Requirement Gathering Templates for Groupware Applications

Victor M. R. Penichet, Maria D. Lozano, José A. Gallud, Ricardo Tesoriero

Computer Systems Department. University of Castilla-La Mancha, 02071 Albacete, Spain

{ victor.penichet, maria.lozano, jose.gallud }@uclm.es; ricardo@dsi.uclm.es

Abstract. This chapter presents a novel approach to gather requirements for groupware applications, i.e. an application designed to be used by several users through a net of computers such as the Internet. In this chapter we propose several extensions to traditional templates typically used to gather requirements in order to include those requirements specifically related to groupware applications that currently cannot be described with traditional templates. The methodology we propose may be integrated in a process model to identify the roles and tasks needed in the following stages of the development process starting from the new requirements specification.

1 Introduction

Groupware applications are becoming more and more usual every day. They are applications where users achieve common objectives by performing tasks through networks. They collaborate, cooperate, coordinate, and / or communicate with each other. Users are not considered as individual members but as members of groups which interact among them. Such applications have particular features that if they were taken into account explicitly from the beginning could improve the quality of the final system [4, 5, 6, 7, 8, 10, 13, 14, 15].

In this chapter we present a proposal to gather such particular features. Traditional techniques such as brainstorming or interviews may be used to collect the information. Then, templates are used to describe the information to be gathered [3]. Some specific metadata are proposed regarding the most important features concerning groupware applications.

We have applied this approach in a complete process model to collect the requirements of groupware applications in the first stage of the software life cycle: requirements gathering. This metadata is integrated in the whole process model. Moreover, the traceability among the different stages has been accurately defined. Requirements and actors are identified and described in the requirements gathering stage and they are used to identify tasks and roles in the analysis stage. It is an automated process which may even generate use case diagrams automatically. In this chapter we describe the metadata concerning groupware applications that we consider important to specify such particular software.

The rest of the chapter is structured as follows: Section 2 briefly shows the use of templates in the requirement gathering stage. The templates and the metadata we propose in the requirements gathering of groupware applications are described in Section 3 including some examples. Finally, some conclusions about the work are presented in Section 4.

2. Templates to specify the requirements

Some process models for Requirements Engineering provide three stages: requirements gathering, requirements analysis, and validation. We have based our proposal on the Amador Duran's process model [3], which matches with this three-stage approach and proposes some techniques in each one. In the first stage, he proposes some templates to gather information to specify a system. We extend these templates with some other metadata we consider important for the specification of groupware applications, as well as some new templates have been developed following the same purpose.

Known techniques for requirements gathering may be used in this stage to identify important data about the system:

- Interviews are a natural way of communication among people. Some authors such as [12] consider three steps: the preparation of the interview, the interview itself, then the analysis of the results.
- The Joint Application Development (JAD), the IBM technique developed in 1977, usually provides good results; however it takes too much time and too many people.
- The brainstorming is widely used because it is very easy to implement and ideas are generated easily.
- Use Cases, which were proposed by Jacobson in 1993 [8], are traditionally used in Software Engineering especially in UML [1, 11] for specifying the required usages of a system.
- Observations, study of documentation, questionnaires, immersion, are some other common techniques.

The previous techniques or any other one may identify the information which is necessary to specify a system. Templates are a way to put such information in a semi-formal manner.

Durán proposes several steps to gather the requirements of a system. First you should collect the information regarding the problem domain in the current system or in the current context. Then system objectives are identified and described by means of specific templates. Such system objectives come from the division of the main problem in several sub-problems by following the *divide-and-conquer* technique. Once the main problem has been divided, functional requirements, non-functional requirements and information requirements are identified in every system objective. Such requirements are also described by way of specific templates whose metadata specify every requirement. Finally, requirements and objectives are organized and prioritised. In the later sections the templates and the metadata we propose for groupware applications will be described.

As a result of this requirements gathering stage, Durán proposed a System Requirements Document. This document consists of a manuscript containing all the information which have been gathered organized by means of the several templates. One of the key points are the traceability matrixes, which make explicit some information and relationships among requirements and objectives which is implicit in the specification in order to provide the developer with an easy view.

3. Templates and metadata for groupware applications

Based on Durán templates [3], we have defined the ones we consider necessary in order to gather the requirements for a groupware application. There is a template to gather the information regarding objectives and requirements which consist of three parts: a general template with the common metadata concerning both objectives and requirements, then a specific extension with different metadata for system objectives and for requirements, and finally a CSCW extension with metadata regarding groupware issues in case it is necessary. Every system objective is specified by way of a Use Case Diagram. The most important use cases will be described by means of such templates for requirements.

