TOOLBOX SUPPORTS GROUP AWARENESS IN GROUPWARE

Ouahab Kadri [1,2], Adel Abdelhadi [1,2], Leila-Hayet Mouss [1]

[1] **Laboratory of Automation & Production Engineering, University of Batna, Algeria**
[2] **Department of Computer Science and Math, University of Khenchela, Algeria**

Corresponding author: Ouahab Kadri, ouahabk@yahoo.fr

ABSTRACT: Group awareness tools are developed to minimize the time of cooperative application realization and spare designers a lot of effort devoted to integrating the group awareness aspect into groupware. But these tools have several disadvantages, such as dependence on a single type of application or overloading the minds of users with unnecessary information. From here comes the need to develop a tool that allows to offer information of group awareness configurable and to be both generic and easy to use. Our article presents some tools that have inspired several ideas. It proposes a design of a new toolbox that allows a better interpretation of group awareness information. Finally, it presents a variant of the client/server architecture based on work area.

KEY: Groupware, Group Awareness, Toolbox, Work Area.

1. INTRODUCTION

Group awareness is a fundamental element in all groupware. Even if designers do not take it into account during the development phase, each user develops his particular method of acquiring the actions of the other participants which is more or less interesting. So here the quantity and quality of the captured information is different from one user to another.

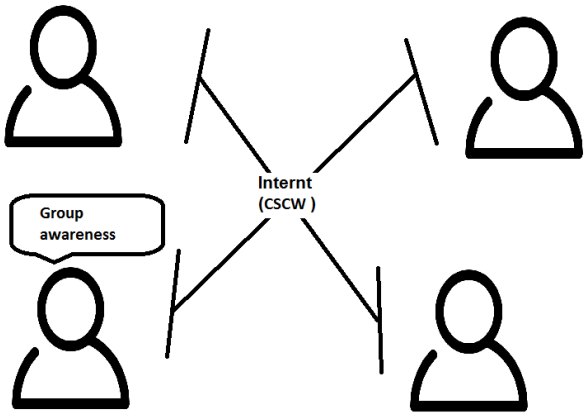

Fig. 1. Group awareness in Computer Supported Cooperative Work

When working in a group, such as co-writing a scientific report, we always have an eye on what other members of the group do to act and react according to the tasks performed, such as removing an issue addressed by other group participant in its section which it is in the drafting. The group is effective, only if there is a team spirit that relies on group awareness (see figure 1).

The term "group awareness" is the most commonly used in the literature, there is another less used term that is "group feedback" [D+13]. Group feedback is often linked to the field of co-writing. To keep the generic character of our tool, we will use the term "group awareness".

Group awareness is a new concept in the field of CSCW (Computer Supported Cooperative Work), which is why there is no conventional definition for this term. Gutwin in [GDB03] says that the term "group awareness" represents the information through which members of a group, while they are training to perform their individual tasks, observe the actions of other participants and adjust their own activities accordingly.

The origin of group awareness is psychology; it is defined by Dourish [DB92] as "the understanding of the activities of others, which gives a context to one's own activity".

The essential element of all given definitions is to construct a common cognitive space, we must transfer information between group members. While in the case of CSCW, the distance between participants will make the construction of this space a bit difficult.

The distance between group members (Tele-presence) often causes loss of information. For example, it can be understood that the poor psychic state of a limb does not allow the work to be accomplished if one is in direct contact with it. Therefore the decision is either to replace the member or to report the work at a later date. But in the case of CSCW, it is necessary to have the techniques and means to transmit this information. So it is very interesting to locate this information and find the right way to transmit it in the right time and with the appropriate method to the group context [KLB03]. And all this must be based on the role definition of each member and the interdependencies that linked him with the other participants.

2. AWARENESS TECHNIQUES

2.1 User interface

The technique of hiding information and showing others is currently used in several software. The secret of this method is to calculate the frequency of use of the functions of the software by the user. So the interface of the application depends on the trends of the user who sees only the functions most frequented by him. But, it is also necessary to leave to the user the possibility of modifying the interface in a direct way.
In applications, the interdependence that exists between the participants makes the previous technique insufficient to the need of the work, since the work space of the member depends directly on the tasks proper to him and also to the events received.

