Bringing Context to CSCW

M. R.S. Borges[1], P. Brézillon[2], J. A. 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 context concept 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. The framework is useful to consider some groupware systems from the context perspective and to obtain some insight on possible improvements for users.

1. Introduction

There has been little emphasis on the understanding of the context concept in the CSCW field. Context has been used in several publications in the area, but it is presented in many cases with different meanings [7]. CSCWD is a good example where the context plays an important role in the specialization of an area. The specialization in this case is defined by the term design, meaning that one is interested in studying the knowledge related to applying CSCW techniques in the area of Design. The contextualization, however, seems so natural that in many cases people lose its real significance.

The context concept has many meanings depending on the area it comes from [5, 12]. On the one hand, there is a series of interdisciplinary conferences on modeling and using context since 1997 [6]. These conferences deal with aspects of context at the highest level of knowledge and reasoning. However, this approach rarely considers practical aspects of context in real-world applications such as collaborative work. On the other hand, in CSCW articles, several issues point to context without being called as such. Context has been applied in group work usually associated with awareness mechanisms. However, few applications use the context concept to guide design decisions, leaving it to be processed mostly by the user. We believe that most misunderstanding is 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 discuss the application of context in the area of CSCW. We aim to guide the designer to the systematic use of context when developing an application. We believe this model is useful not only to understand the use of contextual information but also to relate components of groupware systems.

The paper is organized as follows. Section 2 reviews the context concept. Section 3 presents a framework for understanding how groupware issues relate to context. Section 4 presents the groupware model for awareness mechanisms [8]. In Section 5 we use the model to show some examples where a groupware fails in dealing with these concepts. Section 6 concludes the paper.

2. Context

A context in real life is a complex description of knowledge on physical, social, historical, or other circumstances within which an action or an event occurs. In order to understand many actions or events, it is necessary to have access to relevant contextual information. Understanding the "opening a window" action, e.g., depends on whether a real window is referred to or a window on a graphical user interface (UI) [16]. It is possible (i) to identify various types of context, and (ii) to organize them in a two-dimensional representation, vertically (i.e., depth first) from the more general to the more specific and horizontally (i.e., width first) as a heterogeneous set of contexts at each level [4].

In "depth first", there are different contexts defined by their degree of generality, mainly in highly organized systems. The context of a building is more general (higher level) than the context of an office. According to its depth, a context contains more general information than contexts at a lower level, but it has more specific information than contexts at an upper level. A context is like a system of rules (constraints) for identifying triggering events and for guiding behaviors in lower contexts. In the spirit of [3], one observes that a context at one level contains contextual knowledge, when the application of rules at the lower levels develops proceduralized contexts tailored to lower contexts. Conversely, an upper context is like a frame of reference for the contexts below it.

Each actor has its context in "width first". User's context contains information on the reasons for a move, the results of a meeting with a customer, etc. The context of a communicating object contains knowledge on its location, how to behave with the other communicating objects. At a given level of the context hierarchy, there is thus a set of heterogeneous contexts.

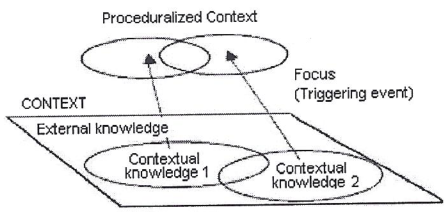

Figure 1. Contextual knowledge and proceduralized context [5]

Brézillon & Pomerol [5] distinguish the relevant and the non-relevant parts of the context at the step of the performing task. The latter part is called *external knowledge*. The former part is called *contextual knowledge*. At a given step of the task, a part of the *contextual knowledge* is proceduralized. The *proceduralized context* is a part of the *contextual knowledge*, which is invoked, structured and situated according to a given focus (Figure 1).

Proceduralization also means that people use contextual knowledge into some functional knowledge or causal and consequential reasoning. This proceduralization obeys to 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 [13].

There are various views: context as conceptual drift (a context engine), context as an implicit input to an application, context as a medium for the representation of knowledge and reasoning, context as what surrounds the attention focus, etc. All these context concepts have been formalized and used in knowledge base applications. However, these views are rather individual. An analysis of shared context and its use in group work is also necessary. In the next section we present a framework that can be seen 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 [9]. Although the contextual elements in some situations are stable, understandable and predictable, there are some situations when this does not occur. Cases with apparently the same context can be different.

