, qui nous a permis de vérifier le bon fonctionnement de nos algorithmes.

Sur un dispositif de présentation donné, par exemple un écran, nous pourrions *classer* les différentes u.s. affichées. De la sorte, il serait possible d'affecter une *priorité* à chaque u.s. Ainsi, les u.s. les plus prioritaires (par exemple, des informations sur des départs imminents, ou sur des disparitions d'enfants) apparaîtraient en premier. De même, il pourrait y avoir des priorités entre utilisateurs. Par exemple, les handicapés ou les souscripteurs de formules *privilégiées* auraient une plus grande priorité. De plus, lorsque la présentation simultanée d'un grand nombre d'u.s. serait impossible, les priorités aideraient à choisir lesquelles seraient présentées en priorité.

Si un utilisateur est seul devant un écran, alors seules ses informations sont affichées, par exemple sa destination. Cela peut permettre à des personnes mal intentionnées d'apprendre des informations privées sur cet utilisateur. Ce problème peut être résolu en *brouillant* l'affichage, c'est-à-dire en affichant dans ce cas quelques informations non pertinentes de façon à assurer le respect de la vie privée.

A court terme, nos expérimentations grandeur nature nous permettront d'évaluer l'utilisabilité de ce type de systèmes d'information très dynamiques, ainsi que l'utilisabilité de la plate-forme du point de vue des développeurs. Nous espérons en déduire des lignes directrices pour la conception d'applications grâce à la plate-forme PRIAM.



## 7. Bibliographie

- Bass L., Faneuf R., Little R., Mayer N., Pellegrino B., Reed S., Seacord R., Sheppard S., Szczur M. R., « A Metamodel for the Runtime Architecture of an Interactive System », *SIGCHI Bulletin*, vol. 24, n° 1, p. 32-37, 1992.
- Coutaz J., « PAC, an Object-Oriented Model for Dialog Design », in H.-J. Bullinger, B. Shackel (eds), *INTERACT 87 - 2nd IFIP International Conference on Human-Computer Interaction*, North-Holland, p. 431-436, September, 1987.
- Ducatel K., Bogdanowicz M., Scapolo F., Leijten J., Burgelman J.-C., Scenarios for Ambient Intelligence in 2010, Final report, Information Society Technologies Advisory Group (IS-TAG), European Commission, February, 2001.
- Harter A., Hopper A., « A Distributed Location System for the Active Office », *IEEE Network*, vol. 8, n° 1, p. 62-70, January, 1994.
- Hull R., Neaves P., Bedford-Roberts J., « Towards Situated Computing », *ISWC '97*, IEEE Comp. Soc., Washington, DC, USA, p. 146, 1997.
- Kindberg T., Barton J., « A Web-based nomadic computing system », *Computer Networks (Amsterdam, Netherlands : 1999)*, vol. 35, n° 4, p. 443-456, 2001.
- Krasner G. E., Pope S. T., « A cookbook for using the model-view controller user interface paradigm in Smalltalk-80 », *Journal of Object Oriented Programming*, vol. 1, n° 3, p. 26-49, 1988.
- Long S., Kooper R., Abowd G. D., Atkeson C. G., « Rapid Prototyping of Mobile Context-Aware Applications : The Cyberguide Case Study », *Mobile Computing and Networking*, p. 97-107, 1996.
- Souchon N., Limbourg Q., Vanderdonck J., « Task Modelling in Multiple Contexts of Use », *Proceedings of the 9th International Workshop on Interactive Systems. Design, Specification, and Verification*, Springer-Verlag, p. 59-73, 2002.
- Streitz N. A., Röcker C., Prante T., Stenzel R., van Alphen D., « Situated Interaction with Ambient Information : Facilitating Awareness and Communication in Ubiquitous Work Environments », *HCI International 2003*, June, 2003.
- Teil D., Bellik Y., « Multimodal Interaction Interface using Voice and Gesture », in M. M. Taylor, F. Néel, D. G. Bouwhuis (eds), *The Structure of Multimodal Dialogue II*, John Benjamins Publishing, chapter 19, p. 349-366, 2000.
- Vogel D., Balakrishnan R., « Interactive public ambient displays : transitioning from implicit to explicit, public to personal, interaction with multiple users », *UIST '04*, ACM Press, Santa Fe, NM, USA, p. 137-146, 2004.
- Weiser M., « Some computer science issues in ubiquitous computing », *Communications of the ACM*, vol. 36, n° 7, p. 75-84, 1993.
- Zock M., Sabah G., « La génération automatique de textes », in M. Fayol (ed.), *Production du langage*, Hermès, p. 263-285, 2002.