HAL
open science

Providing a Progressive Access to Awareness Information

Manuele Kirsch-Pinheiro, Marlene Villanova-Oliver, Jerôme Gensel, José Valdeni de Lima, Hervé Martin

To cite this version:

Manuele Kirsch-Pinheiro, Marlene Villanova-Oliver, Jerôme Gensel, José Valdeni de Lima, Hervé Martin. Providing a Progressive Access to Awareness Information. 2003, pp.336-353. hal-00120272

HAL Id: hal-00120272

https://hal.science/hal-00120272v1

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.

Providing a Progressive Access to Awareness Information

Manuele Kirsch Pinheiro[1], Marlène Villanova-Oliver[1], Jérôme Gensel[1], José Valdeni de Lima[2], and Hervé Martin[1]

[1] Laboratoire LSR – IMAG
BP 72 – 38402 Saint Martin d'Hères Cedex, France
{Manuele.Kirsch-Pinheiro, Marlene.Villanova, Jerome.Gensel, Herve.Martin }@imag.fr
[2] Instituto de Informática – UFRGS
CP 15064 – Av. Bento Gonçalves 9500, Campus do Vale, Bl. IV – Porto Alegre, Brazil
valdeni@inf.ufrgs.br

Abstract. Awareness has now become a widely accepted central feature in groupware systems. In this context, providing an awareness support consists in delivering to users involved in a collaborative task, information concerning their group, other users, their past, present or future roles or activities, etc. However, one of the potential underlying risks of such a support is to produce too much information. As a matter of fact, this amount of information may reveal itself cumbersome, leading to a cognitive overload for users which slows down the global activity of the group. In this paper, we propose to address this issue by stratifying the awareness information space into different levels of details. Such stratifications give users a progressive access to information: first, they can access to the most relevant information linked to their current activity and profile, and then, they can gradually access to the rest of awareness information. The proposed architecture relies on a generic Data Model for Awareness Information (DMAI) and on a Progressive Access Model (PAM), we have previously designed in the context of Web-based Information Systems. We show, on an example, how the coupling between the DMAI and the PAM allows groupware systems designers to describe a progressive access to awareness information. The defined stratifications can also be used as filters which save a given user, involved in a given role and performing a given activity, from useless, irrelevant or forbidden information.

Keywords: Awareness, groupware systems, progressive access, adaptability.

1 Introduction

When working cooperatively, users share a context of work in which their activities are "immersed". In fact, according to Schmidt [14], cooperating actors, even while doing their individual tasks, take heed of the context of their joint effort. Still according to this author, they try to align and integrate their activities with those of their colleagues without interrupting each other. This context, in Computer Supported Cooperative Work (CSCW) research area, refers to the notion of *awareness*. Awareness, according to Dourish [6], is "an understanding of the activities of others, which provides a context for your own activity. This context is used to ensure that

R. Meersman et al. (Eds.): CoopIS/DOA/ODBASE 2003, LNCS 2888, pp. 336–353, 2003.
© Springer-Verlag Berlin Heidelberg 2003

individual contributions are relevant to the group's activity as a whole and to evaluate individual actions with respect to the group goals and progress".

Awareness support appears to be a central feature in the design of collaborative work systems, since being aware of the other contributors and of their activities makes the work more natural and fluid [8]. This support intents to help the group to reach its goals and to increase the whole quality and efficiency of the collaborative work by providing a better understanding of the work context. By improving each individual contribution, awareness support also enhances the whole group contribution.

In groupware systems, users are implicitly or explicitly assigned a set of roles. Each role describes a set of activities and defines a group of users who are given rights to perform this set of activities. The efficiency of an awareness support depends also on its ability to deliver to users only the relevant information for the role they fulfill at a given moment. Nevertheless, even when information is targeted for a role, awareness support mechanisms often produce and manage large amounts of information, which may appear to be cumbersome in the achievement of the tasks.

In this paper, we propose to organize the information delivered by the awareness support into different levels of detail, in order to limit the amount of information presented. As a consequence, we expect to prevent groupware users from suffering from an information overload. The idea is to offer users a progressive access to information, by supplying them first, the most relevant information and by allowing them to access gradually to more information if needed.

This paper is organized as follows. In section 2, we advocate for the needs of a gradual delivery of awareness information in order to limit the risk of information overload, and present in section 3 an application scenario. Then, we present in section 4 a Progressive Access Model, called PAM, which enables to stratify any information space into different levels of detail [18]. In section 5, we show how to apply the PAM to a Data Model for Awareness Information, called DMAI, which describes users, roles and activities of a groupware system. In section 6, we illustrate our approach by applying the PAM to the application scenario introduced in section 3. Some related work are presented in section 7 before we conclude.

2 Towards Adaptability in Awareness

Awareness refers to an understanding of the group and its activities, which provides a context for individual activities [6]. The context of an action is what allows people to identify the meaning of this action [5]. Generally speaking, the term 'awareness' refers to actors taking heed of the context of their joint effort. In CSCW context, 'awareness' refers to actors who are or become aware of something, align and integrate their activities with the activities of others [14]. However, there is no single definition for awareness, and the term is used in very different situations [10]. In this paper, we decide to adopt the definition given by Dourish [6] and presented above in the introduction.

Awareness is widely accepted as a central feature in groupware systems. It deals with the delivery of information (about groups, users, activities, etc.) to each user involved in a collaborative task. However, when this information is not well-adapted

to each particular individual needs, it turns out to be disturbing and prejudicial to each user's contributions, and as a consequence, to the group's ones.

When information is not appropriate, users are confronted to a cognitive overload due to a too massive and difficult to understand quantity of information [17]. Such a cognitive overload, even when it concerns just a few group's members, may have a negative impact on the whole group's work by penalizing other members in their work. In order to protect the group from these drawbacks, some adaptability in awareness support is required. Indeed, the execution of some activities can be disturbed, and the exchange of information about these activities can be slowed down.