In order to reduce this impact, we use a conceptual framework aimed to identify and classify the most common contextual elements in groupware tools [17]. The goal of the framework is to provide guidelines for research and development in groupware and context.

The conceptual framework considers the relevant elements for analysis of the use of context in groupware applications. The contextual information is clustered in five main categories: (1) people and groups, (2) scheduled tasks, (3) the relationship between people and tasks, (4) the environment where the interaction takes place and (5) tasks and activities already concluded. These clusters were borrowed from the Denver Model [18].

In synchronous environments, group members need to work at the same time, but in asynchronous environments, there might be a time lag between interactions. The needs for each type of environment are different, especially related to contextual information and the awareness required in each situation [14].

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

The various types of contextual information will be grouped by the framework into the five categories. According to McCarthy [10], the size of the contextual dimension is infinite. Thus, the framework considers only contextual elements most relevant to task oriented groups, i.e., contextual knowledge and proceduralized context [3].

The first category is the information about group members. It concerns information about the individuals and the groups they belong to. The knowledge about the group's composition and its characteristics is important for the understanding of the potential ways the project or task will be developed. The knowledge about the characteristics of individuals and the group as a whole encourages interaction and cooperation [14]. This category is further divided into 2 types of context. The individual context carries information about the individual who is a member of a group. It includes information about his abilities, interests, location, previous experience, personal data and working hours.

The group context carries information about the characteristics of the team. The data is similar to the aforementioned, but related to the group. It includes the composition of the team, its abilities and previous experience as a group, the organizational structure.

The second category is the information about scheduled tasks. Independent of how the interaction occurs, the group members need to be acquainted with task characteristics. Task context is the name given to this context. Its goal is to identify tasks through its relevant characteristics. Among these characteristics, we can select the task name, its description and goals, the deadline, the predicted effort, the technology and other requirements and pre-conditions.

The third category concerns the relationship between the group members and the scheduled tasks. Its goal is to relate the action of each group member and the interaction he is involved in. This interaction begins with an execution plan and terminates when the task has been concluded, passing through a sequence of actions required for carrying out the plan. If the interaction is interrupted before the task is concluded, the reasons for the premature ending are also part of the context and are relevant to understand the interruption rationale. This category is further divided into two types of contexts: the interaction context and the planning one. The interaction context consists of information representing the actions, which took place during the task completion. When interaction is synchronous, it is worth to be aware of details of the activity at the time it occurs, while in asynchronous interactions it matters to provide an overview of activities.

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

The fourth category groups information about the environment. It represents the aspects of the environment where the interaction takes place. It covers both the organizational issues and the technological environment, i.e., all information outside the project but within the organization that can affect the way the tasks are done. The environment gives some additional indications to group members about how the interaction will occur. For example, the quality control patterns are part of this context. Strategy rules, policies, financial restrictions and institutional deadlines are other examples of this context.

The last category gathers all information about concluded tasks. Its goal 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 calls this set of information "historical context". This information is important for understanding errors and successful approaches in previous projects to be used in 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.

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 [15]. According to commonly used definitions, data are facts, which have not been analyzed or summarized yet; information is data processed into a meaningful form, and knowledge is explained as the ability to use information.

When people are working as a group, context becomes especially relevant. Not only individual contexts need to be proceduralized, but also 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? Figure 2 shows our proposed model. It is basically a *knowledge processing* procedure. People individually create knowledge, which is communicated to the rest of the group as well as being presented in a UI 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 proposals. Part of this information is stored, according to previous conditions, e.g., "all contents 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 provide indication on which part of the document the user works. 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.

The *awareness* step consists of the processing of information to provide it to the other participants. Note 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, e.g., when an agent decides to distribute a summary report on recent work in asynchronous systems. Finally, notice there is group context received as input. This is needed as important information to process the rest of the inputs.

