Groupware System Design and the Context Concept

Marcos R.S. Borges[1,*], Patrick Brézillon[2], Jose Alberto Pino[3],
and J.-Ch. Pomerol[2]

[1] NCE&IM, Universidade Federal do Rio de Janeiro, Brazil
mborges@nce.ufrj.br
[2] LIP6, Université Pierre et Marie Curie, France
{Jean-Charles.Pomerol, Patrick.Brezillon}@lip6.fr
[3] DCC, Universidad de Chile, Chile
jpino@dcc.uchile.cl

Abstract. The concept of context can be advantageously applied to the Computer-Supported Cooperative Work field. The term *awareness* has traditionally been used in this area without explicit association to context. This paper attempts to clarify the relationship between these two concepts. In particular, a framework is proposed to understand context and awareness as connected to other concepts used in group work as well. The framework is useful to consider some groupware systems from the perspective of context and to obtain some insight on possible improvements for users. Two examples illustrate the application of the framework.

1 Introduction

The concept of context has not been well understood in the Computer-Supported Cooperative Work (CSCW) field. Context has been used in several publications in the area, but with several different meanings associated to it [8]. CSCWD (Computer-Supported Cooperative Work in Design) is a good example where context plays a role in the specialization of an area. Specialization, in this case, means the knowledge related to applying CSCW techniques in the area of Design. Nevertheless, contextualization seems so natural that people often lose sight of its real significance.

The meaning of the context concept depends on the subject area [6], [13]. On the one hand, there have been several conferences on modeling and the use of context since 1997 [7]. These events deal with aspects of context at the highest level of knowledge and reasoning. However, this approach rarely takes the practical aspects of context in real-world applications, such as collaborative work, into consideration. On the other hand, in CSCW articles, several issues point to context without referring to it as such. Context has been applied in group work and is usually associated with awareness mechanisms. Few groupware systems use the context concept to guide design decisions, leaving it to be processed mostly by users. Most misunderstandings are caused by not explicitly recognizing and representing the notion of context and its association with other elements of groupware systems.

We present a framework for understanding the concept of context in group work, and we also discuss the application of context in the area of CSCW. Our aim is to

---

* On sabbatical leave at DSIC-Polytechnic University of Valencia, Spain.

guide the designer to the systematic use of context when developing an application [3]. We believe this model can be useful not only to understand the use of contextual information but also to relate components of groupware systems.

This paper is structured as follows: Section 2 reviews the concept of context; Section 3 presents a framework for understanding how groupware issues relate to context; Section 4 presents the groupware model for awareness mechanisms [3]; Section 5 uses the model to show cases where groupware fails in dealing with this concept; and Section 6 presents our conclusions.

2   Context

Context in real life is a complex description of knowledge about physical, social, historical, or other circumstances within which an action or an event occurs. Access to relevant contextual information is required in order to understand many actions or events. Understanding the "opening a window" action, e.g., depends on whether a real window, or a window on a graphical user interface is referred to [17]. It is possible (i) to identify various context types, and (ii) to organize them in a two-dimensional representation: vertically (i.e., depth first), from more general to more specific; and horizontally (i.e., width first), as a heterogeneous set of contexts at each level [5].

In the vertical representation ("depth first"), there are different contexts defined by their level of generality, mainly in highly organized systems. For example, the context of a building is more general (a higher level) than the context of an office. Contexts at a higher level contain general information, while contexts at a lower level contain more specific information. A context is like a system of rules (constraints) to identify triggering events and to guide behaviors in lower contexts. Based on Brézillon [4], it can de observed that a context at a general level contains contextual knowledge. The application of rules at more specific levels develops proceduralized contexts. A higher context is like a frame of reference for the contexts below it.
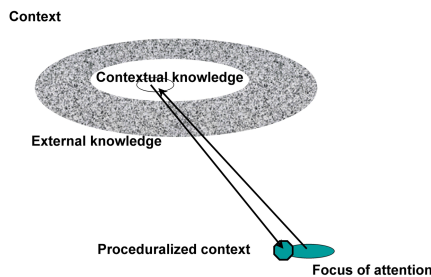

Fig. 1. Contextual knowledge and proceduralized context [6]

Each actor has its context in the horizontal representation ("width first"). The user's context contains specific information; for example, the results of a meeting with a customer, the reasons for changing offices, etc. The context of a communicating object contains knowledge about its location, and how to behave with the other

communicating objects. Thus, at a given level of the context hierarchy, there is a set of heterogeneous contexts.

