Design and Evaluation of a Service Oriented Architecture-Based Application to Support the Collaborative Edition of UML Class Diagrams

Victor M.R. Penichet, Jose A. Gallud, Ricardo Tesoriero, and Maria Lozano

LoUISE Research Group – Computer Science Research Institute (I3A), Castilla-La Mancha
University, 02071 Albacete, Spain
{vpenichet, jgallud, mlozano, ricardo}@dsi.uclm.es

Abstract. Developers in modern and geographically distributed software companies need to collaborate because most of them are part of big development teams involved in large projects. CASE tools are commonly used to model software applications. However, most of them are not really prepared to support collaboration in the sense that they do not provide developers with a real collaborative environment. In this paper, we show the design and the implementation of a cooperative CASE tool called CE4WEB to support the edition of UML diagrams using the Service Oriented Architecture (SOA) paradigm. The tool described in this paper demonstrates the successful use of groupware tools and Software Engineering techniques. An evaluation method is also presented to show the validity of the proposal.

Keywords: groupware, SOA, collaborative CASE tool.

1 Introduction

Nowadays, software development tends to be decentralized; online collaboration among different development teams is something that software companies wish to incorporate in their development environments.

Positive effects of the globalization phenomenon are, among others, the possibility of configuring distributed development teams belonging to different countries that collaborate in the same project. SourceForge or Linux are good examples of the distributed software engineering [8].

CSCW research field helps people to work better by means of computers. Software development is one of the scenarios where groupware may improve the way programmers work together providing collaborative tools and methods [7].

In this paper, the design and the implementation of CE4WEB (Cooperative UML Editor for the Web) is presented. CE4WEB is a cooperative CASE tool that supports the edition of UML [12] diagrams. This tool demonstrates the emerging use of groupware tools and techniques in software engineering.

The structure of the paper is as follows: Section 2 describes how collaboration is supported by the common CASE tools. Section 3 describes the design and the implementation of our proposal. Section 4 shows how the tool works in real cases. Section 5

describes the evaluation method we have performed to validate the tool and Section 6 shows the metrics applied and the results obtained from the experiment. Finally, some conclusions and future works are presented in Section 7.

2 Collaborative CASE Tools: An Overview

Traditional CASE tools provide some collaborative functionality; therefore we can consider them as collaborative tools. However, from the CSCW research field point of view, most common CASE tools do not provide a real cooperative scenario for software development. In this section, we describe the way in which these tools support collaboration in the modeling process of software applications with UML.

Modelling provides the first step to improve workgroup, though market demands more powerful tools to support UML modelling by combining the functions of traditional UML applications with the advantages of real time collaboration [9].

The evolution of UML modelling tools covers these stages: standalone tools, repository-based model sharing, Web-based model sharing, Real-time model sharing and collaboration. The main motivation of our work is the lack of UML editors specially designed to support collaboration. Many authors identify this lack as one of the future needs of CASE tools [1, 2, 5, 13].

Rational Rose supports some kind of collaboration by allowing the parallel development of a model by dividing it in versionable units, sharing models among different teams and a complete integration with common version control tools. The Teamwork release of Visual Paradigm for UML offers a similar support for version controlling. Microsoft Visio offers collaboration support by means of SharePoint Portal Server to publish and share diagrams among the different users. EclipseUML is integrated inside the Eclipse framework. Team's members can access to the artefacts from different stages of the development at any time. Enterprise Architect for UML is one of the most powerful tools from the team support point of view. This environment offers the usual version control system and other additional functions to maintain all members of the team informed and connected with the project. PoseidonUML and Koneso are well-known environments that offer an advanced collaboration support for development teams. They supply advanced functions as real-time collaboration, versioning control, central repository, news board, etc.

All these tools have common features: support for UML diagrams, reverse engineering, code generation, version controlling, and so on, but collaboration support is poor or incomplete. CO2DE [11] is the tool we have used as reference. It is a cooperative class diagram editor where users can work simultaneously in the same diagram keeping a version list.

