



HAL
open science

BW-M: A framework for awareness support in web-based groupware systems

Manuele Kirsch-Pinheiro, Marlene Villanova-Oliver, Jérôme Gensel, Hervé Martin

► **To cite this version:**

Manuele Kirsch-Pinheiro, Marlene Villanova-Oliver, Jérôme Gensel, Hervé Martin. BW-M: A framework for awareness support in web-based groupware systems. 2005, pp.240-246. hal-00120284

HAL Id: hal-00120284

<https://hal.science/hal-00120284v1>

Submitted on 13 Dec 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BW-M: A Framework for Awareness Support in Web-Based Groupware Systems

Manuele Kirsch-Pinheiro, Marlène Villanova-Oliver, Jérôme Gensel, Hervé Martin

Laboratoire LSR – IMAG

BP 72 – 38402 Saint Martin d'Hères Cedex, France

{kirsch, villanov, gensel, martin}@imag.fr

Abstract

In this paper, we present a framework for context-based awareness support designed for web-based groupware systems. Nowadays, such systems can be accessed through web-enabled mobile devices, such as PDAs and cellular phones, which allow their users to perform simple tasks, such as contacting colleagues or reading messages. However, these devices present some constraints related to their intrinsic characteristics (reduced display size, limited memory capacity...). These constraints make more acute the need for information filtering. We present a framework for awareness support which is able to adapt the awareness information to the user's context, in order to reduce the risk of information overload. This framework, called BW-M, selects among the available awareness information that which matches the user's, represented through pre-defined profiles, according her/his current context.

1. Introduction

Web-based mobile devices, such as PDAs, cellular phones and even laptops, bring new opportunities for collaborative work, and consequently, for web-based groupware systems. These opportunities come from the fact that users involved in a collaborative task can actually use such devices to work.

Perry *et al.* [12] have studied the behavior of mobile workers. These authors have noticed that such workers need to keep contact with their office when they are involved in some collaborative process (such as, for example, to develop a product or to coordinate a project). They need to be kept aware of any relevant information related to this process (concerning their colleagues activities, the process deadlines, etc.), in order to improve it. In other words, they need some *awareness support*. Awareness in groupware systems refers to cooperating actors taking heed of the context of their joint effort [14]. It refers to the knowledge and the understanding a user has about the group itself and her/his colleagues activities, providing a shared context for individual activities in the group [2].

Many works in the CSCW (*Computer Supported Cooperative Work*) community have stressed the importance of the awareness support for collaborative

work, underlining the problems an inappropriate awareness support may cause (see [3], [6] and [14], as an illustration). One of these problems is referred as the information overload, which arises when users are confronted to a too massive and difficult to understand amount of information [15].

In this paper, we present a framework for awareness support, called BW-M, which is designed to help developers to build web-based groupware systems, particularly those which support asynchronous work and which can be accessed through web-enabled mobile devices. These systems are naturally exposed to the problems these devices may cause (see [4], [13]). When regarding the intrinsic physical constraints of these mobile devices (their reduced display size, limited memory and battery capacities, frequent network disconnection, and so on), we observe that mobile users are more vulnerable to the information overload problem than users in a fixed device (*e.g.* a desktop PC), which do not have to face the same constraints.

We consider a mobile user as someone who disposes of multiple devices, among them mobile devices, and thus is able to alternate between different places and devices. The situation in which the user is currently accessing the system, including the device she/he is using, her/his location, the activities she/he is performing, and many other aspects, can be seen as the context of this user. If information presented to such a user is not related to her/his current context, it can be considered as noisy (inappropriate or disturbing). In fact, users generally use mobile devices to consult some available information which is needed in the current context (*e.g.* when away from her/his office, a user may use her/his PDA to consult only high priority messages related to her/his work group). They typically use these mobile devices in brief time intervals, in order to perform urgent tasks or to consult small, but relevant, amount of information.

