KUP: A MODEL FOR THE MULTIMODAL PRESENTATION OF INFORMATION IN AMBIENT INTELLIGENCE

Christophe Jacquet[1], **Yacine Bellik**[2] **and Yolaine Bourda**[1]

[1]Supélec, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette Cedex, France
[2]LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France

Keywords: Ubiquitous computing, Multimodality, Mobility

Abstract

This paper deals with the design of multimodal information systems in ambient intelligence. Its agent architecture is based on KUP, an alternative to traditional software architecture models for human-computer interaction. The KUP model is accompanied by an algorithm for choosing and instantiating interaction modalities. The model and the algorithm have been implemented in a platform called PRIAM, with which we have performed experiments in pseudo-real scale.

1 Introduction

Users of public places often have difficulties to obtain information that they need, especially when they do not know the premises. For instance, when a passenger arrives in an airport, he does not know where his boarding gate is located. So as to provide users with potentially useful information, staff generally place *information devices* in specific locations. These can be screens, loudspeakers, interactive information kiosks, or simply display panels. Thus, monitors display information about upcoming flights in an airport, maps show the location of the shops in a shopping mall, etc.

However, these information sources give non-targeted, general purpose information suitable for anyone. In consequence, they are generally overloaded with information items, which makes them difficult to read. Yet, a given user is generally interested in only one information item: finding it among a vast quantity of irrelevant items can be long and tedious.

Indeed, it is no use presenting an information that nobody is interested in. Therefore, we propose an ubiquitous information system that is capable of providing personalized information to mobile users.

For instance, monitors placed at random in an airport could provide nearby passengers with information about their flights. Only the information items relevant to people located in front of the screens would be displayed, which would improve the screens' readability and and reduce the users' cognitive load.

As we have just seen, all users are faced with difficulties when they are seeking information and have to move around in an unknown environment. However, these tasks are all the more painful as people have disabilities. Indeed, classical information devices are often not suited for handicapped people. For instance, an information screen is useless to a blind person. Similarly, a deaf person cannot hear information given by a loudspeaker.

For these reasons, we focus on *multimodal* information presentation. One given device will provide information to a user only if its output modality is compatible with the user's input modalities. This way, the system will avoid situations in which people cannot perceive the information items.

Besides, we wish to avoid any initial configuration of the system. In [6], we have proposed a framework to have display screens cooperate with each other, as soon as they are placed close to one another. In this paper, we build on this zero-configuration system and add multimodal adaptation features.

Section 2 gives a short review of related work. Section 3 introduces a new software architecture model for ambient intelligence systems, called KUP. An agent-based embodiment of this model is introduced in Section 4. In Section 5 we propose an algorithm for choosing modalities when creating information presentations. Finally, Section 6 gives the result of experiments that have assessed the benefits of using our framework.

2 Related Work and Objectives

Since the mid-1990s, several research projects have attempted to provide information to mobile users. In general, the resulting systems are built around Personal Digital Assistants (PDAs): the Cyberguide [10] is a museum tour guide; CoolTown [8] displays web pages to users depending on their location.

These approaches suffer from one major drawback: they force users to carry with them a given electronic device. Even if almost everyone owns a mobile phone today, it is painful to stop to look at one's phone screen, especially when one is carrying luggage. For instance, if someone is looking for their boarding desk in an airport, they would find it disturbing to stop, put down their luggage and take out their mobile phone.

Yet, a few recent systems, such as the Hello.Wall [13], aim at using large public surfaces to display personal information. However, to respect people's privacy [15], the information items cannot be broadcast unscrambled. Thus, the Hello.Wall displays cryptic light patterns that are specific to each user. This limits the practical interest of the system, which is rather an artistic object than a usable interface.

We do not wish to broadcast *personal* information, but rather to *perform a selection* among the whole set of available information, which limits the scope of the privacy issues. Presentation devices will provide information relevant only to people located at proximity.

We have already proposed a model and algorithms that enable to use diverse public screens to display information to several mobile users [6]. This is a kind of Distributed Display Environment (DDE) [5]. However, whereas usual DDE systems are based on static configurations of screens (see for instance [11]), we have introduced a model in which the assignation of information to screens changes in a purely dynamic way.

The present article takes the idea further, and introduces a notion of double *opportunism* when providing and presenting information. Besides, beyond simple content layout, we wish to take several modalities into account, which is not dealt with by DDE problematics. Thus, the present article also focuses on the negotiation of multimodal content between heterogeneous users and devices.

Note that the topic here is *not* to specify a general-purpose framework for building contextual or ambient applications. Rather, the applications that it describes may be built *on top* of such existing frameworks, for instance those described in [4] or [7].