Since the participants of a groupware application should be taken into account as members of groups interacting among them and they are the key point in such systems, it is very important to gather information about them from the beginning. Some templates are proposed to accomplish this situation. First the participants as part of something else are considered by means of the organizational structure template. Then every participant is specified by specific templates. They may be actors, groups, individuals, users, or agents.

The aforementioned templates will be described in the following sub-sections and the methodology to accomplish this requirement gathering will be described later.

3.1. General template for system objectives and requirements

The general template for system objectives and requirements contains the common metadata which describes the sub-systems in which the system has been divided, the functional requirements, the non-functional requirements, and the information requirements:

- Unique Identification is *OBJ-<id>, IR-<id>, FR-<id>,* or *NFR-<id>* depending on the element to be described.
- *Version* allows changes and a record of them.
- *Author* addresses who is the person that has gathered this piece of information.
- *Sources* provides the origin of the information.
- *Importance* indicates how important this objective or requirement is.
- *Urgency* shows if the development of this objective or requirement should be carried out immediately.
- *State* indicates the current level of development if necessary.
- *Stability* shows the necessity of future changes.
- *Comments* provide a place where to note something else which is relevant.
- *Related Objectives* and *Related Requirements* bring system objectives and requirements into relationship.

Previous metadata were proposed by Duràn. Other two metadata have been introduced to consider awareness and who are the participants in the system:

- *Awareness issues* provide a way to know who should be informed about what, how, etc. Awareness is becoming more and more important due to the number of participants in groupware applications, and due to the amount of data managed. Such issue should be taken into account from the beginning. A metadata in this template is a first step. It is necessary to consider what is important to whom, i.e. a description of the information itself, the way to they should be aware of such information, i.e. web notifications, email, avatars, etc.; when they should be know the piece of information, where, and why. This metadata could be allocated in the CSCW extension to the general template because concerns to such kind of information, however we have decided to introduce it on the general template because sometimes this information may be required form objectives or requirements which are not typically collaborative.
- *Participants* is a metadata which depicts who or which are doing something in the system. The most important thing is to know what they are going to do in the groupware application.

3.2. Objective, requirements, and CSCW extensions

In this sub-section the specific extensions for objectives and requirements, as well as those regarding CSCW issues, will be described. Metadata regarding requirements are the traditional information that should be considered when gathering them (they have been introduced in the templates by Durán), whereas the CSCW ones are part of our proposal. All these metadata contribute to complete the general template for system objectives and requirements.

The specific metadata for *objectives* are three: a *description* about the objective, and *up-dependences* or *down-dependences* which describe the hierarchical relationships among the different system objectives. As mentioned before, a system objective represents a sub-problem in which the main problem has been divided.

Information requirements have also three metadata: a specific *description*, information concerning *specific data*, and a *temporal range*, i.e. past and present or only present.

Functional requirements consider traditional metadata such as *pre-condition*, *normal sequence*, *post-condition*, *exceptions*, *frequency*, *performance*, and *description*.

Non-functional requirements only consider a *description* as a particular metadata extension.

When an objective or a requirement regarding groupware application issues is described by means of the templates, it is also necessary to consider the special groupware applications' features. The metadata proposed as a new extension is the following:

- *CSCW description*. In addition to the previous general description in the general template, it is possible to complete the specification of the objective and/or the requirement considering another description only with CSCW information. Such explanation describes their collaborative nature.
- Environment description. The environment where the requirement will be carried out should also be well-known: rooms, equipment, resources, etc.
- *CSCW features*. Those concerning coordination, collaboration, cooperation, and/or communication.
- *Time/space features*. Groupware applications could be synchronous or asynchronous, i.e. in real time or not, and in the same place or in different places.
- *Demand level*. For example, a surgical procedure made in the distance through a groupware application must work in real time and without any delay because a human life could be at stake. However, some delays may be allowed in a chat application.

From the requirements identified in this requirement gathering stage, the analysis stage will be able to identify the different tasks to be performed in the system.

As an example of use, Table 1 shows the specification of a functional requirement called *document edition* by means of the metadata of the general template and the extensions introduced for to consider the specific features concerning CSCW systems. *Document edition* is a requirement of a bigger case study about a system which provides users with a mechanism to elaborate and review documents among several people considering groupware issues.