Work dealing with adaptability in groupware systems mainly focus on the tailorability of the *functionalities* of the system. Tailorability refers to some extensions added to an application during its use in order to cover requirements that where not present in the original design [7][16]. In this paper, we propose to address the adaptation of the *information* delivered by the awareness support. Here, adaptability refers to the capacity the awareness support has to provide each user with awareness information which is useful and essential for her/him and the specific tasks she/he performs.

Obviously, collaborators do not need all the same information, neither do they need all the available information all the time [17]. Thus, one has to determine what information users actually need, when and how it should be delivered to them. Also, the relative importance of the information has to be determined, and exploited to choose a proper notification method [10].

When working cooperatively, users play, implicitly or explicitly, roles inside the group. Since each role encompasses a specific set of activities, all roles do not require the same information to be fulfilled. The content of awareness information delivered to a user has to be closely related to the role she/he plays. For instance, a user playing an author role in a cooperative authoring environment needs some information different from the one required by the team coordinator. Obviously, an author needs information related to the part of the document she/he writes, whilst a coordinator should be informed about the overall work progress. The former is interested in the text adjunctions or modifications made by other authors. The later needs a global view of the group activities, the defined tasks and deadlines, to take her/his decisions and guide the team efforts. Many research works (such as [4]) have emphasized the fact that coordinators need well-suited awareness information for this role to be better played.

Therefore, it is necessary to limit the awareness information presented to a user to the set of information which is relevant to the role she/he plays. Also, users whose role grants no right on a shared object in the workspace should not be notified about the operations that concern this object. Nevertheless, to limit the information to present to the user only considering what is relevant for her/his role may be not satisfactory for this user. It is important to consider that users, even when working cooperatively, have their own interests and personal goals, which affect their work inside the team. According to Schmidt [14], cooperative workers, in their "natural attitude", are not ignorant or disinterested spectators, they are engaged in and with the world in which they cooperate. As such, they have interests, things to do, things to achieve. Thus, we believe that personal interests of the users should be considered when selecting awareness information, and that groupware applications should identify the user's characteristics and interests. As Liechti [10] affirms, in some cases it is preferable to identify who is the user and what are the artifacts that are

"interesting" for the user. When this is done, only a subset of the available information can be identified as critical and notified to the user.

However, even when reducing awareness information to the one related to user's interests and roles, the amount of available information may still be large. In this case, presenting the whole flow of information to the user decreases the time and attention she/he can spend on her/his work and finally submerges her/him.

Further, we strongly believe that the member's ability to assimilate information can be improved if the cognitive load is better controlled. One way of achieving this is to present to a user the most important information first and to offer her/him the possibility to progressively access information of lesser importance if needed.

Therefore, one can imagine that, ideally, a system should prioritize the information which is relevant to the role played by the user. Most of the time, this task is performed by the designer of the collaborative system or by the coordinator who assigns the roles to the different members. One can also imagine that an authorized user could be able to assign by herself/himself some priorities to information she/he wants to receive. In each case, one needs to describe the way information is structured in different levels of priority. In order to achieve such a structuration of information, we propose to apply and adapt to awareness in collaborative work the notion of *progressive access* [18] we have previously introduced in the field of Web-based Information Systems. We show in the example scenario described in the next section how this notion of progressive access can be relevant in the field of CSCW.

3 Application Scenario

Groupware systems, particularly web-based ones, have many similarities with Web-based Information Systems (WIS), which have inspired the proposition of the Progressive Access Model (cf. section 4). One of these similarities is the risk of cognitive overload, as discussed by Villanova-Oliver [18]. Based on these similarities, we think that coupling the notion of progressive access to the awareness support inside groupware systems can bring many advantages to these systems. The example below demonstrates some of these benefits.

Let us consider a web-based groupware system, for instance, a shared workspace, such as BSCW [2]. Such systems have often shared repositories, some awareness support, usually provided by subscription/notification methods, and some communication tools. In this example, we consider that this groupware offers such functions, and relies on a meeting tool (such as a Microsoft© NetMeeting) for communication purpose. Then, one can imagine that if a team member has missed the last group meeting, the awareness support may inform her/him about this meeting. The system may keep track of a lot of information about a meeting, but we consider that the system keeps only the following ones: the meeting goals, the decisions that have been taken, the discussions that have been raised (the chat register, for instance), and a multimedia resource associate to it, such as a video of the meeting or a slide presentation made during the meeting.

However, if the system presents to this user all the available information about the last meeting, together with all other events produced in the shared workspace, this team member may feel herself/himself overloaded by all this information, and she/he may miss some important one. For instance, she/he may not pay attention to the

meeting decisions, which can seriously influence the group work. Then, how can the system prevent such an overload? One may suggest that the system could automatically resume the meeting content. Unfortunately, this solution usually involves complex algorithms to be integrated in the groupware. An alternative is to separate the information in levels of details, in order to present the most important one first. Thus, the meeting information can be divided into three levels: first, the user receives only the meeting goals and decisions. Only if user requests more information, she/he receives the discussions (second level), and at last, the multimedia resource associated (third level). This division will reduce the amount of information presented to the user and improve the assimilation of the most important information.

Let us carry on with this scenario, considering that the groupware has its own adaptation proxy, which is able to adapt web-oriented content to mobile devices. If the system stratifies the meeting information in levels of detail, the proxy will not be obliged to adapt all meeting content to such devices. If user does not request the third level, for instance, the proxy will not adapt the multimedia resource.

Thus, in order to organize the awareness information into these levels, we propose to adapt to awareness support the notion of progressive access. In the next section, we introduce this notion and a model that implements it, before showing how this Progressive Access Model can be exploited to adapt awareness information.

