Context Models for Managing Collaborative Software Development Knowledge

Renata Mendes de Araujo[1,3], Flávia Maria Santoro[1,3], Patrick Brézillon[2], Marcos Roberto da Silva Borges[3] and Márcio Gonçalves Pereira da Rosa[3]

[1] Department of Applied Informatics/UNIRIO, Av. Pasteur 458, Térreo, Urca, CEP 22290-240, Rio de Janeiro, RJ, Brasil
{renata.araujo, flavia.santoro)@uniriotec.br
http://www.uniriotec.br
[2] LIP6, University Paris VI, 8 rue de Capitaine Scott, 75015, Paris, France
Patrick.Brezillon@lip6.fr
[3] Post-Graduate Program in Informatics, Universidade Federal do Rio de Janeiro, PO Box 2324, CEP 20001-970, Rio de Janeiro, RJ, Brazil,
mborges@nce.ufrj.br, marciorosa@frb.br

Abstract. The aim of this paper is to sum up our first observations on the use of the context concept in software development applications – specially the collaboration supporting tools – in order to improve software organization's knowledge management and, furthermore, its organizational learning. This work proposes a framework for identifying and organizing contextual information within the scope of software development organizations and the overall domain area. The paper discusses how this framework can be used to help designers to introduce contextual information into collaborative software development tools.

1 Introduction

Software development processes involve several professionals, typically working in teams, and a wide set of distributed shared knowledge [1]. This knowledge constitutes much of the team experiences and background, and it is built during the project development. Who the specialists in certain technique are, which techniques have been successfully applied in development projects, which mistakes made by the teams should not be repeated, what effective solutions have been adopted, what are the software organizations strategies, business objective, domain market etc. are a few examples of the kind of knowledge that could be provided in order to make software development activities more efficient.

The memory of software organizations is built, managed and retrieved collaboratively by the many project participants, including the organization clients and suppliers. Also collaborative in its nature are the great part of the managerial or technical activities that take place within a software development project. This collaborative aspect has motivated the proposal of several groupware research prototypes.

However, so few software engineering tools facilitate the access to the necessary, appropriate or possible organizational knowledge for improving the enactment of each software development activity they support. In fact, providing appropriate and effective organizational knowledge within the use of any computational system is quite a challenge.

The research work on the concept of context [3][4][5] has discussed the issue that explicit organizational knowledge capture, management and retrieval cannot be dissociated from the context when it was created and the (most of the times different) context when it will be effectively used. This lack of knowledge contextualization can lead to knowledge misuse or may cause its wrong application.

Studies on how context information can be modeled and manage in collaborative applications have also started to be carried out [6][7][8]. However, there is still a need to detail which context information can be retained, recorded and managed in specific domains, such as software engineering activities. Furthermore, it is still an open issue how software engineering tools can provide context information in order to better support collaborative knowledge use and management.

The aim of this paper is to start the discussion on the application of the context concept in software development applications – specially the collaboration supporting tools – in order to improve software organization's knowledge management and, furthermore, its organizational learning. This work proposes an overview of a framework for organizing contextual information within the scope of software development organizations and the overall domain area. Additionally, the paper discusses how this framework can be used to help the designers to introduce contextual information into collaborative software development tools.

The Section 2 discusses knowledge management within the scope of software development domain. In the Section 3, the concept of context is reviewed, as well as its dynamics and how it can benefit knowledge management. In the Section 4, the contextual information framework for software development organizations is presented. The Section 5 concludes the paper.

2. Knowledge Management in Software Development

Software development processes involve several professionals, and a great need for sharing knowledge among them. This knowledge relies on each member's previous skills, experiences and background, as also on the activities, conditions, facts, and situations faced during a project development. This set of knowledge constitutes what can be called the organizational memory (OM) [2].

From the OM, professionals extract exchange and assemble knowledge to perform their operational activities. In the same way, a group performing its collaborative task at a certain moment of the development process is both a source and a consumer of knowledge that can be stored in and retrieved from the OM.

Specifically considering the nature of software organizations, we can argue that there are other aspects that surround their organizational memories, determining the way it must be managed: the need for defining and enacting work processes and its collaborative aspect.

