Supporting Cooperative Requirements Engineering with an Automated Tool

Denise Franzotti Togneri[1,2], Ricardo de Almeida Falbo[1],Crediné Silva de Menezes[1]

[1]UFES - Universidade Federal do Espírito Santo
Centro Tecnológico - Mestrado em Informática
Avenida Fernando Ferrari, s/n, Vitória-ES, Brazil, CEP: 29060-900
togneri@terra.com.br, {falbo, credine}@inf.ufes.br

[2] FAESA - Faculdades Integradas Espírito-santenses
Colegiado do Curso de Ciência da Computação
Rua Anselmo Serrat, 199, Vitória-ES, Brazil, CEP: 29041-010

Abstract. Requirements Engineering – one of the macro-activities of Sofware Engineering – is a systematic process of capturing, modeling and documenting requirements through an iteractive and cooperative approach[13]. It's an activity which requires team work. Groups of people formulate requirements, design and implement a system and execute quality assurance activities during the software process[28]. The cooperative work becomes necessary since solutions require different knowledge and professionals become more specialized[3]. Viewing these aspects, it's worthwhile to use computational resources to support the cooperative Requirements Engineering activities. The area of Computer-Supported Cooperative Work (CSCW) is interested in how groups of people working on a cooperative basis can be assisted by computational support[28]. This work investigates how solutions from the CSCW area can support Requirements Engineering activities and introduces CRETA – a Cooperative Requirements Engineering support tool, which intends to integrate groupware applications into applications supporting Requirements Engineering. This tool was developed for the Web, using the object-orientation paradigm.

Keywords: Cooperative Requirements Engineering, Requirements Engineering, Groupware, CSCW (Computer-Supported Cooperative Work), Software Engineering.

1 Introduction

In the development of complex systems, the definition of requirements stands out as one of the activities which presents the biggest problems, when done in an inadequate way [12].

Requirements Engineering is a macro-activity of Software Engineering which covers all the activities involved in the elicitation, documentation and maintenance of the group of requirements of a software system [29] and it requires team work.

Cooperative groups of people formulate requirements, project and implement a system and execute quality assurance activities along the software process [28]. Cooperative work becomes necessary once the problems are becoming larger and more complex, professionals are becoming more and more specialized and solutions require different knowledge. Besides, cooperation involves heterogeneous groups, with different knowledge levels and interests, belonging to the same organization or not.

The success of this phase is also determined, among other aspects, by the level of cooperation present in the development team and not only by the individuals that compose it [3].

Bearing this in mind, it is worthwhile to use computacional resources to support, on a cooperative-basis, the Requirements Engineering process. That support has been given through a great number of CASE (Computer-Aided Software Engineering) tools, which, in general, help the software engineers in several activities of the software process [25]. Specific CASE tools for Requirements Engineering basically support the development and validation of models of the software as well as the documentation and management of change of the requirements [18].

In general, the existing tools and environments do not offer appropriate support for the cooperation and interaction processes, which are vital in Requirements Engineering. In that particular point, the potential of computer support to the cooperative-based activities has been recognized since the mid-80s. This area of interdisciplinary research, called Computer-Suported Cooperative Work (CSCW), has been making fast progress, englobing researchers from the areas of computer science, cognitive sciences and social sciences. CSCW is interested in how groups of people working on a cooperative basis can be assisted by a computacional support [28]. Therefore, this work investigates how the area of CSCW can support Requirements Engineering and introduces CRETA - a Cooperative Requirements Engineering Support Tool, which intends to integrate requirements engineers, domain specialists, users, managers and sponsors involved in a software development process.

This paper is structured as follows: section 2 discusses Cooperative Requirements Engineering and Computer-Supported Cooperative Work; in section 3 the requirements of a support framework to Cooperative Requirements Engineering and Computer-Supported Cooperative Work are introduced; section 4 introduces CRETA, a Cooperative Requiremens Engineering Support Tool; section 5 discusses related works and compares them; finally, in section 6, the conclusions of this work are presented.