Pomerol and Brézillon [16] distinguish between the non-relevant and the relevant parts of the context for each step of a task. The non-relevant part is called *external knowledge*. The relevant part is called *contextual knowledge*. At a given step, a part of the *contextual knowledge* is proceduralized. The *proceduralized context* is the part of *contextual knowledge* that is invoked, structured and situated according to a given focus (Figure 1). Proceduralization means that people use contextual knowledge in functional knowledge or causal and consequential reasoning. This proceduralization fulfills the need of having a consistent explicative framework to anticipate the results of a decision or action. This consistency is obtained by reasoning about causes and consequences in a given situation [14].

There are several views of context: context as conceptual drift (a context engine); context as a medium for the representation of knowledge and reasoning; context as what surrounds the focus of attention, etc. All these context concepts have been formalized and used in knowledge-based applications. However, these views are rather isolated. An analysis of shared context and its use in group work is also necessary. In the following section we present a framework that can be considered as a first step towards this goal.

3   Understanding Context in Group Work

A context may be seen as a dynamic construction with five dimensions: (1) time, (2) usage episodes, (3) social interactions, (4) internal goals, and (5) local influences [10]. Although the contextual elements in some situations are stable, understandable, and predictable, there are some situations where this does not occur. Cases having apparently the same context can be different. In order to reduce this, we use a conceptual framework whose objective is to identify and classify the most common contextual elements in groupware tools [18]. The goal of this framework is to provide guidelines for research and development in groupware and context.

According to McCarthy [11], the size of the contextual dimension is infinite. Thus, the framework considers only those contextual elements that are most relevant to task-oriented groups, i.e., contextual knowledge and proceduralized context [4]. The contextual information is clustered into five main categories: (1) people and groups, (2) scheduled tasks, (3) the relationship between people and tasks, (4) the environment where the interactions take place and (5) the tasks and activities that have already been completed. These clusters were borrowed from the Denver Model [19]. In synchronous environments, group members need to work at the same time; however, in asynchronous environments there might be a time lag between interactions. The needs of each type of environment are different, especially those that are related to contextual information [15].

The framework is a generic classification of contextual elements. It does not cover the peculiarities of a certain domain nor does it apply to a specific type of groupware. This generic framework is a starting point for a classification of contextual elements in specific domains, where new contextual elements may be considered relevant.

The first category provides information about the group members; it contains information about the individuals and the groups they belong to. The knowledge about the group's composition and its characteristics is important to be able to understand the potential ways in which the project or the task will be developed. This knowledge encourages interaction and cooperation [15]. This category is sub-divided into two types of context. The individual context carries information about each of the individuals who are members of a group. The group context data is similar to the aforementioned, but relates to the group as a whole. It includes the composition of the team, its abilities and previous experience as a group, and the organizational structure.

The second category provides information about scheduled tasks. Independently of how the interaction occurs, the group members need to be acquainted with the task characteristics. Task context is the name given to this context. Its goal is to identify tasks through their relevant characteristics: the task name, its description and goals, the deadline, the predicted effort, the technology, and other requirements.

The third category provides information about the relationship between the group members and the tasks. The goal of this category is to relate the action of each group member and the interaction s/he is involved in. This interaction begins with an execution plan, goes through a sequence of actions required to carry out the plan, and terminates when the task has been completed. If the interaction is interrupted before the task is completed, the reasons for the premature termination also form part of the context and are relevant to understanding the reason for the interruption.

This category is sub-divided into two types of contexts: the interaction context and the planning context. The interaction context consists of information that represents the actions that took place during task completion. When the interaction is synchronous, the details of the activity must be known at the time that it occurs. When the interaction is asynchronous, the overview of activities is what is most relevant.