3 The KUP Model

3.1 Requirements

As people rapidly move from place to place in public spaces, they will not necessarily be able to perceive a presentation device (look at a monitor or listen to a loudspeaker) when a given information item is made available. In consequence, the system must ensure that this information item is presented to them *later*, when a suitable device becomes available.

This leads us to consider two unsynchronized phases:

- in a first phase, an information item is *"conceptually"* provided to the user,
- in a second phase, this information item is *physically* presented to the user, through a suitable device and modality (text on a screen, speech synthesis from a loudspeaker, etc.)

To *"conceptually"* provide information to the user, the latter must be explicitly represented by a logical entity in the system. This entity is introduced by the KUP model.

3.2 Knowledge Sources, Users and Presentation Devices

The KUP model is a software architecture model for ambient intelligence systems. It takes three logical entities into account:

- knowledge sources, for instance the information source about flight delays in an airport. They are denoted by K_ℓ,
- logical entities representing users, denoted by U_ℓ,
- logical entities representing presentation devices, denoted by P_ℓ.

These logical entities correspond one-to-one to physical counterparts, respectively:

- the spatial perimeter (zone) in which a certain knowledge is valid, denoted by K_φ,
- human users, denoted by U_φ,
- physical presentation devices, denoted by P_φ.

Most software architecture models for HCI (e.g. MVC [9], Seeheim [12], ARCH [2] and PAC [3]) rely on logical representations for the functional core and the interface only (see fig. 1). There is no active logical representation of the user. In contrast, this entity lies at the center of the KUP model (see fig. 2):

- in the first phase, a knowledge source K_ℓ sends an information item to the user entity U_ℓ,
- in the second phase, the user entity U_ℓ asks a presentation entity P_ℓ to present the information item. This results in a presentation device P_φ presenting the information for the human user U_φ.

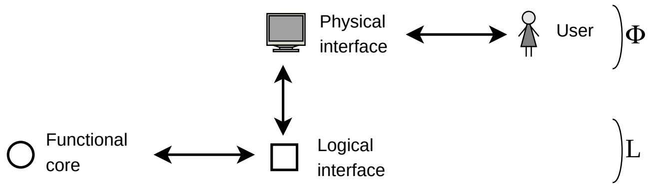

Figure 1: In classical architecture models, the user is not logically represented. The Φ and L letters respectively denote the physical and logical layers.

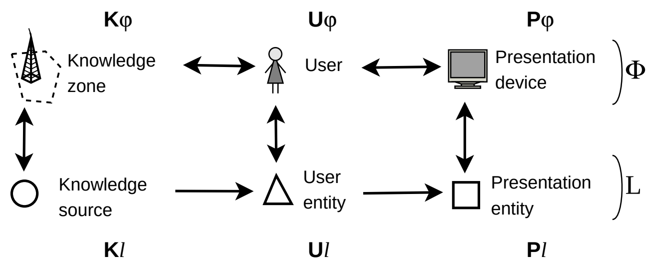

Figure 2: In KUP, a user entity lies at the center of the system. The Φ and L letters respectively denote the physical and logical layers.

3.3 Radiance Spaces and Perceptive Spaces

The physical entities have *perception* relationships with each other. For a given entity, its *radiance space* is the set of positions from where another entity can *perceive* it. Conversely, its *perceptive space* is the set of positions where it can perceive another entity.

In these definitions, *perception* means *sensory perception*. For instance, the perceptive space of a sighted user contains the screens in front of him, located at reading distance, and the loudspeakers nearby. However, the perceptive space of a blind user located at the same place contains the loudspeakers only. Therefore, perception depends not only on proximity, but also on orientation and sensory capabilities. Nevertheless, we use the terms *proximity* or *closeness* to mean *inclusion in the perceptive space*.

Proximity relationships originate in the physical world, and then are mirrored to the logical entities, that are said to share the *same* relationships.

3.4 An Opportunistic and Proximity-Based Information Presentation System

Information items are formally called *semantic units*. They are elementary pieces of information, capable of being transmitted over a network, and of expressing themselves into a number of modalities.

We have seen above that there are two phases in an interaction: information *providing* and information *presentation*. The first phase can be very simple: when a user enters the perceptive space of a knowledge source, the knowledge source may send a semantic unit of interest to the logical entity U_ℓ. We will not give more details on this phase. Rather, we will focus on the second phase.

