Chapter 4

MULTIMODAL PRESENTATION OF INFORMATION IN A MOBILE CONTEXT

Christophe Jacquet, Yolaine Bourda
SUPELEC
3 rue Joliot-Curie, 91192 Gif-sur-Yvette Cedex, France
Christophe.Jacquet@supelec.fr, Yolaine.Bourda@supelec.fr

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

A final version of this document was published in:

Advanced Intelligent Environments,
Achilles D. Kameas, Victor Callagan, Hani Hagras, Michael Weber, Wolfgang Minker, editors
ISBN: 978-0-387-76484-9 (Print) 978-0-387-76485-6 (Online)
Springer, 2009

Abstract This chapter deals with the design of multimodal information systems in the framework of 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.

Keywords: Ubiquitous computing, Multimodality, Mobility

Introduction

Users of public places often have difficulties obtaining information that they need, especially when they are not familiar with the premises. For instance, when a passenger arrives at an airport, he does not know where his boarding gate is located. So to provide users with potentially useful information, the staff generally place *information devices* in specific locations. These can be screens, loudspeakers, interactive information kiosks, or simply display panels. For example, monitors display information about upcoming flights at 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. As a 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 information that nobody is interested in. Therefore, we propose an ubiquitous information system that is capable of providing personalized information to mobile users. The goal is not to provide *personal* information, but rather to perform a *selection* among the set of available information items, so as to present only those *relevant* to people located at proximity.

For instance, monitors placed at random at 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 screen's readability and and reduce the user's 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 for people with 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 one of its output modalities is compatible with one of 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 specific configuration of the system. In (Jacquet et al., 2006), 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 chapter, 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.

1. Related Work and Objectives

Computers, which were initially huge machines gathered in rooms dedicated to being *computer rooms*, made their way to the desktop in the 1980s. Then, as the use of microcomputers was becoming commonplace, people started imagining systems in which computerized information would be available everywhere, any time, and not only when one was sitting at one's desk. Hence came the notion of *ubiquitous computing* (Weiser, 1993). This notion is also referred to by the terms *pervasive computing*, *disappearing computer* and *ambient intelligence*.

As a consequence, 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). For instance, the Cyberguide (Long et al., 1996) pioneered the field of a museum tour guides, which has seen more recent contributions (Chou et al., 2005). Some of them simply resort to displaying web pages to users depending on their location (Kindberg and Barton, 2001; Hlavacs et al., 2005).

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 have to stop in order to look at one's phone screen, especially when one is travelling, and thus carrying luggage. For instance, if someone is looking for their boarding desk at an airport, they would find it disturbing to stop, put down their luggage and take out their mobile phone.

A few recent systems, such as the Hello.Wall (Streitz et al., 2003), aim at using large public surfaces to display personal information. However, to respect people's privacy (Vogel and Balakrishnan, 2004), 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 more an artistic object than a usable interface. Similarly, the use of the ceiling to convey information

through patterns has been investigated (Tomitsch et al., 2007). The concept of *symbiotic displays* (Berger et al., 2005) enables users to use public displays as they move for various applications such as e-mail reading. However, due to the sensitive nature of this application, they are obliged to blur the most private details, that the user must read on another, personal device (mobile phone or PDA). This makes the solution cumbersome because using two different devices is quite unnatural.

In contrast, 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 support the use diverse public screens to display information to several mobile users (Jacquet et al., 2006). This is a kind of Distributed Display Environment (DDE) (Hutchings et al., 2005). However, whereas usual DDE systems are based on static configurations of screens (see for instance (Mansoux et al., 2005)), we have introduced a model in which the assignation of information to screens changes in a purely dynamic way.

In this chapter, we take the idea further, and introduce a notion of double *opportunism* when providing and presenting information. Indeed, information items are first *opportunistically* provided by the environment, before being *opportunistically* presented onto various devices.

Besides, beyond simple content layout, we wish to be able to use several modalities. This is not dealt with by DDE studies, which focus on the physical layout of *visual* items (i.e. belonging to only *one* kind of modality). Thus, we also focus on the negotiation of multimodal content between heterogeneous users and devices. This way, we explain how a given information item can be presented on a wide range of devices with various characteristics and using various modalities. This relates to the notion of *plasticity* (Calvary et al., 2002), which studies the automatic reconfiguration of graphical user interfaces across heterogeneous devices (desktop PCs, PDAs, mobile phones, projection screens, etc.).

The methods described here are close to media allocation techniques such as those exposed in (Zhou et al., 2005). This article describes a graph-matching approach that takes into account user-media compatibility and data-media compatibility. However, in addition to these, our solution considers *user-device* compatibility: this is the key to building an opportunistic system that can use various devices incidentally encountered as the user moves.

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 (Dey et al., 2001) or (Jacquet et al., 2005).

2. The KUP Model

In this section, we introduce a conceptual model that enables applications to provide users with personalized information in public places.