The planning context consists of information about the project execution plan. This information can be generated at two different points. For ad-hoc tasks, the information appears as a result of the interaction. For scheduled tasks, it is generated at the time of the plan, i.e.; when the tasks are defined and the roles are associated to them. The planning context can include rules, goals, deadline strategies, and coordination activities.

The fourth category provides information on the environment. It represents the aspects of the environment where the interaction takes place. It covers both organizational issues and the technological environment; i.e., all the information outside the project (but within the organization) that can affect the way the tasks are performed. The environment gives further indications to group members about how the interaction will occur; for instance, quality control patterns are part of this context. This context also includes strategy rules, policies, financial restrictions and institutional deadlines.

The last category provides all the information about tasks that have already been completed. The goal of this category is to provide background information about the experiences learned either from the same group or similar tasks performed by other groups. It should include all contextual information about previous projects. The framework refers to this set of information as "historical context". This information is important for understanding errors and to be able to apply successful approaches from previous projects to current tasks. It can also be used out of the context of a project to

provide insight into working practices and team cooperation. A summary of the framework is shown in Table 1.

Table 1. Conceptual framework for the analysis of context in groupware [18]

Information type	Associated Contexts	Goals	Examples of contextual elements
Group Members	Individual (Synchronous & Asynchronous)	To identify the participants through the representation of their profiles.	• Name • Previous experience • Working hours
	Group (Synchronous & Asynchronous)	To identify the group through the representation of its characteristics	• Members • Roles • Organizational Structure
Scheduled Tasks	Task (Synchronous & Asynchronous)	To identify the tasks through the representation of their characteristics.	• Goals, deadlines • Estimated effort • Activities
Relationship between people and tasks	Interaction (Synchronous)	To represent in detail the activities performed during the task completion.	• Exchanged messages • Presence Awareness • Gesture awareness

	Interaction (Asynchronous)	To represent an overview of the activities performed during the task completion.	• Artifacts generated
			• Activities completed
			  • Author
			  • Results

	Planning (Synchronous & Asynchronous)	To represent the execution plan of the task to be performed.	• Interaction roles • Rules • Strategies • Procedures
Setting	Environment (Synchronous & Asynchronous)	To represent the environment where the interaction occurs; i.e., characteristics that influence the task execution.	• Quality patterns • Policies • Financial constraints • Standard procedures
Completed Tasks	Historical (Synchronous & Asynchronous)	To provide understanding about tasks completed in the past and their associated contexts.	• Task Name • Versions of the artifacts • Contextual elements • Working Plan

4   Context and Awareness in Groupware

Proceduralization of context involves the transformation of contextual knowledge into some functional knowledge or causal and consequential reasoning in order to anticipate the result of actions [16]. When people work as a group, context becomes especially relevant. Not only do individual contexts need to be proceduralized, but so does the group context. As described in the framework, group context is not simply the union or intersection of individual contexts. For instance, a specific person may work differently with a certain group of colleagues than with another one.

How is context processed when doing group work? Fig. 2 shows our proposed model. It is basically a *knowledge processing* procedure. People create knowledge individually. It is then communicated to the rest of the group as well as being presented in a User Interface (UI) and eventually stored. The *generation* step consists of a person contributing information to the group. This information could be contents

for the group's output or it could be related information, such as questions, suggestions, or proposals. Part of this knowledge is stored to satisfy the condition "all contents must be saved".