4 Progressive Access: Notions and Model

The central idea behind the notion of *progressive access* is that the user of an information system does *not* need to access *all* the information *all* the time. The goal is to make the system able to deliver *progressively* a personalized information to its users: first, information considered as essential for them is provided, and then, some complementary information, if needed, is available.

4.1 Notions Related to the Progressive Access

The notion of progressive access is related to the one of *Maskable Entity*. A *Maskable Entity* (*ME*) is a set of at least two elements (*i.e.* $|ME| \geq 2$) upon which a progressive access can be set up. The progressive access to a ME relies on the definition of *Representations of Maskable Entity* (*RoME*) for this ME. These RoME are subsets of the ME ordered by the set inclusion relation. Two kinds of RoME, *extensional* or *intensional*, are distinguished. *Extensional RoME* are built from the *extension* (*i.e.* the set of elements) of the ME. In the case where the ME is a set of structured data having the same type, *intensional RoME* can be built from the *intension* of the ME. The intension of a structured ME is here defined as the set of descriptions of variables (or slots, or fields) which constitute the structure of the ME. Whatever its nature – extensional or intensional –, each RoME of a ME is associated with a level of detail (or relevance, or priority, etc.). $RoME_i$ is defined as the RoME of a ME corresponding to the level of detail i, where $1 \leq i \leq max$, and max is the greatest level of detail available for this ME.

The Fig. 1 shows a ME with three associated RoME. Some rules impose that a $RoME_{i+1}$ – whether it is extensional or intensional – associated with the level of detail

i+1 ($1 \leq i \leq max-1$) contains at least one more element than $RoME_i$. This way, extensional RoME can be seen as different ordered *masks* on the extension of a ME, while intensional RoME can be seen as different ordered *masks* on the intension of a ME.

A stratification for a ME is a sequence of RoME ordered by set inclusion as illustrated in the right part of the Fig. 1. Please note that several stratifications can be defined for a given ME.

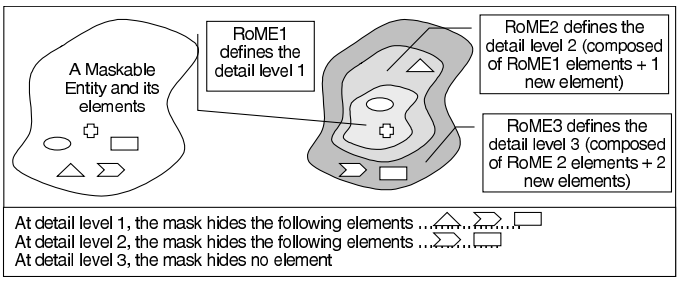

Fig. 1. A Maskable Entity with three RoME corresponding to three levels of detail.

The progressive access relies on two functions which allow to switch from a RoME to another within a given stratification:

- from a $RoME_i$, at level of detail *i*, the *masking function* gives access to the $RoME_{i-1}$ at level of detail i-1:

$$masking(RoME_i)=RoME_{i-1}, \text{ where } 2 \leq i \leq max.$$

- from a $RoME_i$, at level of detail *i*, the *unmasking* function gives access to the $RoME_{i+1}$ at level of detail i+1:

$$unmasking(RoME_i)=RoME_{i+1}, \text{ where } 1 \leq i \leq max-1.$$

4.2 The Progressive Access Model

The Progressive Access Model (PAM) is defined as a generic UML class diagram (see Fig. 2) which allows the description of well-formed stratifications for maskable entities.

Each notion presented in section 4.1 is implemented by a stereotyped class. An instance of the class "*Maskable Entity*" is, at least, composed of two instances of the class "*Element of Maskable Entity*". An instance of the class "*Stratification*" is represented as an aggregation of, at least, two *ordered* instances of the class "*Representation of Maskable Entity*". An instance of this class is linked by the association *adds* to one or more instance(s) of the class "*Element of Maskable Entity*" which are the elements of the ME added by the $RoME_i$ at the level of detail *i*. The dependency relation ({*subset*}) ensures that the added elements belong to the set of elements corresponding to the ME.

An important characteristic of the PAM is the ternary association called *definition* which links the classes "*Stratification*" (S), "*Maskable Entity*" (ME) and "*User Category*" (U). The class "User Category" is an abstract class which allows the connection with any user model so that each described user can benefit from a personalized progressive access to a given maskable entity.

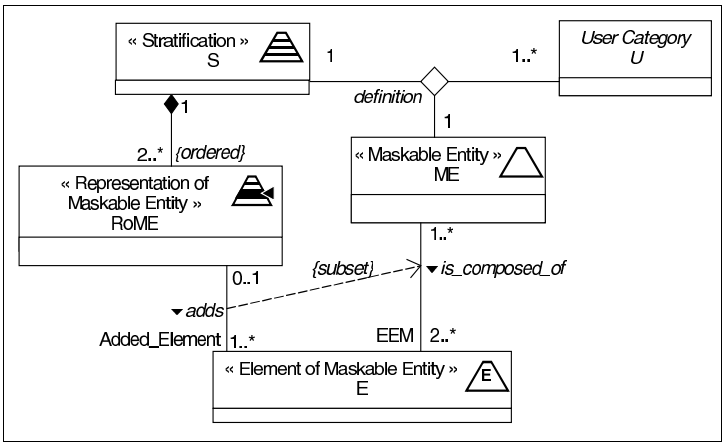

Fig. 2. The Progressive Access Model described using UML stereotypes

It is worth noting that maskable entities can be of several kinds. Basically, every data model (*e.g.* a class diagram) and every constituent (*e.g.* a class, an association…) of this model can be considered as maskable entities and, therefore, can be stratified (see [18] for a description). Also, the progressive access approach can apply to information represented by a non-structured data model, as well as to a semi-structured (XML) or to a structured (object or relational) one. For instance, in [17] the authors have shown how to couple the PAM with an object-based knowledge representation formalism in order to offer a progressive access to the objects of the knowledge base. In the next section, we explain how the PAM described above can be used to offer a progressive access to a data model supporting awareness information in collaborative systems.