2.1 Requirements

As people rapidly move from place to place in public spaces, they will not necessarily be able to perceive a given presentation device (look at a monitor or listen to a loudspeaker) at the precise moment when a given information item is made available. As a 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. This does not correspond to a physical action, but rather to an exchange of data between computer systems, corresponding respectively to an information source and to the user. More details are given below,
- in a second phase, this information item is *physically* presented to the user, through a suitable device and modality (text displayed on a screen, speech synthesized and emitted from a loudspeaker, etc.)

To *"conceptually"* provide information to the user, the latter must be explicitly represented by a logical entity in the system. Therefore, the KUP model introduces such an entity.

2.2 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 at 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 (Krasner and Pope, 1988), Seeheim (Pfaff, 1985), ARCH (Bass et al., 1992) and PAC (Coutaz, 1987)) rely on logical representations for the functional core and the interface only (see fig. 4.1). There is no active logical representation of the user. In contrast, this entity lies at the center of the KUP model (see fig. 4.2):

- in the first phase, a knowledge source K_ℓ sends an information item to the logical 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.

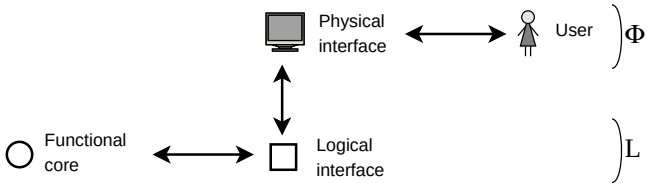

Figure 4.1. In classical architecture models, the user is not logically represented. The Φ and L letters respectively denote the physical and logical layers.

2.3 Radiance Spaces and Perceptive Spaces

The physical entities are located in a space (denoted by $\mathcal{S}$). They have *perception* relationships with each other. Let us define these notions more formally.

Perceptive Space. Informally, we wish to define the *perceptive space* of a physical entity e as the set of the points in space where an entity can be perceived by e. For instance, the perceptive space of a human being could coincide with his or her visual field. However, this definition is too restrictive:

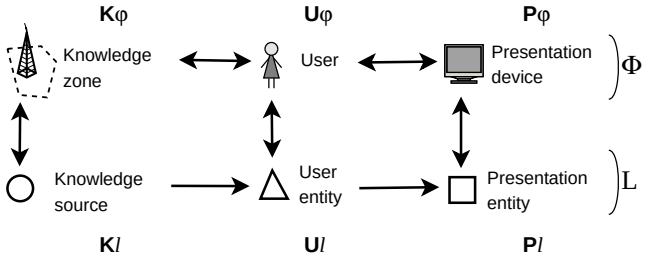

Figure 4.2. In KUP, a user entity lies at the center of the system. The Φ and L letters respectively denote the physical and logical layers.

1 a human being has several senses, which have various perception characteristics. For instance, the visual field of a person does not coincide with his/her auditory field. For example, a man cannot perceive a screen located 2 m behind him, but can generally perceive a sound emitted at the same place,

2 perception depends on the *orientation* of people, which means that a point in space $\mathcal{S}$ should not only account for one's position (x, y, z) but also for one's orientation,

3 perception depends too on the *attributes of the modalities*. For example, a phone ringing 50 m away cannot generally be heard, but a siren wailing 50 m away can be heard without any problem.

As a consequence, to precisely define the notion of perceptive space, we must take modalities and their instantiations into account. Thus, we introduce the notion of *multimodal space*, or *m-space*. An m-space is the cartesian product of the space $\mathcal{S}$ with the set of all possible instantiated modalities.

For instance, let us suppose that the relevant modalities are as follows:

- a *telephone ring*, with an attribute *volume* with continuous values ranging from 0 to 100,
- a *text*, with an attribute *size* with continuous values ranging from 10 to 60 points, and an attribute *color* whose possible values are *red*, *green* or *blue*.

Examples of points in this m-space would be:

- the point at 4623"32' N, 102"56'E, with a text of size 23, colored in green,

- the point at 4507"19' N, 201"32'E, with a text of size 59, colored in blue,

- the point at 4623"32' N, 102"56'E, with a ring of volume equal to 61.

Formally, the *perceptive space* of a physical entity e can now be defined as a subset of an m-space $\mathcal{M}$, which contains the points that e can perceive (perception being defined as above). We denote by $\mathcal{PS}(e, x)$ the perceptive space of an entity e located at $x \in \mathcal{S}$.

We have just seen that the *perception* of a physical entity e can be described in an m-space. Conversely, a physical entity can be seen as a *source of multimodal content*, whose extent is a subset of the m-space, of the form $\{(x_i, m_i)\}$ where $\{x_i\}$ is the subset of $\mathcal{S}$ corresponding to the physical extent of the object, and $\{m_i\}$ the set of the entity's modalities. The set $\{(x_i, m_i)\}$ is called *location* of the entity e, and is denoted by $\ell(e)$.

Note that the perceptive space depends on the particular person considered. 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.

Radiance Space. The perceptive space of an entity describes its perception capabilities, in other terms its *behavior as a receiver*. We can now define the inverse notion, in order to describe its *behavior as an emitter* of multimodal content.