Focus on the Process: The idea of having a defined process is being strongly demanded by the software development market [9][10]. Specially concerning KM, the existence of a defined working process within the organization serves as a reference to identify, acquire, develop, disseminate, use and preserve the organizational knowledge in a systematic way. When associated to a process automation tool (such as a workflow system) the possibilities for KM can be amplified [11][12]. The process becomes the infrastructure for the organization to execute its activities, learn with its execution, observe flaws and needs for change and to collaborate.

Group support: From the initialization till the deployment phase of a software development project, most of the necessary activities – requirements specification, modeling, documentation, planning, meetings, programming, reviews, decision making, training etc. – are performed collaboratively. Thus, supporting group interactions assumes an important role in improving the process productivity and software quality. Additionally, while interacting in groups, software process participants communicate and share both tacit as explicit knowledge that should also be captured into the OM.

Communities of practice: After integrating knowledge and working tools, software organizations can benefit of this knowledge infrastructure as a learning environment [18]. Solutions for collaborative and project-based learning [19] are suggested to compose the communities of practice within a software organization. Inside this community, a professional can find resources for the execution of a specific task, meets specialists, and have access to important aspects and historical data provided by other professionals along the execution of their activities (Figure 1).

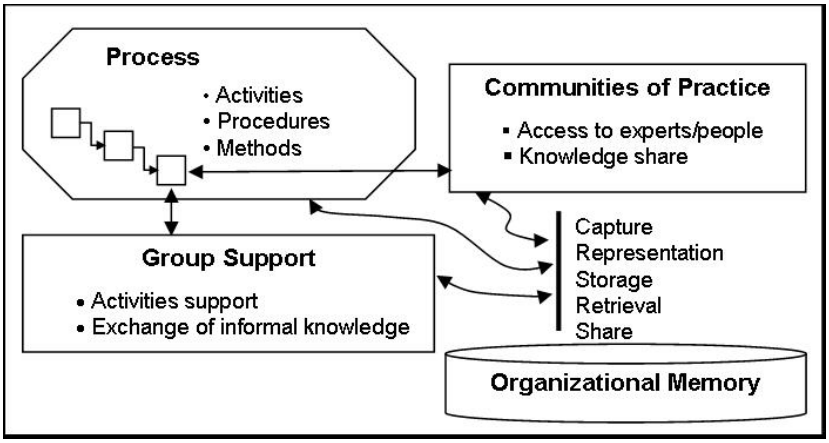

Figure 1. Perspectives for using the organizational memory in software organizations

In general terms, the purpose of the organizational memory is to guarantee that the desired knowledge can be recovered at the right time and in the right place by whoever needs it while he/she is following a work process, interacting within a group activity or aiming to learn based on the organization experiences.

However, the effective use of knowledge is just not a matter of storing and making it explicit into an OM structure. It is also a matter of giving it meaning, to understand its context. What is the meaning of having information if it cannot be effectively understood and appropriated by who is receiving it? For instance, what is the relevance of knowing that a development project lasted 3 months if we are not aware that it comprised highly skilled programmers and/or that the contract was restricted to delivering the system within this period of time and/or all the recommended organizational quality practices were abandoned?

In the case of collaborative interactions such as software development, this context is collaborative created at a given focus or activity and must also be shared in order to be effectively used. Thus, the process of converting knowledge from and into the organizational memory is basically dependant on how contextual information surrounding this knowledge can be effectively provided.

3 Context

In the real world, context is a complex description of the knowledge shared on physical, social, historical and other circumstances where actions or events happen. All this knowledge is not part of the actions to execute or the events that occur, but will constrain the execution of an action and event interpretation [5]. For the total understanding of several actions and events, it is necessary to have access to important contextual information.

At a given step of a task performing, Brézillon and Pomerol [14] 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 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.

Although the contextual knowledge exists in theory, it is actually implicit and latent, and is not usable unless a goal (or intention) emerges. When an event occurs, the attention of the actor becomes focused on it. At this moment, part of his knowledge, the one that may have direct relationship to perform the current task is concentrated as its contextual knowledge. Part of this contextual knowledge will be proceduralized while the actor effectively performs the task. When the task proceeds from one step to another, there is a movement between the contextual knowledge and the proceduralized context because a new item enters or leaves the focus of attention. Thus, context is dynamic within the scope of a specific task (Figure 2).