Some authors establish that the main collaborative features that are missed in these known tools [3] are the following:

- Collaboration: Although PoseidonUML and Koneso support collaboration, most of them do not provide real-time collaboration.
- Perception: The tools revised do not support user perception of the work developed by other members of the team.

- Communication: Users communication is not supported.
- Workgroup memory: There is no support for storing the information that the workgroup produces when different people are working in the same project.

3 Designing and Implementing the Collaborative Editor

In this section, we describe the design and implementation of our cooperative editor called CE4WEB. The description is focused on the cooperative aspects of the system.

As a result of the former analysis of the most common UML tools, we have defined the following list of the most important requirements that a cooperative CASE tool should include:

- Allow the geographical distribution of users working in the same diagram.
- Support real time cooperation (synchronous working).
- Support informal and formal communication.
- Store and share a collective memory.
- Provide a way to let users know the work developed by others (awareness).

The first requirement made us to select the Web platform, as some authors recommended [4, 6]. In this way, we can support the distribution of users and their need of synchronous and asynchronous collaboration. The second requirement implies the use of a concurrency control system which includes coordination services and data consistency. Regarding the third requirement, we decided to include the following units: an alarm and notifications system, an asynchronous communication system (electronic mail), a comment and annotations system and a synchronous communication system (chat). The fourth requirement implies the use of a repository or a group memory space implemented by a versioning system. The tool also provides a log file to store all users' actions. And the last requirement can be reached by providing a list of participants, a friendly interface where the changes introduced by any user are immediately reflected on the interface. We have also decided to apply the telepointer technique used in CO2DE [11].

The system should also accomplish a series of non functional requirements such as portability, easiness, low response time, availability, and generality (not only for UML diagrams).

The main components of the system are the following. An UML Editor that allows the creation and modification of UML class diagrams in a cooperative way, a component responsible for the synchronous cooperative process of the UML Editor (SCS), a reduced version of a control system that supports maintenance and management processes (Versioning Control System, SCV), the user management system responsible for controlling and maintaining users and workgroups (SGU), and a system to control the access to the Database containing both the application data and the information generated by workgroups (CAD).

The general view of the system architecture containing the components described above is shown on Fig. 1.

Fig. 1. System architecture

4 Collaborative Editing of UML Diagrams: CE4WEB at Work

We have followed the typical schema of three-layer architecture in the implementation of the system. The layout layer is implemented by the UML Editor and the Web Application. The business layer is implemented by the Users Management System, the Versioning Control System and the Synchronous Collaboration System. The data layer is implemented by the Data Access Component and the database.

The UML Editor allows a real-time interaction among multiple users to create or modify an UML class diagram. The editor offers two main behaviours: as a UML editor, it offers a limited functionality compared to Rational Rose or MS Visio; as a collaborative tool, it offers a shared blackboard, a list panel of connected users, a list panel of actions, the position of the users' pointers, a computer log for actions and changes, and a chat for communicating.

A group of users that want to work together in the same project have to log on the system. The *User and Group System* offer users the possibility of organizing themselves in groups sharing resources and a common space. This part of the system is responsible for group and security management.

When users log on the system, different options are offered to them. Users can access a previous project or create a new one. Users can see the information about the group they belong. An interesting point here is the possibility for a user to join an active session and cooperate with other users in the same group or starting a new session.

The *Versioning Control System* is responsible for offering a shared workgroup memory. The information and artifacts are organized in the following way: a project is composed by a set of diagrams. Each diagram has associated a set of versions. A version is created in a session. A new version is generated when the diagram is modified. Fig. 3 shows a versioning tree, where each node represents a different version of a diagram and the number represents the order in which they were created.

Fig. 2. UML editor's interface

A user can see the information of a specific version by selecting one of the nodes. The information of a version is shown in Fig. 4. This figure has several tabs containing different functionality; reviewing the status of the diagram, showing the list of users' actions, showing a list of the chat's posts, and the users' annotations.