Table 1. Description of the functional requirement called *Document edition* with the proposed template

RF-8	**Document Edition**	
Version	1 (04/06/2007)	
Author	• Victor M. R. Penichet (Researcher)	
Sources	• Textual description from meetings	
Related objec-tives	• #OBJ-3: Synchronous elaboration of documents to be published	
Related require-ments	• #{RI-<*id*>, RF-<*id*>, RNF- <*id*>} (<*requirements_name*>)	
Importance	Very High	
Urgency	High	
State	Specified; To be implemented	
Stability	Could change, but currently stable	
Awareness issues	The following actors should be aware of this requirement: • #G-1 (AUTHORS): - *What*: an actor is modifying part of the current document - *How*: current modification is showed graphically - *When*: in real-time - *Where*: in the same workspace, in the same window - *Why*: to know who is modifying what and not to interfere - *What*: an actor modified a document - *How*: a past modification is showed by e-mail - *When*: after saving the current version, asynchronously - *Where*: in the actor's intranet and by e-mail - *Why*: to know who modified the document and what part	
Participants	Actors which collaborate: • #G-1 (AUTHORS)	
Comments	None	
Description	The system will behave as follows when a user belonging to the group #G-1 (AUTHORS) edits the document.	
Pre-condition	The document exists; the user is a registered one belonging to the group #G-1 (AUTHORS); a work session has been started	
Normal sequence	**Step**	**Action**
	1	The user select the document to be modified
	2	The use case RF-7 (Session validation) is performed
	3	The use case RF-9 (Document validation) is performed
	4	The user selects the tool to be used
	5	The user marks where the change will be done in the document
	6	The user makes the modifications
	7	The system notifies the modifications to the authors
	8	The user saves the modificaitons
	9	The system notifies the modifications by e-mail and in the intranet
Post-condition	Save or lose changes	
Exceptions	**Step**	**Action**
	-	-
Performance	**Step**	**Time**

	7	Real-time
	9	Asynchronously
Frequency	Several times in every session and in every user	

CSCW description	Because of the collaborative nature of the current requirement: • Notifications are necessary for user awareness • Insertion, modification, and modification in a document are issues to be careful. Awareness in real time is important. Some actions such as deleting an image could be too fast for the rest of authors to be aware. They should be aware in some way. • Real-time feeling in the document elaboration is important but not vital.
Environment description	The environment will be: • --
Coordination	No
Cooperation	Yes. A document is elaborated in real-time by several authors. There are critical sections where a user cannot modify anything if another does: check in / check out.
Collaboration	No
Communication	No
Space	Different
Time	Synchronous when editing a document: authors are aware of modifications in real-time. Asynchronous: when a document is saved, changes are sent by e-mail.
Demand level	Not so high.

3.3. Templates for the organizational structure, actors and extensions for groups

As mentioned before, application's users may be geographically distributed, working in a synchronous way, and so forth. In the specification of software systems, social structures and users' collaborations are taken more and more into consideration so that the analysis of cooperative concerns can be done in depth. The organizational structure of the participants in a collaborative system represents the distribution of its elements.

Participants of the system and the relations among them are described by means of two templates: a template for the organizational structure of the participants of the system, and a template for each participant. If the participant is a group, an extension is provided.

Participants or organizational items compose the organizational structure of the participants of an organization, that is, they compose the logical structure of the users that will use a collaborative application:

- An *actor* is one or several persons or another external system that interacts with the system. Actors directly interact within the collaborative system to accomplish individual tasks, but they can also interact with each other, through the

system, to perform cooperative tasks. This is a key aspect in the difference between human-computer interaction and human-computer-human interaction (CSCW).
This concept was firstly introduced by the object-oriented software engineering paradigm [8]. Nowadays, UML 2.0 [11] also uses this approximation. However, as it was pointed by [2], the use that Jacobson made of this term could be confusing because actor seems to mean the same as role. Constantine indicates that, colloquially, 'actor' refers to the person playing a part, and 'role' refers to the part being played.
The approach we use is closer to Constantine's one, thus actor does not represent the roles played by humans, hardware and other external systems, but such humans, hardware and other external systems.
An actor is a general concept which can be concretized into a group or into an individual item. In the same way, an individual item could be concretized into a user or an agent (defined later).

- A *group* is a set of individual or collective actors which play roles. Such a set of actors needs to interact together and to collaborate with each other in order to achieve a common objective. Common objectives would not be reachable without such collaboration. Groups can also be defined as part of other groups, so the whole organization could be seen as the largest group. A group itself can play a role.
- An *individual item* is a unique actor which plays a role. In other words, it is not a set of actors, but just one.
- A *user* is a human individual item who interacts with the system. We understand that some other artefacts could also interact with the system, and these artefacts could not be people. Accordingly, an agent is a non-human individual item. All the users are actors. All the agents are actors. But neither all the actors are users, nor agents.

From the description of every participant in the system, the analysis stage will be able to identify the different roles to be played by the participants in the system.
The template to describe the organizational structure of the participants includes the following metadata:

- *Version*, *author*, *sources*, and *comments* metadata have the same meaning that the ones in the general template for requirements.
- Then *actors*, *groups*, *individual*, *users*, and *agents* are identified separately.
- Description outlines how the participants are organized, their hierarchical relationships, and other associations.

The template to describe the each participant includes the following metadata:

- *Version*, *author*, *sources*, and comments metadata have the same meaning that the ones in the general template for requirements.
- *Description* of the actor.