Brézillon [15] also points out that it is possible to identify different types of context and to organize them in different levels, 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. The former case defines that contexts are different by their granularity. For example, the context of an enterprise is more general than the context of its employee. The latter case defines that each actor within a specific scope has its

own context. For instance, a research project gathers different specialists with different backgrounds, abilities and culture.

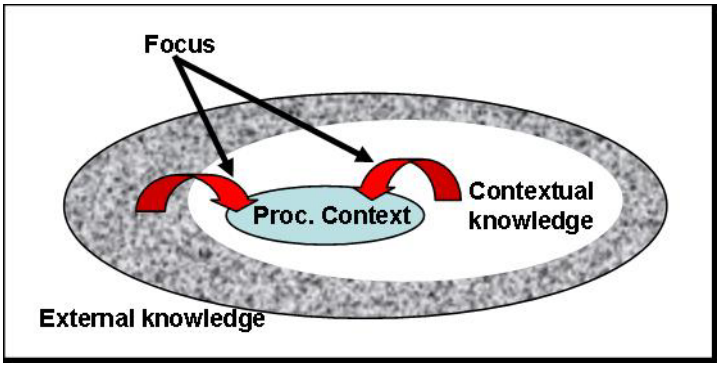

Figure 2. Context dynamics

3.1. Context and Knowledge Management

Considering the cycle of knowledge creation [20], knowledge is the appropriation and reasoning of information by an actor. This information had, in its turn, been gathered by the interpretation made by this actor over a set of available data (Figure 3).

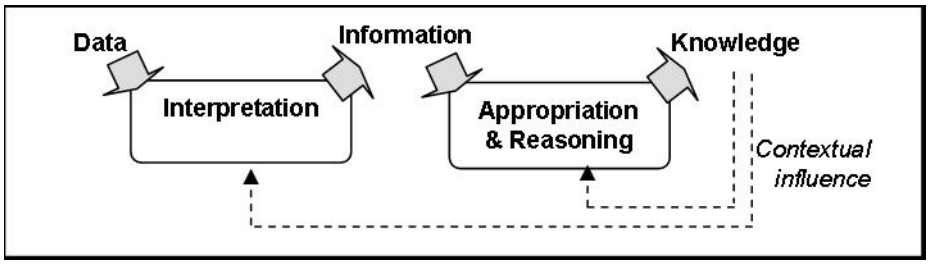

Figure 3. Knowledge creation process

The amount of knowledge gathered by an actor influences his/her data interpretation and information reasoning, delivering new knowledge. Therefore, the existing knowledge constrains the interpretation and appropriation activities of an actor. The concept of context, as mentioned above, tries to give meaning to this influence made by knowledge over the creation of new knowledge. When facing a given problem, task or focus, an actor has to interpret and reason over the available data and information. In order to do this, he uses his previous knowledge but not all of it, just the knowledge that is strongly related to the necessary interpretation and reasoning (contextual knowledge). To interpret data, to reason and to appropriate information during a specific focus means to proceduralize it, creating new knowledge that will take part of the actor's contextual knowledge.

Context is a way of giving knowledge focus and meaning, and the most focused and understandable, the most effective it can be captured and used in a given situation. In the case of collaborative interactions, where tasks and their underlying knowledge must be shared among participants, this facet of the task shared knowledge - its context – is important for effective collaboration.

Moreover, we argue that it is possible to identify what are the elements that comprises the contextual knowledge for task performing in specific domains. By identifying and organizing these elements into a framework or model, this model could be used to guide the design of applications in order to consider the context aspect.

3.1. Context in Collaborative Interactions

Dourish and Belloti were the first ones to establish the connection between context and awareness mechanisms in groupware applications [13]. Borges, Brézillon and Pino defined a framework to relate these concepts [7]. The following definition can be used: "*Context is composed by the set of information that helps to characterize the task of the group. Its objective is to offer conditions to the group members to notice and understand all the factors that influence their interaction*".

Working in group supposes to manage context explicitly. Not only individual contexts need to be proceduralized but also the group context. However, the group context is not simply the union or intersection of individual contexts. 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 [16].