2 Software Requirements Engineering and Computer-Supported Cooperative Work

Requirements Engineering can be described as a process, in other words, an organized group of activities, methods, techniques, practices and transformations, which must be followed to originate, validate and maintain the artifacts generated. These processes vary a lot in an organization, or from one organization to another, from non-structured processes, based only on the experience of the people involved, to systematic

processes, based on the application of some methodology. The definition of an organizational process to the Requirements Engineering can bring many benefits. However, it does not make sense to talk about an ideal process or to define one and impose it to an organization. Instead, the organizations should begin with a generic Requirements Engineering process and instance it to a more detailed process, appropriate to its needs [29].

Although different projects require processes with specific features to contemplate its peculiarities, it is possible to establish a group of basic activities that should be considered in the definition of any Requirements Engineering process. In the literature [4, 7, 18, 19, 22, 29, 30], there are several groups of activities proposed for the process. Based on the proposal by Kotonya & Sommerville [18], in general, the activities of the Requirements Engineering process involves:

- **requirements elicitation**: involves the discovery of the functional and non-functional requirements that the software should attend, taking into account the different points of view from the participants of the process [19, 22, 29, 30],
- **requirements analysis and negotiation**: engineers analyze the documented requirements up to that moment, searching for problems and conflicts of the most varied types. Once found, the problems are listed and then, users, customers, domain specialists and engineers of requirements negotiate to reach an agreement about the changes to be made [4, 18, 19, 25],
- **requirements documentation**: once agreed, the requirements are documented in an appropriate level of details,
- **requirements validation**: there should be a careful check of the requirements for consistency, completeness and accuracy and should be concerned with answering the question 'have we got the requirements right?' [18]. It examines the specification to ensure that all requirements have been stated unambiguously, that inconsistencies, omissions, and errors have been detected and corrected, and that the work products conform to the standards established for the process, the project, the product [25],
- **requirements management**: running at the same time all the activities described above are carried out, it concerns the management of changes in the requirements and has the objective of keeping track of the changes and ensure that they are made to the requirements document in a controlled way.

There are not very defined boundaries among the proposed activities; in fact, they are interleaved and there is a high degree of iteration and feedback among them. In general, the process is executed up to when all the users are satisfied and in agreement with the requirements or when the pressure of deadlines precipitates the beginning of the Software Design phase, what is actually undesirable [18].

Requirements Engineering is a non-trivial, interactive task which involves intense human communication and cooperation [13] and there are several problems involving Requirements Engineering activities [6, 18,33]. However, the failures occurring in Requirements Engineering process can be attributed, partly, to the difficulty of the development team in working on a cooperative-basis, since, in general, software engineers are not used to share information and to interact, thus regarding cooperation as a distressful need [28]. The problems originated from the lack of cooperation tend to worsen as the number of professionals involved in the software process increases

[20]. Thus, computer support to cooperative activities can be helpful in the solution of some of these problems.

Computer-Supported Cooperative Work (CSCW) is a wide designation applied to the research area that investigates the work in group, addressing researches on the nature of the organizations and work places, embracing from sociological analysis and anthropological descriptions of work to the necessary technologies to support working in group [14].

The term groupware refers to the CSCW applications built to support and to promote the work in group more efficiently and effectively, increasing the exchange and the sharing of information, reducing communication overheads, promoting coordination and allowing collaboration inside and among organizations [5].

There are several different and discordant definitions of the terms cooperation and collaboration. In this paper, the definition adopted is the one proposed by Barros, mentioned by Nitzke et al. [21], which states that collaboration is related to contribution, while cooperation, includes not only the meaning of collaboration, but it is also a co-accomplishment work, which involves collective work to reach a common objective. Thus, the concept of cooperation is more complex, since it includes collaboration, while the opposite is not true.

In what concerns functionality, groupware applications can be classified in electronic mail, text-based conference, which can be can be synchronous, usually in the form of chat, or asynchronous, usually in the form of forum, videoconference, softwares of electronic meetings, sharing of documents and collaborative authorship, electronic document management systems, groups coordination systems, intelligent agents and workflow management systems [5, 10]. Synchronous interactions are those that happen when people are executing activities at the same time, and asynchronous interactions are those executed in different times [10, 24].

