# Context-Aware Filtering for Collaborative Web Systems: Adapting the Awareness Information to the User's Context

Manuele Kirsch-Pinheiro
LSR Laboratory - IMAG
BP 72 – 38402 St.M. d'Hères
France
+33 4 76 82 72 11
Manuele.Kirsch-Pinheiro@imag.fr

Marlène Villanova-Oliver
LSR Laboratory - IMAG
BP 72 – 38402 St.M. d'Hères
France
+33 4 76 82 72 80
Marlene.Villanova@imag.fr

Jérôme Gensel
LSR Laboratory - IMAG
BP 72 – 38402
St.M. d'Hères - France
+33 4 76 82 72 80
Jerome.Gensel@imag.fr

Hervé Martin
LSR Laboratory - IMAG
BP 72 – 38402
St.M. d'Hères - France
+33 4 76 82 72 80
Herve.Martin@imag.fr

## ABSTRACT
We propose a context-based filtering process which aims at adapting the awareness information delivered to mobile users by collaborative web systems. This filtering process relies on a model of context which integrates both a physical and an organizational dimensions and allows to represent the user's current context as well as *general profiles.* These profiles are descriptions of user's potential contexts and express the awareness information filtering rules to apply when the user's current context matches one of them. These rules reflect the user's preferences given a context. We describe how the filtering process performs in two steps, one for identifying the general profiles that apply, and a second for selecting the awareness information. We also discuss the patterns matching algorithms used in the filtering process to compare the contexts descriptions.

## Categories and Subject Descriptors
H.4.1 [**Information System Application**]: Office Automation – *gropware* H.5.3 [**Information Interface and Presentation**]: Group and Organization Interface – *computer-supported cooperative work, collaborative computing.*

## General Terms
Algorithms, Management, Design.

## Keywords
User adaptation, context-aware computing, collaborative web systems, awareness support.

## 1. INTRODUCTION
The introduction of new web-enabled mobile devices, such as laptops, PDAs and cellular phones, entails more flexibility for mobile users who may easily access collaborative web systems from any place using these devices. Nevertheless, these mobile devices, despite their evolution, still have some limitations. We may cite, for instance, their reduced display and memory capacities or their limited wireless bandwidth and battery life [5]. In addition, the circumstances under which users access those systems are constantly changing: users move, changing their physical location, they use different devices and they are involved in various collaborative processes. For these reasons these systems should now be able to adapt the delivered information (and the services they offer) to the users according to their context of work.

Recently, some works have been proposed in this direction (*e.g.* [9][11]). However, these works try to adapt the information delivered to the user by selecting or transforming its content, according the user's physical context (*i.e.* her/his current location, device, etc.). They give a limited importance to the user's preferences and to the collaborative processes in which she/he participates.

In this paper we propose a new approach relying on a *context-based filtering* process which is based on *general profiles* which describe both the user's current context and her/his preferences. We adopt an object-oriented modeling of the user's context [6], which takes into account both the user's physical and organizational contexts. The latter refers to the knowledge about the collaborative process in which the user is involved, including concepts like groups, roles, activities, etc. We consider that this knowledge should be part of a mobile user's context since this user, when accessing collaborative web systems, is also concerned by some collaborative process. In our approach, we specially focus on collaborative web systems which support asynchronous work, such as BSCW [1] and Toxic Farm [14]. These systems are more and more accessed through mobile devices, and we believe that delivering an informational content adapted to the user's context and to her/his preferences may help her/him to optimize her/his work, and consequently, the entire group's work.

We propose here a context-based filtering process, which first selects, according to the current user's context, the predefined user's preferences for this context, and then filters the available information according to these preferences. We show how this process can be used by the awareness support component of collaborative web systems[1]. This component handles the


*SAC'05*, March 13-17, 2005, Santa Fe, New Mexico, USA.


[1] We consider systems which are architecturally made of components, such as communication or concurrency control, as proposed by the ANTS framework [10]. The awareness support is one component of this architecture.