5 Progressive Access to Awareness Information

In order to integrate the Progressive Access Model into the awareness support, we first define a data model dedicated to the awareness information. The different ways of stratifying this model are presented in section 5.2. Then, a stratification process which combines different kinds of stratifications is presented in section 5.3.

5.1 A Data Model for Awareness Information

The PAM presented in section 4 generally applies to a data model which describes the application domain. Using the PAM, stratifications can be defined for each user or group of users. These stratifications give them a progressive access to the information contained in the data model. Since our objective is to provide a progressive access to awareness information, we propose here a simplified data model, called Data Model for Awareness Information (DMAI), which describes the concepts related to awareness information and the relations between these concepts. Although this model

is quite simple, we think it is sufficient enough for introducing our approach. It describes the basic concepts linked to the awareness definition given by Dourish [6], and it can be used or extended in order to be exploited by a groupware system. The different components of the DMAI (groups, roles, users, activities, and profiles) are represented in the UML description we give in Fig. 3.

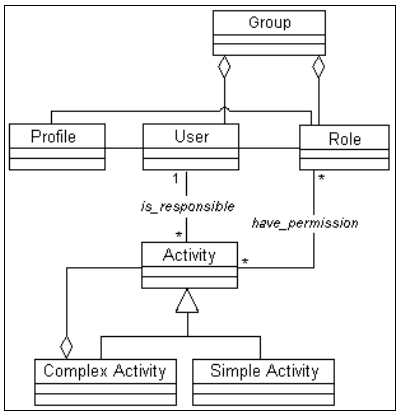

Fig. 3. Data Model for Awareness Information

As stated before, awareness information must help in understanding the group and the activities performed by its members. Then, we claim that the awareness information should be represented as the information about *groups*, *users*, *roles* and *activities*. A group is defined by a set of users and a set of roles. Users who are authorized to perform this set of roles are members of the group. A user can belong to one or more group(s) and a role can also be linked to one or more group(s). A user can fulfill one or more role(s). A role is possibly fulfilled by several users and defines the rights and responsibilities concerning a set of activities. An activity is an elementary task a user can perform when fulfilling a given role. Several users can perform the same activity, but only one of them is responsible for this activity. The activities can range from simple tasks to more complex ones which are composed of others tasks.

Another important characteristic when dealing with awareness information is the representation of both users and roles characteristics (cf. section 2). These features are represented through the notion of *profile*. A profile can be associated either with a user or with a role. In the former case, a profile represents the user's interest in terms of needs and preferences. In the later, a profile describes the requirements to meet in order to fulfill an associated role. A user's profile can be independent from or linked to the role fulfilled by the user. When linked, the profile represents the user's preferences associated to a specific role. Moreover, we consider that each profile gathers a set of activities that are important for the user or role, and also a time interval, which constrains this set of activities to take place during a certain period of time.

5.2 Possible Stratifications for the DMAI

In order to apply the principles of the PAM to the DMAI, one has to identify the possible Maskable Entities to be stratified. Three kinds of stratification are possible:

1. **Extensional stratification of the DMAI.** In this case, the DMAI as a whole is seen as a ME whose set of elements is the set of classes, associations and classes-associations which constitutes the DMAI. Then, an extensional stratification of the DMAI can be performed. It consists in defining subsets of the DMAI components (classes, associations and/or classes-associations) ordered by the set inclusion subsets, providing so a progressive access to each subset of this sequence. For instance, one extensional stratification S of the DMAI is illustrated by Fig. 4 (please note that only the components 'class' are considered in this example). The stratification is defined as S = {{User, Role}, {User, Role, Activity}}. S contains 2 RoME: $RoME_1$ = {User, Role}, and $RoME_2$ = {User, Role, Activity}. At the first level of detail, $RoME_1$ gives access to all the instances of the classes User and Role. At the second level of detail, $RoME_2$ gives also access to the instances of the class Activity. In this example, instances of the classes Profile and Group remain inaccessible.

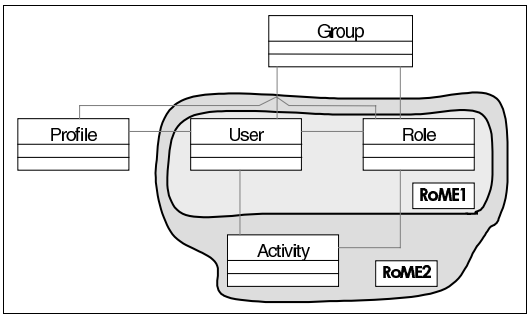

Fig. 4. Extensional stratification of the Data Model for Awareness Information.

2. **Extensional Stratification of a DMAI component.** In this case, components of the DMAI (classes, associations and classes-associations) are seen as ME whose set of elements is a set of instances. Then, an extensional stratification of a DMAI component can be performed. It consists in selecting some instances of the DMAI component and in ordering by the set inclusion subsets of this set of instances in order to provide a progressive access to each subset of this sequence. For instance (Fig. 5), one extensional stratification S of the class Activity for which ten instances $\{a_1, a_2, \ldots, a_{10}\}$ exist can be defined as S = $\{\{a_4, a_7, a_3\}, \{a_4, a_7, a_3, a_1, a_9\}, \{a_4, a_7, a_3, a_1, a_9, a_6\}\}$. S contains 3 RoME. At first level of detail 3 activities are visible, at level two, two more activities are visible, at level three six activities are visible. In this example, activities a_2, a_5, a_8 and a_{10} are always masked and therefore can not be accessed when using the unmasking function (right part of Fig. 5)

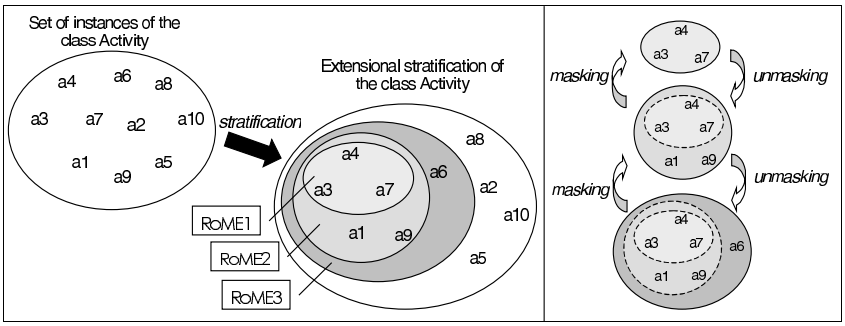

Fig. 5. Extensional Stratification for the class Activity.