The user is mobile: when he or she receives a semantic unit, there is not necessarily a presentation device available at proximity. But when at a given moment, one or more devices becomes available, the user entity will try have the semantic unit presented on one of them. There are two interdependent sub-problems:

1. if there are several devices available, one of them must be chosen. This topic has been dealt with in [6],

2. for a given presentation device, the user and the device must agree on a modality to be used to convey the semantic unit. Indeed, the system presented here is *multimodal* because it can successively use diverse modalities. However, it is not designed to mix several modalities to convey one given semantic unit. This behavior is called *exclusive multimodality* [14].

The two phases that we have seen make the system's behavior opportunistic in two respects:

- with respect to information providing: the user receives semantic units when she enters specific areas, while moving around,

- with respect to information presentation: semantic units are presented when the user stumbles upon a presentation device.

4 Architecture

It would have been possible to build a system based on a *centralized* architecture. However, we think that this has a number of shortcomings, namely fragility (if the central server fails, every entity fails) and rigidity (one cannot move the knowledge sources and presentation devices at will). In contrast, we wish to be able to move, remove and bring new entities without having to reconfigure anything. The system must adapt to the changes by itself, without needing human intervention.

That is why we propose to implement logical entities by software agents: *knowledge agents* (K), *user agents* (U) and *presentation agents* (P), respectively associated with the logical entities K_ℓ, U_ℓ and P_ℓ. Proximity relationships are sensed in the real world, and then mirrored to the world of agents.

We suppose that agents can communicate with each other thanks to an ubiquitous network. This assumption is realistic since the advent of wireless and mobile networks. Besides, agents are defined as *reactive*. An agent stays in an idle state most of the time, and can react to two kinds of events:

- the receipt of an incoming network message from another agent,

- a change in its perceptive space (i.e. another agent/entity comes close or moves away).

Since all agents are only reactive, events ultimately originate in the real world. In the real world, users are proactive[1]: they move, which is mirrored in the world of the agents, and hence triggers reactive behaviors.

The events happening in the real world are sensed by physical artifacts. For instance, RFID technology can be used to detect proximity, and hence to construct perceptive spaces. This way, monitors could detect users approaching in an airport thanks to the passengers' tickets, provided that the tickets are equipped with RFID tags. Other possible techniques include computer vision and Bluetooth.

5 Choosing a Modality

5.1 Taxonomy of Modalities

We call *modality* a concrete form of communication using one of the five human senses [14]. Examples of modalities are speech, written text or music.

Before reasoning about modality and making a choice, we have to determine the list of available modalities. To this end, we propose to build a taxonomy of modalities. Figure 3 is a partial example of such a taxonomy. It is only an example: the taxonomy can be adapted to the particular needs of any given system, enhanced, refined, etc. It contains modalities for two senses only (visual and auditory), but could include the tactile sense as well.

[1] Presentation devices and knowledge sources may be proactive too. They can be moved, yet at a different pace and rate. For instance, staff can move monitors in an airport, or can change the radiance space of a knowledge source so as to reflect a new organization of the airport.

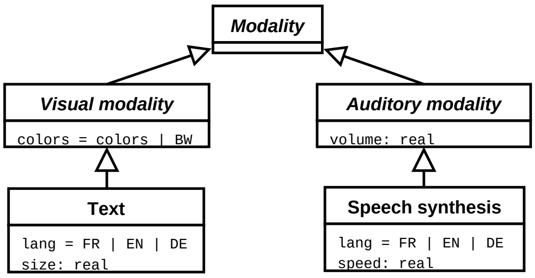

Figure 3: Excerpt of a taxonomy of modalities.

In the taxonomy, all modalities are classified in a tree. Leaves represent concrete modalities, whereas internal nodes represent abstract modalities, that correspond to groups of (sub-)modalities. The root of the tree is an abstract modality that encompasses every possible modality. The second-level abstract modalities correspond to the senses of human beings.
Modalities have *attributes* that characterize a concrete presentation using this modality. Attributes can have discrete or continuous values. For instance, the language for a text must be selected in a finite list, whereas the text size can take any value in a given interval.
Before presenting an information using a modality, the values for the modality's attributes have to be determined first. This step is called *instantiation* [1].

5.2 Profiles

The problem that we have to solve is as follows: a given user wishes to have a given semantic unit presented on a given presentation device. The system must choose a modality, and instantiate it, in order to present the semantic unit. The modality and its instantiation must be compatible with both:

- the user's capabilities (e.g. one cannot use a visual modality if the user is blind) and preferences (e.g. if a user prefers text to graphics, the system must try and satisfy this wish),
- the presentation device capabilities (e.g. a monochrome screen is not capable of performing color output),
- the semantic unit's capability to convey its informational content in different modalities.