We define the *radiance space* of an entity e, with respect to an entity d, as the set of points $x \in \mathcal{S}$ from where d can perceive e, i.e. for which e is in the perceptual space of d located in x:

$$\mathcal{RS}(e|d) = \{x \in \mathcal{S} | \ell(e) \cap \mathcal{PS}(d, x) \neq \emptyset\}$$

Proximity. The above definitions have introduced a notion of *proximity*. Proximity certainly depends on geographic locations, but it also depends on the multimodal capabilities of the entities. In the remainder of this paper, we may use the terms *proximity* or *closeness* to mean *inclusion in the perceptive space*, and they must be understood as *sensory proximity*.

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

2.4 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. However, when at a given moment, one or more devices become available, the user entity will try to 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 (Jacquet et al., 2006),

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* (Teil and Bellik, 2000). In the future, we plan to study how to use several modalities in a complementary, redundant or equivalent way (Coutaz et al., 1995).

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 he/she enters specific areas, while moving around,
- with respect to information presentation: semantic units are presented when the user stumbles upon a presentation device.

3. Software 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 has become realistic since the advent of wireless (e.g. WiFi) and mobile (e.g. GSM) 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 contrast, in the real world, users are proactive[1]: they move, which is mirrored in the world of the agents, and hence trigger 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 at an airport thanks to the passengers' tickets, provided that the tickets are equipped with RFID tags. Other possible techniques include computer vision, Bluetooth and other wireless protocols.

4. Algorithms for Choosing and Instantiating a Modality

Our system must be capable of providing users with multimodal content. As users have different needs and wishes regarding modalities, it is necessary to choose a modality and instantiate it when interacting with a user. To begin with, we define a taxonomy of modalities.

4.1 Taxonomy of Modalities

We call *modality* a concrete form of communication using one of the five human senses (Teil and Bellik, 2000). 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. Thus, we propose to build a taxonomy of modalities. Figure 4.3 is a partial example of such a taxonomy. It is nothing more than an example: the taxonomy can be adapted to the particular needs of any given system, enhanced, refined, etc.

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 human beings' senses.

This differs from Bernsen's own taxonomies of modalities (Bernsen, 1994), in which modalities are grouped according to their *arbitrary*, *linguistic*, *analogue* or *explicit* nature, and not according to the corresponding human sense. Indeed, in our taxonomies, subsumption between modalities of the first and second levels corresponds to subsumption between sensory capabilities. However, at deeper levels, our taxonomies are closer to those of Bernsen.

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 item using a modality, the values for the modality's attributes have to be determined first. This step is called *instantiation* (André, 2000).

4.2 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 each of the following:

- 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 contents using various modalities.

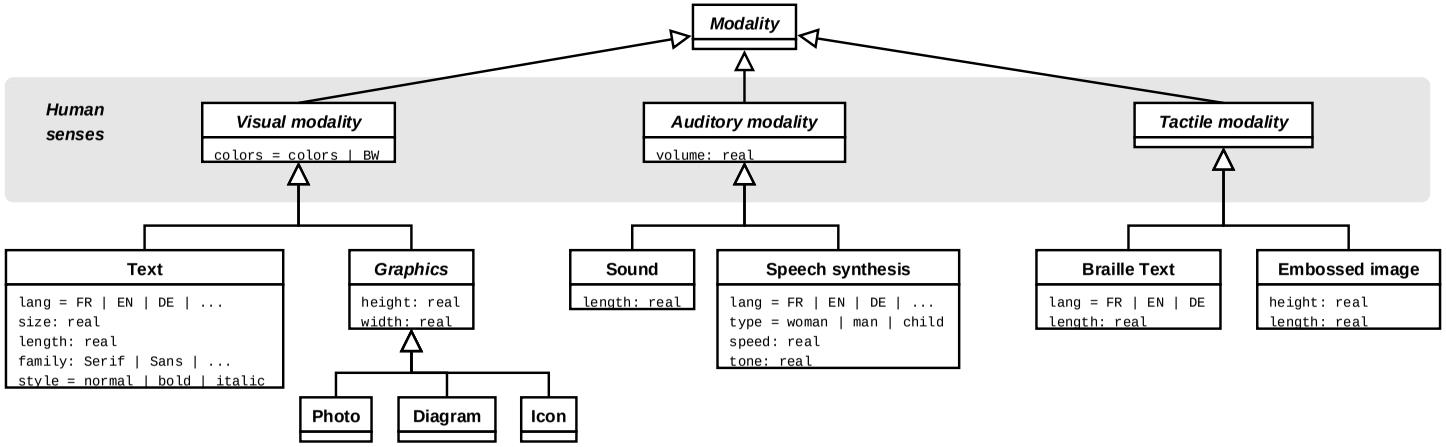

Figure 4.3. Example of a taxonomy of modalities.

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, which maps every possible value to a weight, again a real number between 0 and 1.