*awareness information*, which stands for the knowledge a user has about the group itself and her/his colleagues' activities, providing this way a context for individual activities. This context is used to ensure that individual contributions are relevant to the group's activity as a whole and to evaluate individual actions with respect to the group's goals and progress [3]. We assume that our context-based filtering process is embedded in such a component for performing the filtering of the awareness information before sending it to the user.

This paper is organized as follows: in Section 2 we introduce the context-based filtering process. We present the model of context we adopt and describe in details the two steps that compose the process. In Section 3, we show some preliminary results. We compare our proposition with other related works, in Section 4, before we conclude.

## 2. CONTEXT-BASED FILTERING

The context-based filtering process we propose is based on an object-oriented model of context (cf. Section 2.1). This model is used to represent, on the one hand, the real *current user's context* and, on the other hand, *general profiles*. A general profile is the description of a potential context that might characterize the user's real situation and expresses filtering rules that should apply when this happens (i.e. when the user's current context matches the general profile context description (see Section 2.2)). The filtering rules reflect the user's preferences considering the context associated with the general profile (for instance, what is the awareness information she/he wants to be informed of). Please note that we assume that awareness information is represented by events following the model proposed by [7]. Basically, an event describes some information related to a given process which can be useful for the collaborative process. The filtering consists in selecting relevant event instances from the available set of events generated by the system (see Section 2.3).

### 2.1 Representing the User's Context

A collaborative web system, in order to exploit the user's context, has obviously to represent somehow this notion of context. In this work, we adopt an object-oriented model [6] using a UML schema in which classes represent both the user's physical context (*location, device, application*) and the user's organizational context (*group*, *role, member*, *calendar*, *activity*, *shared object* and *process*). The Figure 1 shows how the concept of context we use is represented by a class *context description* which is a composition of both physical and organizational concepts mentioned above. These concepts are represented through a common superclass, called *context element*.

The Figure 2 focuses on the classes related to the user's organizational context (*group*, *role, member*, *calendar*, *activity*, *shared object* and *process*) and their relationships. This view allows us to show that each element of context is not an isolated information but does belong to a more complex representation of the user's situation. We claim that these concepts and their relations bring some knowledge that can help to determine the relevance of an event for a mobile user. For instance a user may be interested in some awareness information about an activity *only if* it is performed by a given colleague of her/his before a certain date.

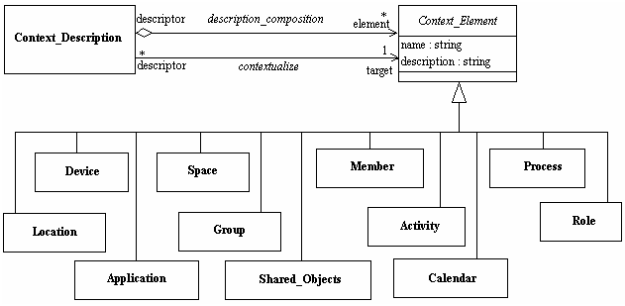


**Figure 1. The composition of the context description**

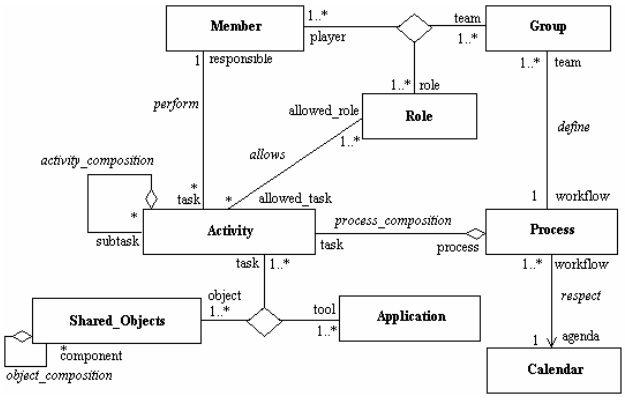


**Figure 2. Elements of the user's organizational context.**

We have implemented the context model using the java API of the AROM system [12]. AROM is an object-based knowledge representation system, which adopts classes/objects and associations/tuples as main representation entities. Using AROM, we created a knowledge base (KB) whose content are the classes and the instances of this UML schema.