All these considerations highlight the necessity of designing new web-based groupware system, which are able to adapt the information delivered to user, in order to match this information to the user's context and to reduce the risk of the information overload problem. One way to perform this adaptation is to select only information which can be considered as relevant for the current user's context (see [1] [10]). We focus on this approach by proposing a context-based filtering process inside the

BW-M framework. This filtering process intends to select relevant awareness information, with regard to the user's context.

We have proposed in [5] a representation of context, described in UML, which differs from those traditionally encountered in context-aware systems (concerned only by the user's *physical context* – device, location, etc. [1] [8]) by taking into account aspects related to the collaborative process (group, roles, activities...). These aspects are very important for mobile users who are involved in such a process.

The filtering process uses this representation, in the one hand, to represent the current user's context, and, on the other hand, to associate an application context to pre-defined profiles. These *profiles* represent the preferences of the user when in a given context. The filtering process selects among the pre-defined profiles those which match the current user's context, and then, it filters the available awareness information according to the selected profiles.

By using this filtering process, the BW-M framework proposes a context-based awareness support, which is an evolution of our previous works [6] [7]. It can be exploited in order to develop new groupware systems, which are concerned with mobile users. We consider that these systems respect a component-based architecture, such as the one proposed by ANTS [9], in which the BW-M represent a component. We propose here this framework by emphasizing its context-aware features.

The paper is organized as follows: first, we present the overall structure of the BW-M framework (Section 2). Then, we present the adopted representation of context (Section 3) and, next, the proposed context-based filtering process (Section 4). Finally, we present how to exploit this framework (Section 5) and present our conclusions.

2. The BW-M framework: overall structure

The BW-M framework adopts an *event-based model*. In this model, all the awareness information delivered to the user is supported by events, which are defined by the system developer. These *events* may contain any useful information (for the group's members) about a specific topic related to the collaborative process. As a result, we are able to describe events that are interesting for users in a nomadic situation, such as the presence of a member of the group in a given location.

The BW-M framework has four main packages: Interface, Storage, Control and Kernel (see Figure 1, more details about these packages can be found in [6]). Each package manages specific and relevant aspects of the awareness support: the *Interface* package delivers the awareness information to the final user; the *Storage* package handles the storage of the awareness information; the *Control* package manages this support by processing the requests performed by the groupware; and the *Kernel* package represents the data model, defining

the awareness information manipulated by the other packages. We have introduced in [7] a fifth package, called *PAM*, for adaptation purposes. This package acts in collaboration with the Interface package supplying users with a progressive access to awareness information, based on the definition of a personalized organization of the data model into successive levels of details

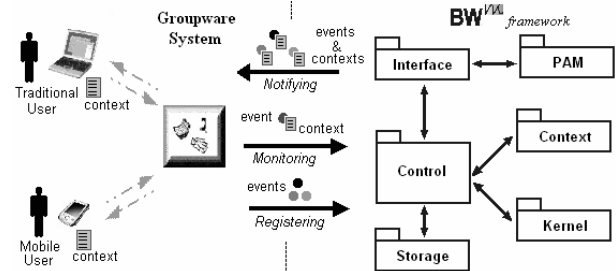


Figure 1. The structure and the work phases of the BW-M framework.

The BW-M framework works in three phases (see Figure 1): Registering, Monitoring and Notifying. Firstly, the groupware registers all the event types that will be monitored (the *Registering* phase). Then, it detects the event occurrence and generates the corresponding objects. Once the events have been produced, the groupware inform the framework, by sending to it these event objects (the *Monitoring* phase). The framework then saves these events, if they correspond to registered event types. At any moment, when a user requests (or the system decides to supply him/her with) awareness information, the groupware can transfer this request to the Control package, which receives and handles it (the *Notifying* phase). This package filters the available events according to the user's preferences, expressed by Profiles (cf. Section 4), delivering the selected events to the user through the Interface package (see also [6] [7]).