Most software support only the interaction between user and system, offering minimum support to human-human interactions, which is needed, because a great part of the activities people execute is in group. To support group interactions, the following should be considered [5, 9, 10, 23]:

- **computer mediated group communication**: it allows the sending and the exchange of information with other people; can be synchronous or asynchronous [5, 10],
- **computer mediated coordination**: it is the act of making sure that a team is executing their activities effectively, reaching their objectives, reducing conflicts and repetitive actions, and contributing to an increasing communication and collaboration [10]. It still encompasses management of dependence among activities [2],
- **computer mediated cooperation**: it is the act of joint cooperation among individuals in the execution of an activity or in the resolution of problems and it is, together with communication, a main feature in group activity [5],
- **awareness**: it allows individuals to understand the activities of the others and to adjust their own activities, structuring them and avoiding work duplication. Awareness information is also required to coordinate group activities [9],
- **knowledge management**: it is the formal management of knowledge resources so as to facilitate its access and reuse, typically using adequate information

technology [23], avoiding loss of memory and making it possible to identify history, argumentation used and chain of thoughts behind each decision, procedure, technical and business strategy [17].

3 Requirements of a Support Framework to Cooperative Requirements Engineering

Having defined Requirements Engineering as a cooperative process, it should be investigated how the CSCW area can support Requirements Engineering. Therefore, the main goal of this section is to enumerate the requirements of a computational support framework to the cooperative work accomplished by the teams involved in a Requirements Engineering process. These support requirements include:

- **Documentation of Templates, Resources and Knowledge about the Requirements Engineering Process**: it should allow the registration and sharing of elements, such as knowledge on paradigms, activities, processes, life cycle models, artifacts and procedures, document templates, available resources, work teams, its components and performing roles, so as to be reused consistently in the definition of the Requirements Engineering process, which will be adopted in the development of a project,
- **Definition of the Requirements Engineering Process**: a project to be developed should be registered, so that it can be controlled effectively and the stored knowledge should be used by the project manager to support the definition of the Requirements Engineering process, the life cycle model and the activities to be performed. The process should then be customized and refined, allowing the inclusion of sub-activities, the schedule of activities and the ones in charge of them, the resources to be used, the procedures to be adopted and the artifacts to be produced and required in each activity. It should still allow to define the document templates to be adopted in the artifacts elaboration, as well as those which will guide the execution of activities [11],
- **Tracking of the Requirements Engineering Process**: it is important to facilitate the tracking of the activities, so as to allow the participant to: (1) customize their work environment, to receive messages of automatic notifications, indicating, for instance, which activities under their responsibility are delayed or close to their deadlines; (2) obtain statistics indicating, for instance, the percentage of accomplishment of the activities and of the project, receiving automatically motivational or alert messages for the work teams and (3) view the process so as to have a perception of how their work depends on and affects the work of others,
- **Planning and Scheduling of Activities**: once several activities are accomplished in group with strong interaction among the participants, it becomes necessary to plan them and schedule them previously. Thus, among others, the requirements of that category are: (1) to allow the participants to schedule individual and group activities, verifying automatically the availability of schedules of those involved and suggesting alternative schedules; (2) it should allow the ones allocated to perform a particular activity to receive an automatic message of notification of the

scheduling, that, if accepted, should, by its turn, generate an entry in the personal agenda; (3) to allow the planning of an activity to be made through synchronous or asynchronous discussions and (4) to allow group activities to have their results registered so as to promote their sharing,

- **Interactions**: to support group interactions, three key areas should be addressed: communication, cooperation and coordination. Awareness and organizational memory should also be considered:
  - *Communication*: it should allow participants geographically distributed or not to communicate in a synchronous way, through their workstations, viewing instantaneously the exchanges of information and the presentation of ideas. Asynchronous meetings should also allow participants to think over their opinions, information and ideas available, so as to allow the registration of their opinions in the most convenient moment to them. It should also allow the sending of asynchronous messages, addressed to specific participants in such a way that it is possible for both the sender and the receiver to track the group of actions executed by the receiver,
  - *Awareness*: it should allow the participant to perceive his role and who his partners are in the work team and in the execution of each activity, to view the active participants, to perceive who accomplishes each activity, as well as the situation, instructions and time foreseen for execution of such activities, avoiding duplications. It should also allow the automation of the collection and presentation of the information about the participants' activities, inside a shared workspace, where continuous updating of the accomplished actions and of the global group progress is possible,
  - *Organizational memory*: it should allow the registration, sharing and reuse of the generated artifacts and of the memory of the group's interaction process, including communications, discussions, comments, justifications, lessons learned and the steps taken,
  - *Cooperation*: to allow cooperation, support must be given to communication, to the sharing of stored information, awareness of the changes and activities of other people of the group, coordination of the activities and registration and sharing of knowledge in the organizational memory,
- **Requirements Engineering Process Execution**: it is necessary to support the execution of several activities, such as:
  - *Registration of the project's information sources*: the different information sources identified in a project, such as books, people, existent systems, manuals, forms and documents, should be registered and made available, allowing the subsequent identification of the requirements sources,
  - *Construction of a common vocabulary*: to facilitate the understanding of the specialized terms used by the users, colect, registration, organization and approval of a common vocabulary must be supported and made available to the whole organization,
  - *Application of elicitation techniques*: an adequate approach to support interviews [16] and meetings [22] consists in allowing them to be virtual and held either synchronously or asynchronously. Furthermore, it should allow the registration and sharing of discussions, decisions taken, participants and results