Table 1. Conceptual framework for the analysis of context in groupware applications [17]

Information type	Associated Contexts	Goals	Examples of contextual elements	
Group Members	Individual (Synchronous & Asynchronous)	To identify the participants through the representation of their personal data and profiles.	• Name • Qualifications • Interests • Formal Education	• Previous experience • Location • Working hours • Web page
	Group (Synchronous & Asynchronous)	To identify the group through the representation of its characteristics	• Name • Members • Roles • Abilities	• Previous experience • Organizational Structure • Location • Working hours
Scheduled Tasks	Task (Synchronous & Asynchronous)	To identify the tasks through the representation of its characteristics.	• Name • Description • Goals • Deadlines	• Estimated effort • Activities • Restrictions • Workflow

Relationship between people and tasks	Interaction (Synchronous)	To represent in detail the activities performed during the task completing.	• Group in-charge
			• Messages exchanged
			• Presence Awareness
				• Gesture awareness
				• Concluded Activities
				  • Author
				  • Goal
				  • Report

	Interaction (Asynchronous)	To represent an overview of the activities performed during the task completing.	• Group in-charge
			• Artifacts generated
			  • Versions
				• Activities completed
				  • Author
				  • Goal
				  • Report

	Planning (Synchronous & Asynchronous)	To represent the Execution Plan of the task to be performed	• Interaction roles • Rules • Aim • Responsibilities	• Strategies • Coordination Procedures • Working Plan
Setting	Environment (Synchronous & Asynchronous)	To represent the environment where the interaction occurs; i.e., characteristics that influence task execution.	• Quality patterns • Rules • Policies • Institutional deadlines	• Organizational structure • Financial constraints • Standard procedures • Standard strategies

Completed Tasks	Historical (Synchronous & Asynchronous)	To provide understanding about tasks completed in the past and their associated contexts.	• Task Name
			• Activities
			  • Author
			  • Goal
			  • Justification
			  • Date
				• Versions of the artifacts
				• Contextual elements used to carry out the task
				• Working Plan
				• Task Goals

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

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: the interpretation process. A person performs this step when, considering the visualized information and her individual context, she assimilates the presented information into knowledge. This is needed by the person to generate new contributions, 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 UI or a complex query specification.

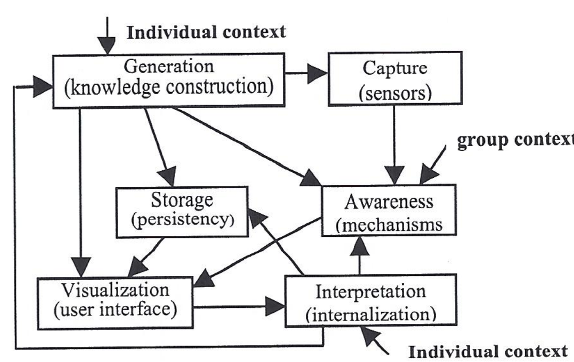

Figure 2. Context knowledge processing in group work

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 intended to support the group discussion occurring before an actual meeting; and CO2DE [11], a cooperative editor that handles multiple versions as a way to deal with conflicting views. Both systems support groups with a common task – opinions about agenda items (SISCO), and one or more versions of a collaboration diagram in a software engineering project (CO2DE). Neither of the systems supports context explicitly but they 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 context at that time and the current context.

If the goal is the realization of the solution, it is also important to account for individual contexts. A specialist can propose a solution from her field of domain. Yet, another specialist may give constraints. In such a case, the first specialist will modify her context to include the fact the pair (problem, solution) in her domain must be changed to the triple (problem, context, solution).

By working together, each person will increasingly share more experience with others. Thus, their individual contexts will have a non-empty intersection making their interaction short and efficient.

In SISCO, an individual context is used to select meeting participants, although this process is not supported by the system. The selection is based on the contextual knowledge each participant has about the meeting agenda items, as well as the diversity of individual contexts, as the goal is to have a broad discussion. The contributions are shared among group members to reduce repetitions and also to increase the quality of the contributions by making explicit other participants' ideas. This sharing promotes the internalization and idea generation processes.

A reduction occurs when a person is not working with a group but individually. Then, the awareness step is dropped; the capture may still be needed, but it becomes trivial, and probably it will be just 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 her. This keeps the contextual knowledge uniform among group members even when they stay disconnected from the system during long periods. Perhaps no one has a complete knowledge of the contributions. Thus, the system has to make contributions persistent and provide awareness mechanisms to allow users to update their individual contexts with the group context represented by the set of contributions.

The task context covers as much as possible the wide range of options and arguments related to the agenda items. During the discussion 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 own views, but those which are logically consistent with the task context. In this way the system intends to disassociate opinions from individual contexts and move them towards the task context. A way of achieving it is by removing authorship from the contributions.

Another form 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 example, the coordinator role is provided with a *participameter*, a widget informing 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 when, for example, the level of participation in a certain item is low: remind people, promote discussion or even drop the topic.

