Context-Based Awareness in Group Work

P. Brézillon[2], M. R.S. Borges[1], J. A. Pino[3] and J.-Ch. Pomerol[2]

[1]NCE, Universidade Federal do Rio de Janeiro, CP 2324, 20001-970 Rio de Janeiro, Brazil (mborges@nce.ufrj.br)
[2]LIP6, Université Paris 6, 8 rue du Capitaine Scott – 75015 Paris, France {Patrick.Brezillon , Jean-Charles.Pomerol}@lip6.fr
[3]DCC, Universidad de Chile, Casilla 2777 Santiago, Chile (jpino@dcc.uchile.cl)

Abstract

The concept of context can be used with advantage in the area of Computer-Supported Cooperative Work. For many years, the awareness term has been used in this area without explicit association to context. This paper attempts to clarify their relationship. In particular, a framework is proposed to understand context and awareness as connected to other concepts used in group work as well, such as user interface and storage. The framework is useful to consider some groupware systems from the context perspective and to eventually obtain some insight on possible improvements for users.

Introduction

The terms context and awareness have been used together in several publications in the area of CSCW, sometimes meaning similar ideas and others complementing each other. In some cases they are presented as quite different and conflicting concepts (Dourish, 2001). The concept of context has many meanings, depending on the area it comes from (Brézillon, 1999; Moran and Dourish, 2001). Bazire, Brézillon and Tijus (2003) show that all the definitions found on the web can be assembled around six questions: Who? What? When? How? Where? And Why?

Now, research deals with different aspects of the concept of context at the highest level of knowledge and reasoning.

When one reads articles in the CSCW forum, a number of issues appear related, directly or indirectly, to the concept of context. Awareness mechanisms and awareness information are the common terms to deal with context in groupware. The group memory also combines both context and content information, sometimes without an appropriate relationship between them. In groupware, one must deal with several contexts at different granularity: The context of the group (why this group is constituted), the individual contexts of the members (e.g. their technical origins), and the context of the project (e.g. the product to build).

Awareness is one of the mechanisms of groupware aimed at *providing* context to group members. Consider a session in a cooperative editor, for example. The part of the text a team member is working on is contextual information that is made available to other members connected to the session through a user interface device called telepointer. The telepointer is managed by an awareness mechanism that receives the information from the cursor sensor.

In order to address these problems we propose a framework representing the groupware mechanisms associated with an explicit representation of context. We believe that this model is useful not only to understand the use of contextual information but also its relation to components of groupware applications. Without properly separating and understanding these concepts, the groupware may not only confuse the user but also miss an important opportunity to improve the results of group work. We believe that the explicit representation of context, at various levels - individual, task and team – will bring much benefit to the interaction among group members. The general framework we propose will provide a representation of both context and awareness to induce an adequate treatment of them when developing groupware.

The rest of the paper is organized into five sections. In the two following section we review the concept of context, focusing on its use on group work, and the awareness concept, showing examples where it is confused with context. In the fourth section, we present the groupware model that combines the groupware components with contextual information. In the following section, we use our model to explain why some groupware fail in dealing with these concepts.

Context

Introduction

In real life, a context is a complex description of shared knowledge about physical, social, historical, or other circumstances within which an action or an event occurs. In order to fully understand many actions or events, it is necessary to have access to relevant contextual information. For example, understanding the action of “opening a window" depends on what is referred: a real window or a window on a graphical user interface (Rittenbruch, 2002).

In HCI, a definition is that a context feature is any information that can be used to characterize and interpret the situation in which a user interacts with an application at a certain time. In the context-aware applications area, Dey, Salber and Abowd (2001) define context as any information that characterizes a situation related to the interaction between humans, applications and the surrounding environment. In AI, Brézillon (1999) defines context as what does not intervene explicitly in a problem solving but constrains it. All these definitions are very close and mainly differ by their context of use.

Three types of context

At a given step of a task performing, Brézillon and Pomerol (1999) distinguish between the part of the context, which is relevant for the current focus of attention, and the part, which is not relevant. The latter part is called *external knowledge*. The former part is called *contextual knowledge* because it has strong connections with the current focus although not directly considered in it. Always at a given focus, part of the *contextual knowledge* is proceduralized. This *proceduralized context* is a part of the *contextual knowledge, which* is invoked, organized, structured, and situated according to the focus and used in the task performing at this focus.

The context evolves with the focus. This dynamic of the context can be identified at the level of a movement between the contextual knowledge and the proceduralized context. Thus, a part of the context is static, e.g. the context at a step of the focus of attention is defined by a fixed number of contextual elements and a fixed proceduralized context, but the overall focus of attention is associated with a dynamic context through this movement between the contextual knowledge and the proceduralized context.