This KB is populated by three kind of knowledge. First, the classes and associations related to the system and the working environment are defined and instantiated. For instance, the process is defined as well as its component activities, the group's members (i.e. users), the application (services) the system offers, etc. This knowledge constitutes the awareness information basis. Second, the KB also stores the descriptions of potential contexts associated with general profiles established for the different users. These descriptions represent the situations in which a general profile is valid (i.e. should be used to filter the available events if the user's current context description matches). Third, the KB also keeps the instances of context description which represent the current context of the active users. These instances represent a knowledge dynamically updated by the system according to each user's behavior, and which can be discarded once the user is no longer active.

It is worth noting that the AROM system gives us some interesting advantages: its Java API allows an application to handle and to modify a KB during the execution time. Thus, a collaborative system may adjust the KB by creating new

instances, by modifying existing ones, or even by introducing new classes, associations or attributes. This allows the system, for instance, to dynamically register the battery level of a device.

## 2.2 Step 1: Profile Selection

The proposed filtering process is based on the concept of *general profiles*. We have previously explained that a description of some potential context is associated with such a general profile in order to represent a situation which can be encountered by the user and thus to apply adequately a filtering process. Please note that general profile do not only concern users, but aim at representing the preferences and the constraints the system should satisfy for any given context element (user, group, role, device…).

For mobile users and group roles, this concept specializes in *preferences* (see Figure 3), describing the preferences of the user or her/his role concerning the informational content that should be delivered by the system. For devices, it specializes in *characteristics*, describing the capabilities of the referred device (similar to the profile schemas defined by [9]).

We assume that each user (or the system designer) may define several profiles and the situations in which they are valid (i.e. the description of the potential context, called the *application context*). This means that each user may define what information is relevant to her/him and under which circumstances. Hence, we define the general profiles as a set of the components (see Figure 3): 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.

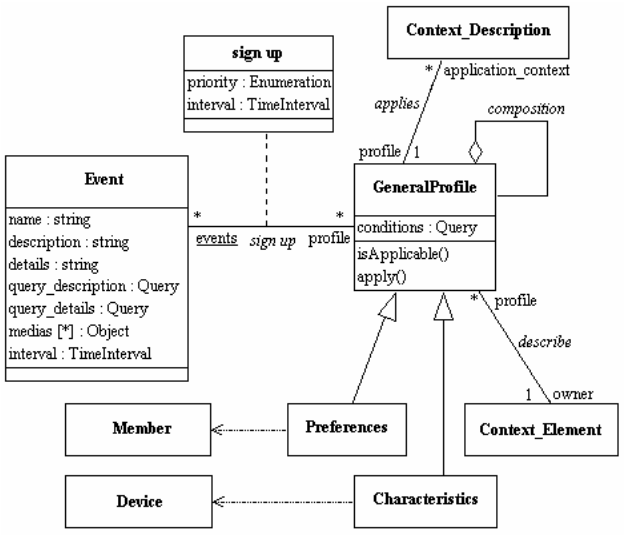


**Figure 3. UML schema describing the General Profiles.**

General profiles are also classes of the knowledge base (see Section 2.1), and they are associated with the context description class (as the *application context* is an instance of *context description*). In addition, it is worth noting that each general profile may have multiple application contexts, *i.e.* multiple situations in which the profile is valid. The set of event types that composes each profile is used to indicate what informational content is considered as relevant, that is, what types of events should be delivered to the owner (see association *sign up* in Figure 3). General profile may also indicate a priority order for these event types, as well as a time interval in which their instances are suitable for the owner. Finally, a set of conditions related 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 given shared object) is represented as an attribute of the class *GeneralProfile*.

These elements (i.e. the set of event types with the associated priority order, time interval and conditions) constitute the 'rules' of each general profile. These rules are the ones applied in the second step of the filtering process (cf. Section 2.3).