Figure 4.4 is an example of a partial profile (the underlying taxonomy is a subset of the taxonomy of Figure 4.3: it contains two concrete modalities only). The profile describes a user with a visual impairment, whose native tongue is English, who speaks a little French but no German[2]. 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).

The examples given here are quite simple. Up to this point, we have not studied how node and attribute weights may be fine-tuned to depict real-world situations. Indeed, to take full advantage of our model, one needs methods to define weights that perfectly match the capabilities and preferences of the entities described. This issue will have to be studied from a methodological perspective.

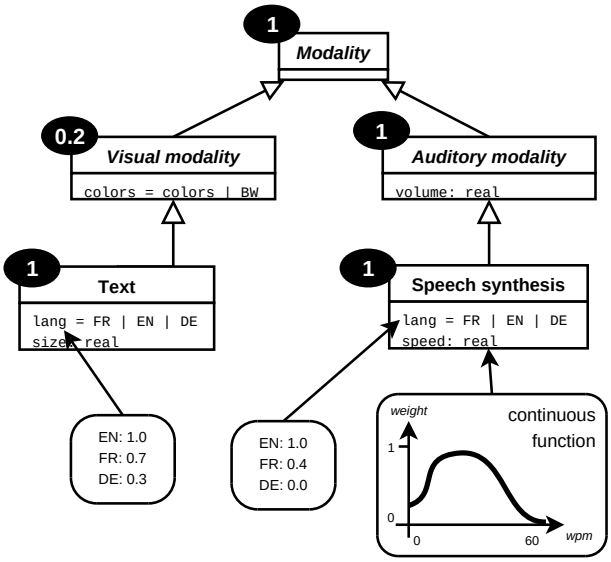

Figure 4.4. A partial profile (for the sake of clarity, some attribute weight functions are not shown).

4.3 Choosing a Modality

This section explains how the profiles can be used to determine the best possible modality instantiation when presenting semantic units. Figure 4.5 gives an overview of the various steps described below.

To select a modality, the system has to take the three profiles into account (user, presentation device, semantic unit). Thus, we define the notion of the *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 the 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 an *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

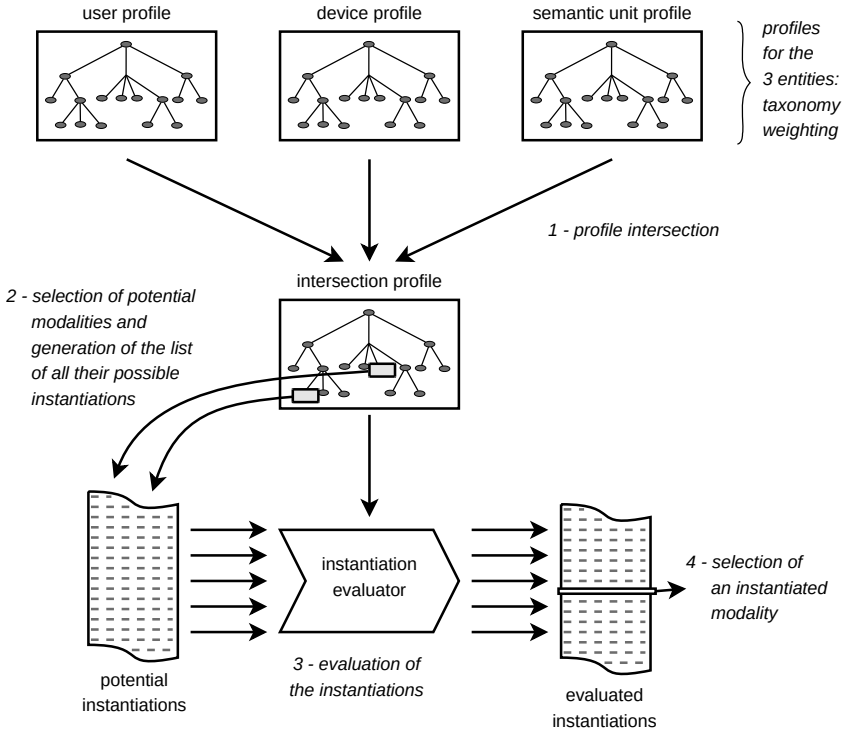

Figure 4.5. Overview of the algorithm for choosing a suitable modality. First, profiles are intersected, which gives out a list of usable modalities. Each possible instantiation of these modalities is *evaluated*, so as to choose the best one.

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 valuation 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 be valued at zero. We could therefore define the valuation 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 be valued at low values than leaves with fewer ancestors.

To avoid this shortcoming, and to account for the various numbers of ancestors of the leaves, we define the *valuation* 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 valuation 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 valuation.

Figure 4.6 illustrates profile intersection and modality evaluation on one simple example. In this case, the system would choose to use the modality that evaluates at 0.65.

