Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1999 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1999

Jade: Learning the Problem Domain through Collaborative Modeling

Christian Rauh
ITA - Technological Institute of Aeronautics, Brazil

Clovis Fernandes
ITA - Technological Institute of Aeronautics, Brazil

Jade: Learning the Problem Domain through Collaborative Modeling

Christian Rauh and Clovis Torres Fernandes, ITA, Brazil, clovis@comp.ita.cta.br

Abstract

Software development depends on the ability of analysts to obtain knowledge from domain experts and learn the problem domain. Traditional methods for eliciting information do not fulfill the needs of analysts. In this paper, we describe Jade, a collaborative application to assist on the development of object oriented software. Adopting Jade, analysts may shift the task of modeling to domain experts. Hence, they can focus on learning the domain as the domain experts create and test the model.

Introduction

Software development is a collaborative task where a group of analysts learn the problem domain and create a logical model to solve it. This learning process is impaired by the fact that the domain information is not explicit. On the contrary, it is implicitly stored in the rules, habits, and minds of the people who deal with this domain on a daily basis, the domain experts.

To overcome the communication barrier between analysts and domain experts, software methodologies are bringing the latter's active participation into the development process. This union of analysts and experts is necessary, but not sufficient, for a good understanding of the problem. Many problems arise when domain experts participate on the development process. We propose a collaborative system that removes some of the drawbacks of traditional methods for eliciting domain experts' knowledge, and allows analysts to concentrate on the task of learning the problem domain instead of managing the development process.

We start by stating the problem in the context where it arises and the traditional methods for solving it. Then an overview of the Jade application and the four steps taken to create it are presented, followed by a section describing each step. Finally, we draw our conclusions.

Context

This work is based on the Exploratory Phase with Scenarios (EPS) of the CRC/WB+ software development methodology (Fernandes 1996) being developed at ITA - Technological Institute of Aeronautics, Brasil.

In EPS the analysts start with a set of responsibilities that the system must fulfill and a set of classes that should execute those responsibilities. The responsibilities are assigned to the classes based on domain knowledge so that the classes form a coherent model of the system.

The difficulty of performing the tasks of EPS resides in the disparity between the owners of domain knowledge and the people creating the model. That is, the domain experts have the domain knowledge but the analysts are the ones responsible for creating the model. To create the model, the analysts perform a double role: managing the process of obtaining information and learning the problem domain. Analysts must learn the information as they obtain it and vice-versa. This method of learning while developing is inefficient, usually ending in poor models with little aggregated knowledge.

A traditional, and largely employed, method of obtaining knowledge from the domain experts is through a series of individual interviews, conducted by the analysts, with each of the domain experts. After finishing the interviews, the analyst creates the model based on the information he has obtained, i.e. the analyst himself assigns the responsibilities to the classes. Since, the analysts must learn as they gather knowledge and vice-versa the interviews are far from exhaustive. Much of the domain knowledge is not obtained and the analyst resolves conflicts afterwards with incomplete information.

To bypass the problems inherent to the individual interviews a recent trend is to use Joint Application Design (JAD) sessions (Martin 1993). The CRC/WB+ methodology adheres to this practice. JAD sessions have two objectives: take the information gathering responsibility from the analysts and obtain information conflicts from the domain experts. Adding a facilitator and putting the modeling task in the hands of the domain experts accomplishes the first. Concentrating all domain experts in the same place at the same time, i.e. scheduling daylong group meetings, does the second.

EPS, in particular, uses a JAD session to create a CRC card model of the system that is being analyzed (Wilkinson 1995). Briefly, cards representing classes are distributed among the domain experts. The facilitator goes through each responsibility on the set, asking the experts to create a description and assign it to a class. The owner of the card representing that class writes the responsibility down. In this way the domain experts model the system. Eliciting and resolving conflicts of opinion is transparent since descriptions and assignments must be unanimous. The analysts participate as simple spectators, simply absorbing the information put forth by the experts and registered on the model that is being created.

It is easy too see the advantages and drawbacks of the JAD session. On one hand, the information gathered is of much higher quality then that from the interviews, since all conflicts are elicited in real-time by the domain experts who, contrary to the analysts, are knowledgeable on the domain. This quality enhancement is proved by the fact that the domain experts already validate the CRC model obtained while creating it (Ambler 1995).

On the other hand, there is the practical problem of arranging large group meetings, especially of experts on the same domain. Few people can spend whole afternoons modeling a system, specially when the domain experts view this activity as “analyst’s job.” Even if a meeting happens, there is a whole myriad of human factors that hinder the development process, such as inter-departmental feuds, personal feelings towards others, shyness, hierarchical power, pulling rank, etc.

We propose a system for a computer-mediated execution of ESP. This system aims to provide the enhanced quality of information gathered through a JAD session combined with the flexible schedule and comfortable isolation provided by individual interviews. Following this proposal, we reach to provide analysts with a tool that they can use to concentrate on learning the domain. This will be done through a rich collaborative environment where knowledge can be obtained through the model being created, through direct contact with the domain experts, but especially through close following of the collaborative session.

Jade

Our system was named Jade, after “electronic JAD”. Four steps were considered to create the system:

- Modification of the JAD/CRC session flow
- Development of a framework for a distributed infrastructure that would allow for separable and adaptable interfaces.
- Development of generic collaborative awareness widgets
- Development of a specific collaborative application for the assignment of responsibilities to classes

Each of these steps will be described in more detail in the next sections.