The *Synchronous Collaboration System* supports the communication, coordination and information sharing processes that allow users to work synchronously when they are editing an UML diagram. This component guarantees shared data integrity and is responsible for propagating users' actions efficiently.

Fig. 3. Information page about diagram versions (right) and version tree (left)

The communication between the Web client and the server is implemented by means of Web services. In this way, the UML collaborative editor takes advantage of the SOA paradigm.

The CE4WEB has been developed with C# using a special library called *Netron Graph*. The client application does not need any *ActiveX* or *COM* component but it

uses a technology called Fusion that allows embedding *WinForm* controls in a HTML page. This technology is similar to *Java Applets* or *JavaBeans*. These technologies can be used to build smart Web clients. In this case, the UML editor is encapsulated in a DLL file called *ControlUML.dll* and it is stored in the server, together with the other application components. The assembled is embedded in the Web page *Editor.aspx* using the `<object>` HTML tag.

```
<OBJECT id="EditorUML" standby="Loading UML Editor..."
classid="http:ControlUML.dll#ControlUML.Control_UML"
VIEWASTEXT></OBJECT>
```

The *classid* parameter indicates: the name of the assembled: *ControlUML.dll* and the class implementing WinForm: *ControlUML.Control_UML*. When the *Editor.aspx* is invoked, the assembled travels together with the HTML code to the client browser and is executed in the client side.

5 Evaluating the Collaboration in the CE4WEB Tool

Several interesting quality aspects of the CE4WEB tool could be evaluated. We may divide the system in different layers and select some parameters to be measured in each layer. If we consider architecture, we should evaluate the tool performance by monitoring the CPU and network utilization among other parameters. On the other hand, we might consider collaboration aspects only and focus the evaluation on how the CE4WEB tool helps people to work together. General users' satisfaction should also be measured as an important quality dimension.

First of all, we had to define the objective of our evaluation (hardware, software, and architecture or usability aspects) and then we had to determine the evaluation method including the specification of the different experiments to be carried out and the metrics we should use in those experiments.

CE4WEB was designed to allow software designers to work in groups; therefore our main interest was to know if the tool actually helped people to work together in the same project, modelling the same diagram. Thus, assessing the collaborative issues of the CE4WEB CASE tool was the main goal of our evaluation.

The next step is to find out how to evaluate groupware users' satisfaction. It must be noted that user's satisfaction and group's satisfaction cannot be obtained in the same way, although it is possible to adapt some traditional HCI evaluation methods to test groupware systems.

Taking all these considerations into account, we describe the evaluation method we defined in order to assess the main aspects of groupware systems, as is cooperation, coordination, information sharing and collaboration together with the spatial and temporal dimensions.

Groupware main features were assessed during the process of modelling a unique medium-sized UML diagram. This artefact was concretely the class diagram corresponding to a booking system for sport facilities and other associated materials (rackets, balls, etc). The diagram has nine classes, nine relationships and a set of methods and attributes associated to each class. Users can introduce more classes and relationships if needed.

Regarding people involved in the experiment, we decided to test the CE4WEB tool using groups of 2, 3, 4, and 5 people. Testing groupware tools take an extra time

because users have to learn previously the different tasks and the very tool as well. In order to reproduce a realistic scenario, we defined different roles on each experiment. Two roles were defined: Depicting classes, and depicting relationships.

Other tasks, as introducing methods, attributes and comments were left out of the roles and could be completed by whoever remaining idle, once completed the classes and the relationships.

After describing important aspects of the evaluation, it is time to present the evaluation method, which covers the following phases. Warming-up phase: Before beginning the experiment, each user learns the basic functions of the system in order to avoid measuring the user's experience instead of the degree of collaboration supported by the tool. In this phase users employed main functionalities of the CE4WEB tool. These functions were the same as the ones to be used during the experiment. Lab 1: Synchronous modeling of the UML diagram by a group of two people. Lab 2: Synchronous modeling of the UML diagram by a group of three people. Lab 3: Synchronous modeling of the UML diagram by a group of four people. Lab 4: Synchronous modeling of the UML diagram by a group of five people.

