A Proposal for Model-Based Design and Development of Group Work Tasks in a Shared Context

Fernando Gallego[1], Ana I. Molina[1], Crescencio Bravo[1], and William J. Giraldo[2]

[1] Computer Science Department, University of Castilla - La Mancha
Paseo de la Universidad, 4 - 13071 Ciudad Real (Spain)
{Fernando.Gallego,AnaIsabel.Molina,Crescencio.Bravo}@uclm.es
[2] Ingeniería de Sistemas y Computación, Universidad del Quindío, Quindío, Colombia
wjgiraldo@uniquindio.edu.co

Abstract. The design and development of groupware systems is a difficult task, especially due to their multidisciplinary nature and the technical complexity of these kinds of systems (e.g. distribution, data sharing, multi-user interfaces). A model-driven development approach could help to deal with this research problem. This paper presents an approach to tackling the design and development of groupware applications. This approach is part of a framework for the model-based user interface development of collaborative applications (called CIAF; *Collaborative Interactive Application Framework*), which includes issues of particular relevance based on the use of several models and notations for representing the collaborative and interactive aspects of this kind of systems.

Keywords: groupware design, model-based design, groupware production.

1 Introduction

In order to speed up and facilitate the development of groupware applications by the developers of such systems, the goal is to automate, as far as possible, the creation of groupware, in such a way that the developer can handle models and higher levels of abstraction, and thus obtaining quality collaborative applications.

Molina et al. [1] presented a review of the main notations and approaches for the analysis and modeling of the interaction and collaboration within groupware systems. Also, a review of current literature relating to the model-based design and development of applications providing support to group work [2-8] has been undertaken. None of these works tackle the development of groupware applications via models that are capable of generating both the user interface and the functionality of the application. CIAF (Collaborative Interactive Application Framework)[9] is established in this line of work. CIAF proposes a model-based methodological approach to the development of the user interfaces of collaborative applications. CIAF proposes the integration of the specified information by means of a notation known as CIAN (Collaborative Interactive Application Notation)[7], dedicated to the modeling of the interactive aspects of group work, and using information gathered from UML models. However, this proposal focuses primarily on the more interactive aspects of the application (its user interface) at a very high level of abstraction,

Y. Luo (Ed.): CDVE 2010, LNCS 6240, pp. 11–18, 2010.

leaving aside many of the more practical aspects. Also, this framework does not allow designing purely collaborative tasks in which the users access shared context simultaneously. Therefore, in our work we aim to take this framework as a starting point and to extend it to also consider these practical aspects. Therefore, our research hypothesis can be formulated as follows: "*it is possible to propose a methodological framework for the model-based design of groupware applications which is capable of generating fully functional and complete collaborative applications*". In order to validate it, our main goal is to develop a framework for designing group work tasks that supports the generation of a fully functional collaborative application from this design. Due to the lacks detected in CIAF, our first partial goal is the model-based design of work tasks in a group with a shared context. And, the second partial goal is know the collaboration mechanisms of the low-level abstraction that are involved in these subtasks, both functional and more interactive level. In this paper, we only focus on the model-based design of tasks for group work in a shared context.

This article is organized into four sections. Section 2 describes in depth how CIAF is complemented by our proposal for the design and modeling of the purely collaborative (or functional) aspects of the end user interface. Lastly, Section 3 presents the conclusions that have been reached and the future lines of research derived from this work.

2 Towards a Model-Based Framework for Design and Development of Group Work Task in a Shared Context

As a solution to the hypothesis presented above, we present an approach organized into stages and based on components for the design and generation of groupware applications. To design the most relevant aspects of an interactive system for supporting group work, already existing notation will be used or new notations will be proposed where necessary to represent certain aspects. From these models, the generation process will take place, based on the selection of those collaborative components that will eventually form the user interface of the final tool. The steps that make up this proposal are explained below, along with the selection method that our framework uses to decide upon the most appropriate components to form part of the final user interface.