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 serve as an indication of which part of the document the user is working on. In another example, a camera can capture the physical movements of a person; these movements might be important for another user, who may be wondering why the first person is not answering his/her questions.
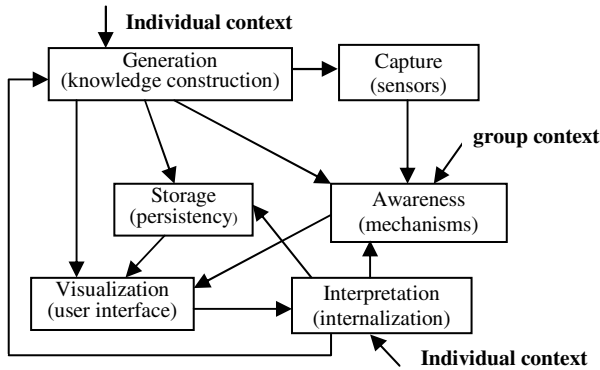

Fig 2. Context knowledge processing in group work [3]

The *awareness* step consists of the processing of information to be communicated to the other participants [9]. Note that it has several inputs. The first input is information from the generation step. An example would be a contribution that has just been 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, this step 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. The awareness step 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, it should also be noted that there is a group context that is received as input. This represents important information that is needed to process the rest of the inputs.

The *visualization* step generates the user interface. It provides users with a physical representation of knowledge: icons, text, figures, etc. Input to this step can come from the generation procedure: the physical feedback a user receives when s/he contributes to the group.

Capture, storage, awareness and visualization are all processing steps that are performed by the system on the basis of users specifications and pre-established rules. Besides generation, there is another human processing step: the interpretation process. The person performs this step by visualizing the information and combining it with his/her individual context to transform it into knowledge. This is needed by the person to generate new contributions to the group and close the cycle of processing context. A person might need some information from storage and can request it; this petition

might be as simple as a mouse click on a button on the UI or it might be a complex query specification.

5   Contexts and Awareness in Practice

We use two groupware systems to illustrate the use of the framework and the contextual knowledge model: SISCO [2], a meeting preparation asynchronous system that is intended to support the group discussion occurring before an actual meeting; and CO2DE [12], a cooperative editor that handles multiple versions as a way to deal with conflicting views. Both systems support groups working with a common task. SISCO provides the organization of opinions about agenda items, and CO2DE provides one or more versions of a collaboration diagram in a software engineering project. Neither of the systems explicitly supports context, but they both use several context elements to support group work.

Notice that making context explicit is a way to remember, not only the way in which a solution was developed, but also the alternatives at the time of solution building, existing constraints, etc. Thus, awareness is achieved by comparing the context used at that time with the current context.

If the goal is to find a solution, it is also important to account for individual contexts. A specialist might propose a solution from his/her field of domain. Yet, another specialist may give constraints. In such a case, the first specialist will modify his/her context from the pair (problem, solution) to the triple (problem, context, solution). By working together, each person will be able to share more knowledge with the other members. Thus, their individual contexts will have a non-empty intersection, making their interaction short and efficient.

In SISCO, since the goal is to have a broad discussion, the selection is based on the contextual knowledge that each participant has about the meeting agenda items, as well as the diversity of individual contexts. The contributions are shared among group members to reduce repetitions and also to increase the quality of the contributions by making other participants' ideas explicit. This sharing promotes the internalization and idea generation processes. Since a repetition occurs when a person is working individually, the awareness step is dropped. The capture may still be needed, but it becomes trivial, and will probably just be presented on the UI.

SISCO must provide persistency of contributions to the discussion as well as awareness of the discussion contents. Whenever a member logs in, the system generates a schematic view of the discussion contents, indicating what is new to him/her. This keeps the contextual knowledge uniform among group members even when they have not connected to the system for long periods. Perhaps no one has complete knowledge of the contributions. Thus, the system must make contributions persistent and provide awareness mechanisms to allow users to update their individual contexts with the group context that are represented by the set of contributions.

The task context covers as much of the wide range of options and arguments related to the agenda items as possible. During the discussion, which is supported by SISCO using an IBIS-like argumentation model, most contributions are based on participants' individual context. Thus the authorship provides some hints about the associated context. SISCO also encourages participants to express not just their own

views, but to express those that are logically consistent with the task context. In this way, the system attempts to disassociate opinions from individual contexts and move them towards the task context. One way of achieving this is by removing authorship from the contributions.