The first step of the proposed filtering process consists in selecting the general profiles which are valid with regard to the user's current context. This selection is performed by comparing the *application context* related to the available user's general profiles with the user's current context. Please note that these two kind of context are both instances of the class context description in our model. For each general profile, we test if one of its *application contexts* has the same content or is a subset of the *user's current context description*. If the test is positive, then the profile is selected to be applied. In order to perform this subset relationship, we consider that each *context description* instance together with its components context elements instances define a graph, where the nodes represent the instances and the edges between them represent the tuples of associations involving these instances. Thus, a context $C$ is a sub-context of a context $C'$ whenever the graph corresponding to $C$ is a subgraph of the graph corresponding to $C'$.

The subgraph relationship is established using a quite simple pattern matching algorithm. This algorithm is based on the operations *equals* and *contains*: *i)* A node N is considered as *equal* to a node N' if the instances that define them have the same values for the same variables. *ii)* An edge E is *equal* to an edge E' if they connect nodes that are equal. And *iii)* a node N *contains* a node N' if, for each edge E' connecting N' to a node N", there is an edge E connecting N that is equal to E'. Then, a graph C is a subgraph of C' if the latter contains the former.

As an illustration, let us consider a team coordinator ("Alice") that is accessing a web-based system which supports shared repository, collaborative editing and asynchronous communication. She is accessing the system from the company central office using her pocket PC, in order to consult the latest notes about a report that her group is writing. When she asks for these notes, the context description instance that represents her current context is the one represented in Figure 4.

Considering that Alice has two available profiles, one related to the report activity and another related to her personal office, these profiles are associated to the context description instances represented in the Figure 5, which contain, for the first profile, context elements referring to Alice's role and to the 'report' activity (instances of role and activity classes, respectively), and for second profile, context elements related to Alice's office and desktop PC (instances of location and device classes).

The selection process will compare these instances of context description with the one currently related to Alice. It will select only the first profile, because its context description is a subset of Alice's current context, as we can see in Figure 6. In fact, all

instances belonging to the context description of the first profile have equal instances into Alice's current context. In the other hand, there are instances in the context description related to the second profile that are not present in Alice's context: the instances representing Alice's office and her desktop PC have no equal instances in the Alice's current context (the *location* instance 'central office' differs from 'Alice's office', and the *device* instance 'pocketPC' differs from 'desktopPC'). That is why the second profile is not selected in this first phase of the proposed filtering process.

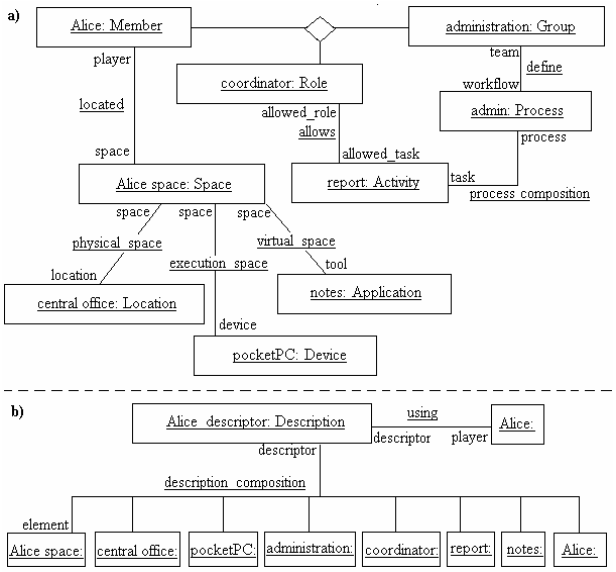


**Figure 4. The context description related to the user Alice's current context (b) and the associations among the elements of this description (a).**