Rosa et al. [6] consider that the important elements for the composition of the group context are divided into five categories (Figure 4): (1) information about people, (2) information about the scheduled tasks, (3) information about the relationships between people and tasks, (4) information about the environment where the tasks are accomplished and (5) information about concluded tasks. These elements form the base of their proposed framework for identifying and organizing contextual information, as detailed in [21].

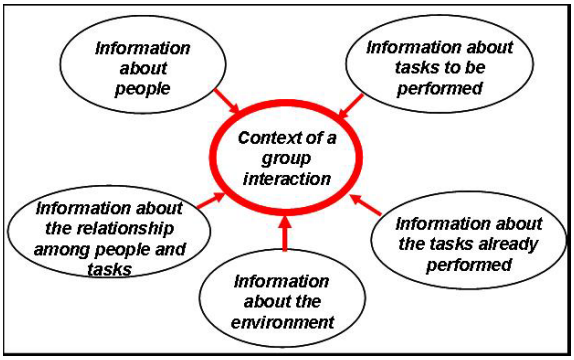

Figure 4. Composition of contextual knowledge for a group task [21]

4 Contextual Information in Collaborative Software Development

The concepts presented above - context and how it can be characterized in group interactions - can be applied generally in any domain where knowledge management and group interaction is needed. However, it will be more effectively applied if it is taken into account the specific sources of knowledge within a domain. In the case of software organizations, if we consider an "in depth view" (see section 3), contextual information can be characterized into many dimensions as depicted in Figure 5:

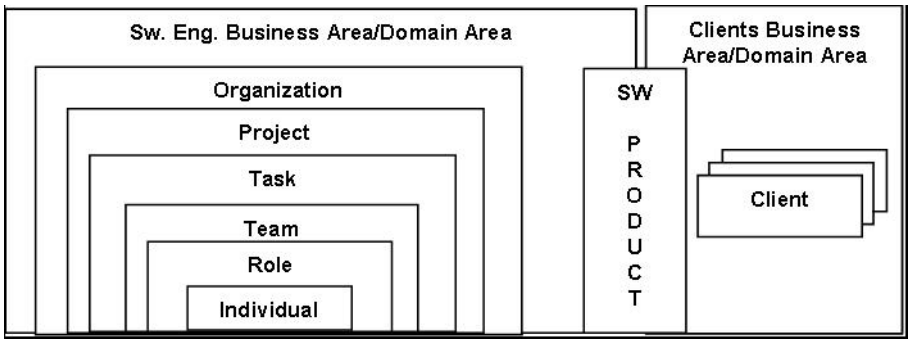

Figure 5. Dimensions for contextual information in software development

1. **Individual/personal context**. The first context information a software developer can have is about its individual or personal context. Individuals have specific experiences, expertise, motivations and interests, independently of the organization, project or roles he can assume. Each person uses a personal body of (contextual) knowledge for reasoning, interpreting, recognizing information in order to integrate the new information in this body of knowledge.
2. **Roles**. Roles are a way to specify the types of knowledge, experience, etc. that are needed to accomplish the task. For instance, managers need information about status, budget and deadlines while software designers need information about the product and its requirements. Also, individuals assuming a same role can demonstrate different characteristics and attitudes.
3. **Team**. The context of a team reflects the aggregation of its many participants and roles. Teams are established within a development project to perform specific tasks such as: planning, testing, specification, programming etc. Teams built with different participants will plan and perform their tasks in a different manner. The successful or unsuccessful outcomes of their tasks are new knowledge that can be stored in the organizational memory.
4. **Tasks**. The context of distinct tasks requires specific information and knowledge: testing software, for example, requires the availability of practices, tools and procedures totally different form software requirements elicitation. Additionally, the flow of activities within a project can influence each activity outcome while performed by different teams, with different skills, experiences and motivations.