Static and dynamic parts of the context are intertwined and must be considered jointly.

A bi-dimensional representation of context

Brézillon (2003a) points out that it is possible (1) to identify different types of context, and (2) to organize them in a two-dimension representation, namely in depth first, from the more general to the more specific and in width first, as a heterogeneous set of contexts at each level.

In "depth first", contexts are different by their granularity, mainly in highly organized systems. For example, the context of an enterprise (with its tradition, habits, rules, etc.) is more general (at a higher level) than the context of an employee. In this case, context has strong relationships with the enterprise organization in terms of roles (Brézillon and Marquois, 2003). According to its depth, a context contains more general information than contexts at a lower level. However, context at one level is not a simple instantiation of the context at the upper level (Brézillon, 2003b). A context is like a system of rules (constraints) for identifying triggering events and for guiding behaviors in lower contexts. A context at one level contains contextual knowledge when the application of rules at the lower levels develops proceduralized contexts tailored to the lower context. A context (the contextual knowledge part) is like a frame of reference for the contexts below it. For example, a French driver going into Great Britain knows that he must drive on the left side of the road (contextual knowledge in the context of the country), and develop a special attention on his driving (proceduralized context in his individual context).

In "width first", each actor has its own context. For example, a European project meets together different specialists from different countries, each with his culture, his habits for working, etc. Actor's context contains information on the reasons for his move, the results of his meeting with the customer, etc. The context of the software agent possesses information on the available means for the accomplishment of the task, the access restriction to the databases, a user model, etc. For a given granularity of the context, there is thus a set of contexts rather heterogeneous, and the horizontal movement from one individual context to another one goes through either the upper context (e.g. the group context) or a lower context (e.g. the project context). Note that at the group level, a group is, recursively, like an actor with his individual context and interacting with other groups in other contexts. Brézillon, Adam and Pomerol (2003) discuss the importance for a firm to develop a coherent individual context for evolving in a market and facing concurrency.

Context and Awareness in Groupware

Awareness in Group Work

The concept of awareness has been widely used in CSCW research and applications (Borges and Pino, 1999). A well-known definition by Dourish and Bellotti (1992) states that awareness is a*n understanding of the activities of others, which provides a context for your own activity.* Furthermore, they say 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 group goals and progress.*

Awareness is a key activity when the work is cooperatively done by a group of people in a computer-supported environment. Otherwise, there will not be actual joint work, but an incoherent set of isolated pieces. However, people can do parallel work and still be part of a collaborative effort: the *divergent* activity will probably be followed by a *convergent* synchronizing activity (Sharples et al., 1993).

Awareness information may be about people who are working together as well as on their doings. However, one has to deal with the difficult tradeoff between providing information and the privacy people are entitled to have. Awareness on what co-workers have done or are doing is also very important (Gutwin, Roseman and Greenberg, 1996).

In some cases, the result of individual work needs to be known by the rest of the co-workers. In others, it is *meta-information* or *aggregated meta-information* what is needed. Consider the example of a distributed asynchronous discussion on a certain subject being done by a group of people. Every person should read each contribution of the other participants in the discussion, but also it is relevant to provide the reader easy-to-grasp information on which contributions are *new* and which are *unread* from the last session, the person logged in (meta-information), etc. Furthermore, the person may also appreciate if the system tells him how many contributions he has made and how that relates to the number of contributions provided by the other participants (aggregated meta-information).

When we contrasted to the previous discussion about context, we observe that a group member needs to have some knowledge about other members, but also the context in which this knowledge is operational. This allows each member to know about the other but also to interpret and extrapolate the other's behavior.

Context-Based Awareness in Group Work

Pomerol and Brézillon (1999) discuss the *transformation of contextual knowledge into some functional knowledge or causal and consequential reasoning in order to anticipate the result of* actions. Data are facts, which have not been analyzed or summarized yet (e.g., see Watson, 1998); information is data processed into a meaningful form, and knowledge is explained as the ability to integrate the information in his body of knowledge.