In order to perform this selection, we propose a new filtering process, which handles the user's context (see Section 4). This filtering process exploits a representation of context in order to define the relevance of a set of events, and consequently, to determine which events should be delivered to the user at a given moment. The next section presents this representation.

3. A notion of context in an object-oriented view

We have proposed in [5] an object-oriented representation for the notion of context, which focus on the awareness support for mobile users in a Web-based groupware system. This representation differs from other context representations adopted by context-aware systems, which hold only elements related to the user's physical context: her/his location, the characteristics of her/his devices, etc. (see [1] [8]). These elements are very important in order to determine the relevance of a piece of

information for a mobile user. Beside, we claim that other elements are also taken into account. When mobile users are involved in a collaborative process, some information to the group, such as its composition, its activities, its goals, etc., can be considered as relevant for such users, and consequently should be included into the user context description. These elements form what we call the *organizational context*, which refers to the knowledge about the collaborative process in which the user takes part [5].

This knowledge is strongly related to the notion of awareness support, which is intrinsic to the collaborative process. We believe that this knowledge can be very useful in order to determine whenever an information is relevant or not to a specific user. The BW-M framework exploits this knowledge, during the filtering process.

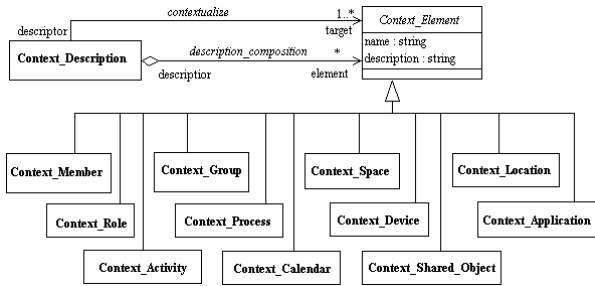


Figure 2. Context description composed by context elements.

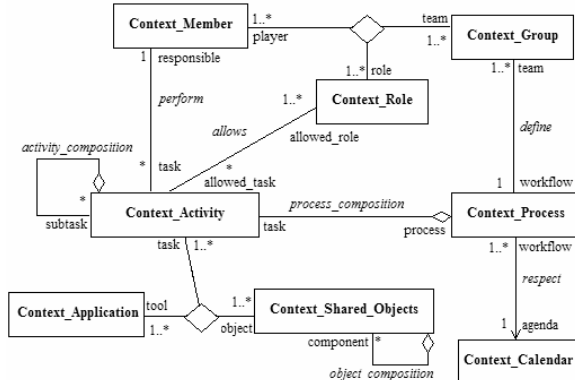


Figure 3. Context elements forming the organizational context.

In this representation (see Figure 2), the concept of context is represented by a class *Context Description*, which is a composition of both the physical and the organizational elements (*Location, Device, Space and Application*, for the former; *Group, Role, Activity, Shared Objects*, etc., for the latter). They are represented by a common superclass, called *Context Element*.

Furthermore, these context elements are related to each other, defining associations between the corresponding concepts (a complete description is given in [5]). Thus, referring only to elements related to the notion of organizational context (see Figure 3), we consider, for instance, that a member belongs to the group through the

roles she/he plays in this group, and that each group defines a process, which is composed of a set of activities (or tasks, also composed by subtasks), and so on.

These classes form the schema of a knowledge base (KB), which allows to describe the context of a user accessing the groupware system. The context of a user (*Member* of a group) is represented by an instance of the class *Context Description*, which is linked by composition to instances of the class *Context Element* (see Figure 2). Additionally, we propose to associate those *Context Description* instances to *profiles* (see Section 4) and to *events*. This last reflects the fact that events should occur in a certain context.

We have implemented this representation using the AROM system [11]. AROM is an object-based knowledge representation system which adopts classes/objects and associations/tuples as main representation entities. We used the Java API supplied by the AROM system, which allows also the manipulation of these objects from any Java application, such as the BW-M framework. This allows the designer of a groupware system to easily adjust this representation to the groupware and its purposes.