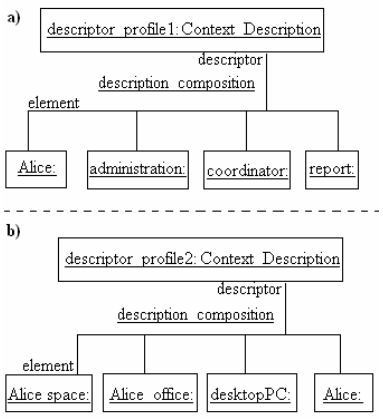


**Figure 5. The context descriptions related to two different instances of General Profile.**

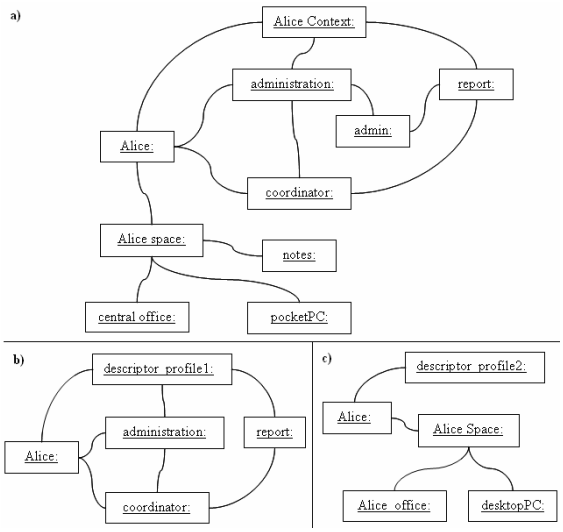


**Figure 6. Graphs generated by the context description instances associated to Alice's current context (a) and her profiles (b) (c) .**

## 2.3 Phase 2: Filtering Events

Once all the applicable profiles have been selected, the second step of the filtering process compares the criteria defined in these profiles (event type, time interval and context conditions) to the information carried by the available events. Thus, among all events, the process selects only those which correspond to these criteria. In other words, this step applies the 'rules' defined in each selected general profile to the set of available events.

As we stated before, events carry the awareness information that can be delivered to any user. Each event represents a set of useful information about a topic, which can be statically defined by the system when generating the event instance, or dynamically defined through queries in the knowledge base. We have defined a basic *Event* class which contains some attributes that we consider as primordial (see Figure 7): an event name, a description, some details about it, a time interval in which it occurs (or has occurred), some media describing its content, and two query string, which can be used to dynamically capture content for the 'description' and 'details' attributes. We also consider the event as referring to one or more elements of the knowledge base, since it carries on some information about a topic. In addition, each event instance is associated with a context description instance, representing the context in which the event has been (or should be) produced.

Therefore, the event filtering is achieved as follows: for each selected profile, the algorithm selects from the set of available events all the events whose type corresponds to a type signed up by the profile. Then, it restricts the selected events to those which have occurred (or should occur) in the interval indicated by the profile. Next, the algorithm applies the set of conditions related to the context of the event expressed in the profile. 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 handles a shared object). Finally, the algorithm orders

all the events that satisfy all these criteria according to the priority order defined in the profile.

To illustrate, considering the user Alice, we assume that her selected profile (the one related to the 'report' activity) includes the event types "new comments" and "modified document", with a time interval corresponding to the last week and, as condition, the fact of handling the shared object "report04.html". In this case, after the execution of this filtering second step, Alice will receive all events that have occurred last week and that describe new comments about or modifications of the document "report04.html" (i.e., events whose context descriptions include the instance of shared object representing this document).

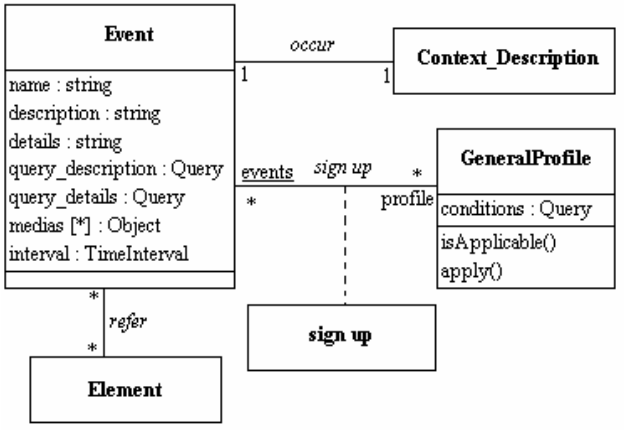


**Figure 7. UML schema describing the Event class.**

At the end of the proposed filtering process, an ordered (by priority) set of selected events will be available for delivering to the mobile user. This set will probably better suit the current mobile user's context since it accords the user's profiles defined for this context. Additionally, this set will have fewer elements than the available set of events, reducing the risk of an information overload to the user. However, before delivering the set of selected events, we strongly suggest to apply over this resulting set, other adaptation algorithms, such as those proposed by [9], [13] or by [15], mainly in order to adapt the presentation of the selected events to the mobile device.