5. **Project**. A specific project comprises an also specific context in respect to the product to be constructed, its objectives and the process to be followed. Good or bad project results better describes the context of an organization. The relationship or production of an specific individual in a project or within the organization provides new context information related to the practices performed, his performance and the need for reviewing the existing practices in order to improve them and make them tailored to the developers needs.
6. **Organization.** The context of the organization describes its business targets and its standard practices. The software development market and domain defines context characteristics and information such as new proposed practices and historical results of their application. The results obtained by organizations while providing services to their clients generate historical information of successful or unsuccessful use of practices and methodologies suggested in the market or in specific business domains.
7. **Client.** Each specific client also has his business scope, domain area, priorities and business objectives.
8. **Product.** The kind of requirements, the scope of the product that will be built, its elements and the interdependence among them are a new set of relevant context information.
9. **Software Engineering Domain:** software engineering is an area where new technologies emerge very often. In order to execute an specific task, developers must be skilled with the available technologies to conduct it as also managers must understand its impacts and cost.
10. **Client Business Domain Knowledge:** software development requires access to the knowledge about the domain where the product will be applied. At any task within the development process, having this knowledge or part of it can be crucial for building the software product.

The proposal of a framework that details the presented contextual dimensions and combines them with the dimensions presented in the framework for group work can be a way for defining specific contextual information that can be generated, captured, stored, notified and visualized by software development participants within a software organization. Although not detailed yet, we show, using the following example, how the framework helps to consider contextual information within the scope of a specific tool or environment

4.1. Designing for context management

Collaborate to Design (CO2DE) is a collaborative UML diagram editor [17]. It was originally design to allow participants to share the same drawing workspace and to control different versions of the document being constructed. In its original design as a groupware tool, it provides functionalities for helping individuals to be aware of some contextual information related to the specific task being performed. For example: information about the team composition (Figure 6 – (1)), information about

each members' position (Figure 6 – (2)), information about each new document version being simultaneously discussed (Figure 6 – (3)), among others.

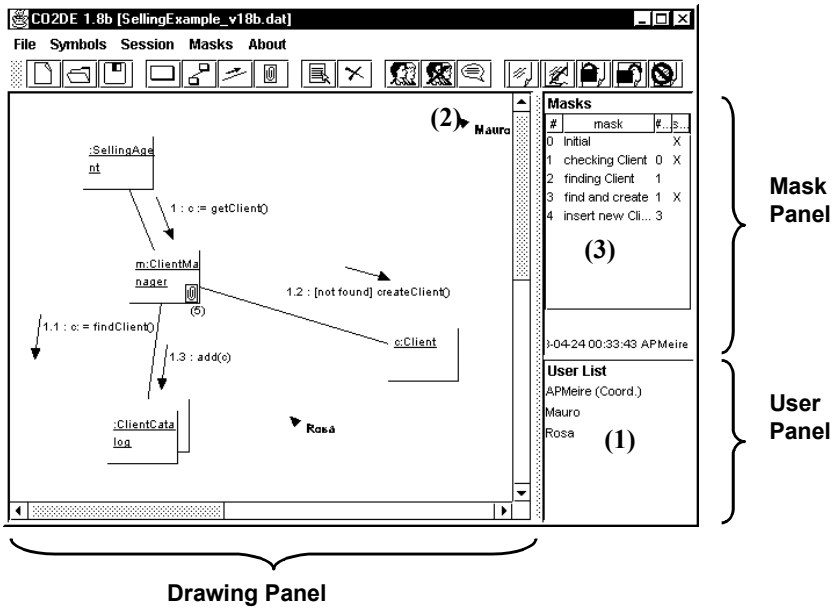

Figure 6. CO2DE main window

While applying his proposed framework for contextual information in groupware, Rosa [21] was able to identify other functionalities that could improve collaboration in CO2DE. His findings were corroborated by observing case studies and by analyzing questionnaires where participants' answers indicated that some functionalities related to other context dimensions were still missing. CO2DE was then redesigned, to incorporate, for instance: detailed information about group members; the identification of a group; the task objective (Figure 7); details about the current document version (Figure 8); etc.