- *Supergroups* express belonging relationships between actors and groups and their relationships.
- *Up-hierarchy* and down-hierarchy depict the hierarchical relationships among actors. For example, a user could be the boss of another one.
- *Other associations* among actors.
- *Capacities*. A capacity is an ability or a responsibility associated to an actor, which allows him to play certain roles and to perform tasks. Actors have capacities. Roles could require an actor to have some capacities in order to play such a role.

If the actor to be described is finally a group, it is also necessary to use the following metadata:

- *Common objective*. From the definition of group: *a group is a set of individual or collective actors which play roles. Such a set of actors needs to interact together and collaborate with each other in order to achieve a common objective*.
- *Membership*. Since a group is a set of individual or collective actors, membership outlines such relationship.
- *Laws*. A law is a rule required by a group. This law conditions which actors could belong to the group according to social rules, cultural rules, actor's capacities, and so forth.

4. Conclusions

In this chapter a proposal to the requirements gathering stage for groupware applications has been presented.

The approach extends the proposal of Amador Durán [3] who uses templates to gather the information at the very beginning after using traditional techniques such as brainstorming or interviews. We have modified, elaborated and extended such templates with particular metadata regarding groupware applications.

The templates and the metadata we propose provides information about the system objectives, the information requirements, the non-functional requirements, the functional requirements, the organizational structure of the participants in a system, and the participants of the system. A participant may be identified as an actor, a group, or an individual item, i.e. a user or an agent.

This proposal has been used in a complete process model to collect the requirements of groupware applications in the first stage of the software life cycle: requirements gathering. This metadata was integrated in the whole process model. Requirements and actors are identified and described in the requirements gathering stage and they are used to identify tasks and roles in the analysis stage. It is an automated process which even may generate use case diagrams automatically.

In this chapter we describe the metadata concerning groupware applications that we consider important to specify such particular software.

Acknowledgements

This work has been partially supported by the grants JCCM PAI06-0093-8836 (Junta de Comunidades de Castilla-La Mancha) and TIN2008-06596-C02-01 (Ministerio de Ciencia e Innovación).

References

1 Booch, G.; Rumbaugh, J.; Jacobson, I.: The Unified Modeling Language User Guide. Addison–Wesley, 1999
2 Constantine, L. L. and Lockwood, L. A. D., Software for use: a practical guide to the models and methods of usage-centered design, Addison Wesley, Reading, Mass, 1999.
3 Durán, Amador: Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información. Doctoral Thesis. University of Sevilla. 2000
4 Greenberg, S.: The 1988 conference on computer-supported cooperative work: Trip report. ACM SIGCHI Bulletin, 21(1), pp. 49-55, July 1989.
5 Greif, I.: Computer-Supported Cooperative Work: A Book of Readings. Morgan Kaufmann, San Mateo CA, 1988
6 Grudin, J. CSCW: History and Focus. University of California. IEEE Computer, 27, 5, 19-26. 1994
7 Horn, Daniel B., Finholt, Thomas A., Birnholtz, Jeremy P., Motwani, D., Jayaraman, S.: Six degrees of Jonathan Grudin: a social network analysis of the evolution and impact of CSCW research. ACM conference on Computer supported cooperative work, 2004. p 582 – 591
8 Jacobson, I.; Christerson, M.; Jonsson, P. y Övergaard, G.: Object–Oriented Software Engineering: A Use Case Driven Approach. Addison–Wesley, 4a edición, 1993
9 Johansen, R. (1988): Groupware: Computer support for business teams. New York: The Free Press
10 Johnson-Lenz, P. and Johnson-Lenz, T.: Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium. In Hiltz, S. and Kerr, E. (Ed.), New Jersey Institute of Technology, Newark, New Jersey, 1981.
11 Object Management Group. UML Superstructure Specification, v2.0; 2005
12 Piattini, M. G.; Calvo-Manzano, J. A.; Cervera, J.; Fernández, L.: Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. ra–ma, 1996
13 Poltrock, S. and Grudin, J. 1994. Computer Supported Cooperative Work and Groupware. In Conference Companion on Human Factors in Computing Systems (Boston, Massachusetts, United States, April 24 - 28, 1994). C. Plaisant, Ed. CHI '94. ACM Press, New York, NY, 355-356
14 Poltrock, S. and Grudin, J. 1999. CSCW, groupware and workflow: experiences, state of art, and future trends. In CHI '99 Extended Abstracts on Human Factors in Computing Systems (Pittsburgh, Pennsylvania, May 15 - 20, 1999). CHI '99. ACM Press, New York, NY, 120-121
15 Poltrock, S. and Grudin, J. 2005. Computer Supported Cooperative Work and Groupware (CSCW). In Interact 2005. Rome, Italy.