## 3. PRELIMINARY RESULTS

We have implemented the proposed filtering process using the AROM system and the awareness model proposed by the BW framework [7]. We have built a knowledge base in which we keep the instances of the context model as well as the profiles and the events. We have performed some tests simulating some situations encountered when using a collaborative web system which proposes shared repository and synchronous/asynchronous communication features. The simulation uses five users and fifteen profiles definitions, and we have evaluated different types of pattern matching algorithm acting on the *equals* and *contains* operations. We have employed two versions of the *equals* operator (one that considers only perfect match – objects must be exactly equal – and another which allows to define a minimum percentage of similar attributes in the objects), and two versions of the *contains* operator (one that compares only equal instances, and another that compares the graph from two instances, even if they are not equal).

These tests have showed the validity of our filtering process with regard to a user's current context. They have also demonstrated some critical points. First, using distinct versions of the *equals* and *contains* operators presents different results: the most satisfactory came from the most flexible versions (*equals* with predefined limit, and *contains* with the comparison of different instances). Then, the definition of the *application context* related to the general profile is more or less critical given the version of the operator. In fact, defining a detailed application context causes the non selection of the profile in most cases when using the first version of the operators. On the other hand, defining a too reduced application context causes its selection in almost all cases, especially when using the second version of the operators. As a result, in the former case, we have fewer selected events than expected, since we have fewer selected profiles to filter the events. In the latter case, mobile users will risk to be overloaded by the events, since we will probably select more events by applying more profiles.

## 4. RELATED WORKS

Regarding the context-aware computing literature, many works propose to adapt the information presented to a mobile user according to her/his context. We may enlighten the propositions made by [9], [13] and [2]. The first two works propose to adapt the presentation of the delivered content according to the physical capabilities of the client device, whereas the latter proposes to adapt it by selecting the information according to the user's location. These works, as the majority of context-aware systems, adopt a representation of context which is limited to the user's physical context. Our proposition differs from these works by adopting a more complete representation of context, which takes also into account the user's organizational context. This is interesting for users of collaborative web systems, who are involved in some collaborative process. We are not concerned about the adaptation of the presentation, which is the case of [9] and [13].

Further, [11] presents a notion of context which includes some concepts related to the user's organizational context, particularly the role played by the user inside the collaborative environment. The authors propose a contextual instant messaging system, which allows users to specify a set of circumstances for a message to be delivered. Unfortunately, concepts like the activities and the collaborative process are not considered, and no formal model of the adopted notion of context is given. Additionally, authors do not propose any mechanism to allow a user to define what kind of messages she/he wants to receive, which is the case of the general profiles described here.

Finally, considering works which deal with awareness support in mobile devices, we may enlighten the system proposed by [4]. This system intends to provide means for opportunistic communication by integrating awareness information into mobile user's messages. This work differs from our by its notion of awareness, which is limited to the information about colleagues and their availability (corresponding to the notion of "group awareness" given by [8]). We do not have this limitation, adopting a larger notion of awareness, defined by [3], near to the

notion of "contextual awareness" given by [8]. Moreover, this system does not have any filtering feature, even if these authors highlight its need [4]. We consider it then as an example of system which may benefit from the proposed filtering process.

## 5. CONCLUSIONS

In this paper, we have presented a context-based filtering process which proposes to adapt the information delivered to mobile user by filtering it according to the current user's context. This process adopts an object-oriented representation of context, which has been implemented using the AROM system [6]. We have performed some preliminary tests using this knowledge base and the awareness model proposed by the BW framework [7]. These tests have showed the validity of this filtering process and how it can be adapted to different situations, and consequently, to different collaborative web systems.