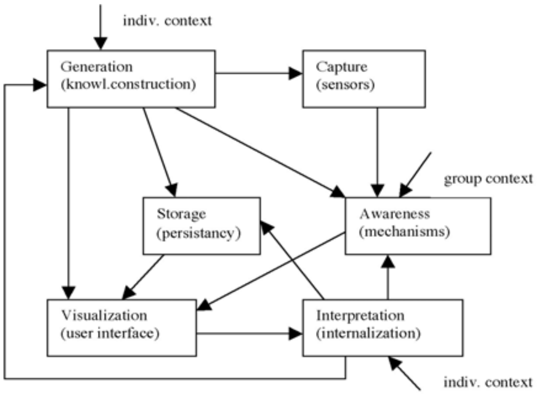


Figure 1. Context for knowledge processing in group work

Working in a group supposes to manage context explicitly. Not only individual contexts need to be proceduralized, but also the group context. Group context involves all the knowledge relating to the group, including group composition, rules, roles, goals, strategies, coordination procedures, etc. Therefore, group context is not simply the union or intersection of individual contexts (the whole is not the sum of the parts).

How is context processed when doing group work? Figure 1 illustrates the framework we propose. It essentially presents a *knowledge processing* procedure. People individually create knowledge, which is communicated to the rest of the group as well as being presented in a user interface and eventually stored. The *generation* step consists of a person contributing information to the group. Of course, this information may be contents for the group's output or related information, such as questions, suggestions, or procedural proposals. Some of this information is stored, according to pre-established conditions, e.g., "all contents information must be stored".

The *capture* step consists of procedures to gather some physical data from the generation step. For instance, in the case of joint text editing, the movement of the user's mouse may be of interest in some cases, since it may provide information on which part of the document the user is intending to work. In another example, a camera may capture the physical movements of a person; these movements may be important for another user who may be wondering why the first person is not answering her questions. A third example of capturing interesting knowledge may be the presence/absence of users during a work session. Contextual information can be obtained by different means.

The *awareness* step consists of the processing of information to provide it to the other participants. Note that it has several inputs. The first is information from the generation step. An example would be a contribution just written by a group member. This information needs to be transformed in some way, perhaps summarized or filtered to make it available to other people. In fact, it takes into account the processing specifications given by individual users. Another type of input is from the capture step; again, this information will probably be processed to avoid information overload. It also receives information from the storage step. This occurs, for example, when an agent decides to distribute a summary report on recent work in asynchronous systems. Finally, there is group context received as an input. This is needed as important information to process the rest of the inputs.

The *visualization* step provides the user interface. It gives users the physical representation of the knowledge: icons, text, figures, sounds, etc. Input to this step can come from the generation procedure: it is the physical feedback that a user receives when she contributes to the group through the system.

Capture, storage, awareness and visualization are all processing steps done by the system on the basis of user's specifications and pre-established rules. Besides generation, there is another human processing step. It is the interpretation process. A person performs this step when, taking into account the visualized information and her individual context, she assimilates the presented information into knowledge. Of course, this knowledge is important for the person to generate new contributions, and thus, closing the cycle of processing context to do participatory work within a group. A person may need some information from the storage, requesting it; this petition may be as simple as a mouse click over a button on the user interface or a complex query specification.

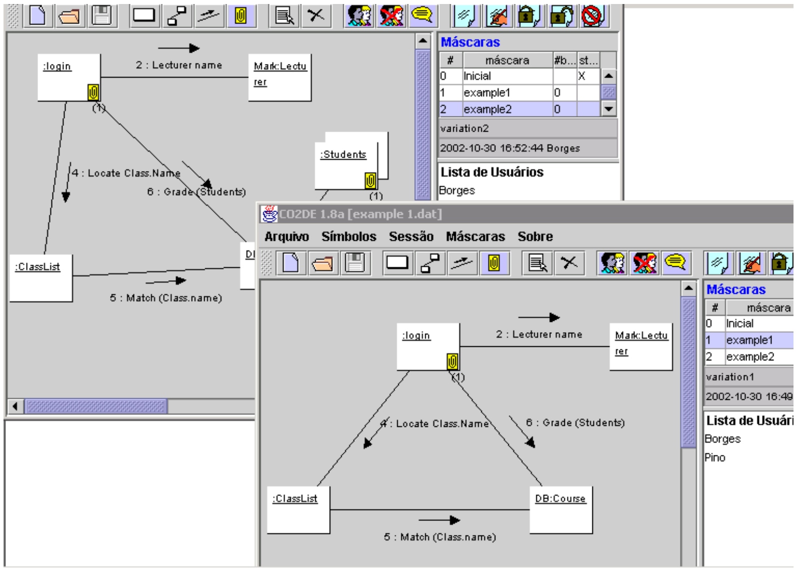
An interesting reduction occurs when a person is not working with a group but rather individually. In such a case, the awareness step must be eliminated; the capture may still be needed, although it may become trivial, and probably it will be directly presented on the user interface.

