Towards an Engineering Approach for Groupware Development

Marco Aurélio Gerosa

Software Engineering Laboratory (LES) – Computer Science Department
Catholic University of Rio de Janeiro (PUC-Rio)
R. Marquês de São Vicente, 225, Gávea, Rio de Janeiro – RJ, 22453-900, Brazil
gerosa@inf.puc-rio.br
http://groupware.les.inf.puc-rio.br

Abstract. Software Engineering, which has advanced substantially in the development of single-user applications but only recently started addressing the human factor problem, fails to support group aspects so needed in collaborative applications. The groupware engineering enhance software engineering with CSCW concepts to develop collaborative applications. The groupware engineering cycle is based on the 3C collaboration model.

1. Research Problem

A Groupware Engineering approach is being proposed in order to address collaboration aspects during software development. Groupware Engineering formulates systematic and disciplined approaches to the development and maintenance of groupware, based on Software Engineering principles enhanced by concepts originated from the fields of CSCW and related areas. The groupware engineering approach is based on the 3C collaboration model, which analyzes collaboration in terms of communication, coordination and cooperation concepts. The concepts of groupware engineering are being applied in the development of a new version of the AulaNet learningware.

Figure 1. Groupware development cycle.
Collaborative systems are especially prone to failure [6]; hence demand iterative evaluation during their development. Ideally, groupware should be prototyped [12], but given the excessive cost of throwing code away, as demanded by “pure” prototyping, an incremental model is more adequate. The groupware engineering cycle is based on the spiral software development model [2], which combines the classical sequential model and the iterative behavior of incremental prototyping. The Groupware Engineering cycle is presented in Figure 1.

2. Stage of the Research

Software engineering principles and techniques were analyzed and adapted to groupware development, as can be observed inside the arrows presented in Figure 1.

The domain analysis of Groupware Engineering is supported by the 3C collaboration model, which is based upon the concepts of communication, coordination and cooperation. General groupware requirements that are elicited in the requirement analysis phase seldom are clear enough to enable a precise specification of system behavior. Incremental prototyping makes it possible to constantly evaluate the design and implementation, thus counterbalancing the necessity of having a complete set of requirements to start the design. Groupware design patterns, UML extensions, groupware architecture and frameworks are used in the design phase. Toolkits and groupware components are used during the groupware implementation. Groupware heuristics guide the system testing.

The 3C model was refined based on its application in the design and evaluation of groupware and distance education courses. The 3C diagram can be observed in Figure 2. This diagram is an extension of models found in the literature, such as the model proposed by Ellis et al. [3] and the Clover design model [7].

![Figure 2. Overview of the 3C collaboration model](image)

According to this model, the communicational apparatus transmits and registers information and the group interprets the messages, updating their commitments and knowledge. That triggers reactions and negotiations take place. This way, the group moves into an argumentation process where they negotiate commitments and, therefore, their knowledge.

The group has to coordinate the ensuing activities designed to enforce the fulfillment of the commitments. Coordination organizes the group to avoid the loss of communication and cooperation efforts and to ensure that the tasks are carried out in the correct order, at the right time and in compliance with restrictions and objectives [10]. Coordination involves the pre-articulation of tasks, their management and post-articulation. Pre-articulation involves actions that are necessary to prepare collaboration, normally concluded before cooperation begins. The post-articulation occurs after the end of the tasks and involves the evaluation and analysis of the collaborative process. The management of the carrying out of the tasks is the act of managing interdependencies between tasks that are carried out to achieve an objective [8].
Cooperation is the joint operation within a shared workspace. Group members cooperate by producing, manipulating and organizing information and building and refining cooperation objects. Expression elements are the means for acting upon cooperation objects, while awareness elements display the results of a participant action (feedback) and the action of their colleagues (feedthrough).

A new AulaNet environment architecture, illustrated in Figure 3, was designed based on the 3C model principles and is being prototyped.