obtained. The use of questionnaires [16] can be supported through the registration of the questionnaire to be applied, of the participants' answers and, in the case of objective answers, of the automatic tabulation of the results. In what concerns prototyping [4, 16], prototypes to be evaluated should be registered and made available, as well as an indication of which requirements and functionalities were liberated for evaluation and which problems were solved in the liberated version. The users evaluate the prototype and register a report indicating the problems found and the suggestions for improvements. To support the technique of document investigation [16], the images of the documents should be made available for analysis by the participants of the project, that can register a report on the results of the investigation. Observation [16] can be supported allowing its results to be documented by the requirements engineer and consulted by the other participants,

- *Requirements documentation*: the Software Requirements Specification (SRS) should be registered and be in agreement with the previously defined template. Through appropriate notations, it should allow for the documentation of functional and non-functional requirements, requirements sources, problems to be solved, reasons for choosing a solution or a decision taken and other alternatives which were considered,
- *Requirements analysis and negotiation*: each requirement documented in the SRS's rough draft should be analyzed by the requirements engineer and the problems and conflicts which may come up should be registered in a report. This report should serve as base for negotiation with the users, which can be carried out with the users present or virtually,
- *Requirements validation*: Formal Technical Revision is a widely used technique to validate requisites. Several of its tasks can be supported, such as [18]: (1) scheduling of the revision meeting; (2) distribution of documents for validation; (3) registration, in an individual pre-revision report, of problems identified in each requirement by each reviewer; (4) holding of the revision meeting, physically or virtually, when each requirement showing problems is evaluated and a group of solutions and actions are brought to an agreement and registered; (5) management of the these actions and (6) revision of the requirements document,
- *Requirements management*: it should allow the registration of the requirements change solicitation, and, then, an automatic notice of the registration should be sent to the project manager. To support the evaluation of the impact of the change, tracking information should be maintained and the Impact Evaluation Report should be registered, indicating which requirements will be affected, the cost, the resources and time needed to implement the change. If approved, the change activity is scheduled and the ones in charge of the change are allocated. Finally, it should allow the change in the requirements to be accomplished and approved and a new version of the SRS should be generated.

4 CRETA - A Cooperative Requirements Engineering Support Tool

To meet the requirements enumerated in section 3, a system called CRETA - a Cooperative Requirements Engineering Support Tool - was developed. It intends to integrate groupware applications with the area of Requirements Engineering, supporting the work of the knowledge managers, requirements engineers, domain specialists, users, project managers and sponsors [31]. The goal is to support the main activities of the Requirements Engineering process and to promote mechanisms for sharing information, and facilitating communication, coordination and cooperation among people, as well as awareness and knowledge management.

In CRETA, the existent knowledge on paradigm, process, life cycle models, activities and artifacts, among others, is registered by the knowledge manager, as well as the templates that can be adopted. The project to be developed is registered, its process is defined and people are allocated to activities.

The participants make use of cooperative tools available – such as electronic agendas, electronic mail and appointments, discussion lists, synchronous (chat) and asynchronous (forum) virtual meetings – to improve communication, cooperation and interaction. Documents and prototypes for evaluation can be made available using the tool. The several artifacts generated in each activity, such as plans of interviews, meetings minutes and Software Requirements Specifications, can be registered so as to promote their sharing, validation and management.