Now, considering the fact that CO2DE is to be used by a software development team within a software organization, other contextual aspects inherent to the software development work should be discussed too, as outlined above in this section. We argue that it is possible to identify what would be the contextual information that could be retrieved from the OM and made available to this development team at the right moment they perform the diagram editing task. For instance, participants should also be able to understand where this activity is inserted within the overall project process being conducted – its deadlines, its relevance and its impact into the process. Chunks of information about UML and how it is being used into the organization, what were the successful and unsuccessful cases of using this technique; relevant aspects of the client's business domain etc could also be available for contextualizing this group activity. Additionally, as in software development projects, participants

assume specific roles – project manager, business analyst, programmers, user etc – and along with the need of being identified as so by the team, they also need different information from the current editing task.

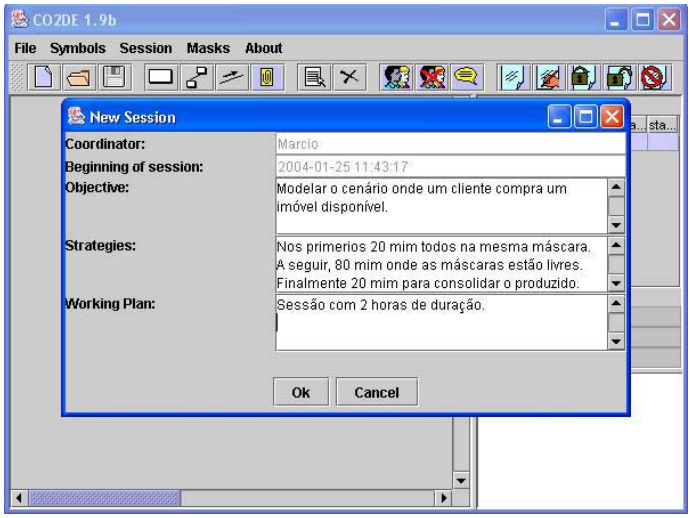

Figure 7. Detailed information about a group member

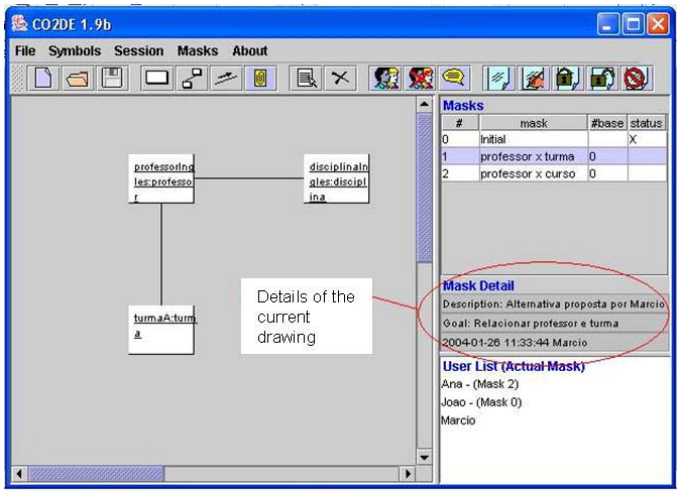

Figure 8. Details for the current mask

5 Conclusion

Software engineering projects are inherently collaborative and needs effective support to knowledge management. The elements for knowledge management in

software development comprise the existence of an organizational memory, the focus on its management through process, group support and communities of practice.

Context management has been considered as a relevant concept for knowledge management. In order to be effectively used, each knowledge domain area should consider its own contextual elements and what is the relationship among them. This work presented the basis of our framework of contextual information for software development organizations.

We are using this framework as a guideline on how to design environments or tools tailored to presenting or capturing contextual information. It also serves as a parameter, for instance, for analyzing commercial CASE tools on how they deal with contextual information or how they can be extended to favor it.

We are also investigating other aspects concerning software development work culture that can also influence the design of context management mechanisms. One special characteristic, for instance, is the feeling of "forever emergency" in software development projects. Even to those organizations where project management practices are well established, emergencies happen and must be treated like that, due to the critical dependency we have nowadays on software. Maintenance activities are a typical example. Some maintenance activities must be solved in timely manner and that involves retrieving relevant context information for taking fast decisions, planning for action and coordinating the team to perform them.

References

[1] Rus, I., Lindvall, M. (2002), "Knowledge Management in Software Engineering", *IEEE Software*, May/June, pp. 26-38.

[2] Ackerman, M.S. (1994), "Augmenting the Organizational Memory: A Field Study of Answer Garden", *Proc. Conf. on Computer-Supported Cooperative Work (CSCW'94)*, pp. 243-252.