2.2 Event Management

2.2.1 Dissemination of events

Dissemination of events in a working group is a primary task. It is necessary to ensure that the member is aware of partner's activities. It is also essential to know who is among them is sharing the action with him on his objects. This information which tells him the location of his tasks in the total group work, it is to enable him to better coordinate.
There are two models of development tools used for events diffusion, the first one is based on a server that binds all the participants, its main role is to publish all the events and to receive queries on these events and thus to create categories for the classified. The classification into categories is based on the roles of users and obviously each category must have a subject described according to its content. The second model used for the development of these tools is based on the group decomposition into sub-groups according to the workspace and the proximity of the work areas [C+12].
In our work, we use a hybrid model based on a super server. There is one server for each work area and another for coordination between the different zones. The user must be given the opportunity to know the activity of the whole group since the global vision is an essential element in group awareness.

2.2.2 Types of Events

In the development phase of an application, actions are associated with events caused by the user or the environment. The great utility of these events is at the moment of their triggering. On the other hand, in a CSCW application, the utility of an event can be increased with the passing of time. For example, the frequency of consultation of an article by the co-editors indicates the importance of the article to the new participants. This type of event is called past event.
To show the importance of past events in group awareness, it is enough to imagine an application that does not support the possibility of viewing past events. This absence creates several problems, among others: in a cooperative work, if a member does not have the history of an experience of another participant for the realization of a common task, he will repeat the same errors.
Until today there is no classification of past events. In a synchronous application, one is often interested in the near past as in the case of the Telepointer trace. The trace makes it possible to understand the actions of the other participants and also to remain connected with their current actions. But if you keep the pointer position for a long time, it will clutter the scene and alter the cooperative work. On the other hand, in an asynchronous application, one has more flexibility to go through the history by looking for the most relevant actions of the other participants.
To make a classification using past events, we will be based on the classification of the cooperative applications "space and time" of Ellis [EN12], (see table 1). Although this classification has been criticized, it is still used since there are no other proposals that call into question this classification.

Table 1. Space - time classification

	The same time	Different time
A single space	Face to face work	Asynchronous work
Several spaces	Synchronous and distributed work	Asynchronous and distributed work

We can distinguish three types of past events: close past in the case of a synchronous application, medium and far in the case of asynchronous applications. But currently, since the core of cooperative applications offers both modes of communication (synchronous and asynchronous) then they can contain all three types. In the next section, we will present two tools of cooperative work that have inspired several ideas for the development of our tool.

3. AWARENESS TOOLS

3.1 MAUI toolkit

MAUI (Multi-user Awareness) is a toolbox based on JavaBeans, which allows a large re-use of code. It offers shared objects as all tools specific to groupware development. It also provides a shared version of traditional human-machine interface components such as buttons and input fields. The highlight of this toolkit is the use of telepointers.

The MAUI toolkit consists of five main parts: a component set, an event template, a custom runtime service, and a communications interface and a participation manager [HG04].

3.2 Community Bar

Community Bar (CB) is a CSCW tool that provides group awareness to a small, distributed group of collaborators. The CB interface is structured around the following principles. It appears as a sidebar divided into frames. Each frame contains several media articles that can be viewed in several ways. CB uses small images, icons and words to pass information about other members. It also offers the possibility to communicate to the participants via the chat service and access to certain shared objects [RMG13].

4. PROPOSED APPROACHES

A groupware must be extensible either by the ability to activate functions or by the integration of new modules. The latter is invalid unless the application is open and has a standard input / output interface with the software environment, for example XML files.
This makes it possible to select one or more values for certain parameters of an application in order to specialize it and make it more suitable to the needs of the user. Generally the development of a groupware is done from scratch. Since most dedicated tools do not allow transforming the group awareness information to knowledge for the participants and all this due to the absence of a system of capture and filtering of the data and also the interpretation of these data filtered in relation to the work context [KLB03]. The tools cited above represent an attempt in this direction. But each of the two tools has several disadvantages. For example, MAUI asks the developer to go down to a low level of detail in programming and CB does not use any elements of synchronous groupware like telepointers [HG04] [RMG13].
Several approaches are used in the development of groupware always respecting the difference that exists between a traditional application is a CAW application since the latter always consists of three parts (production, communication and coordination). These approaches can influence the appreciation of working group members to the groupware developed by these approaches.
The first approach is Development from nothing. As indicated by his name; this method gives the person responsible for development the freedom to choose the method of design and how to implement it. The disadvantages of this approach are closing the application and not reusing existing code. The result will be an application that does not allow great flexibility and risks limiting
The second characteristic is the parameterization in coordination among the participants. And we often use this approach to forget functions if the problem is complex [H+04].
The second approach is Development using toolkits. A toolbox is a set of classes that can be reused independently or in small groups for application development. The purpose of using toolkits is to spare the developer a lot of programming effort. The toolkits dedicated to the development of groupware offer in addition standard components of programming, a set of specialized elements to ensure the cooperation. The disadvantage of this approach is the lack of functionality that was not foreseen [H+04]. From here comes the need to develop toolboxes that allow for both scalability and flexibility. In the next section, we will present the most important elements in our future toolkit.