If there are several possibilities, the system should choose the user's *preferred solution* among them.
To solve this problem, we associate a *profile* with the user, the presentation device and the semantic unit. These profiles describe interaction capabilities and possibly preferences, i.e. which modalities can be used, which attribute values are possible. The solution will have to comply with *each* profile, therefore it will lie at the "intersection" of the three profiles.
We define a profile as a weighting of the modality tree. A real number, comprised between 0 and 1, is associated with each node of the tree. 0 means that the corresponding modality (or the corresponding sub-tree) cannot be used; 1 means that it can be used; values in-between can indicate a preference level. For instance, in the profile of a blind person, the sub-tree corresponding to visual modalities is weighted by 0, so that it cannot be used. Likewise, in the profile of a monitor, only the sub-tree corresponding to visual modalities is weighted by a non-null value.
The nodes' weights will determine the choice of a modality. Similarly, attributes are "weighted" too, which will help instantiating the chosen modality. More precisely, each possible value of an attribute is given a weight between 0 and 1, with the same meaning as above. Formally, a *weight function* is associated with the attribute, whose domain is the attribute's possible values, and whose codomain is the $[0, 1]$ interval.
Figure 4 is an example of a partial profile (the underlying taxonomy is a subset of the taxonomy of Figure 3. It only contains two concrete modalities). The profile describes a user with a visual impairment, whose native tongue is English. The node weights are shown in white characters inside black ovals. Since the user is visually impaired, but not blind, the weight of the visual modality is low, but not zero. The weight functions of the attributes are depicted inside boxes with rounded corners. Discrete functions are associated with attribute whose values are discrete. For instance, weights are given to any possible value of the `lang` attribute. Continuous functions are associated with attributes with continuous values. For instance, a function maps a weight to any speed, expressed in words per minute (wpm).

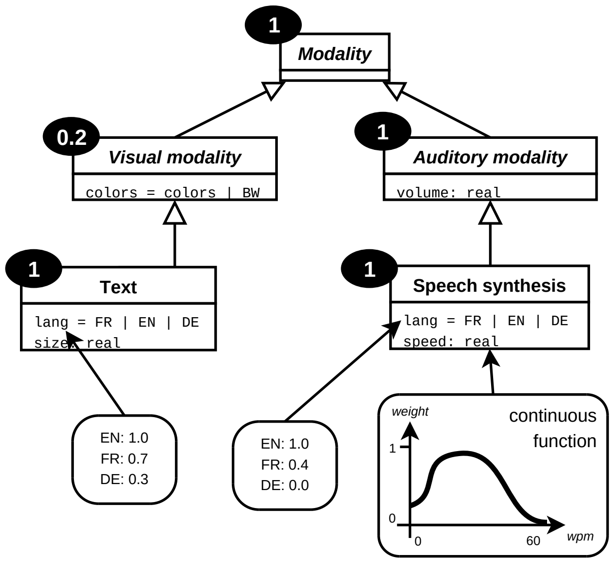

Figure 4: A partial profile (for the sake of clarity, some attribute weight functions are not shown).

5.3 Choosing a Modality

This section explains how the profiles can be used to determine the best possible modality instantiation when presenting semantic units. Figure 5 gives an overview of the various steps described below.