As it can be noted, all the experiments were carried out at the same time, which is known as synchronous cooperation. Regarding the spatial dimension, each user was in his/her office and all the experiments were performed on the Internet. It is important to remark this fact, because most of the known collaborative CASE tools only work in an intranet because of the chosen technology.

One important consideration we took into account when performing the experiment was the fact that users could only communicate by means of the functionality provided by the tool. This requirement was important in order to evaluate the quality of the groupware features supported by the tool itself without any external help.

Asynchronous experiment was not considered in this occasion, because conditions of synchronous collaboration are much harder and interesting than the asynchronous ones in order to assess the CE4WEB tool.

Next section presents the metrics employed and the outcomes of the experiment.

6 Presentation of Metrics and Results

The evaluation process started with the Warm-up phase. Each user was guided by a personal instructor who taught him/her the basic functionality of the CE4WEB tool. A correct instruction of users takes over 30 minutes and also some extra time for users to learn by themselves. To perform the different experiments, users received a list of 15 tasks that should be completed in 10 minutes.

Before presenting the results of the experiment, we will talk about the selected metrics to measure the quality degree of the cooperation in CE4WEB.

Users involved in the different experiments should write down the next values: Number of tasks completed in the defined period; Number and description of errors introduced, if any; and a list of suggestions and improvements and Fill in the satisfaction questionnaire (10 questions).

All users completed all tasks in the predefined time. Additionally, the more users were involved in the experiment the faster tasks were performed. This is an interesting aspect when it is needed to point out some benefits of groupware systems. In this

case, we are talking about obtaining a higher throughput when the number of users that are working together is higher.

Some errors were detected during the execution of the experiments. The most important one is related to the need of using a communication mechanism from the very beginning, just when users log into the system. The CE4WEB tool offers a Chat tool but it is only available once users have entered to the drawing area to start the creation of the model.

The satisfaction questionnaire was defined using the recommendations of the ISO/IEC 9126-4, part 4 about Quality in Use [10]. The formula to measure users' satisfaction can be shown here:

$$S_q = \sum_{l=1}^L \frac{l}{L} \cdot \frac{x_l}{N} \tag{1}$$

Where S_q is the satisfaction level of question q , L is the amount of possible answers (in this case 5), x_l is the amount of people that answered l in question q , and N is amount of people that filled in the questionnaire.

The questionnaire was composed by ten questions evaluating whether the system is capable of performing the task, whether the system works properly, whether the tasks have been performed on time, there have been problems to perform tasks, the problems found could be solved, the cooperative task has been a richer experience than if it would have been done in an individual way, the communication among the members of the group was correct, the tool provides the sufficient cooperative functions needed to perform tasks in a cooperative way, the tool allows information exchange among the different users and whether the tasks have been performed synchronously.

The users' satisfaction degree is expressed in 5 levels. The satisfaction degree mapping is the following: 1-absolutely disagree, 2-disagree, 3-indifferent, 4-agree, 5-absolutely agree.

This questionnaire was filled in by 3 university professors and 3 PhD students related to the collaborative research field. The results obtained applying this expression to the questionnaire that the users filled in can be seen in Fig. 4.

Fig. 4. Users' satisfaction results (values for each question)

Only in question number 6 users were completely agree with the system. This question was “The collaborative task has been a richer experience than if it would have been done in an individual way”. On the other hand, we can note the very low value in question 7. This question was “The communication among the members of the group was correct” and this was due to the lack of suitable tools inside the tool to enable communication from the very beginning. This lack of communication systems before starting the drawing process made users to use additional communication tools to bypass some complicated tasks. A complementary view of the former results can be found in Table 1. In this case we can see the results of users’ satisfaction according to each participant.