The underlying assumption of this representation is that the groupware system detects dynamically the elements of the user's context. For mobile users, it means that the system has to identify the user and the activities she/he is performing, her/his physical location, her/his client device, etc. Sometimes, the system cannot determine each of these context elements. For instance, a system that uses the GPS technology for location detection is unable to determine the context element *Location* for a mobile user who is not equipped with a GPS receiver. The adopted representation (and its implementation with AROM) allows the groupware system to handle these situations by omitting elements from the user's context description (or by leaving attributes unvalued in these elements), representing the fact that the system does not have any or enough knowledge to represent the element (or parts of it).

This ability to omit partially or entirely a context element gives to the groupware system more flexibility when describing the user's context. We are able, for instance, to build a partial *Context Description* instance composed only by instances referring to the organizational context, or/and referring to elements of the physical context (location, device, and application), or/and referring to activities performed by the user without defining their details (an attribute of the *Activity* class).

4. Filtering the awareness information

In order to reduce the risk of information overload for mobile users, the BW-M framework filters the events (which encapsulate the awareness information), according

to the user's context. This filtering process is performed by the *Control* package, during the *Notifying* phase (see Figure 1), and it intends to determine, using the context representation previously presented, which events the user considers as relevant in her/his current context.

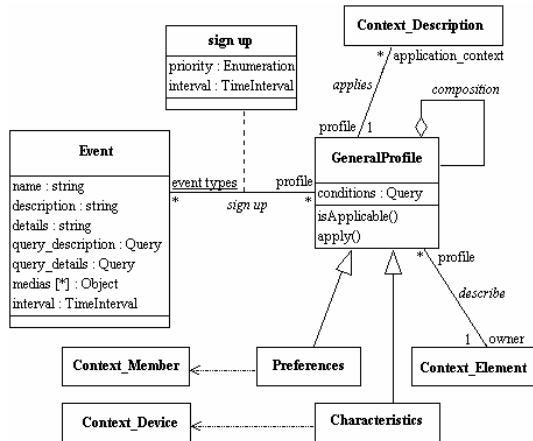


Figure 4. UML schema describing the General Profiles.

This filtering process uses the concept of *general profiles*. General profiles represent the preferences and the constraints the system should satisfy for any given element (a group member, a role, a device...). They can be seen as a set composed by (Figure 4): an owner (for who/what the profile is defined), the application context to be considered, a set of event types to be selected, and a set of conditions to be checked. The application context represents the context in which this profile can be applied, *i.e.* the situations in which the profile is valid (association *applies* in Figure 4). The set of event types is used to indicate which awareness information is considered as relevant, that is, which types of events should be delivered to the owner. A general profile may also indicate a priority order for these event types, as well as a time interval during which their instances are suitable for the owner (see association *sign up* in Figure 4). Finally, a set of conditions, represented by an attribute of the class *GeneralProfile*, refers to the context in which an event takes place (for instance, if the event has been produced in a given location, or if it has handled a shared object).

We assume that each group member should be able to define for herself/himself a set of profiles and circumstances (represented by the application context) in which each profile is valid. In other words, through the profile definition, we allow the user (or the system designer or its administrator) to identify common situations in which she/he access the system, and then to describe which event types (and under which conditions) she/he considers relevant for her/him when she/he finds herself/himself in these situations. In fact, one can observe that mobile users generally access web-based

groupware systems in well defined situations: when traveling (in the airport or in the train), at home (or at the hotel), using his/her laptop, or domestic PC, etc. They rarely access the system from completely different situations. Thus, by defining profiles for these common situations, the users (and the designer) are able to cover most of the use cases and to adapt the system behavior to them.

Using the current user's context, the BW-M framework performs the filtering process in *two main steps*: firstly, it selects all the applicable profiles, and secondly, it combines and applies all these profiles. The first step selects the profiles the framework should apply in order to filter the awareness information. It identifies the event types the user considers as relevant in her/his current situation. The second step consists of the application of the selected profiles, actually filtering the available set of events.