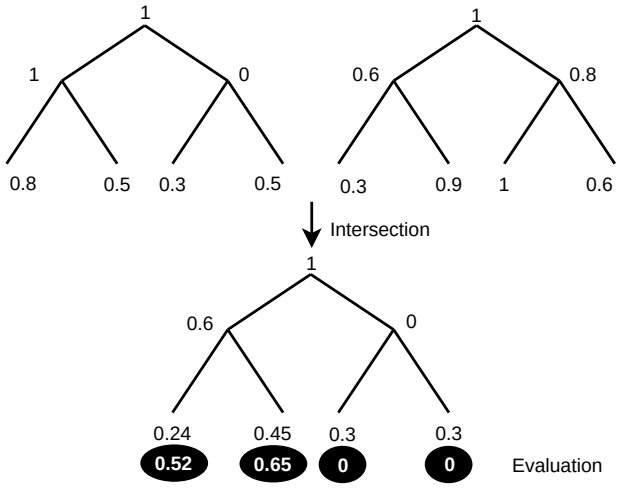

Figure 4.6. Intersection and evaluation.

4.4 Instantiating the Chosen Modality

Once a modality has been selected, the system has to determine values for its attributes. Of course, the weight functions of $p_\cap$ must be taken into account. Moreover, there must be a *global trade-off* between the needs and preferences of *all* the users located at a certain 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.

From here on we will start reasoning at device-level. 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 the*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 valuations 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 $d_1 \times d_2 \times \cdots \times d_p$:

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

As seen in the preceding section, 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 *valuation* of ω and is denoted by $e(\omega)$. For a given combination of attribute values, $e(\omega)$ is the list of valuations of the combination, *starting with the worst valuation.*

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 valuations 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 valuations given by the three users. The results are summarized on Table 4.1.

In Table 4.1, the first column contains the attribute combinations. The next three columns contain the corresponding user valuations, and

ω – **Values**	e_A	e_B	e_C	$e(\omega)$ – **Valuation**
(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 4.1. Formalization of the example situation.

the last column the global valuation 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 valuations are at least 0.7 in this solution).

4.5 Points of View

In the above sections, the profiles are *static*: for instance, a given user can require a minimum font size for text displays, but this size is always the same. However, depending on the distance between the user and the screen, the minimum font size should be different. Texts should be bigger, and similarly, sound should be louder as people are farther away from (respectively) a monitor or a loudspeaker.

For this reason, we introduce the notion of *points of view*. There are two possible points of view:

- *the presentation device's point of view*: constraints on attributes (i.e. weight functions) are expressed with respect to content synthesis on the presentation devices. For instance, in this point of view, font sizes are expressed in pixels or centimeters, because these are the units used by the screen controllers to generate an image. The devices and semantic units' profiles are expressed in this point of view,

- *the user's point of view*: constraints are expressed with respect to what is perceived by the user. For instance, from this point of view, font size are expressed as perceived sizes. Perceived sizes can be expressed as angular sizes (see fig. 4.7). In this way, measures correspond to what users can actually perceive, independently of the distance to the presentation devices. Only the users' profiles are expressed in this point of view.

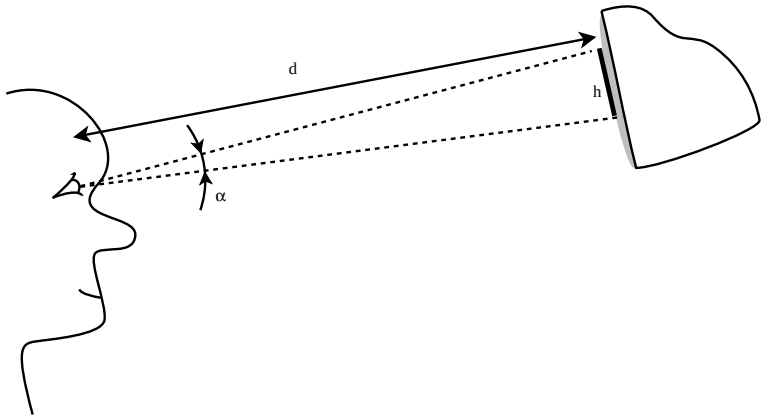

Figure 4.7. α is the angular size of the object of height h located on the screen at a distance d from the user.

As a consequence, the three profiles must be converted to a common point of view. As the point of view which will eventually be used to synthesize a presentation is that of the presentation device, we simply propose to convert the user profile into this one. Let us now see an example of how this works.

Let us convert a font size, expressed as an angular size in the point of view of a user, into a linear size in the point of view of a screen (see fig. 4.7). The user is at a distance d from the screen, the angular size is α and the linear size is h. Then we have:

$$\tan\left(\frac{\alpha}{2}\right) = \frac{h}{2d}$$

Thus, knowing the distance d, it is very easy to translate constraints expressed in the user's point of view into the screen's point of view. Similar formulae can be found for other modalities and quantities. This allows users with sensory impairments to express constraints that will ensure that they can perceive information, regardless of their distance to the presentation devices.

5. Implementation and Evaluation

To evaluate the theories exposed above, we have implemented the models and algorithms, and then used this implementation to carry out experiments with real users.

5.1 PRIAM: A Platform for the Presentation of Information in Ambient Intelligence

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[3].