Context -Based Awareness in Applications

We developed several applications using context explicitly. First, we design and develop a system for incident management on a subway line (Brézillon, Pomerol and Pasquier, 2003). From this application, we develop now a context-based formalism (called Contextual Graphs) for representing knowledge and reasoning in context (Brézillon, 2004). We thus represent contextual elements about which operators are aware during an incident solving. We also developed two groupware systems: SISCO (Borges et al. 1999), a meeting preparation asynchronous system aimed at supporting the group discussion that occurs before an actual meeting; and CO2DE (Borges, Meire and Pino, 2003), a cooperative editor that supports multiple versions as a way to deal with conflicting views when building a diagram. Both systems support groups with a common task – a report on the opinions about meeting agenda items, in the case of SISCO, and versioning of a collaboration diagram in a software engineering project supported by CO2DE. This section analyzes how context is represented and used at different levels in the CO2DE system.

The CO2DE editor allows to join individual contexts into a single diagram by providing a synchronous cooperative edition facility and a WYSIWIS interface (see Figure 2). Although it also allows asynchronous interaction, it does not focus particularly on it. The diagram works as the persistency of the latest group context, in this case the union of individual contexts. However, the notion of context is not explicitly treated by CO2DE. The resulting diagram is considered simply as the result of a group work.

When conflicting views arise on elements of the diagram, most cooperative editors support users to reach a consensus by means of a communication mechanism, e.g. a chat. The resulting element is then expressed in the diagram associated with the corresponding discussion.



Máscaras

#	máscara	#b...	st...
0	Inicial		X
1	example1	0	
2	example2	0	

variation2

2002-10-30 16:52:44 Borges

Lista de Usuários

Borges

Máscaras

#	máscara
0	Inicial
1	example1
2	example2

variation1

2002-10-30 16:49

Lista de Usuári

Borges

Pino

Figure 2. The user interface in the CO2DE system

CO2DE has a different approach to deal with conflicts. It allows several versions of the diagram to co-exist. It organizes the versions into a tree to associate each version to its origin, its alternative versions resulting from conflict and its further decomposition originated from another conflict. In none of these cases, however, the system represents contextual information, for example, what was the conflict and which assumption a version was based on. This information is kept within each individual context and is not stored by the system. If a person wants to understand the rationale behind the creation of a new version, he has to ask its creator.

During the elaboration of the diagram, several versions may co-exist. It is left to participants to solve the conflicts and express the resulting consensus in a single version. One may argue that this is similar to solve the conflicts as they arise. The CO2DE approach has the advantage of allowing users to represent their personal views in a more comprehensive format, since a single conflict in many cases involves several elements of the diagram. It is like discussing two or more options using the complete picture, instead of discussing element by element. Another advantage is the representation of the work evolution by means of a set of refined versions.

The approach also supports a mental comparison of two alternatives. With a simple click of the mouse one can rapidly perceive the differences between diagrams.

The framework presented in the previous sections helps to visualize a possible improvement to CO2DE. When many versions of a diagram are present, it would be nice to have the rationale of each version stored with it, since even its creator may forget it and also for convenience. Naturally, this explanation should be hidden behind the diagram version representation to avoid information overload. Note this context is not awareness information. The system should be extended to handle these hidden explanations and allow the user to retrieve them by clicking over certain button in the version representation.

Conclusion

Work on context and CSCW has largely been done independently. One could think this has not been a good idea for groupware designers, who might benefit from research in contexts. The framework presented in this paper may be a first step to narrow that gap by relating the concepts of context and awareness to other terms widely used in CSCW, such as user interfaces, automatic capture and storage.

The framework presents group work as a knowledge-processing job with some activities possible to do by machine as support to the human tasks. This dataflow-type modeling is novel, as well as the presentation of context as the knowledge flowing among processing activities.

The framework can be applied to get some insight into some groupware designs. In particular, by considering context as knowledge to be applied during the group work, one can have a wider perspective than just focusing on the information provided to users by awareness mechanisms, as illustrated in the previous chapter. Many other groupware designs would probably be possible to analyze from this viewpoint.

Acknowledgments

This work was partially supported by grants from: CNPq (Brazil) No. PROSUL AC-62, Fondecyt (Chile) Nos. 1000870 and 7000870.

References

Bazire, M., Brézillon, P. and Tijus, Ch. 2003. Eléments intervenants dans le contexte d'une activité finalisée. Symposium "Tâche, Activité et Contexte" (TAC). In: J.M.C. Bastien (Ed.), Actes des Deuxièmes Journées d'Etude en Psychologie Ergonomique – EPIQUE'2003, INRIA, pp. 281-286.