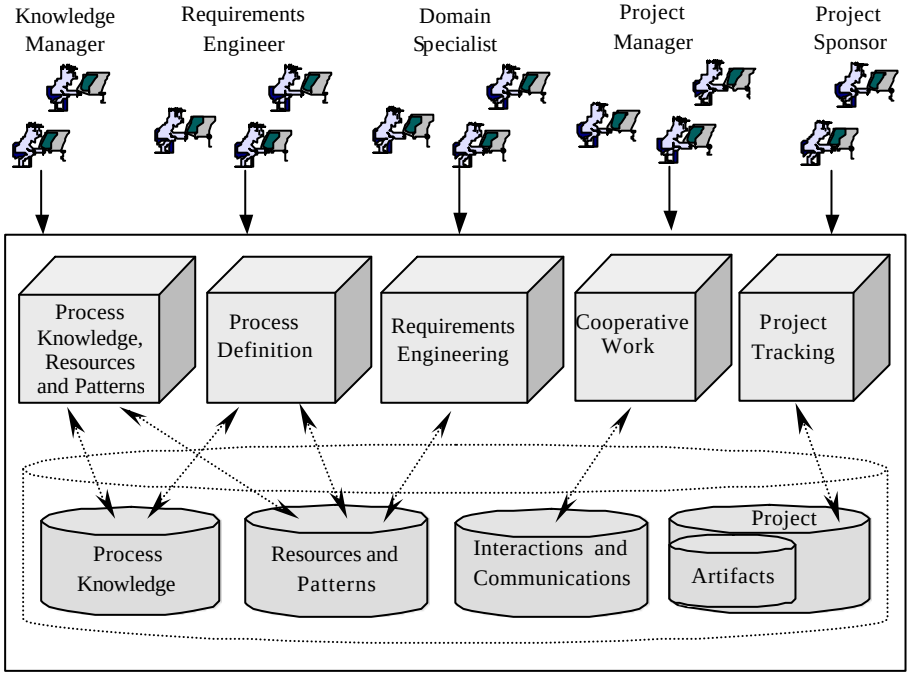

Fig. 1. CRETA's Architecture

CRETA's architecture is presented in figure 1. It is composed by five components:

- **Process, Resources and Template Knowledge Component**: it is used by the knowledge manager to register the organizational knowledge, including the requirements engineering standard process, resources and templates. The requirements engineers and the project managers use that component for querying the organizational memory,
- **Process Definition Component**: it is used by the project manager to register the project's features, and to define and customize the Requirements Engineering process for a project. The project sponsors, requirements engineers and domain specialists use that component only for queries,
- **Requirements Engineering Component**: it is used by the project participants to support the elicitation, analysis, negotiation, documentation, SRS validation and requirements management activities,
- **Cooperative Work Component**: it supports cooperative work, involving functionalities which allow the participants to use electronic mail, electronic scheduling of appointments, discussion lists, group scheduling, synchronous (chat) and asynchronous (forum) virtual meetings, and to view the active participants in a given moment,
- **Project Tracking Component**: it is used by the project sponsor and by the project manager to manage the course of the activities.

The implementation of CRETA's architecture was based on the Web, so as to allow the participants to be geographically dispersed and to use the tool inside the organization, through Intranet, as well as outside the organization, through Internet. Technologies to facilitate its implantation in different hardware and software platforms were adopted, such as JSP (JavaServer Pages) and Sun's Java 2. The objects persistence was implemented using SQL Server 7.0. However, any other relational DBMS could be used, once the Structured Query Language was used. The database was only used as a data repository. No type of processing was attributed to it.

Figure 2 presents CRETA's main page, which shows the appointments and mail that was not read, the activities scheduled to that specific date, and the participant's private agenda. From the main page, the functionalities available in CRETA can be accessed.

The groupware functions are located in the superior right corner of CRETA. The available links allow to access the group calendars, chats and forums, the electronic appointment and mail and to manage the discussion lists.

From the General Register menu it is possible to register the companies, the human resources, the work teams, the hardware and software tools, the patterns and the available knowledge in the organization.

From the Project Register menu it is possible to register and customize a project indicating, among others, the customer, the name of the project, the foreseen initial and end dates, the size, the complexity, the allocated teams, the process of requirements engineering associated, the discussion list of the project, the activities that should be executed, the stakeholders and the schedule.