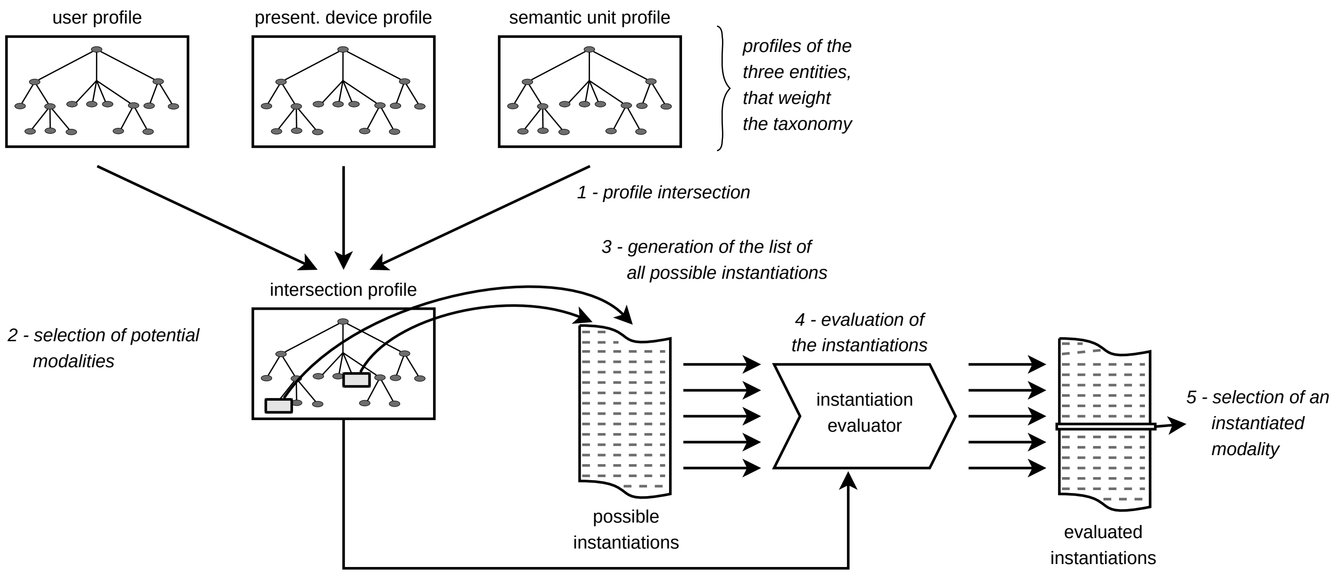

Figure 5: Overview of the algorithm for choosing a suitable modality. First, profiles are intersected, which gives out a list of usable modalities. Each possible instantiation of these modalities is *evaluated*, so as to choose the best one.

To select a modality, the system has to take the three profiles into account (user's, presentation device's, semantic unit's). To this end, we define the notion of *intersection* of profiles. The *intersection* of n profiles $p_1, \ldots, p_n$ is a profile (i.e. a weighted modality tree), in which weights are defined as follows:

- the weight of a node is the product of the n weights of same node in the profiles $p_1, \ldots, p_n$,
- the weight function of an attribute is the product of the n weight functions of the same attribute in the profiles $p_1, \ldots, p_n$.

We call it *intersection* because it has natural semantics. Indeed, a given node is weighted by 0 in the resulting profile if and only if there is at least one of the intersected profiles in which the given node is weighted by 0. The resulting profile is called $p_\cap$. $p_\cap$ contains information about which modalities can be used to present a given semantic unit to a given user, on a given presentation device. It also contains information to determine the values of the attributes of the chosen modality (instantiation, see below).

First, the system has to choose a concrete modality, i.e. one of the leaves of the tree. To do this, it *evaluates* each leaf. The evaluation of a leaf is a real number that accounts for the weights that have been assigned to all its ancestors in the weighted tree. If an internal node has a null weight, it means that the corresponding sub-tree cannot be used, so all its leaves must evaluate to zero. We could therefore define the evaluation of a leaf to be equal to the product of all the ancestor node weights. However, in this case leaves with many ancestors would by nature be more likely to have a small evaluation than leaves with fewer ancestors.

To avoid this shortcoming, we define the *evaluation* of a concrete modality (i.e. a leaf), to be the *geometric mean* of all its parent modalities' weights (including its own weight). More precisely, if $w_1, \ldots, w_m$ are the node weights along a path going from the root (weight w_1) to the concrete modality (weight w_m), then the evaluation is:

$$e = \sqrt[m]{w_1 \times w_2 \times \cdots \times w_m}$$

From that, we decide to choose the concrete modality with the highest evaluation.

Figure 6 illustrates profile intersection and modality evaluation on one simple example. In this case, the system would choose to use the modality that evaluates to 0.65.

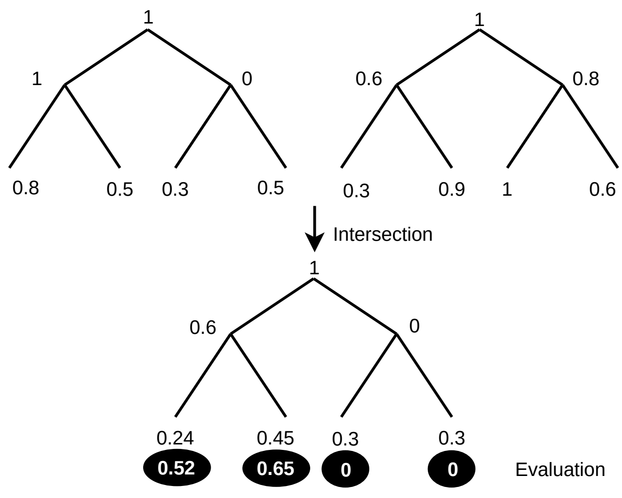

Figure 6: Intersection and evaluation.