The *first step*, the profile selection, is performed by comparing the content of the Context Description instances of both the active user and the profile. The filtering process compares the application context (a Context Description instance) of the profile with the current user's Context Description instance, by looking for a subset relationship between these instances. As a matter of fact, each Context Description instance forms a graph with the other instances of the KB which are connected to it by the existence of tuples binding them. In this graph, the nodes represent the instances, and the edges, the tuples connecting these instances. The filtering process searches then for a subgraph relationship between these graphs. If the graph representing the application context of the profile is a subgraph of the user's context description, then this profile will be select to be applied in the second step.

Let us consider for instance a team coordinator (named "Alice") who is accessing the groupware from the company central office using her PDA, in order to consult the last information about a report that her group is writing. When Alice logs in the system, it detects her current context, creating then the Context Description instance represented in Figure 5a. Considering that Alice has a profile related to the report activity, this profile will be associated to a context description which contains context elements referring to the coordinator role and to this activity (Figure 5b). As a result, this profile will be selected as applicable because its application context is a subset of the Alice's current context (see Figure 5).

In the *second step*, once all the applicable profiles have been selected, the BW-M framework compares the conditions associated with these profiles to the information carried by the set of available events. Thus, among all events, the framework selects only those which match these conditions. This matching is achieved using a simple algorithm which processes one profile at a time.

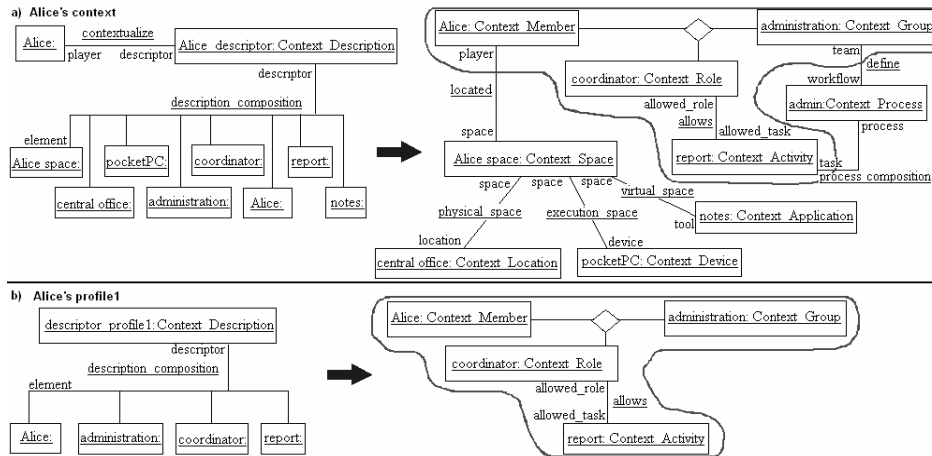


Figure 5. Alice's current context description (a), with the context elements which compose it (on the left side) and the relationships between them (on the right side). In (b), the application context of an Alice's profile and its context elements.

For each selected profile, this algorithm executes the following steps: first, it selects from the available events all events whose type corresponds to a type represented in the profile by tuples of the association *sign up* (Figure 4). If these tuples are also labeled by a time interval, the algorithm restricts the selected events to those which have occurred (or should occur) during this interval. Then, the algorithm applies the conditions, expressed in the profile, related to the context of the event. This means that it checks, for each selected event, whether its context description satisfies these conditions (for instance, if the event was produced in a given location or refers to an activity). If the event satisfies these conditions, it will be delivered to the user. Otherwise, the filtering process will discard it. Using this algorithm, the filtering guarantees that the conditions expressed in each profile will be applied to the corresponding events.