5. ESSENTIAL ELEMENTS OF THE TOOLKIT

5.1 Telepointer

Telepointer is a remotely manipulated cursor by a user to designate positions on the screen of the local user. Telepointer is a very useful element in synchronous groupware [DYC04]. It is simple in its representation but it provides several information of awareness like the presence or not presence of the participants. But the use of telepointers in complex networks often suffers from several problems due to network congestion and overloading of the scene by the presence of several participants in the same work area at the same time. The two frequent problems are the telepointer jump and the slowing of its movement which lead to transmitting a false representation of the reality and a misinterpretation of the gestures of the user by the other participants.

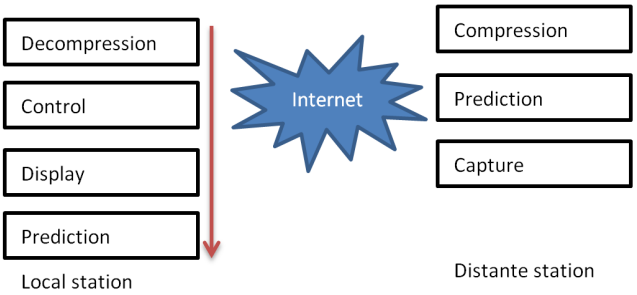

Fig. 2. Exchange of telepointer messages

Despite the increase in the Internet speed, the problem will not be solved soon, since the traffic also increases and because of the existence of users who use limited connections especially in Third World countries.

We will use the algorithm of dead reckoning [GDB03] which aims to limit the problems related to the network by reducing the emission of update messages. Its principle is to predict the status of a telepointer until an update arrives. Thus, the machine with a telepointer at its charge must send a message only when the distance between the actual telepointer state and the prediction exceeds a certain threshold (see Figure 2).

One of the essential elements of a telepointer that supports group awareness is the trace that is the interaction history of the other participants. Traces can show the recent movements of a person in the shared work area.

The origin of this technique is cartoons. Artists in this field have long thought of solving the problem of showing the history of movements in a convincing and understandable way, even with low rates of static images. The best idea is the lines of motion. Its principle is to use one or more lines tracing the path of the moving object. This technique has several advantages for visualizing previous movements of and therefore to understand the activities of other participants. But it also suffers from several problems, especially in the case of presence of several participants in the same work area [DGM12] [GDB03].

Indeed, we cannot reason with the history of all users at the same time. On the other hand, the user prefers to examine the previous facts of all users to better understand its activities and objects handled. For this reason, we combined this technique with that of radar view. The principle of our proposal is to let the user define the number of participants present in his work area to switch to radar view [GG16] [SMI92].

In computer dictionaries the radar view means the view that allows to locate the other collaborators and to measure the extent of their zone of activity. But in this article, one more of these functions, it shows the trace of a selected telepointer and also offers the user the opportunity to rebroadcast the activities of the participant in question for a specified time previously. Thanks to this technique, the user's mind will not be overloaded by unnecessary information but he only sees the desired information with a quantity relating to his reflection speed.

5.2 History of Objects

In software where group members do not intervene simultaneously (asynchronous work) [H+04], it is essential to know what changes were made during our absence. And also to adjust our future actions with the work done, we must at least know the modified objects. Therefore, for each new connection, the system must present to the user the history of objects in its work area.

The history must include, the list of participants who have tried or modified an object, the different steps that have undergone the object while arriving at the current state and with what means are carried out these modifications.