To design an application with PRIAM, one has to define classes for the agents that will provide information (K), present information (P), and model users (U). This can be done by sub-classing high-level abstract classes, or simply by reusing (or adapting) classes from a library of common entities: user, information screens, simple knowledge sources, etc.

To assess the validity of our approach, we have implemented an on-screen simulator that represents an environment where people and devices are in interaction with each other (fig. 4.8). Proximity relationships can easily be manipulated by dragging-and-dropping objects. This has enabled us to debug and fine-tune our algorithms before conducting pseudo real-scale evaluations.

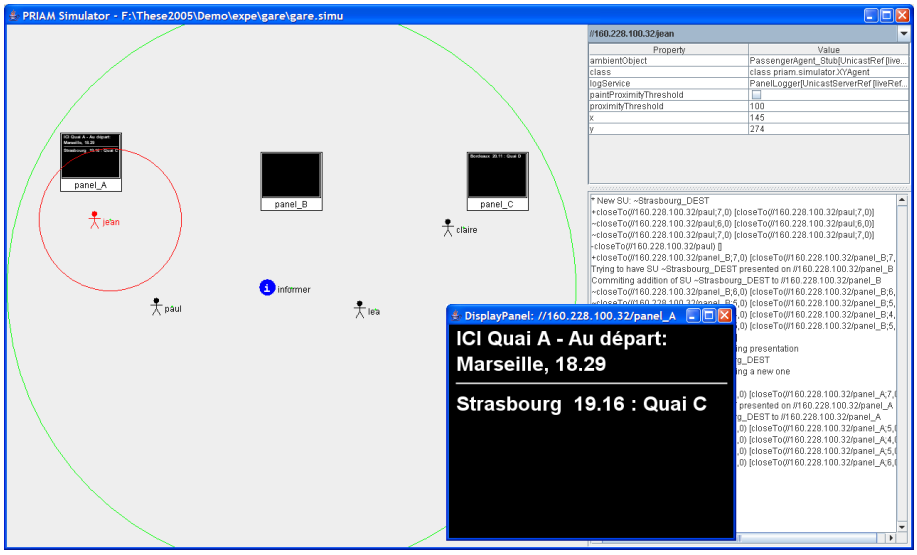

Figure 4.8. Screenshot of the simulator.

The goal of the evaluations is to demonstrate the interest of dynamic information presentation systems for mobile users. They were conducted in our laboratory, with real users. The evaluations are based on screen displays. Proximity among screens and users can be read by sensors

thanks to infrared badges. Other techniques could have been used, such as RFID, but infrared presents a significant benefit: they not only allow the detection of people's proximity, but also of people's orientation. In 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*.

5.2 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. Sixteen subjects had to find an information item among a list of other similar items. We proposed two different tasks: to find an exam results from a list (after sitting for an exam) and to find the details about a flight. We measured the lookup time for each user, with respect to the number of users simultaneously standing in front of the list. There were 1 to 8 simultaneous users (see fig. 4.9), which seems to be realistic of the maximum number of people who can gather around the same display panel.

Figure 4.9. Looking up exam results in a list (in the back) or on a screen (on the left). This is a picture from the experience video.

In control experiments, users were presented with fixed-size dynamic lists, containing 450 examination marks (see fig. 4.10) or 20 flight details. When using the dynamic system, the display panel showed only the information relevant to people standing at proximity (i.e. 1 to 8 items), see fig. 4.11.

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

- as for the exam results (see fig. 4.12), lookup times were 51 % to 83 % shorter (depending on the number of users), and in average 72 % shorter,

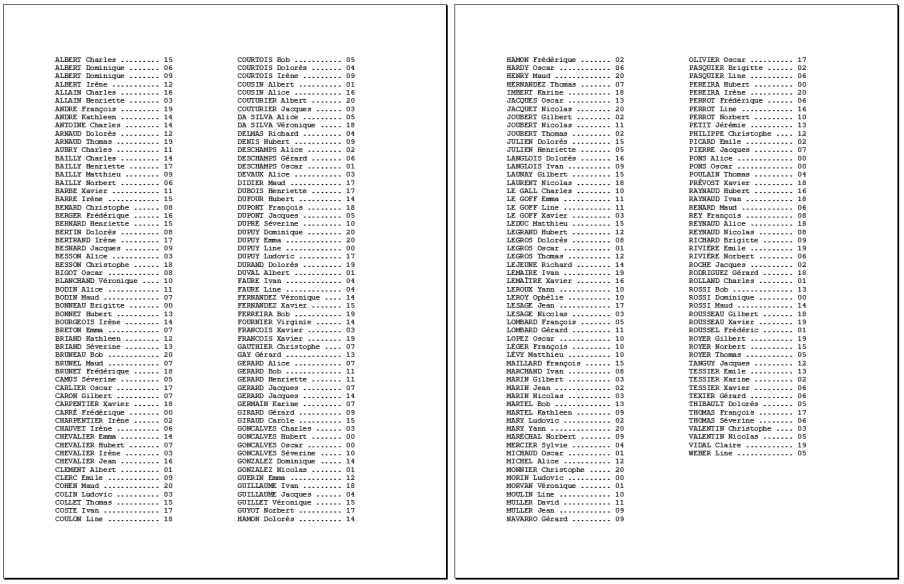

Figure 4.10. A static list of marks, on paper sheets.