[3] Brézillon, P. (2002) "Making context explicit in communicating objects". In *Communicating Objects*, C. Kintzig, G. Poulain, G. Privat, P.-N. Favennec (Eds.), Hermes Science Editions, Lavoisier.

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

[5] Brézillon, P. (1999) "Context in problem solving: A survey". *The Knowledge Engineering Review*, vol. 14, n°1, pp. 1-34.

[6] Rosa, M. G. P., Borges, M. R. S., Santoro, F. M. (2003) "A conceptual framework for analyzing the use of context in groupware", *Lecture Notes in Computer Science*. Paris:, v. 2806, pp. 300-313.

[7] Brézillon, P., Borges, M. R. S., Pino, J. A., Pomerol, J.-Ch., (2004) "Context-awareness in Group Work: three case studies", In: *Proceedings of the IFIP Int. Conference on Decision Support Systems (DSS2004) Decision Support in an Uncertain and Complex World*, Prato, Italy, July 1-3, pp. 115-124..

[8] Brézillon, P., Borges, M.R.S., Pino, J., Pomerol, J-Ch., (2004) "Context-based awareness in group work." In: *Proceedings of the 17th International Flairs Conference*, AAA Press (CD-ROM), Miami, Florida, USA, pp. 575-580.

[9] "Capability Maturity Model-Integrated / CMM-I" Software Engineering Institute http://www.sei.cmu.edu (last visit: 24th april 2004)

[10] ISO 9001:2000, International Organization for Standardization http://www.iso.ch

[11] Wargitsch, C., Wewers, T., (1997), "WorkBrain: Merging Organizational Memory and Workflow Management Systems". *Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises*, In conjunction with the 21st Annual German Conference on AI '97 (KI-Jahrestagung '97).

[12] Maus, H., (2001), "Workflow Context as a Means for Intelligent Support". *Proc. Third International and Interdisciplinary Conference (CONTEXT 2001)*, In: Akman, V., Bouquet, P., Richmond, T., Young, R.A. (eds) Modeling and Using Context, Springer, LNAI 2116, pp 261-274.

[13] Dourish, P.E., Belloti, V., (1992), "Awareness and Coordination in Shared Workspaces". In: *Proc. of the CSCW'92*, Toronto, Canadá, pp 107-114. ACM Press.

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

[15] Brézillon, P., (2003), "Individual and team contexts in a design process". *Proc. of the 36th Hawaii Int.Conf. on Systems Sciences*. HICSS-36, Track "Emerging Technologies", January, R.H.Sprague (Ed.), Los Alamitos: IEEE, CD-Rom.

[16] Borges, M.R.S, Brézillon, P., Pino, J.A., Pomerol, J.-Ch, (2004), "Bringing Context to CSCW". In: *Proc. of the 8th Int. Conference on Computer Supported Cooperative Work in Design*, Xiamen, P.R. China, May 26-28, International Academic Publishers, Beijing World Publishing Corporation, IEEE Press, vol II, pp. 161-166.

[17] Meire, A.P., Borges, M.R.S, Araujo, R.M., (2003), "Supporting Collaborative Drawing with the Mask Versioning Mechanism", *Proceedings of the International Workshop on Groupware CRIWG 2003*, Autrans, France.

[18] Wenger, E., McDermott, R. A., Snyder, W. (2002), *Cultivating communities of practice: a guide to managing knowledge*. Boston, Mass., Harvard Business School Press.

[19] Santoro, F.M., Borges, M.R.S, Santos, N. (2003), "Learning through collaborative projects: The architecture of an environment", In: *International Journal of Computer Applications in Technology – IJCAT: Special Issue on Computer-Supported Cooperative Work in Design.* Vol. 16, No 2, pp. 127-141, ISSN 09252-8091.

[20] Davenport, T.H., Prusak, L., (2000), *Working Knowledge*. Boston, Mass., Harvard Business School Press.

[21] Rosa, M.G.P., (2004), "Inserting Context in Groupware", MSc, Thesis, Post-Graduate Program in Informatics, NCE-IM, Federal University of Rio de Janeiro (in Portuguese).