Another way of supporting task context is through the definition of roles. When playing a role in SISCO, an individual is given a narrower context with specific awareness mechanisms. For instance, the coordinator role is provided with a *participameter*, a widget that informs about the level of participation in the discussion [1]. The *participameter* is considered a kind of group or task context and provides the coordinator with elements to decide on what to do. For example, when the participation level in a certain item is low the possible actions to be taken are: remind people, promote discussion, or even drop the item.
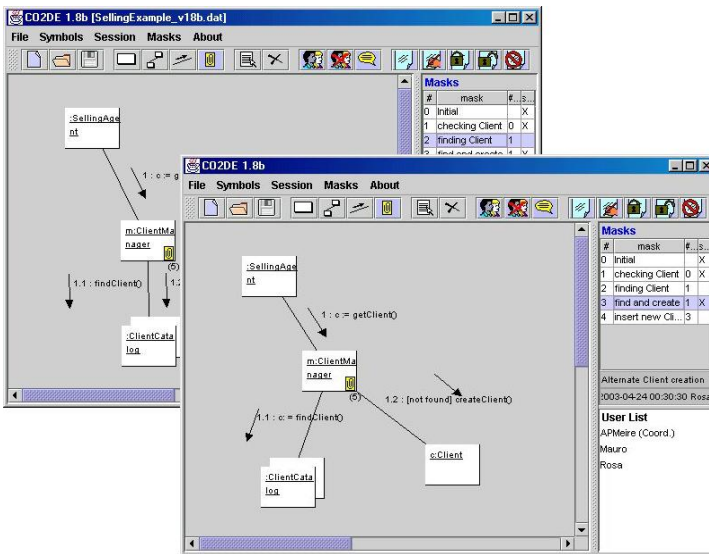

Fig 3. CO2DE user interface [12]

The CO2DE editor allows for individual contexts to be joined into a single diagram by providing a synchronous cooperative edition feature and a WYSIWIS interface (Fig. 3). Although this also allows asynchronous interaction, it does not focus on it. The diagram functions as the memory of the latest group context, which is the union of individual contexts. However, the context notion is not explicitly treated by CO2DE.

When conflicting views arise in a diagram, most cooperative editors encourages users to reach a consensus by means of a communication mechanism, e.g., a chat. CO2DE deals with conflicts in a different way. 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 the conflict, and its further decomposition originated from another conflict. In none of these cases, however, does

the system represent contextual information; e.g., the conflict and the assumptions for a version. This information is kept within each individual context and is not stored.

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. The CO2DE approach has the advantage of allowing users to represent their views in a more comprehensive format, since a single conflict usually involves several elements of the diagram. It is like discussing two or more options using the complete picture, instead of discussing each element one at a time. Another advantage is the representation of the work evolution by means of a set of step-refined versions. The approach also supports a mental comparison of two alternatives. With a simple click of the mouse the user can rapidly perceive the differences between diagrams.

The framework presented in this paper indicates a potential for improvement to CO2DE. When many versions of a diagram are present, it is desirable to have the rationale for each version stored with it, since even its creator may forget what it was. This context is not awareness information. The system should be extended to handle these explanations and allow the user to retrieve them by clicking on a specific button in the version representation. This is equivalent to the "requesting additional information" arrow from "Interpretation" to "Storage" in Figure 2.

6   Conclusions

The study of context and CSCW has largely been done independently. Perhaps this has not been beneficial for groupware designers, who might profit from research in contexts. This framework may be a first step in narrowing this gap by relating the concepts of context and groupware. The model representing how the awareness mechanism can carry contextual information illustrates how the notion of context is related to other widely used terms in CSCW, such as user interfaces, automatic capture, knowledge construction and storage.

The context process model presents group work as a knowledge-processing task that has some activities that can be performed by a machine as support to the human tasks. This dataflow-type modeling is novel. The presentation of context as knowledge flowing among different processing activities is also new.

The framework and the model can be applied together to obtain some insight into certain groupware designs. By considering context as knowledge that can be applied during group work, there can be a wider perspective than just focusing on the information provided to users by awareness mechanisms. Other groupware designs would probably be suitable for analysis and improvement from this viewpoint.