The CO2DE editor allows for individual contexts to be joined into a single diagram by providing a synchronous cooperative edition facility and a WYSIWIS interface (Figure 3). Although it also allows asynchronous interaction, it does not focus 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.

When conflicting views arise on a diagram, most cooperative editors support 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, the system represents 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 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 step-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.

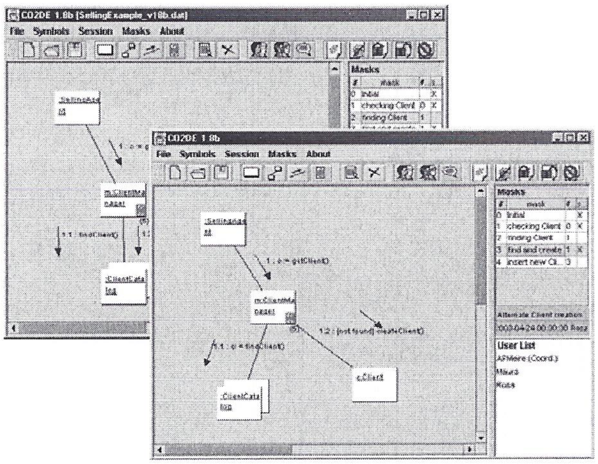

Figure 3. CO2DE user interface [11]

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

6. Conclusions

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

The context process model 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 knowledge flowing among processing activities.

The framework and the model can be applied together to obtain some insight into some groupware designs. In particular, by considering context as knowledge to be applied during group work, one can have a wider perspective than just focusing on the information provided to users by awareness mechanisms, as shown in the previous section. Other groupware designs would probably be suitable for analysis from this viewpoint.

Acknowledgments

This work was supported by grants from: CNPq (Brazil) PROSUL AC-62, Fondecyt (Chile) No.1040952.

References

[1] Borges, M.R., Pino, J.A.: Awareness Mechanisms for Coordination in Asynchronous CSCW, Proc. WITS, 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*, 27(3), 1999, 271-289.

[3] Brézillon, P. Individual and team contexts in a design process. HICSS-36, 2003, IEEE Computer Soc. Press, CD-R

[4] Brézillon, P. Using context for Supporting Users Efficiently. HICSS-36, 2003, IEEE Computer Soc. Press, CD-Rom.

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

[6] Context 2003:http://context.umcs.maine.edu/CONTEXT-03

[7] Dourish, P. Seeking a Foundation for Context-Aware Computing. *Human-Computer Interaction*, 16(2-4), 2001.

[8] Dourish, P., Bellotti, V. Awareness and Coordination in Shared Workspaces. CSCW, 1992, ACM Press, 107-114

[9] Greenberg, S., Context as a Dynamic Construct. *Human-Computer Interaction* 16 (2-4), 2001, 257-268.

[10] McCarthy, J., Notes on formalizing context, Proceedings of the 13th IJCAI, 1993, Vol. 1, 555-560.

[11] Meire, A., Borges, M., Araujo, R. Supporting Collaborative Drawing with the Mask Versioning Mechanism. Lecture Notes in Computer Science 2806, 2003, 208-223.

[12] Moran, T.P. and Dourish, P. Context-Aware Computing. *Human-Computer Interaction*, 16(2-4), 2001, 87-94.

[13] Naveiro, R., Brézillon, P., Souza, F. Contextual knowledge in design: the SisPro project. In "Espaces Numériques d'Information et de Coompération", C. Simone, N. Matta & B. Eynard (Eds.) Hermès, 5(3-4), 2002, 115-134.

[14] Pinheiro, M.K., Lima, J.V., Borges, M.R.S. A framework for awareness support in groupware systems. *Computers in Industry* 52 (1), 2003, 47-57.

[15] Pomerol J-Ch., Brézillon P. Dynamics between contextual knowledge and proceduralized context, Lecture Notes in Artificial Intelligence 1688, 1999, 284-295.

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

[17] Rosa, M.G.P., Borges, M.R.S., Santoro, F.M. A Conceptual Framework for Analyzing the Use of Context in Groupware. Lecture Notes in Computer Science 2806, 2003, 300-313.

[18] Salvador, T., Scholtz, J., Larson, J. "The Denver Model for Groupware Design", *SIGCHI Bulletin* 28 (1), 1996.