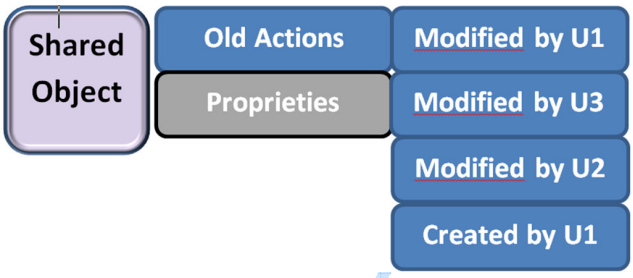

Fig. 3. Object's history in CSCW

The object's history implements a detail level management for viewing changes. The level is directly adjusted by the user according to his needs (see figure 3). The first level presents simple information such as the date of the last modification and the number of modifications. On the other hand, in the last level (the most detailed), you can find all information about the changes made to the object. Statistical information is also found as the most active member and the rate of change of the object by each participant in relation to the set. And always in this last level the user can rebroadcast video scenes.

5.3 History of avatars

Avatar is a graphical representation of a participant in a groupware. Several systems make it possible to choose an avatar and to avoid situations where the representation of two persons is identical by accompanying the avatar of discriminating information [KLB03] [SG04].

An avatar is not sufficient for an exhaustive representation of the identity of a participant. Other information should be (Email, phone, ...).

In order for the system to display a participant's unavailability, it is based on the importance of the tasks it is undertaking. But it is also necessary to let the user indicate his availability in an explicit way since any time the system decisions are relative.

We apply the same principle as presented in the previous section but this time on the avatars instead the objects. This time the participant activity rate will be represented by a curve in order to allow the other participants to extract information concerning the altitudes of the user (for example: the user is more active in the morning).

Information is very useful in group awareness; it is the ratio between the frequency of connection and the number of tasks performed by the user to distinguish the most assets.

Based on a filtering tool, the history of the avatars will be represented on several levels of detail. Ranging from simple information such as the frequency of connection when arriving at a level in which one indicates the objects manipulated by the user. In this last level, information about the workplaces from which the user has connected. This last information tells us the working context or the user is more active.

6. SOFTWARE ARCHITECTURE

6.1 Architecture 3 Servers

A toolbox must be independent of the software architecture. In this section, we propose a software architecture that allows us to better exploit the functionalities offered by our toolbox. The application consists of three parts of server type and one of client type. In reality the notion of client / server is exceeded since the exchange of service is done in both directions in the Most Internet applications.

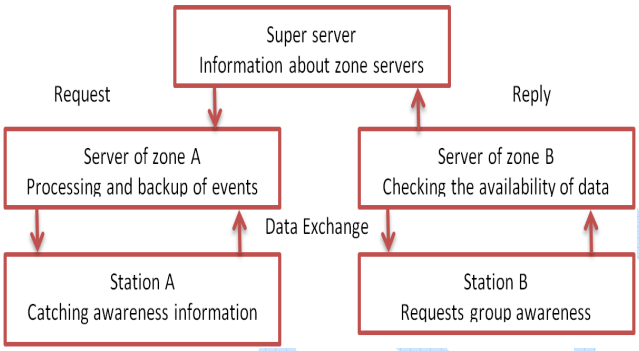

Fig. 4. Exchange of group awareness data between two work areas

The main server (super server) acts as coordinator. It ensures communication between the different work area servers. While the latter are used to save and broadcast messages and events (see Figure 4).

6.2 Managing Events and Messages

As mentioned, the main functions of the servers are the backup and dissemination of events and messages. These functions ensure the exchange of awareness information among the participants according to an approach based on the user role and its working area.

A copy of the participant's records is saved in the work area server. The latter remains attentive to the requests of the participants to offer them the information in question. To reduce the size of information changed in the network, the client receives only the result of its query. The data is processed before it is saved in the database to facilitate and minimize the execution time of the queries.

As in the case of telepointer messages, a compression algorithm has been applied to some types of messages (the choice of type is predefined).

6.3 Viewing group awareness information

Two approaches will be used to visualize the group awareness information (graphic and textual). In the textual approach, to arrive at the representation of the group awareness data requested by the user in a clear and readable way, one must go through three steps: Data capture, transfer of data and finally filtering and classification of data. In this approach, data capture is done by identifying the object being manipulated and its current state (the values of its properties). All this data is transferred to the server for further processing. The classification of the data according to the chronological order serves only to minimize the execution time of the requests.