Table 1. Users’ satisfaction by participant

Participant	1	2	3	4	5	Mean	Std. Dev.	Max	Min
Satisfaction	3.8	3.9	3.9	3.9	3.7	3.84	0.08	3.9	3.7

Analyzing Table 1 we can conclude that users are satisfied enough with the collaboration quality offered by the CE4WEB tool as the values obtained are near 4 that means “Agree”. As it can be noted, the standard deviation is very low, what shows the coincident opinion among the different users.

7 Conclusions and Future Work

The wide use of broadband communications technology propitiates the growing of collaborative systems. This is particularly useful in the field of Software Engineering where developers and software companies need modern collaborative tools in order to face complex projects in a globalized world. Many experts have noted that most of the problems appearing in software projects are due to the lack of communication, collaboration and coordination among the members of the team.

This paper presents a cooperative editor to support the process of modelling UML diagrams called CE4WEB. This CASE tool supports the main CSCW features: coordination, communication and information sharing. This tool has been developed as a SOA application in order to cover the real needs demanding by development teams when they are physically distributed in different countries.

The tool has been evaluated using a special evaluation method defined to measure the quality of the collaboration, communication, coordination and information sharing provided by the tool. The evaluation outcomes demonstrate the validity of the implemented tool and point out the need of this kind of collaborative tools.

The paper shows also the effort put in the definition of the evaluation method. However it is possible to make a recurrent question about the validity of the method applied and additional tools should be introduced to prove that formally. This is part of the future work.

References

1. Altmann, J., Weinreich, R.: An Environment for Cooperative Software Development. Realization and Implications. In: *Thirty-First Annual Hawaii International Conference on System Sciences*, vol. 1, pp. 27–37 (1998)
2. Amin Farshchian, B.: A Framework for Supporting Shared Interaction in Distributed Product Development Projects. Department of Computer and Information Science. Norwegian University of Science and Technology (Norway) (2001), Online proceedings <http://csgsc.idi.ntnu.no/2001/?page=proceedings>
3. Borges, M., Araújo, R.M., Dias, M.: A Framework for the Classification of Computer Supported Collaborative Design Approaches. In: *III CYTED-RITOS International Workshop in Groupware – CRIWG 1997*, El Escorial (España), pp 91–100 (1997)
4. Borges, M.: Designing collaboration through a web-based groupware infrastructure. *International Journal of Computer Applications in Technology* 19, 175–183 (2004)
5. Cook, C., Churcher, N.: An Extensible Framework for Collaborative Software Engineering. Software Visualization Group, Department of Computer Science, University of Canterbury (Nueva Zelanda) (2003)
6. Goguen, J.A., Lin, K., Gea, M., Cañas, J.: Web-based Support for Cooperative Software Engineering. In: *Proceedings International Symposium on Multimedia Software Engineering*, Taipei (Taiwan), pp. 167–191(25) (2000)
7. Grudin, J.: CSCW: History and Focus. University of California. *IEEE Computer*, 27, 5, 19–26 (1994)
8. Herbsleb, J., Mockus, A., Finholt, T., Grinter, R.: An Empirical Study of Global Software Development: Distance and Speed. In: *Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001*, pp 81–90 (2001)
9. Hurwitz Report: Collaborative UML Development. CanyonBlue Incorporated (2001)
10. ISO/IEC TR 9126-4:2004. Software engineering.Product quality.Part 4: Quality in use metrics. JTC 1/SC 7; ISO Standards. ICS: 35.080. Stage date: 2004-03-17
11. Meire, A.: Suporte à Edição Cooperativa de Diagramas utilizando Versões. Universidade Federal do Rio de Janeiro (2003)
12. Object Management Group. UML Superstructure Specification, v2.0 (2005)
13. Saeki, M.: Communication, Collaboration and Cooperation in Software Development - How Should We Support Group Work in Software Development? In: *Proceedings of the 2001 Asia Pacific Software Engineering Conference, APSEC 2001*, pp 12–20 (2001)