Back to our example, let us consider that Alice has two applicable profiles: one (*profile1*) defining that she is interested in events related to comments about the report made last week (*i.e.* a profile signing up the event type "new comments" with the time interval corresponding to the last week and the condition "related to 'report' activity"), and another one (*profile2*) describing that she wants to know the availability of her colleagues in her current location (*i.e.* a profile which signs up the event type "user availability" and the condition "event location is equal to the current location"). The application of these profiles will cause the selection of all events concerning new comments about the report made last week, and all information about the colleagues which are also located in the central office. Any event concerning her colleagues located in other locations, neither comments about other documents (or documents concerning other activities than the report) will be discarded by the filtering process.

At the end of this filtering process, the Control package will retain a reduced set of events, which is delivered to the Interface package. At its turn, this package may apply other adaptation methods, particularly the one defined in [7].

5. Exploiting the BW-M framework

The BW-M framework, like any framework, must be connected to a system to be functional. In order to use the BW-M framework, a designer, when developing a web-based groupware system, must implement some "entry points", which should be connected to the groupware system and cope with its purposes. There are three main entry points: the events, the interface and the storage.

First of all, the system designer has to establish the set of possible *events*. This entry point consists in determining which information could be (or is important to be) delivered to the users of the groupware system, *i.e.* the information about the collaborative process should compose each *event type*. For instance, let us consider the design of a web-based groupware system which supports collaborative editing and asynchronous communication. Such a system may represent as events situations like the modifications of a document, comments about it, or the availability of the users, by defining (and registering) three types of events.

In addition, the system designer should determine whether and how the users will be able to define their own profiles. We strongly suggest to the system designer to propose a set of default profiles which could be modified by the users, according to their needs and interests. For example, the system previously mentioned may allow the users to define their own profiles, allowing then someone to create a profile which signs up the first event type (related to the modification of the document), ignoring other event types, and also restricting these

events to those performed by a given user and during a given period.

The second entry point is the *interface*. In fact, the system designer should connect the Interface package of the BW-M framework to the groupware interface. She/he should do so by implementing the abstract classes that compose the Interface package, probably connecting this package to other components of the groupware system, particularly those for adaptation purposes.

Finally, the third entry point refers to the storage medium used by the BW-M framework to store the awareness information. The Storage package contains an abstract class which should be implemented in order to determine the storage medium really used by the groupware system. This way, the designer is free to choose how and where the awareness information will be stored. An option is to use the AROM system for this purpose, since the adopted context representation (cf. Section 3) has many common elements with the Kernel package, which describes the data model used by the BW-M framework (see [5] and [6]). These common elements (*Member*, *Group*, *Role* and *Activity*) refer to elements also represented in the organizational context. Thus, the system designer may use the same KB which contains the context representation to keep also the elements of the Kernel package, which includes events and profiles.

6. Conclusions

In this paper, we have presented a framework for context-based awareness support which uses an object oriented representation of context in order to perform a filtering process, which takes into account the user's context in order to better filter the available awareness information. This framework was designed for Web-based groupware systems with the main objective of reducing the risk of information overload for mobile users by delivering them only most relevant information for their current context.

We have implemented this framework using Java, which facilitates the integration of the adopted context representation (cf. Section 3). We are now studying, based on some previous experiences [6], how to encapsulate this framework in a web service, in order to facilitate its use on web based groupware systems. In fact, even if there are only a few entry points to be checked by the designer, she/he still has to learn how to use the framework. By proposing a web service with its functionalities, we expect to reduce this learning phase.

We also intend to extend the filtering process by refining its first step (cf. Section 4). We are interested in defining a method which calculates an acceptable semantic distance between the objects, in order to check if they are sufficiently similar (and not necessarily equal) in

order to establish a subgraph. We are also interested in improving the profile structure in order to reduce the problems caused by an incorrect profile definition (for instance, the definition of too strict profiles, whose context description includes many different elements, may lead to the selection of none, but also the selection of too many profiles may lead to a similar situation). We intend to improve particularly the conditions expressed in the profile, given to the user and to the system designer more flexibility. Finally, we are studying the possibility to include in the profiles elements related to a user model, such as the proposed in [15].