CIAF allows for the modeling of individual and group-work tasks. The specification for the first type can handle different levels of abstraction, down to the most detailed levels, in which the CTT notation[10] is used to specify particular aspects of human-computer interaction. Using CTT a developer can achieve high levels of detail describing the model of interaction, and the obtaining of the final design of the user interface is facilitated. As for the modeling of the group-work tasks, CIAF makes a distinction between cooperative and collaborative tasks. This distinction is based on the definitions of the terms cooperation[1] and collaboration[2] as given by Dillenbourg[11]. CIAF proposes breaking down cooperative group-work

[1] Cooperation implies that group member pursue the same goals, but act independently in terms of their own work but in separate parts within a shared context.

[2] Collaboration implies that individual members work together within a common space of representation.

tasks into independent individual tasks because CTT only allows the modeling of cooperative tasks that are specified by a cooperative model that relates individual tasks. However, CTT cannot be used for modeling collaborative tasks and shared contexts. Obtaining the final user interface, for individual and cooperative tasks alike, presents no problem since for both types of tasks a CTT model is created, allowing the appropriate corresponding user interface to be obtained. However, obtaining the interface for collaborative tasks is somewhat more complex because they cannot be broken down into individual, independent tasks. As such, we have found that the CIAF methodological framework does not allow for the generation of user interfaces for purely collaborative tasks in which there exists a shared context and less coordinated and independent interactions between users, who access this context simultaneously. To address this problem, we propose the method described below, using the CIAF methodological framework. These phases that make up this method are: **(1)** *Modeling of the group-work tasks*, **(2)** *Modeling of the interaction of group-work tasks*, **(3***)* *Relating the group-work tasks and the application domain* y **(4)** *Designing the interaction of group-work tasks*. Before explaining in detail what the steps above consist and how each stage is carried out, we explain how the selection of components for supporting the group-work tasks is made. This is necessary in order to better understand the proposed process.

2.1 Selection of Components for Supporting Group Work Tasks

This section explains how to perform the selection of components for supporting group-work tasks. This selection is critical to the design of group-work tasks and allows one to identify which components will form part of the user interface.

Firstly, we conducted a systematic review of those groupware tools already documented in current literature. From each one of these tools we identified the visual components from the graphical user interface that support group work, making a distinction between those of them without any connection with the domain of the tool providing support and those of them linked to the domain. For example, a communication component, such as a chat feature, has no connection with the application domain whereas another component, such a specific collaborative editor, does have a connection with this domain. The aforementioned review was also intended to measure the degree of coverage that each of the components had in terms of the six dimensions of the Zachman framework [12]. This taxonomy consists in a two-dimensional matrix, where the columns show different *views* of a system (*What, How, Where, Who, When, Why*) and, the rows indicate the different levels of specification or *perspectives* to take into account in its development (technological, system, and business level). As such, our next step consisted in classifying the identified components according to the cells of the Zachman taxonomy in the system's perspective, where the relevant aspects are designed according to the technology that is to be used. After analyzing the components that usually form part of end collaborative user interfaces, the next step is to define a set of *decision rules* to makes possible, from a set of parameters, to decide which component is more suitable to provide better support for some specific collaboration requirements. As decision parameters we take the views of the Zachman framework. Up to now, we focus on the views *What*, *How* and *When*. Focusing on the *How* view, we consider the various collaborative mechanisms proposed by Pinelle in [13], defined as the basic operations and interactions to be performed by the workgroup to carry out a task that must take place to get a job done within a collaborative framework. These collaborative mechanisms can be modeled using the method proposed in the phase *Relationship between the group-work tasks and the application domain* since this relationship is determined by the action carried out by the user upon the task, which will imply the way in which the data will be handled. To define the actions that can be carried out upon the found collaborative components, certain changes have been made to the table of collaboration mechanisms provided in [13]. In Table 1 below you can see the different collaboration mechanisms sorted by categories, along with the action that each performs and the symbol that is used to model it:

Table 1. Collaborative Mechanisms

Category	Symbol	Mechanism	Action
Communication	💬	Spoken messages	Conversational. Verbal shadowing
	💬	Written messages	Conversational. Persistent
	💬	Video messages	Conversational
	💬	Manifesting actions	Stylized actions
	💬	Deitic references	Pointing conversation
Information Gathering	◆	Visual evidence	Normal actions
	◆	Basic awareness	Observing who is in the workspace, what they are doing and where they are working
	◆ R	Feedthrough	Changes to object. Characteristics sings or sounds
Coordination	⚙	Basic coordination	Allows users to be coordinated
Shared access	R	Obtain resource	Physically take objects or tools. Occupy space
	R	Set resource	Physically set objects or tools
Transfer	⬇	Deposit	Place object and notify
	⇄	Handoff object	Physically take/give object. Verbally offer/accept object

The next parameter to consider in the designed decision system is related to the *What* view. Therefore we considered those activities in which the use of groupware components is most common. We have analyzed each component of those under review, considering the essential task for which that particular component is used. The following list of tasks has been compiled: search, share, communicate, coordinate, decide, draw, discuss, edit, report, swap, browse and plan. Last of all we must consider the *When* view. In this case we consider the type of collaboration that is to take place: whether the interaction is to occur synchronously or asynchronously. Information pertaining to these two dimensions, *What* and *When*, is added to the models created in the phase *Configuring the interaction of group-work tasks*.

After analyzing each component, along with the considered parameters, we obtain a system of decision rules which allow us to select any collaborative component out of those collected. The inference rule for the components is as follows:

What x How x When => Component

Or expressed in another way:

Activity x Collaborative mechanism x Collaboration type => Component

Thus, when modeling a group-work task, configuring the type of activity it will be used to complete, the way in which the interface information collaboration will occur, and whether this activity is synchronous or asynchronous, the most appropriate graphical component for supporting said collaboration can be generated. For example:

Edit x Set resource x Synchronous => Edit panel

Draw x Gestural messages x Synchronous => Annotation panel

From this moment onwards, the groupware application continues to be developed following the guidelines and phases of the CIAF framework, which were explained above. To solve the identified problem, namely that the generated collaborative application must be functional and complete, all of the graphical components of the user interface are generated with their own functionality, as a *black box*. Thus, the final user interface product will be composed of fully functional components, in turn ensuring that the final product itself is also fully functional.

2.2 Stages of the User Interface Generation Process for Groupware Systems

This section explains in depth what each one of the phases that make up the proposed process covered above consist of.

Modeling of the Group Work Tasks. This phase is the starting point for our proposal The purpose of which is to increase the level of detail in the specification of group-work tasks. These tasks are broken down into individual tasks and group-work tasks of a lower level of abstraction, for which CTT[10] task diagrams will subsequently be created. These subtasks form an activity graph via which the dependencies that exist between the subtasks can be modeled. At this stage, just as proposed in CIAM [7], it will be possible to specify the tools used to support the collaborative activity (e.g. chat feature). These components will only be able to be added in group-work tasks or subtasks.

Modeling of the interaction of Group Work Tasks. The purpose of this phase is to create the interaction model associated with each of the individual and group-work subtasks to represent the computer-person interaction between the user and the interactive system. Its objective is to describe the division of the activity and its classification of interaction and implementation types according to who is responsible for leading a task. At this level, the breakdown of the subtasks is represented using CTT task models, since they are highly suitable for designing usable interfaces. This proposal adds a new kind of task to the CTT models, allowing for the modeling of those tasks that require interaction within a shared context (collaborative tasks), since CTT only allows for the modeling of individual and cooperative tasks.

Relating the Group Work Tasks and the Application Domain. This stage specifies the way in which the interactive part of the system relates to the information. Once the task model has been defined, the data manipulated by the user interface must be connected. From the point of view of the interface, the execution of a task requires a series of actions to be carried out upon the interface elements. As such, in order to select the graphical component that supports group work in a shared context, to the new task type included in the CTT we add the way in which the user interacts with the domain data. Such actions are called *collaborative mechanisms* [13].

Designing the Interaction of the Group Work Tasks. So far, the information that has been provided is not yet sufficient for the generation of the components that support group work within a shared context, since it would not be possible to clearly distinguish the components that are being designed. As a possible solution, at this stage the CTT task is configured for the modeling of the group-work tasks. This configuration uses the provided information that is necessary to clearly distinguish each component. Said information will be the *type of collaboration* to be implemented in the task and the *activity* that will be performed with the task. With this information, along with the modeling done in previous phases, the appropriate component for the support of group work can be identified. In section 3.1, this step is explained in depth.