From the Requirements Engineering menu it is possible to access the elicition functions, such as the information sources documentation, support to the scheduling, accomplishment and registration of the meetings and interviews results, that should

be presential or virtual, synchronous or asynchronous, the questionnaires application and registration, documents investigation and observation reports registration. It is possible to access the functions of requirements documentation, registration of the results of the requirements negotiation and analysis, the software requirements specification validation and the requirements management.

From the Project Management menu it is possible to accomplish the activities execution.

Fig. 2. CRETA's main page

Figure 3 shows one of the Requirements Engineering Component's functionality, the Interview Plan's creation page, which can be accessed by the requirements engineer from the "Requirements Engineering" menu.

Fig. 3. Interview Plan's Creation Page

5 Related Works

A lot of research concerning the integration of the areas of CSCW and Software Engineering and, especially, Requirements Engineering, has been conducted. Also several support tools and environments of cooperative software development have been proposed.

The main goals of Beyond-Sniff project [2] are: to offer a conceptual base and terminology for the Cooperative Software Engineering field, to develop computational tools for Cooperative Software Engineering and to develop a platform for integrating data and control. Beyond-Sniff is an environment for cooperative software engineering, which allows communication through requests and notifications mechanisms, management and integration of plug-and-play distributed components

(tools and services), such as infrastructure services, data dictionary, notation mechanisms and project management services, among others.

The PROSOFT software development environment supports the formal software development, aiming data, control and presentation integration among the tools built in the environment. Reis [27] presented a proposal of evolution of the environment with the objective of establishing support to the cooperative interaction among the developers concerning the use of the tools available in the environment. Thus, the Cooperative PROSOFT was developed to support the cooperative manipulation of the objects produced by the tools of the environment, through the aid of synchronous cooperation and the objects version control.

FLECSE - Flexible Environment for Collaborative Software Engineering [8] integrates several tools supporting requirements analysis, project, code, inspection and tests. Some functionalities available in the environment are: it allows the accomplishment and the registration of flexible electronic meetings, both synchrounous and asynchronous; it allows both remote and local access to work context, and the several artifacts generated are made available through hiperlinks.

Rational Suite Analyst Studio is a solution which encompasses a group of tools developed by Rational [26], such as Rational Unified Process, Rose, RequisitePro, ClearQuest, TestManager and ClearCase. It allows, among other things, to define and manage software processes, to model business processes, to register and to trace changes in the requirements, to control versions, to create test cases and to carry configuration management. It also allows integration with external or internal electronic mail and creation of discussion list for a given requirement, for instance.

CORE Enterprise developed by Vitech [32] is an environment which supplies a flexible combination of modelling and simulation tools supporting the Process and Product Engineering, and allowing the execution of several activities, such as: (1) specification and development of a system; (2) the extraction of original requirements from its source documents, analysis concerning completeness and consistence, possibility of testing and tracking each requirement for a behavioral model; (3) reengineer of business processes; (4) evaluation of impact of change solicitations in relation to planning, risks and cost; and (5) support to revision of requirements through the registration of comments and demarcations in the requirements or concurrently proposal of alternatives.

Beyond-Sniff, Cooperative PROSOFT and FLECSE are environments composed by a group of tools that allow cooperative manipulation of the objects produced and registered in these tools. However, they give little emphasis to the integration with groupware applications, which can facilitate synchronous and asynchronous communication, scheduling of activities and group agenda, among others.

The main features of the Rational Suite AnalystStudio solution and the CORE environment are: (1) support to the analysis of documents, identification, storage and classification of the elicited requirements; (2) support to the documentation of the requirements, making document templates available and offering consistent verifications in the document; (3) support to bi-directional tracking; (4) support to configuration management, allowing version control, access control for modification, and requirement historical report; (5) groupware applications, which basically support the revision, demarcation and registration of comments and users' concurrent access. However, these solutions offer limited support to the other elicitation techniques and

to the registration of the results obtained with their use. They give little emphasis to the use and integration of groupware solutions, such as chats, forums, internal electronic mail and coordination systems, which allow the scheduling of activities and group agendas, applied to a specific project.