5.4 Instantiating the Chosen Modality

Once a modality has been selected, the system has to determine the values of its attributes. Of course, the weight functions of $p_\cap$ must be taken into account. But actually, there must be a *global trade-off* between the needs and preferences of *all* the users located at proximity, the capabilities of *all* the semantic units to be presented, and the capabilities of the presentation device.

For instance, let us suppose that two users each have one semantic unit displayed on a screen, as a line of text. Each of them would like his semantic unit to be displayed in the largest font size possible. However, the surface of the screen is limited, and so are the font sizes for each user. So the system must calculate a trade-off between the attribute values of the two semantic units.

We suppose that there are a number of semantic units to present on a given device, which gives a total of n attributes, whose domains are called $D_1, \ldots, D_n$. We call *attribute combination space* the set of all possible combinations of the attribute values, and we denote it by Ω. $\Omega = D_1 \times D_2 \times \cdots \times D_n$ (Cartesian product).

Some of the elements of this set are not compatible with the constraints of the presentation device. We define $\tilde{\Omega}$ as the subset of Ω whose elements are compatible with these constraints. So the "best" combination of attributes is one of the elements of $\tilde{\Omega}$. Informally, we can define the "best" solution as the solution that gives satisfaction to as many users as possible. Let us see how we can formally define this.

In a similar way as we have defined evaluations above, we define the *evaluation function* of a concrete modality to be the geometric mean of the evaluation functions of the attributes of the concrete modality and its ancestors. If there are p such attributes, of domains $d_1, \ldots, d_p$ and of weight functions $f_1, \ldots, f_p$, the evaluation function of the concrete modality, denoted by e, is defined over $d_1 \times d_2 \times \cdots \times d_p$:

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

So for each user interested in one of the semantic units to present, there is an evaluation function. Let us suppose that there are q evaluation functions, denoted by $e_1, \ldots, e_q$. Let us take one of them, denoted by e_i. e_i is defined on a subset of $\Omega = D_1 \times \cdots \times D_n$, so it can be extended onto Ω or $\tilde{\Omega}$. We denote this extension by $\tilde{e}_i$.

Therefore, we can associate a q-component vector to each element ω of $\tilde{\Omega}$, consisting of the q values $\tilde{e}_1(\omega), \ldots, \tilde{e}_q(\omega)$ sorted by ascending order. This vector is called *evaluation* of ω and is denoted by $e(\omega)$. For a given combination of attribute values, $e(\omega)$ is the list of evaluations of the combination, *starting with the worst evaluation*.

We want to give satisfaction to as many users as possible, so we must ensure that no-one is neglected in the process. For this reason, we decide to choose the combination of attributes whose worst evaluations are maximum. More precisely, we sort the vectors $e(\omega)$, for all ω, by ascending *lexicographical* order. We then choose the value ω with the greatest $e(\omega)$, with respect to this lexicographical order.

Example — let us suppose that a device has to present three semantic units for three users A, B and C. The system has to determine the values of five attributes, given the evaluations given by the three users. The problem is formalized on Table 1.

ω – **Values**	e_A	e_B	e_C	$e(\omega)$ – **Eval.**
(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)

Table 1: Formalization of the example situation.

In Table 1, the first column contains the attribute combinations. The next three columns contain the corresponding user evaluations, and the last column the global evaluation vector, composed of the values of the three preceding columns in ascending order. The chosen solution is the third one, because it maximizes the least satisfied user's satisfaction (all user evaluations are at least 0.7 in this solution).

6 Implementation and Evaluation

We have built an implementation of the framework described in this article. It is called PRIAM, for PResentation of Information in AMbient intelligence. It is based on Java. Network transparency is achieved thanks to RMI[2].

The goal of the evaluations is to demonstrate the interest of dynamic information presentation systems for mobile users. The evaluations are based on screen displays. Proximity among screens and users is sensed thanks to infrared badges. Other techniques could have been used, such as RFID, but infrared present a significant benefit: they not only allow the detection of people's proximity, but also of people's orientation. This way, someone who is very close to a screen, but turning her back to the screen, is not detected. Interestingly, this corresponds to the notion of *perceptual proximity*.

6.1 Information Lookup with Dynamic Displays

We performed an evaluation so as to assess the impact of dynamic display of information in terms of item lookup time. 16 subjects had to find an information item among a list of other similar items. We proposed two different tasks: to find a mark obtained at an examination and to find the details about a flight. We measured the lookup time for each user, with respect to the number of users standing in front of the list. There were 1 to 8 simultaneous users (see fig. 7), which seems to be realistic of the maximum number of people who can gather around the same display panel.

On the one hand, in control experiments, users were presented with fixed-size dynamic lists, containing 450 examination marks or 20 flight details. On the other hand, when using the dynamic system, the display panel showed only the information relevant to people standing at proximity (i.e. 1 to 8 items).