7. References

- [1] J. Burrell, G.K. Gray, K. Kubo, N. Farina, "Context-aware computing: a text case". G. Borriello, L.E. Holmquist, (Eds.), *Ubicomp 2002, LNCS 2498*, Springer, 2002, pp. 1-15.
- [2] P. Dourish, V. Bellotti, "Awareness and Coordination in Shared Workspaces", *ACM Conference on Computer-Supported Cooperative Work*, ACM Press, 1992, pp. 107-114.
- [3] A. Espinosa, J. Cadiz, L. Rico-Gutierrez, R. Kraut, W. Scherlis, G. Lautenbacher, "Coming to the wrong decision quickly: why awareness tools must be matched with appropriate tasks", *CHI Letters, CHI 2000*, vol. 2, n° 1. ACM Press, 2000, pp. 392-399.
- [4] J. Jing, A. Helal, A. Elmagarmid, "Client-server computing in mobile environment", *ACM Computing Surveys*, vol. 31, n° 2, ACM Press, 1999, pp. 117-157.
- [5] M. Kirsch-Pinheiro, J. Gensel, H. Martin, "Awareness on Mobile Groupware Systems", A. Karmouch, L. Korba, E.R.M. Madeira (Eds.), *MATA 2004, LNCS 3284*, Springer, 2004, pp. 78-87.
- [6] M. Kirsch-Pinheiro, J.V. Lima, M.R.S. Borges, "Framework for Awareness Support in Groupware Systems", *Computers in Industry*, vol. 52, n° 3, Elsevier, 2003, pp. 47-57.
- [7] M. Kirsch-Pinheiro, M. Villanova-Oliver, J. Gensel, J.V. Lima, H. Martin, "Providing a progressive access to awareness information", R. Meersman et al. (Eds.), *CoopIS/DOA/ ODBASE 2003, LNCS 2888*, Springer, 2003, pp. 336-353.
- [8] T. Lemlouma, N. Layaïda, "Context-Aware Adaptation for Mobile Devices", *IEEE International Conference on Mobile Data Management*, IEEE Computer Society, January 2004, pp.106-111.
- [9] P.G. Lopez, A.F.G. Skarmeta, "ANTS Framework for cooperative work environments", *IEEE Computer*, vol. 36, n° 3, IEEE CS, March 2003, pp. 56-62.
- [10] M.A. Muñoz, M. Rodriguez, J. Favela, A.I. Martinez-Garcia, V.M. Gonzalez, "Context-aware mobile communication in hospitals", *Computer*, vol. 36, n° 9, IEEE CS, 2003, pp. 38-46.
- [11] M. Page, J. Gensel, C. Capponi, C. Bruley, P. Genoud, D. Ziébelin, D. Bardou, V. Dupieris, "A New Approach in Object-Based Knowledge Representation: the AROM System", *14th Int. Conf. on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, (IEA/AIE2001), LNAI 2070*, Springer, 2001, pp. 113-118.
- [12] M. Perry, K. O'Hara, A. Sellen, B., Brown, R. Harper, "Dealing with mobility: understanding access anytime, anywhere", *ACM Transactions on Computer-Human Interaction*, vol. 8, n° 4, ACM Press, 2001, pp. 323-347.
- [13] B.N. Schilit, J. Trevor, D.M. Hilbert, T.K. Koh, "Web interaction using very small Internet devices", *Computer*, vol. 35, n°10, IEEE CS, 2002, pp. 37-45.
- [14] K. Schmidt, "The problem with 'awareness': introductory remarks on 'Awareness in CSCW'", *Computer Supported Cooperative Work*, vol. 11, n° 3-4, Kluwer Academic Publishers, 2002, pp. 285-298.
- [15] M. Villanova, J. Gensel, H. Martin, "A progressive access approach for web based information systems", *Journal of Web Engineering*, vol. 2, n° 1 & 2, 2003, pp.27-57