**Figure 3. The AulaNet architecture**

A groupware must be sufficiently flexible to be adapted to the group characteristics and to the evolution of work processes. Although there is no way to foresee all the features that will be demanded from a groupware, different groupware products share a number of characteristics. This scenario is suitable for the application of component-based development techniques, which provides the flexibility needed in projects with changing requirements [13]. In such situation, groupware services can be plugged and unplugged from the system. The system architecture comprises component frameworks, which define overall invariants and protocols for plugging components.

In the AulaNet architecture, the AulaNet Component Framework defines the general functionalities common to all services, like the management of services interaction and data sharing. Currently, there are three different families of services: collaboration, administrative and guest services, which corresponds to components frameworks that deal with characteristics specific to each service.

Current AulaNet services are also developed using a component framework-based architecture. There is a common structure implemented by the collaboration framework, which defines the skeleton of the service, and plugged into this framework, there are the communication, the coordination and the cooperation component frameworks, which gives support to each C. Class frameworks are used to implement components, that are plugged into the corresponding C-framework and implement the specific functionalities of the service.
3. Outline of Objectives and Expected Outcome

This research aims to develop a groupware engineering process and implement a flexible architecture that will be used to instantiate the AulaNet learning environment and other groupware systems.

4. State of the Art

There are different techniques suitable for the design phase, namely, groupware design patterns [5] for reusing common approaches of design; UML extensions for representing groupware specific aspects; groupware architectures [14] and groupware-related frameworks [9] for reusing code and infrastructure. For the implementation phase, toolkits [11] and groupware components [14] are alternatives for building collaborative systems. Groupware heuristics [1] may guide experiments to test the system.

5. Methodology

The groupware engineering concepts have being iteratively formulated and tested in the development of a new component-based system architecture to the AulaNet learning environment.

For example, a previous version of the AulaNet Debate service was implemented with a communication component, which implements synchronous communication protocols, and a cooperation component, which implements the shared space, as can be seen in Figure 4. This version of the Debate is a plain chat tool, holding an expression element, where learners type their messages; and awareness elements, where learners participating at the chat session are presented.

![Figure 4. Debate architecture.](image)

This version of the Debate service gives no support to coordination, leaving it to the standing social protocol. However, some courses use a well defined procedure to the debate activity [4]. In order to better support tightly integrated activities, in the current version of the Debate service, coordination mechanisms were implemented. Floor control, participation order and shared space blocking ability were added to the service. The shared space was also enhanced by new awareness elements, like session title and time stamp.
Figure 5. New Debate Architecture

In the new Debate version, the same communication component was used, as the synchronous communication protocols and the characteristics of the messages didn’t change. The cooperation component, which implements the shared space, was enhanced by new awareness elements. And a new coordination component, which implements the coordination mechanisms, was implemented and plugged into the Debate architecture (Figure 5).

6. Conclusion

Based on the 3C model, in order to collaborate, individuals must debate ideas (communicate), be in tune with other participants of the group (coordinate) and operate together in a shared space (cooperate). The groupware component system architecture used in the AulaNet environment mirrors the 3C collaboration model. Communication, coordination and cooperation functionalities are directly mapped into the implementation of AulaNet collaboration services. The redesign of the AulaNet Debate service illustrates this mapping and the modularity achieved using the component system architecture.

The case study shows the need to have a component-based architecture that reflects the collaboration model. Using a groupware system architecture and component frameworks facilitates the task of programmers, who can reuse and extend data structures provided by frameworks, leaving to the infrastructure provided by the groupware architecture the support of some specific multi-user aspects. This research aims to investigate the groupware engineering development cycle, applying its concepts in the design of component based architecture to the AulaNet environment.

Acknowledgement

The AulaNet project is partially financed by the Fundação Padre Leonel Franca, by the Ministry of Science and Technology through its Multi-Agent Systems for Software Engineering Project (ESSMA) grant nº 552068/2002-0. It is also financed by individual grants awarded by the National Research Council to Marco Aurélio Gerosa nº 140103/02-3.

References

5. Groupware Patterns Swiki (2003), http://swiki.darmstadt.gmd.de/gw-patterns