This experiment showed that information lookup was far quicker when information display was dynamic:

- as for mark lookup (see fig. 8), lookup times were

[2]Remote Method Invocation.

Figure 7: Mark lookup in a list (in the back) or on a screen (on the left). This is a picture from the experience video.

51 % to 83 % shorter (depending on the number of users), and in average 72 % shorter,

- as for flight lookup (see fig. 9), lookup times were 32 % to 75 % shorter (depending on the number of users), and in average 52 % shorter.

However, people were generally disturbed by the items dynamically appearing and vanishing, which caused complete redisplays each time, because the lists were constantly kept sorted. This problem could be addressed by inserting transitions when adding and removing items, or by adding new items unsorted at the bottom of the lists.

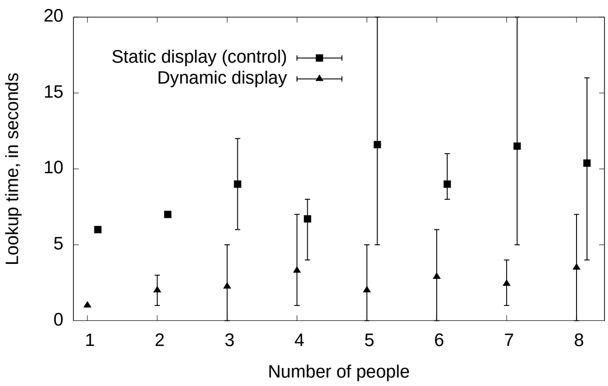

Figure 8: Mark lookup time, with respect to the number of people. The vertical bars represent standard deviations, the dots average values.

6.2 Avoiding Unnecessary Moves in a Train Station

In a second experiment, we *added* dynamic information to initially static display screens, such as those located in train stations' underpasses. In an underpass, a screen is located near the passageway leading to each platform: it displays the departure times for the trains on that platform. However, when a passenger changes trains, he initially has no clue which direction to take, so roughly half of the times, he first walks the whole underpass in the wrong direction, and then has to go back.

Our idea is to display personalized information on *any* screen when a passenger approaches. This information can

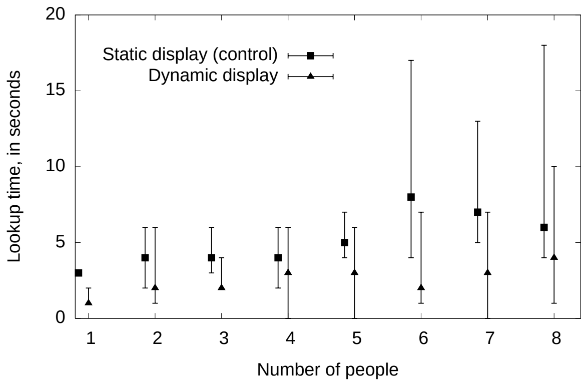

Figure 9: Flight information lookup time, with respect to the number of people. The vertical bars represent standard deviations, the dots average values.

include the platform number, as well as an arrow indicating the direction. It does not *replace* the usual static display of departing trains on the platform associated with screen, but comes *in addition* to that content. We assume that it will help people walk directly to the right platform.

We reproduced a station underpass in a corridor of our laboratory. Five display screens represented platform screens. People started from a random location in the underpass, and had to take a train to a given destination, whose platform passengers did not know. We counted the number of elementary moves of the users (n_u), and compared it to the *optimal* number of necessary elementary moves (n_o). The ratio $\frac{n_u}{n_o}$ is called *relative length* of the paths.

When provided with static information only, people often made mistakes, which resulted in unnecessary moves (table 2). When provided with additional dynamic information however, they *always* followed optimal paths (relative length of 1). These results were confirmed even when several users had to go to different platforms at the same time. People seemed to enjoy using this system, and did not feel disturbed or distracted.

Subject	n_u	n_o	Relative length
a	7	4	1.75
b	3	3	1.00
c	9	2	4.50
Average	—	—	2.42

Table 2: Relative lengths when users are provided with static information only.

6.3 Conclusions on the Evaluations

The evaluations have shown the benefits of dynamic display of information for mobile users. This turns out to allow very quick lookup of information in lists. Moreover, providing mobile users with supplementary personalized direction information enables a drastic decrease in the number of unnecessary moves.

7 Conclusions and Perspectives

We have presented a model and an algorithm that enable the design of multimodal information presentation systems. These systems can be used to provide information to mobile users. They intelligently make use of public presentation devices to propose personalized information. We have performed evaluations in pseudo-real conditions, which leads us to consider the following perspectives.