From this point onwards, and following the CIAF methodological framework, the sequence of generating the graphical interface begins, encompassing the consecutive development of the *abstract interface*, the *concrete interface* and the end interface [14] within the activities of the *conceptual design*, the *detailed design* and the *implementation*, respectively. In order for the user interfaces generated via this proposal to be fully functional and complete, each of the components forming the final user interface will have its own functionality. Thus, all of the components that make up the tool will operate independently of each other, making the final product a complete and functional groupware tool.

2.3 Case Study

In order to provide a better explanation of this process, we use a small case study. A teacher proposes an activity, consisting in completing a writing, and their pupils carry out this activity in groups, each using their personal computer and a collaborative tool designed using our proposal. The students will have the help of a faculty advisor who will guide them in the use of the tool and clarify any doubts. This type of activity was chosen for the case study because of its clear need for collaboration between users and for interaction between each individual user and the designed application. Below we detail the steps that make up our proposal.

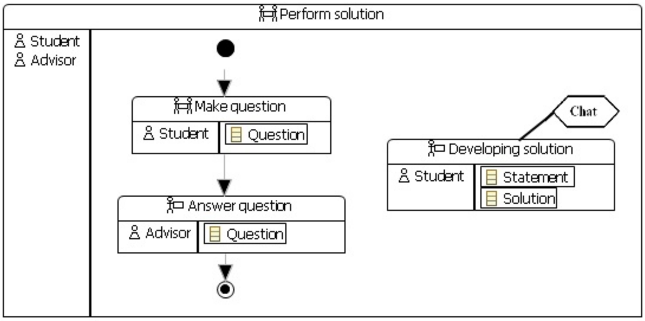


Fig. 1. Modeling group task "Perform solution”

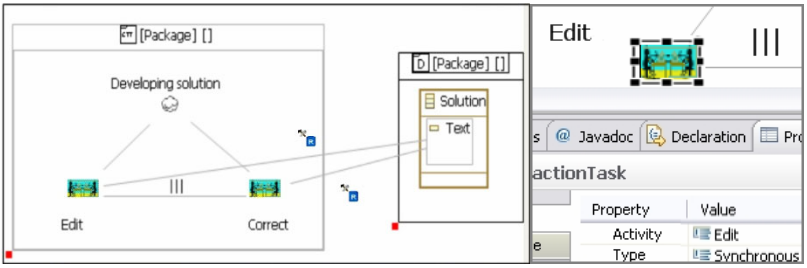
Modeling of the Group Work Tasks. At this point, the tasks that make up our case study are broken down into individual tasks and group-work tasks of a lower level of abstraction. For example, Fig. 1 shows how the group-work task *Perform solution*, which is performed by the roles *Student* and *Adviser*, is designed. This task is broken down into three tasks: a group-work task, *Developing solution*, performed by the *Student* role, and two individual tasks forming an activity sequence, where *Make question* is made by the *Student* role and *Answer question* by the *Adviser* role. Added to the group-work task is a domain-independent collaboration component which facilitates working in groups, in this case a chat feature.

Modeling of the interaction of Group Work Tasks. To illustrate this point in terms of the case study, the *Developing solution* task is broken down into two group-work tasks: *Edit* and *Correct*. For the modeling of these tasks, the new proposed CTT task has been used, since in these tasks users will interact with the shared context, in this case a text editor.

Relationship between the Group Work Tasks and the Application Domain. The relationship between the task model and the application domain for the previous task of this case study is shown in Fig. 2a. These relationships are labeled with the new symbols which indicate the involved collaboration mechanisms. In this case, the relationship type is *Set resource*.

Configuring the Interaction of the Group Work Tasks. The configuration for group work task in a shared context of this case study is shown in Fig. 2b. In this case, the Edit Task, That is a synchronous task and use the mechanism of Collaboration *Set resource.*

With all this information, the modeling framework could generate the component *Edit Panel* for supporting collaboration in this subtask, which would have a fully and independent functionality.