Figure 4.11. A dynamic list of marks, displayed on a screen.

- as for flight lookup (see fig. 4.13), lookup times were 32 % to 75 % shorter (depending on the number of users), and in average 52 % shorter.

5.3 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' subways. In a subway, a screen is located near the passageway leading to each platform: it displays the departure times for the trains on that platform.

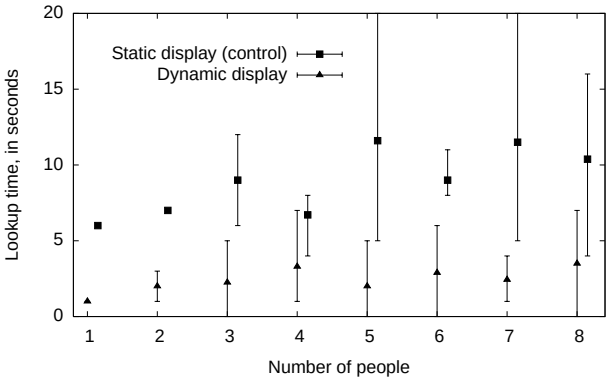

Figure 4.12. Mark lookup time, with respect to the number of simultaneous people. The vertical bars represent standard deviations, the dots average values.

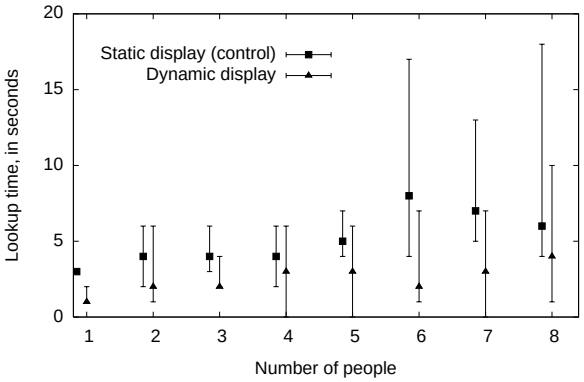

Figure 4.13. Flight information lookup time, with respect to the number of people present simultaneously. The vertical bars represent standard deviations, the dots average values.

However, when a passenger changes trains, he initially has no clue which direction to take, so roughly half of the time, he first walks the whole length of the subway 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 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 assumed that it would help people walk directly to the right platform.

We reproduced a station subway in a corridor of our laboratory. Five display screens represented platform screens. People started from a ran-

dom location in the subway, and had to take a train to a given destination, whose platform was not known by the passengers. When users had found their "platform", they had to raise their hands (fig. 4.14). 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 the *relative length* of the paths.

Figure 4.14. This corridor reproduced an subway in a train station. Display screens were installed at regular intervals, along the wall.

When provided with static information only, people often made mistakes, which resulted in unnecessary moves (table 4.2). 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. Moreover, 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 4.2. Relative lengths when users were provided with static information only.

5.4 Conclusions of 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 on lists. Moreover, providing mobile users with supplementary personalized direction information enables a drastic decrease in the number of unnecessary moves.

However, people were generally disturbed by the items dynamically appearing and vanishing, which caused complete redisplays each time, because the lists were constantly being re-sorted. This problem could be addressed by inserting transitions when adding and removing items, or by inserting new items at the bottom of the lists instead of sorting them. Techniques for automated layout (Lok et al., 2004), and dynamic layout reorganization (Bell and Feiner, 2000) could be investigated.

6. 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 various criteria rather than just alphabetically or in a chronological way. A *level of priority* could thus be associated with each semantic unit. This would 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 or 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.

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. These issues will be the object of future work. A simple workaround would be to display one or two randomly chosen irrelevant items on the screen when only one person is present, thus confusing the malevolent persons. Or instead of displaying relevant items only, we could display all the items and then guide people's gaze to statically displayed items thanks to personal audio clues, in a way similar to the EyeGuide system (Eaddy et al., 2004).

The physical layout of semantic units (i.e. computing the positions of visual units on a screen, scheduling the temporal succession of audio units, etc.) needs to be addressed, and ergonomic considerations need to be taken into account. The algorithm presented in (Zhou et al., 2005) features metrics to coordinate presentations, and is therefore able to enfore ergonomic rules such as ensuring presentation ordering and maintaining presentation consistency. Implementing metrics like these would surely benefit to our system.

Our first experiments took place in *simulated* environments (a room and a corridor in our laboratory). So in the short term, we plan to carry out real-scale experiments, for instance at 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? As we have seen in the experiments performed so far, some people are disturbed by too much dynamicity.
- to study the sociological impact of this system. Does it help people feel at ease when moving around unknown places, or conversely does it infringe on their privacy?
- to test the platform's usability from the point of view of the application designer: is it easy to create an application? what are the guidelines to follow?
- in particular, the problem of assigning the weights for the modalities in the taxonomy needs to be addressed. A multidisciplinary study needs to be performed in order to assess the degree of appropriateness of particular modalities in various contexts.