Borges, M., and Pino, J.A. 1999. Awareness Mechanisms for Coordination in Asynchronous CSCW", Proceedings of the 9th Workshop on Information Technologies and Systems (WITS '99), Charlotte, North Carolina, , pp. 69-74.

Borges, M., Meire, A.P., and Pino, J.A. 2003. An interface for supporting versioning in a cooperative editor, Proceedings of the 10th International Conference on Human - Computer Interaction, Crete, Greece.

Borges, M.R.S., Pino, J.A., Fuller, D., Salgado, A.C. 1999. Key issues in the design of an asynchronous system to support meeting preparation", *Decision Support Systems*, 27(3): 271-289.

Brézillon P. 1999. Context in problem solving: A survey. *The Knowledge Engineering Review*, 14(1): 1-34.

Brézillon, P. 2003a. Individual and team contexts in a design process. Proceedings of the 36th Hawaii International Conference on Systems Sciences, HICSS-36, Track "Collaboration Systems and Technology", January, 2003, R.H. Sprague (Ed.), Los Alamitos: IEEE, CD-Rom.

Brézillon, P. 2003b. Using context for Supporting Users Efficiently. Proceedings of the 36th Hawaii International Conference on Systems Sciences, HICSS-36, Track "Emerging Technologies", January, 2003, R.H. Sprague (Ed.), Los Alamitos: IEEE, CD-Rom.

Brézillon, P. 2004. Representation of procedures and practices in contextual graphs. *The Knowledge Engineering Review* (to appear)

Brézillon, P. and Marquois, E. 2003. Une approche centrée contexte de l'activité. Symposium "Tâche, Activité et Contexte" (TAC). In: J.M.C. Bastien (Ed.), Actes des Deuxièmes Journées d'Etude en Psychologie Ergonomique – EPIQUE'2003, INRIA, pp.263-268.

Brézillon, P. and Pomerol, J.-Ch. 1999. Contextual knowledge sharing and cooperation in intelligent assistant systems, *Le Travail Humain* 62 (3), PUF, Paris, 223-246.

Brézillon, P., Adam, F. and Pomerol, J.-Ch. 2003. Supporting complex decision making processes in organizations with collaborative applications - A case study. In: Favela, J. & Decouchant, D. (Eds.) Groupware: Design, Implementation, and Use. LNCS 2806, 261-276, Springer Verlag.

Brézillon, P., Pomerol, J.-Ch. and Pasquier, L. 2003. Learning and explanation in a context-based representation: Application to incident solving on subway lines. In: R. Jain, A. Abraham, C. Faucher and J. van der Zwaag (Eds.) Innovations in Knowledge Engineering. International Series on Advanced Intelligence.

Dey, A.K., Salber, D. and Abowd, G.D. 2001. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, *Human-Computer Interaction*, 16 (2-4): 97-166.

Dourish, P. 1992. Seeking a Foundation for Context-Aware Computing. *Human-Computer Interaction*, 16(2-4).

Dourish, P. and Bellotti, V. 1992. Awareness and Coordination in Shared Workspaces. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work CSCW'92*, Toronto, Ontario, ACM Press, New York, 107-114

Gutwin, C., Roseman, M. and Greenberg, S. 1996. A usability study of awareness widgets in a shared workspace groupware system. CSCW 96 Conference Proceedings. ACM Press, New York, 258-267.

Moran, T.P. and Dourish, P. 2001. Introduction to this special issue on Context-Aware Computing. *Human-Computer Interaction*, 16(2-4): 87-94.

Pomerol J-Ch. and Brézillon P. 1999. Dynamics between contextual knowledge and proceduralized context. In *Modeling and Using Context* (CONTEXT-99), Lecture Notes in Artificial Intelligence n° 1688, 284-295, Springer Verlag.

Rittenbruch, M. 2002. ATMOSPHERE: A Framework for Contextual Awareness, *International Journal of Human-Computer Interaction*, 14 (2): 159-180.

Sharples, M., Goodlet, J., Beck, E., Wood, C., Easterbrook, S. and Plowman, L. 1993. Research issues in the study of Computer Supported Collaborative Writing. In: M. Sharples (ed.): *Computer Supported Collaborative Writing*. Springer-Verlag, London, 9-28.

Watson, R.T. 1998. Data Management. *Databases and Organizations*. John Wiley and Sons, Inc., Second edition, New York.