3. **Intensional Stratification of a DMAI component.** In this case, components of the DMAI (classes, associations and classes-associations) are seen as ME whose structure is made of a set of variables. Then, an intensional stratification of a DMAI component can be performed. It consists in selecting some (possibly all) variables of the DMAI component and in ordering by set inclusion subsets of this set of variables, providing a progressive access to each subset of this sequence. For instance, let us suppose that the class Activity contains the following variables: 'name', 'description', 'details', 'responsible' and 'time interval'. One intensional stratification S of the class Activity can be defined as S = {{'name', 'responsible', 'time interval'}, {'name', 'responsible', 'time interval', 'description'}, {'name', 'responsible', 'time interval', 'description', 'details'}. S contains 3 RoME. At the first level of detail 3 variables are visible and give some essential information about the activity, at level two, one more variable is visible ('description'), at level three the whole structure of the class is visible (see Fig. 6).

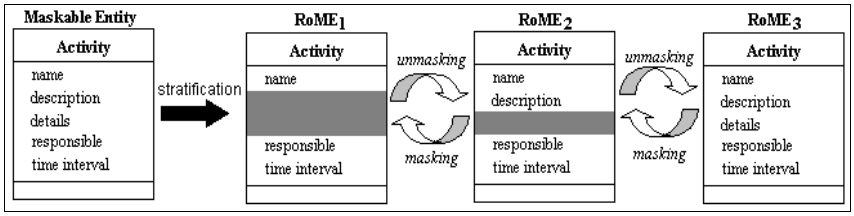

Fig. 6. Intensional Stratification of the class Activity.

We believe that applying only one of the stratification techniques is not enough to significantly reduce the overload risk. For instance, an extensional stratification of a component (e.g., a class) can be compared to a filtering process, since it selects instances of such component. However, we cannot ensure that such selection will sufficiently reduce the amount of instances presented to the user, since awareness mechanism often produces high amounts of it and a large proportion can be selected. Thus, we propose a process that combines some stratification techniques over the data model for awareness information. For instance, it is possible to apply an intensional stratification (selection of ordered subsets of variables), followed (or preceded) by an

extensional stratification (selection of ordered subsets of instances) to a class which belongs itself to a RoME of an extensional stratification of the DMAI. Such a composition of stratifications is illustrated in the next section.

5.3 Stratification Process for Awareness Information

In this section, we show how a groupware designer can use the stratification techniques presented above for awareness information. We address here the delivery of awareness information which is relevant at a given time.

In order to determine this relevance, the awareness support needs to know *who* is performing activities as well as her/his *current role* and also the *tasks* associated with her/his activities. Groupware designer should then determine the target of the stratification, which can be, basically, *a given user, role or activity*. In the first case, stratification consists in providing a progressive access to information taking into account the general characteristics of the user (level of knowledge, presentation preferences, material configuration, etc.). In the second case (*given role*), stratification consists in providing a progressive access to some general awareness information related to this role (the associated activities, the users fulfilling the role, etc.), without considering who is assigned this role. In the last case (*a given activity*), stratification aims at giving a progressive access to the awareness information related to this activity, without considering who is performing this activity and her/his current role (e.g. stratifications given in Fig. 5 and Fig. 6). It is also possible to compose these possible targets, searching then stratifications for *a couple <user, role>*, *a couple <role, activity>*, or *<user, activity>*, or even *a triple <user, role, activity>*. In this last case, the stratification focuses on the awareness information dealing with one activity performed by a given user when she/he performs a given role.

As we discussed in section 2, we believe that awareness support should take into account the user's preferences as well as her/his role's needs. Thus, we suggest that groupware designer should target the triple *<user, role, activity>* when defining the stratifications of the DMAI.

Therefore, we propose to combine the stratification techniques (cf. section 5.2) by applying them in the following sequence: first, extensional stratifications of the DMAI, which organize the model itself in levels of details. Then, extensional stratifications of the DMAI components, which reduce and organize in levels the instances that will be presented to the user. Finally, intensional stratifications of these components, which define for each component levels of details. The Fig. 7 represents this process.

Thus, once defined the stratification target and its process, the groupware designer has to define the system's stratifications. To begin, the designer has to define an extensional stratification of the DMAI in order to serialize subsets of components (classes, association) of this model. For instance, she/he can assign the class Activity to the first level of detail in order to give an immediate access to the set of its instances. She/he can then define, in a similar way, the components of the DMAI that will appear at the other levels.

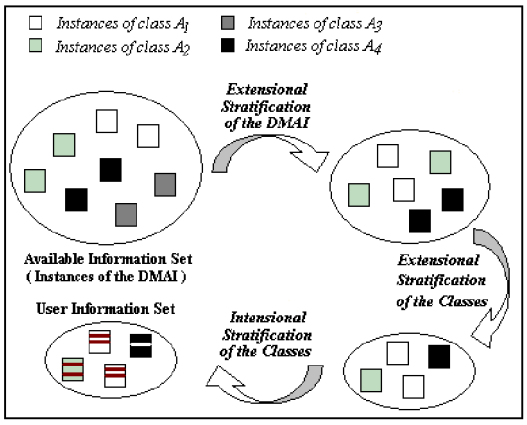

Fig. 7. The stratification process suggested for the awareness information

Then, the designer may apply an extensional stratification to each class (or just a subset) of the model in order to give a progressive access to this set of instances. For instance, she/he can consider that the most relevant is to inform the user about the last performed activities. This consists in associating the corresponding instances of the class Activity (*i.e.* the more recent ones) to the first RoME.