Acknowledgements

References

1. Borges, M.R.S., Pino, J.A.: Awareness Mechanisms for Coordination in Asynchronous CSCW. Proceedings of the Workshop on Information Technology and Systems. Charlotte, NC, USA (1999) 69-74
2. Borges, M.R.S., Pino, J.A., Fuller, D., Salgado, A.: Key issues in the design of an asynchronous system to support meeting preparation. Decision Support Systems, Vol. 27 No. 3. Elsevier (1999) 271-289
3. Borges, M.R.S., Brézillon, P., Pino, J.A., Pomerol, J-Ch.: Bringing Context to CSCW. Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design Vol II. Xiamen, China, IEEE Press (2004) 161-166
4. Brézillon, P.: Individual and team contexts in a design process. Proceedings of the Hawaii International Conference on System Sciences (HICSS-36). Hawaii, USA, IEEE Computer Society Press (2003) CD-R
5. Brézillon, P.: Using context for Supporting Users Efficiently. Proceedings of the Hawaii International Conference on System Sciences (HICSS-36), Hawaii, USA, IEEE Computer Society Press (2003) CD-R
6. Brézillon P.: Contextualizations in a Social Network. Context in Social networks and virtual communities. Revue d'Intelligence Artificielle, Vol. 19 No. 3. Hermès Science Publications, (2005) 575-594
7. Context 2005: http://www.context-05.org/ Accessed on 28 January 2005
8. Dourish, P.: Seeking a Foundation for Context-Aware Computing. Human-Computer Interaction, Vol. 16 No. 2-4. Lawrence Erlbaum (2001) 87-96
9. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proceedings of the Computer-Supported Cooperative Work Conference. ACM Press (1992) 107-114
10. Greenberg, S.: Context as a Dynamic Construct. Human-Computer Interaction, Vol. 16 No. 2-4. Lawrence Erlbaum (2001) 257-268
11. McCarthy, J., Notes on formalizing context, Proceedings of 13th International Joint Conference on Artificial Intelligence. Chambery, France, Morgan Kauffman (1993) 555-560
12. Meire, A., Borges, M.R.S., Araujo, R.: Supporting Collaborative Drawing with the Mask Versioning Mechanism. Proceedings of the 9th International Workshop on Groupware. Lecture Notes in Computer Science Vol. 2806. Springer-Verlag, Berlin Heidelberg New York (2003) 208-223
13. Moran, T.P. and Dourish, P.: Context-Aware Computing. Human-Computer Interaction, Vol. 16 No. 2-4. Lawrence Erlbaum (2001) 87-94
14. Naveiro, R., Brézillon, P., Souza, F. : Contextual knowledge in design: the SisPro project. Espaces Numériques d'Information et de Coopération, Vol. 5 No. 3-4. C. Simone, N. Matta & B. Eynard (Eds.). Hermès Science Publications (2002) 115-134
15. Pinheiro, M.K., Lima, J.V., Borges, M.R.S.: A framework for awareness support in groupware systems. Computers in Industry Vol. 52 No. 1. Elsevier (2003) 47-57
16. Pomerol J-Ch., Brézillon P.: Dynamics between contextual knowledge and proceduralized context. Lecture Notes in Artificial Intelligence Vol. 1688. Springer-Verlag, Berlin Heidelberg New York (1999) 284-295
17. Rittenbruch, M. ATMOSPHERE: A Framework for Contextual Awareness, Int. Journal of Human-Computer Interaction Vol. 14 No. 2. Lawrence Erlbaum (2002) 159-180
18. Rosa, M.G.P., Borges, M.R.S., Santoro, F.M.: A Conceptual Framework for Analyzing the Use of Context in Groupware. Proceedings of the 9th International Workshop on Groupware. Lecture Notes in Computer Science Vol. 2806. Springer-Verlag, Berlin Heidelberg New York (2003) 300-313
19. Salvador, T., Scholtz, J., Larson, J.: The Denver Model for Groupware Design, SIGCHI Bulletin Vol. 28, No. 1. ACM Press (1996) 52-58