Property	Value
Activity	Edit
Type	Synchronous

Fig. 2. a) Interaction diagram for the task *Developing solution*. b) Configuring the task *Edit.*

3 Conclusions and Future Work

In this article, we have proposed a methodological framework for the model-based design of group work tasks in a shared context, which can generate fully functional and complete collaborative applications. The framework allows a developer to can model the collaboration mechanisms that the user can use in during the collaborative tasks, and it can model the group work tasks in a shared context. Therefore, our working hypothesis is validated. Our work has taken CIAF as a starting point, integrating new notations and mechanisms for the modeling of groupware applications.

Additional work for the future is the generation of collaborative applications which, from the CIAF framework now enhanced by this proposal, have as their final product a fully functional, executable application. This will require the development of a complete toolkit of components designed with collaboration in mind.

References

1. Molina, A.I., Redondo, M.A., Ortega, M.: A Review of Notations for Conceptual Modeling of Groupware Systems. In: New Trends on Human-Computer Interaction, pp. 1–12 (2009)
2. Garrido, J.L., Gea, M., Noguera, M., González, M., Ibáñez, J.A.: Una Propuesta Arquitectónica para el Desarrollo de Aplicaciones Colaborativas. In: Interaccion 2004, Libro de Actas del V Congreso Interacción Persona Ordenador (AIPO), pp. 164–171 (2004)
3. David, B., Chalon, R., Delotte, O., Vaisman, G.: Scenarios in the model-based process for design and evolution of cooperative applications. In: Human-Computer Interaction Theory and Practice, London, vol. 1(LEA), pp. 68–72.
4. Bouassida, I., Sancho, G., Villemur, T., Tazi, S., Drira, K.: A Model-Driven Adaptive Approach for Collaborative Ubiquitous Systems. In: Proceedings of the 3rd Workshop on Agent-Oriented Software Engineering Challenges for Ubiquitous and Pervasive Computing, London, United Kingdom (July 2009)
5. Chalon, R., David, B.: IRVO: an Architectural Model for Collaborative Interaction in Mixed Reality Environments. In: Proceedings of the Workshop MIXER 2004, Madeira (January 2004)
6. Penichet, V., Paternò, F., Gallud, J.A., Lozano, M.: Collaborative Social Structures and Task Modelling Integration. In: Doherty, G., Blandford, A. (eds.) DSVIS 2006. LNCS, vol. 4323, pp. 67–80. Springer, Heidelberg (2007)
7. Molina, A.I., Redondo, M.A., Ortega, M.: A methodological approach for user interface development of collaborative applications: A case study. Science of Computer Programming 74(9), 754–776 (2009)
8. Dumont, A., Pietrobon, C.: A Method for Specification of Collaborative Interfaces through the Use of Scenarios. In: Fifth International Conference on Computer Supported Cooperative Work in Design, pp. 15–19 (2001)
9. Giraldo, W.J., Molina, A.I., Collazos, C., Ortega, M., Redondo, M.A.: A Model Based Approach for GUI development in groupware systems. In: Briggs, R.O., Antunes, P., de Vreede, G.-J., Read, A.S. (eds.) CRIWG 2008. LNCS, vol. 5411, pp. 324–339. Springer, Heidelberg (2008)
10. Paternó, F.: ConcurTaskTrees: An Engineered Notation for Task Models. In: The Handbook of Task Analysis for HCI, pp. 483–501 (2004)
11. Dillenbourg, P., Baker, M., Blaye, A., O'Malley, C.: The Evolution of Research on Collaborative Learning. In: Learning in Humans and Machine: Towords an Interdisciplinary Learning Science, pp. 189–211 (1995)
12. Zachman, J.A.: A framework for information systems architecture. IBM Systems 26, 276–292 (1987)
13. Pinelle, D., Gutwin, C., Greenberg, S.: Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. ACM Transactions on Computer-Human Interacion 10, 281–311 (2003)
14. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López-Jaquero, V.: USIXML: A Language Supporting Multi-path Development of User Interfaces. In: Proceedings of EHCIDSVIS 2004, Dordrecht, p. 3425 (2004)