Considering that our target is the triple <*user, role, activity*> and since these extensional stratifications behave as a selection process in the set of available instances, groupware designer can use profiles to define (and allow users to define) these stratifications. A profile (cf. section 5.1) may include some conditions to perform an appropriate selection. Each profile can define information levels, each level being composed by DMAI classes that user/role wants/needs to be aware of and some special conditions (e.g. time intervals) that the instances of these classes should respect to be selected. In addition, by combining the three profile types suggested in section 5.1 (user profile, role profile and user profile linked to a specific role), these stratifications can perfectly match the target, by considering the activity, the role and the user.

For instance, a user may choose in her/his profile to receive at the first level the instances of the activities A_1 and A_2 (extensional stratification of the model) produced in the last week (extensional stratification of the components), while the same level of her/his role's profile includes only the activity A_3. The system could then combine these profiles and present to the user first the instances of the activities A_1, A_2 and A_3 produced in the last week.

Finally, the designer can take advantage of applying an intensional stratification to the structure of a component of the DMAI. This way, the user benefits from, according to her/his roles, more or less complete representations of the instances she/he can access progressively (cf. section 5.2). For instance, considering the Fig. 7, the designer may define that the role's profile determines the extensional stratification of the model and the user's profile, the extensional stratification of the classes, while the intensional stratification is linked to the role.

6 Example

In this section, we present an example of how a groupware designer can use the concepts presented in this paper during the development of a simple groupware tool.

Let us consider again the example presented in section 3 and let us suppose that the groupware system consists here of a simple meeting tool for such environment. Moreover, we suppose that this meeting tool integrates a DMAI in charge of representing awareness and user information. The unique activity handled by the tool is the meeting itself, represented by a Meeting class. Thus, the potential MEs are the model itself and the classes User, Role and Meeting. A meeting object is structured as suggested in section 3, it has a meeting title, goals, decisions, the discussion, and a related document. The discussion refers to the content of a chat session which is automatically registered by the tool. The rest (title, goals, etc.) is supplied by the meeting coordinator.

The Fig. 8 shows the meeting process adopted here: (1) first, the meeting coordinator demands to the meeting manager the creation of a new meeting. The coordinator also provides the manager with the meeting title, goals and the URL of the associated document. The meeting manager creates a new meeting object with such information and starts a new chat session for this meeting (2). Once the meeting has started, other users may join it (3-4), load the associated document (5) and contribute to the discussion (6). The end of the meeting is defined by the coordinator (7), who describes the decisions that have been taken (10). The system registers then the discussions (9) and notifies the awareness manager about it (11).

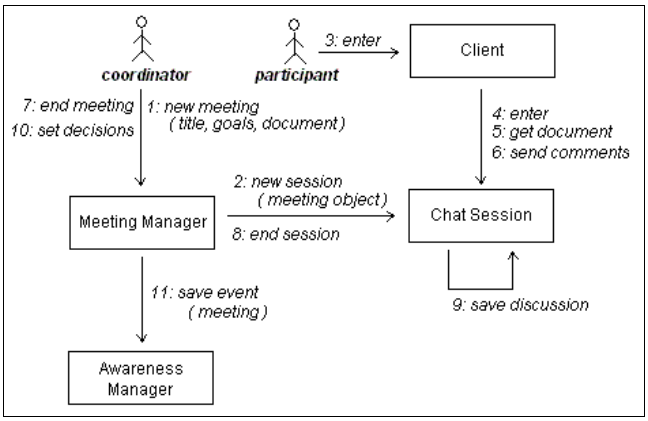

Fig. 8. Meeting process in the Simple Meeting Tool

The awareness support introduced in this tool is limited to inform the users about past (or missed) meetings. In order to facilitate the development of the awareness manager, we use a framework for awareness support called BW [9], we have designed for supplying awareness support about past activities (activities already concluded by the group) in groupware systems.

The BW Framework uses a data model that corresponds to the DMAI and which is stored in a package called Kernel. Beside the description of users, roles and activities

(called "events" by this framework), the Kernel package also includes the definition of profiles for users (personal profiles), roles (role profiles) and users playing a specific role (personal role profiles). These profiles (cf. section 5.1) are composed by a set of relevant activities for the user or role, and an Allen's interval [1] during which they are relevant.

These profiles are used by the meeting tool to define the extensional stratification of the meeting class. This stratification is based on the time interval parameter associated to each object. Thus, each user (participant or the meeting coordinator) may define the time interval in which the information about the meetings is important for her/him.

This extensional stratification results from a filtering process performed by the BW framework. This process aims at reducing the amount of information delivered to the user by limiting the number of instances presented to those that match the profiles. It consists in selecting the available instances by merging the different profiles that concern the user (personal, role and personal role profiles). This merge creates a single set of activities (in our case, the meeting) and a resulting time interval, which is used to select the instances to be presented to the user. This resulting interval is the largest interval that encompasses all composing intervals. So, any instance whose interval is included in the interval of one of these profiles is also included in the resulting interval. Only instances whose time interval occurs during the resulting time interval are really presented to the user. As a result, the set of instances presented is reduced. It is over this resulting set of instances that other stratifications defined for the user and her/his active role apply. The Fig. 9 shows this filtering process, which combines personal and personal role profiles to select only the meeting instances { a_3, a_4, a_5, a_6 }.