6 Conclusions

This work has shown requirements for a support framework to Cooperative Requirements Engineering and presented CRETA - a Cooperative Requirements Engineering Support tool, which intends to support the group interaction held during the Requirements Engineering, facilitating the sharing of information, coordination, cooperation, communication, awareness and organizational memory, in an integrated way. Group communication is facilitated by electronic mail and appointments, discussion lists and by the availability of holding either synchronous or asynchronous virtual meetings.

The system supports cooperation in so far as it allows the sharing of the stored information, supports the coordination of the activities, allows people to be aware of other people's activities, that is, to know about other people's individual and group activities and, finally, it allows the common intellectual resources of an organization - the organizational memory - to be registered, shared and reused.

It is also important to highlight the integration between the CSCW applications and the Requirements Engineering support tools. From any tool functionality, for instance, electronic mail and appointments can be used to support asynchronous addressable communication among participants. The group agenda can be used for the scheduling of meetings, including the invitation of the participants and the automatic sending of an electronic appointment. The several meetings during the Requirements Engineering process can be held virtually, either synchronously or asynchronously. The participants' private address directory supports and facilitates communication. Discussion lists can be created to deepen discussions on specific subjects. The several artifacts generated during the Requirements Engineering process are registered in the system, supporting communication and sharing through an organizational memory.

The requirements pointed for a support framework to Cooperative Requirement Engineering presented in this work, makes way for the application of Computer-Supported Cooperative Work solutions in the development of CASE tools, which effectively support Cooperative Requirements Engineering, facilitating interactions, communication and cooperation among the members of a software team. We hope that in this way, we can improve software quality.

References

1. Ackerman, M. S. Augmenting the organizational memory: a field study of answer garden. In: PROCEEDINGS OF THE CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK (CSCW94), 4., 1994, Chapel Hill, USA. Electronic Proceedings... Available at: <http://www.acm.org/pubs/contents/proceedings/cscw/

192844/>. Accessed on: 02 mar. 2001. p. 243-252.
2. Bischofberger W.R., Kofler T., Mätzel K.-U., Schäffer B. Computer Supported Cooperative Software Engineering with Beyond Sniff. In: PROCEEDINGS OF THE CONFERENCE ON SOFTWARE ENGINEERING ENVIRONMENTS, 7., 1995, Noorwijkerhout, Netherlands. Electronic Proceedings... Available at: <http://www.ubs.com/e/index/about/ubilab/ext/publications/e_bis95c.htm>. Accessed on: 29 dec. 1999.
3. Borges, M., Araujo, R. O uso de groupware em desenvolvimento de software: minicurso. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 13., 1999, Florianópolis, Santa Catarina. Electronic Proceedings... Available at: <http://www.nce.ufrj.br/~chord/ home.htm>. Accessed on: 31 oct. 1999.
4. Brackett, John W. Software requirements. Pittsburgh, Pennsylvania: SEI, jan. 1990. CMU/SEI-CM-19-1.2.
5. Chaffey, Dave. Groupware, workflow and intranets: reengineering the enterprise with collaborative software. Boston: Digital Press, 1998.
6. Christel, Michael G., Kang, Kyo C. Issues in requirements elicitation. Pittsburgh, Pennsylvania: Software Engineering Institute, sept. 1992. Technical Report CMU/SEI-92-TR-12.
7. Davis, A. M. Software requirements: objects, functions and states. Prentice-Hall, New Jersey (1993).
8. Dewan, Prasun; Riedl, John. Towards computer-supported concurrent software engineering. IEEE Computer, New York, v.26, n. 1, p. 17-27, jan. 1993.
9. Dourish, Paul., Bellotti, Victoria. Awareness and coordination in shared workspaces. In: CONFERENCE PROCEEDINGS ON COMPUTER-SUPPORTED COOPERATIVE WORK (CSCW92), 3., 1992, Toronto, Canada. Electronic Proceedings... Available at: <http://www.acm.org/pubs/contents/proceedings/cscw/143457/>. Accessed on: 02 mar. 2001. p. 107-114.
10. Ellis, C. A., Gibbs, S. J., Rein, G. L. Groupware: somes issues and experiences. Communications of the ACM, New York, v. 34, n. 1, p. 38-58, jan. 1991.
11. Falbo, R. de A.; Menezes, C. S.; Rocha, A. R. C. da. Assist-Pro: um assistente baseado em conhecimento para apoiar a definição de processos de software. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 13., 1999, Florianópolis. Proceedings... Florianópolis: UFRGS, II: UFSC, INE, 1999. p. 147-162.
12. Faulk, Stuart R. Software requirements: a tutorial. In: Thayer, Richard H.; Dorfman, Merlin. (Org.). Software requeriments engineering. 2nd edn. Los Alamitos, California: IEEE, 1997. p. 128-149.
13. Graham, Ian. Requirements engineering and rapid development: an object-oriented approach. Addison-Wesley, Harlow, England (1998).
14. Grudin, J. Computer-supported cooperative work : history and focus. IEEE Computer, v. 27, n. 5, p. 19-26, may. 1994.
15. IEEE - Institute of Electrical and Electronics Engineers. IEEE Standard Glossary of Software Engineering Terminology: IEEE Standard 610.12-1990. New York, 1990.
16. Kendall, K. E.; Kendall, J. E. System analysis and design. 4th edn. Prentice Hall, New Jersey (1999).
17. Khoshafian, Setrag; Buckiewicz, Marek. Introduction to groupware, workflow and workgroup computing. John Wiley, New York (1995).
18. Kotonya, G., Sommerville, I. Requirements engineering: processes and techniques. John Wiley, Chichester, England (1998).
19. Lamsweerde, Axel Van. Requirements engineering in the year 00: a research perspective. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE00), 22., 2000, Limerick, Ireland. Electronic Proceedings... Available at: <http://www.acm.org/pubs/contents/proceedings/soft/337180/>. Accessed on: 30 dec.