We expect to extend the proposed filtering process by refining the pattern matching algorithm used to compare instances of the context description class. We are interested in defining different similarity measures, calculating more precisely the semantic distance between two instances, according to the system needs (*e.g.* what is considered by the system as an acceptable distance for a given class). In addition, we are also interested in refining the general profile definition, looking for a definition that allows a more powerful selection of the events, as well as mechanisms to define these profiles automatically, maybe through the analysis of the (evolution of) the user's context.

## 6. REFERENCES


[1] Appelt, W. What groupware functionality do users really use? Analysis of the usage of the BSCW system. *Proceedings of 9th Euromicro Workshop on PDP 2001*. http://bscw.gmd.de/Papers/PDP2001/PDP2001.pdf

[2] Burrell, J., Gray, G.K., Kubo, K., Farina, N. Context-aware computing: a text case. *Proceedings of Ubicomp 2002, LNCS 2498*. Springer-Verlag, 1-15

[3] Dourish, P., Bellotti, V. Awareness and Coordination in Shared Workspaces. *Proceedings of ACM Conference on Computer-Supported Cooperative Work* (CSCW'92). ACM Press, 107-114.

[4] Hibino, S., Mockus, A. handiMessenger: awareness-enhanced universal communication for mobile users. *Proceedings of Mobile HCI 2002, LNCS 2411*. Springer-Verlag, 170-183

[5] Jing, J., Helal, A.S., Elmagarmid, A. Client-server computing in mobile environments. *ACM Comp. Surveys, 31, 1* (Jun. 1999), 117-157.

[6] Kirsch-Pinheiro, M., Gensel, J., Martin, H. Representing Context for an Adaptative Awareness Mechanism. *Proceedings of the X International Workshop on Groupware* (CRIWG'04), *LNCS 3198* (San Carlos, Costa Rica, Sept 5-9 2004) Springer-Verlag, 339-348

[7] Kirsch-Pinheiro, M., Lima, J.V., Borges, M.R.S. Framework for Awareness Support in Groupware Systems. *Computers in Industry, 52, 3* (Sept. 2003) 47-57

[8] Liechti, O. Awareness and the WWW: an overview. *Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW'00), Workshop on Awareness and the WWW*. http://www2.mic.atr.co.jp/dept2/awareness/

[9] Lemlouma, T., Layaïda, N. Context-aware adaptation for mobile devices. *Proceedings of the IEEE International conference on Mobile Data Management* (Berkeley, CA, USA, January 19-22, 2004). IEEE, 106-111.

[10] Lopez, P.G., Skarmeta, A.F.G. ANTS Framework for cooperative work environments. *IEEE Computer, 36, 3* (March 2003), 56-62.

[11] Muñoz, M., A., Rodríguez, M., Favela, J., Martinez-Garcia, A.I., Gonzalez, V.M. Context-aware mobile communication in hospitals. *IEEE Computer, 36, 9* (Sept. 2003), 38-46.

[12] Page, M., Gensel, J., Capponi, C., Bruley, C., Genoud, P., Ziébelin, D., Bardou, D., Dupierris, V. A New Approach in Object-Based Knowledge Representation: the AROM System. *Proceeding of the 14th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems* (IEA/AIE 2001), LNAI 2070. Springer-Verlag, 113-118.

[13] Schilit, B.N., Trevor, J., Hilbert, D.M., Koh, T.K. Web interaction using very small Internet devices. *IEEE Computer, 35, 10* (Oct. 2002), 37-45.

[14] Skaf-Molli, H., Molli, P., Oster, G., Godard, C. Toxic farm: a cooperative management platform for virtual teams and enterprises. *Proceedings of 5th International Conference on Enterprise Information Systems* (ICEIS'03) (Angers, France, April 2003). http://www.loria.fr/~molli/rech/iceis03/

[15] Villanova, M., Gensel, J., Martin, H. A progressive access approach for web based information systems, *Journal of Web Engineering (JWE), 2, 1 & 2* (2003). 27-57.