On a given screen, it could be interesting to *sort* the various displayed semantic units according to different criteria rather than just alphabetically or in a chronological way. A *level of priority* could thus be given to each semantic unit. This would for instance allow higher-priority semantic units (e.g. flights which are about to depart shortly, or information about lost children) to appear first. Similarly, there could be priorities among users (e.g. handicapped people, premium subscribers would be groups of higher priority). Therefore, semantic units priority levels would be altered by users' own priorities.

As seen above, priorities will determine the layout of items on a presentation device. Moreover, when there are too many semantic units so that they cannot all be presented, priorities could help choose which ones should be presented.

In this paper, proximity was *binary*: agents are either close to each other, or away from each other. Actually, it is possible to define several degrees of proximity, or even a measure of distance. These degrees or distances could be used as parameters of the aforementioned instantiation process. For instance, text displayed on a screen could be bigger when people are farther away.

If a user is alone in front of a screen, then only her own information item is displayed, for instance the destination of her plane. This can raise privacy concerns if someone is watching from behind. However, this can be solved by displaying one or two randomly chosen irrelevant items on the screen, thus confusing the badly disposed persons.

Our first experiments took place in *simulated* environments (a room and a corridor in our lab). So in the short term, we plan to carry out real-scale experiments, for instance in an airport or train station.

Their goal will not be to test and validate the algorithms, because we have already verified their behavior with the simulator and the experiments, but rather:

- to evaluate the overall usability of the system: how do users react to such a highly dynamic system?
- to study the sociological impact of this system,
- to test the platform's usability: is it easy to create an application? what are the guidelines to follow?

References

[1] E. André. The Generation of Multimedia Presentations. In R. Dale, H. Moisl, and H. Somers, editors, *A Handbook of Natural Language Processing*, pages 305–327. 2000.

[2] Len Bass, Ross Faneuf, Reed Little, Niels Mayer, Bob Pellegrino, Scott Reed, Robert Seacord, Sylvia Sheppard, and Martha R. Szczur. A Metamodel for the Runtime Architecture of an Interactive System. *SIGCHI Bulletin*, 24(1):32–37, 1992.

[3] Joëlle Coutaz. PAC, an Object-Oriented Model for Dialog Design. In Hans-Jorg Bullinger and Brian Shackel, editors, *INTERACT'87*, pages 431–436. North-Holland, 1987.

[4] Anind K. Dey, Daniel Salber, and Gregory D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human Computer Interaction*, 16(2-4):97–166, 2001.

[5] D.R. Hutchings, J. Stasko, and M. Czerwinski. Distributed display environments. *Interactions*, 12(6):50–53, 2005.

[6] Christophe Jacquet, Yacine Bellik, and Yolaine Bourda. Dynamic Cooperative Information Display in Mobile Environments. In B. Gabrys, R.J. Howlet, and L.C. Jain, editors, *KES2006, 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems*, volume 4252 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 154–161. Springer-Verlag, October 2006.

[7] Christophe Jacquet, Yolaine Bourda, and Yacine Bellik. An Architecture for Ambient Computing. In Hani Hagras and Victor Callaghan, editors, *The IEE International Workshop on Intelligent Environments, IE 2005*, pages 47–54. The IEE, June 2005.

[8] T. Kindberg and J. Barton. A Web-based nomadic computing system. *Computer Networks*, 35(4):443–456, 2001.

[9] Glenn E. Krasner and Stephen T. Pope. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object Oriented Programming*, 1(3):26–49, 1988.

[10] S. Long, R. Kooper, G. D. Abowd, and Ch. G. Atkeson. Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. In *Mobile Computing and Networking*, pages 97–107, 1996.

[11] Benoît Mansoux, Laurence Nigay, and Jocelyne Troccaz. The Mini-Screen: an Innovative Device for Computer Assisted Surgery Systems. *Studies in Health Technology and Informatics*, 111:314–320, 2005.

[12] Günther E. Pfaff, editor. *User Interface Management Systems: Proceedings of the Seeheim Workshop*. Springer, 1985.

[13] N. A. Streitz, C. Röcker, Th. Prante, R. Stenzel, and D. van Alphen. Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments. In *HCI International*, June 2003.

[14] Daniel Teil and Yacine Bellik. Multimodal Interaction Interface using Voice and Gesture. In M. M. Taylor, F. Néel, and D. G. Bouwhuis, editors, *The Structure of Multimodal Dialogue II*, chapter 19, pages 349–366. 2000.

[15] D. Vogel and R. Balakrishnan. 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.