2000. p. 5-19.

20. Lima, C. A. G.De, Reis, R. Q., Nunes, D. J. Gerenciamento do processo de desenvolvimento cooperativo de software no ambiente PROSOFT. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 12., 1998, Maringá - Paraná. Proceedings... Maringá: Ideal Ind. Gráf., 1998.
21. Nitzke, J. A.; Carneiro, M. L. F.; Geller, M.; Santarosa, L. C. Criação de Ambientes de Aprendizagem Colaborativa. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 10., 1999, Curitiba. Electronic Proceedings... Available at: <http://penta.ufrgs.br/pgie/sbie99/acac.html>. Accessed on: 24 apr. 2002.
22. Nuseibeh, Bashar, Easterbrook, Steve. Requirements engineering: a roadmap. In: CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING, 1., 2000, Limerick, Ireland. Electronic Proceedings... Available at: <http://www.acm.org/pubs/contents/proceedings/soft/336512/>. Accessed on: 05 feb. 2001.
23. O'Leary, Daniel E.; Studer, Rudi. Knowledge management: an interdisciplinary approach. IEEE Intelligent Systems. New York, n. 1, p. 24-25, jan./feb. 2001.
24. Palmer, J.; Fields, N.A. Computer-supported cooperative work. IEEE Computer, New York, v.27, n. 5, p. 15-17, may. 1994.
25. Pressman, R. S. Software engineering: a practitioner´s approach. 5th edn. McGraw-Hill, New York (2000).
26. Rational. Rational Suite AnalystStudio. Available at: <http://www.rational.com/products/astudio/index.jsp#text>. Accessed on: 30 may. 2002.
27. Reis, Rodrigo Quites. Uma Proposta de Suporte ao Desenvolvimento Cooperativo de Software no Ambiente PROSOFT. 1998. 177 p. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, 1998.
28. Sommerville, I.; Rodden, T. Environments for cooperative systems development. IEEE Computer, New York, v.26, n. 5, p. 144-155, may. 1993.
29. Sommerville, I., Sawyer, Pete. Requirements engineering: a good practice guide. John Wiley, Chichester, England (1997).
30. Thayer, Richard H., Dorfman, Merlin. (Org.). Software requeriments engineering. 2nd edn. IEEE, Los Alamitos, California (1997).
31. Togneri, Denise Franzotti. Apoio Automatizado à Engenharia de Requisitos Cooperativa. 2002. 138 p. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, 2002.
32. Vitech Corporation. CORE Product Family. Available at: <http://www.vtcorp.com/productline.html>. Accessed on: 30 may. 2002.
33. Zave, Pamela. Classification of research efforts in requirements engineering. ACM Computing Surveys Journal, New York, v. 29, n. 4, p. 315-321, dec. 1997.