Data is an essential element especially to answer specific queries as which are the alterations have the object in question. The query result is displayed in a small transparent window that allows the user to continue to observe the current actions of other participants by reading the information that requested.

Using the information derived from the capture phase in the first approach, the group awareness information can be reproduced and represented graphically as well as in their production state. In the viewing phase, we will use the radar view which allows us to re-broadcast only the actions that concern only the selected participants.

7. FUTURE WORK

The first work remains to be realized is the coding phase using java. Even in the design phase, there is much work to be done. Either by improving objects by adding properties or Evaluation of the toolbox which allows to show all the functionality offered by the toolbox, it is necessary to create several software in various disciplines trying to exploit the maximum the properties and the methods of the objects. In the network side, we will work on problems of congestion and latency by applying the new filtering and compression algorithms. Another essential point in our future work is the proposition of software architectures which allow an optimal exploitation of the information of group awareness.

CONCLUSION

This article presents an introduction to the realization of a toolbox. It must be generic and allows the development of a groupware that supports the group awareness which represents an important part in the collaboration activity. The addition of group awareness to groupware is a classic problem. An attempt has been made to solve it by improving the interface and communication objects. In the first type, we have proposed software components for displaying the group awareness information. They are based on the representation of past information according to textual and graphic approaches. In the second type, an attempt has been made to minimize the size of the information exchanged between the different employees' stations by applying filtering and compression algorithms. The toolkit also saves the developers a lot of effort and makes the task of adding group awareness to a groupware quick and easy.

REFERENCES

[C+12] **B. Cahour, C. Nguyen, J. F. Forzy, C. Licoppe** - *Using an electric car: a situated, instrumented and emotional activity*, in Proceedings of the 30th European Conference on Cognitive Ergonomics, pp. 22-28, 2012.

[DB92] **P. Dourish, V. Belloti** - *Awareness and Coordination in shared workpaces*, in Proceeding of CSCW'92, Toronto, Canada, pp. 107-114, 1992.

[DGM12] **J. Dyck, C. Gutwin, D. Makaroff** - *Adaptive forward error correction for real-time groupware*, in Proceedings of the 17th ACM international conference on Supporting group work, pp. 121-130, 2012.

[D+13] **D. Decouchant, S. Mendoza, G. Sánchez, J. Rodríguez** - *Adapting groupware systems to changes in the collaborator's context of use*, Expert Systems with Applications, vol. 40, no. 11, pp. 4446-4462, 2013.

[EN12] **C. A. Ellis, N. Naffah** - *Design of office information systems*, Springer Science & Business Media, 2012.

[GG16] **C. Gutwin, S. Greenberg** - *Implications of we-awareness to the design of distributed groupware tools,* Computer Supported Cooperative Work (CSCW), vol. 25, 2016.

[GDB03] **C. Gutwin, J. Dyck, J. Burkitt** - *Using Cursor Prediction to Smooth Telepointer Jitter*, in Proceedings of the ACM Conference on Supporting Group Work, pp. 294- 301, 2003.

[HG04] **J. Hill, C. Gutwin** - *The MAUI Toolkit: Groupware Widgets for Group Awareness, Computer- Supported Cooperative Work*, vol. 13 no 5, pp. 539-571, 2004.

[H+04] **T. Herrmann, M. Hoffmann, G. Kunau, K. Loser** - *A modelling method for the development of groupware applications as socio-technical systems*, in Behaviour and Information Technology, vol. 23 no. 2, pp. 119-135, 2004.

[KLB03] **M. Kirsch-Pinheiro, J. V. Lima, M. R. S. Borges** - *A Framework for Awareness Support in Groupware Systems*, Computer in Industry, Elsevier, vol. 52, no. 1, pp. 47-57, 2003.

[RMG13] **N. A. Romero, P. Markopoulos, S. Greenberg** - *Grounding privacy in mediated communication*, Computer Supported Cooperative Work (CSCW), vol. 22, no. 1, pp. 1-32, 2013.

[Smi92] **R. Smith** - *What You See Is What I Think You See*, SIGCUE Outlook, pp. 18-23, 1992.

[SG04] **K. Schneider, C. Gutwin** - *Supporting Group Awareness in Distributed Software Development*, in Proceedings of IBM Eclipse Workshop, International Conference on Software Engineering, EHCI/DS-VIS, pp. 383-397, 2004.