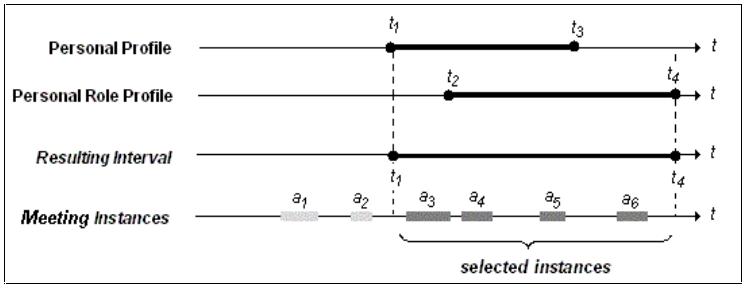

Fig. 9. The filtering process based on the time interval.

In order to define other stratifications (extensional stratification of the model and intensional stratification of the model components), we have designed a package, called PAM, which introduces the progressive access definitions (cf. section 4) and binds them with the BW Framework components, specially the Kernel package, which represents the data model used by the tool.

The PAM package implements the concept of stratification (class PAM_Stratification in Fig. 10), which represents the stratification of the awareness data model. This stratification is connected to the role representation (class BW_Role) through the association "defined_for". This association creates the relationship between stratifications and roles. Furthermore, this package integrates the concept of

RoME, which can be associated with a specific element of the data model to define an intensional stratification (PAM_ElementStratification) of this element.

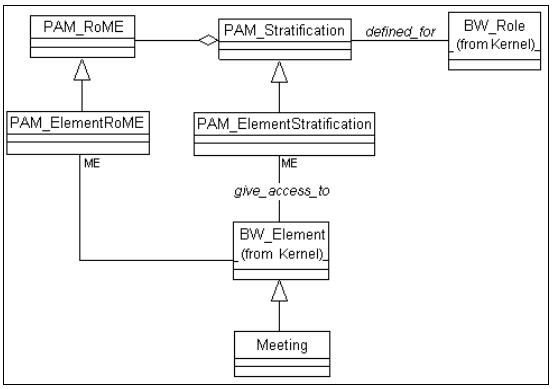

Fig. 10. The PAM package

Through this association between the stratification and the role, we can define, for instance, stratifications for one or more roles. For example, we can define, for the coordinator role, a stratification of the class Meeting, presented in Fig. 10, as S_1 = {{'title', 'time interval'}, {'goals', 'decisions'}, {'document'}, {'discussion'}}.

In order to better coordinate these stratifications with the presentation of the awareness information, the PAM package includes a manager which keeps track of all the defined stratifications. This manager is consulted by the user interface elements in order to know and respect the applicable stratifications. These interface elements receive the awareness information filtered by the BW framework. They consult the PAM manager and, based on the stratification definitions, control the behaviour of the presentation in order to respect these definitions. As a consequence, the awareness information can be supplied progressively to a user, following the stratifications defined for her/his roles.

Thus, through the BW framework and the PAM package, our meeting tool is able to collect the information about the meetings and present it progressively to the user according her/his profiles and the stratification definitions. These definitions are loaded by the tool during its start-up, and with the help of the PAM manager, users may define and load dynamically their own stratifications. A user can, for example, define for herself/himself the stratification S_1 defined above. Moreover, the developer may also define default stratifications. For instance, she/he may define an extensional stratification of the model, where in the first level only information about the meetings is presented, and the information about the users and roles is presented in the second level. We also associated the stratification S_1 to the coordinator role, thus any meeting coordinator will, by default, receive the awareness information through the 4 levels defined for this stratification. Accordingly to this loading policy, the responsibility of creating new stratifications, and then defining what information is relevant to a user or role, is shared between the system designer and the users themselves.

As one could observe, the complexity of our proposition concerns mainly the definition of the stratifications presented above. This definition requires some

knowledge about the data model for the adopted awareness information, referring to the system designer. Then, some flexibility for modifying the stratification definition during the groupware execution is also needed. The expected flexibility must be directed towards system users, first of all towards the team coordinator. However, this flexibility has a cost: an overhead to the user interface design, since this interface should handle different and progressive information set. All this makes us believe that our proposition is better suited to large groupware systems, which generally handle huge and complex awareness information set. In such a system, we expect that the advantages of our proposition will exceed this design and definition complexity imposed by the proposition. Nevertheless, more tests are still needed to better evaluate the impact of this proposition in real-size applications.

7 Related Work

Awareness support is the target of many works that can be found in the CSCW literature. Some examples of groupware systems that are concerned with awareness support are the BSCW workspace [2] and the Alliance editor [12]. These systems try to present awareness information through icons and history windows. Although the Alliance editor proposes a "group awareness agent" to adapt the awareness mechanism to the user's needs by modifying the presentation of its information, none of these systems offers a mechanism for delivering this information gradually when huge amount of information is available.

Furthermore, considering toolkits and frameworks for groupware development, the situation is similar. There are in the literature toolkits and frameworks that can help groupware designers in the system development. We can name, for example, Groupkit [13], ANTS [11] and BW [9]. These tools allow the development of new groupware applications (as ANTS), or the development of awareness mechanism (as BW). Even if these frameworks offer some kind of awareness support (ANTS, for example, presents an awareness component integrated in its infrastructure), they are not really concerned with the adaptation of the system to a high amount of available information, and once again, no mechanism is proposed to gradually access this information, whenever it constitutes a large set.

Concerning the research about tailorability ([16] and [7], just to name a few), these works are mainly concerned with the adaptability of the functionalities of the system. Nevertheless, works that handle tailorability do not directly address the adaptability of the information content, which is our main concern. What we have presented here is a model that allows a gradual access to the awareness information available, adapting the information presented to the user's and role's interests and necessities, in order to prevent some overload in the delivery of information.