Notes

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 at an airport, or can change the radiance space of a knowledge source so as to reflect a new organization of the airport.

2. Only these three languages are given weights here because these are the only possible values for the `lang` attribute *in this taxonomy*. Of course, the system can easily support more languages, provided that they are defined in the taxonomy.

3. RMI: Remote Method Invocation.

References

André, E. (2000). The Generation of Multimedia Presentations. In Dale, R., Moisl, H., and Somers, H., editors, *A Handbook of Natural Language Processing*, pages 305–327. Marcel Dekker, Inc. New York.

Bass, L., Faneuf, R., Little, R., Mayer, N., Pellegrino, B., Reed, S., Seacord, R., Sheppard, S., and Szczur, M. R. (1992). A Metamodel for the Runtime Architecture of an Interactive System. *SIGCHI Bulletin*, 24(1):32–37.

Bell, B. and Feiner, S. (2000). Dynamic space management for user interfaces. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 239–248. ACM Press New York.

Berger, S., Kjeldsen, R., Narayanaswami, C., Pinhanez, C., Podlaseck, M., and Raghunath, M. (2005). Using Symbiotic Displays to View Sensitive Information in Public. In *The Third IEEE International Conference on Pervasive Computing and Communications, PerCom 2005.*, pages 139–148.

Bernsen, N. (1994). Foundations of multimodal representations: a taxonomy of representational modalities. *Interacting with Computers*, 6(4):347–371.

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M., and Vanderdonckt, J. (2002). Plasticity of User Interfaces: A Revised Reference Framework. *Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design table of contents*, pages 127–134.

Chou, S.-C., Hsieh, W.-T., Gandon, F. L., and Sadeh, N. M. (2005). Semantic Web Technologies for Context-Aware Museum Tour Guide Applications. In *AINA '05: Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 709–714. IEEE Computer Society.

Coutaz, J. (1987). PAC, an Object-Oriented Model for Dialog Design. In Bullinger, H.-J. and Shackel, B., editors, *INTERACT'87*, pages 431–436. North-Holland.

Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., and Richard, Y. M. (1995). Four easy pieces for assessing the usability of multimodal interaction: the CARE properties. In *INTERACT'95*, pages 115–120.

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

Eaddy, M., Blaskó, G., Babcock, J., and Feiner, S. (2004). My own private kiosk: Privacy-preserving public displays. In *Proceedings of Eighth International Symposium on Wearable Computers (ISWC 2004)*.

Hlavacs, H., Gelies, F., Blossey, D., and Klein, B. (2005). A Ubiquitous and Interactive Zoo Guide System. In *Proceedings of the First International Conference on Intelligent Technologies for Interactive Entertainment (Intetain 2005)*, volume 3814 of *Lecture Notes in Computer Science (LNCS)*, pages 235–239. Springer.

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

Jacquet, C., Bellik, Y., and Bourda, Y. (2006). Dynamic Cooperative Information Display in Mobile Environments. In Gabrys, B., Howlet, R., and Jain, L., 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.

Jacquet, C., Bourda, Y., and Bellik, Y. (2005). An Architecture for Ambient Computing. In Hagras, H. and Callaghan, V., editors, *The IEE International Workshop on Intelligent Environments, IE 2005*, pages 47–54. The IEE.

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

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

Lok, S., Feiner, S., and Ngai, G. (2004). Evaluation of visual balance for automated layout. In *Proceedings of the 9th international conference on Intelligent user interface*, pages 101–108. ACM Press New York.

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

Mansoux, B., Nigay, L., and Troccaz, J. (2005). The Mini-Screen: an Innovative Device for Computer Assisted Surgery Systems. *Studies in Health Technology and Informatics*, 111:314–320.

Pfaff, G. E., editor (1985). *User Interface Management Systems: Proceedings of the Seeheim Workshop*. Springer.

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

Teil, D. and Bellik, Y. (2000). Multimodal Interaction Interface using Voice and Gesture. In Taylor, M. M., Néel, F., and Bouwhuis, D. G., editors, *The Structure of Multimodal Dialogue II*, chapter 19, pages 349–366. John Benjamins Publishing Company.

Tomitsch, M., Grechenig, T., and Mayrhofer, S. (2007). Mobility and Emotional Distance: Exploring the Ceiling as an Ambient Display to Provide Remote Awareness. In *IE 07, the Third IET Conference on Intelligent Environments*, pages 164–167.

Vogel, D. and Balakrishnan, R. (2004). Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *UIST '04*, pages 137–146. ACM Press.

Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84.

Zhou, M. X., Wen, Z., and Aggarwal, V. (2005). A graph-matching approach to dynamic media allocation in intelligent multimedia interfaces. In *Proceedings of IUI '05, the 10th international conference on Intelligent user interfaces*, pages 114–121. ACM.