Finally, the research involving context-aware systems propose the use of the information about physical context, such as location, to adapt the information delivered to it (see [5] and [3]). An example is Rover [3], which tracks the location of the user dynamically to configure the content delivered to her/him. However, there are only a few studies concerning the context of the work in a cooperative environment, or awareness as defined in the section 2. An example is the AwareNex system [15], which allows a mobile user that is away from her/his desktop to maintain some awareness about her/his colleagues and their availability (know as "group

awareness") and contact them. Unfortunately, AwareNex deals only with these group awareness functionalities, it does not present any other information concerning the context of the group's activities.

8 Conclusion

Awareness support in groupware systems is an important feature which aims at providing users with some knowledge essentially related to a group and to its activities (past, present or futures ones). Such a kind of knowledge constitutes an informational context which improves individual contributions and avoids contradictory interactions between members of the group. One of the well-known drawback of awareness information appears when the amount of delivered information is too large to be efficiently exploited by the user. In order to prevent groupware systems users from suffering from such a cognitive overload, we have proposed in this paper to give them a progressive access to awareness information. This progressive access relies on a stratification of the awareness information space into different levels of detail. We have shown how to link a Progressive Access Model to a Data Model for Awareness Information that we have described. After having defined the different ways of stratifying such a model, we have illustrated strategies the designer can adopt to define stratifications. These stratifications are adapted to a user performing activities when fulfilling a specific role in a groupware system. Finally, we have presented an example of the application of the progressive access in awareness support, by showing the development of a tool offering such a support.

By proposing this progressive access to the awareness information we expect to reduce significantly the risk of cognitive overload. The meeting tool that we have built, although simple, has demonstrated this reduction. We strongly believe that by reducing the cognitive overload it is possible to increase the efficiency of the awareness mechanism, and thus, to increase the performance of the whole group.

This work is just at its beginning. There are still many issues to be investigated concerning the application of the progressive access in awareness information delivery. We should consider, for example, the exploitation of the relationships and dependencies among the activities as a condition for stratification.

References

1. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM, 26(11), ACM Press, 1983, p. 832–843.
2. Appelt, W.: What groupware functionality do users really use? Analysis of the usage of the BSCW system. 9th Euromicro Workshop on PDP, IEEE Computer Society, 2001. Available at: http://bscw.gmd.de/Papers/PDP2001/PDP2001.pdf. Access: February 2003.
3. Banerjee, S., Agarwal, S., Kamel, K., Kochut, A., Kommareddly, C., Nadeem, T., Thakkar, P., Trinh, B., Yossef, A., Larson, R.L., Shankar, A.U., Agrawala, A., Rover: scalable location-aware computing, Computer, 35(10), oct. 2002, IEEE Computer Society, p. 46–53.
4. Borges M.R.S., Pino, J.A.: Awareness Mechanisms for Coordination in Asynchronous CSCW. Proceedings of 9th Workshop on Information Technologies and Systems, 1999.

5. Dourish, P., Seeking a foundation for context-aware computing, Human Computer Interaction, 13(2–4), 2001, Hillsdale, p. 229–241.
6. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proceedings of ACM Conference on Computer-Supported Cooperative Work, ACM Press, Toronto, Canada, 1992, p 107–114.
7. Fernández, A.; Haake, J.M.; Goldberg, A.: Tailoring group work. In: Haake, J.M, Pino, J.A. (eds.), CRIWG 2002, LNCS 2440, Springer-Verlag, 2002, p. 232–242.
8. Gutwin, C., Greenberg, S.: Effects of awareness support on groupware usability. Proceedings of CHI'98 - Conference on Human Factors in Computing Systems, ACM Press, 1998, p. 511–518.
9. Kirsch-Pinheiro, M., Lima, J.V., Borges, M.R.S.: A Framework for Awareness Support in Groupware Systems. 7th International Conference on Computer Supported Cooperative Work in Design, 2002, p. 13–18.
10. Liechti, O.: Awareness and the WWW: an overview. CSCW'00 Workshop on Awareness and the WWW, ACM Press, 2000. Available at: http://www2.mic.atr.co.jp/dept2/awareness/fProgram.html Access: December 2002.
11. López, P.G, Skarneta, A.F.G.: ANTS framework for cooperative work environments. Computer, 36(3), IEEE Computer Society, 2003, p. 56–62.
12. Martínez-Enríquez, A.M., Decouchant, D., Morán, A.L., Favela, J. : An adaptive cooperative web authoring environment. 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Malaga, Spain, 2002.
13. Roseman, M., Greenberg, S.: Building real time groupware with GroupKit, a groupware toolkit. ACM Transactions on Computer Human Interactions, 1(3), 1996, p. 66–106. Available at: http://www.cpsc.ucalgary.ca/grouplab/papers. Access: January 2000.
14. Schmidt, K., The problem with 'awareness': introductory remarks on 'Awareness in CSCW', Computer Supported Cooperative Work, 11(3–4), 2002, Kluwer Academic Publishers, p. 285–298.
15. Tang, J.C., Yankelovich, N., Begole, J.B., Vankleike, M., ConNexus to Awarenex: extending awareness to mobile users, CHI Letters, 3(1), 2001, ACM Press, p. 221–229.
16. Teege, G.: Users as composers: Parts and features as a basis for tailorability in CSCW systems. Computer Supported Cooperative Work, vol. 9, Kluwer Academic Publishers, 2000, p. 101–122.
17. Villanova, M., Gensel, J., Martin, H.: Progressive Access: a Step towards Adaptability in Web-based Information Systems, 8th International Conference on Object-Oriented Information Systems (OOIS 2002), Montpellier, France, September 2–5, 2002, p. 422–433.
18. Villanova-Oliver, M.: Adaptabilité dans les systèmes d'Information sur le Web : Modélisation et mise en œuvre de l'accès progressif, Thèse de Doctorat, Institut National Polytechnique de Grenoble, décembre 2002 (in French). Available at: http://www-lsr.imag.fr/Les.Personnes/Marlene.Villanova/. Access: April 2003.