Modification of the Session

The JAD/CRC session used in the CRC/WB+ methodology is based on the flow graph presented in (Ambler 1995). Ambler’s graph is strictly linear and suitable for the normal JAD style sessions. This linearity, although, seems to restrict the flow of the session since participants work on one responsibility at a time.

To provide the asynchronicity that exists in the individual interviews, the graph was modified to allow for a parallel flow. That is, each participant follows the graph at his own pace and will, regardless of what others are doing. This parallel behavior would certainly disrupt a face-to-face meeting. Yet, with computer-mediated meetings, the system is responsible for managing each user’s progress, ask them to communicate, and synchronize as needed.

Allowing participants to work at their pace creates a better environment for knowledge acquisition. Participants feel more comfortable when able to think as much as needed. They do not feel oppressed by time or are influenced by the opinions of others. As a result, more knowledge about the domain is expressed thus enhancing the model’s content.

Framework

Multiple user profiles is a problem that arises in all computer applications. This problem is complicated in Jade since there is no predictable common aspect to its users. Domain experts can come with any background and have varied experience with computers. While one can assume that analysts will be modestly computer literate, no assumption can be made of the domain experts. Presenting an advanced interface to a domain expert with a great knowledge but little computer experience may shy him away, while a domain expert with little knowledge may be able to compensate through the use of an advanced interface to express a good extent of all his knowledge. Also, because Jade is used rarely by each user, there is no time for extended interface learning as the application must make the most out of each use at every moment. Jade therefore must be able to tailor itself to each user’s needs at real-time.

To create a system that can tailor its interface to the user’s needs, a framework was developed based on the Persona Interface Model (Ortega 1998). It was aptly named Persona Framework (PF).

The central part of PF is the Persona (Reynolds 1997). Persona is an object that stores the user preferences as his state and a tailoring agent as his behavior. Each user on PF has his individual Persona object that persists through same and different application uses. Besides the Persona, the framework has two basic types of objects: interface objects and application objects. Interface objects are responsible for the presentation of data to the user. Application objects are responsible for holding and manipulating data and for communicating with the persona about the tailoring options.

When a PF-based application is started, it communicates with the persona to find out what is the best interface for that user. During execution, the Persona tailoring agent constantly updates the interface based on the use of the application by the user.

Awareness Widgets

We built a set of generic collaborative awareness widgets that enhance the workspace awareness of the participants (Gutwin 1996). The widgets are interface objects that can be used by applications based on the persona framework. The widgets created include the following:

- Workmeter – This widget shows how much each participant is interacting with the system.
- Miniature Telepointer View - This widget presents a reduced view of the workspace and the position of each user. It is intended to provide information on who is working on what object and how.
- Multi-User Radar – This widget shows the position of each user window giving the user the notion of what the others are looking at.

Application

The final step is the actual development of the specific application that executes the modified JAD/CRC session. The application is built upon the Persona Framework and makes use of the generic awareness widgets.

The assigning of responsibilities to classes is done using a shared drawing board metaphor. In a main viewport users are presented with icons representing responsibilities and classes. We chose to use the drawing board metaphor because humans tend to work better with concrete representations than with text and abstract ideas (Greenberg 1995).

Another reason for this option is that analysts can visually follow the assignment of responsibilities to classes by the domain experts. This fact is important to the learning process involved in the development of the model. By watching the domain experts collaborating to create a final model, the analysts can see how the model develops from simple to complex structures. That information reflects directly on the perceived understanding of how each part of the model is related to the other.

The process of assignment with Jade is basically done by the domain experts connecting to the system and receiving a set of tailored interfaces that are most appropriate to his skill. In the drawing board, the domain expert is shown icons of the classes and responsibilities. The domain expert can then click on these representations to connect them. If a domain expert creates a connection that conflicts with an existing one, he has to dialogue with the original connection creator to resolve this conflict.

Conclusions

Jade offers the advantages of both individual interviews and JAD sessions. In addition, many of the problems present in those methods have been overcome, either by advantages of each other or through software enhancements like the interface tailoring and awareness widgets. The adoption of Jade permits the creation of a CRC card model with a high level of confidence and with superior knowledge content than those created through traditional methods. This model, as well as its creation with Jade, gives analysts a better understanding of the domain and, most important, Jade allows analysts to concentrate on learning the domain instead of the information gathering process.

Jade is in development as a prototype. The major obstacles to be overcome are network problems, and multi-platform interface design issues.

While Jade appears to fulfill the basic needs for a successful carrying out of the EPS, further testing is needed to ascertain its the true usability in real conditions.

References

Ambler, S. “Using Use-Cases,” *Software Development* (3:7), July 1995.

Fernandes, C. T. “Object Oriented Design: the CRC/WB+ Methodology,” *Proc. of the II Workshop on Object Orientation*, COPPE/UFRJ, Rio de Janeiro, RJ, December 1996. [in portuguese]

Greenberg, S., Hayne, S. and Rada, R. *Groupware for Real-Time Drawing: A Designer’s Guide*, McGraw-Hill, London, 1995.

Gutwin, C., Roseman, M. and Greenberg, S. “A Usability Study of Awareness Widgets in a Shared Workspace Groupware System,” *Proc. of CSCW’96*, Boston, MA, November 1996.

Martin, J. *Principles of Object-Oriented Analysis and Design*, Prentice Hall, Englewood Cliffs, NJ, 1993.

Ortega, C. and Fernandes, C. T. “A Separable and Adaptable User Interface Model for CSCW Environments,” *Proc. of WORKCOMP ’98 – Workshop on Computer Sciences*, ITA, São José dos Campos, SP. October 1998.

Reynolds, C. “A Critical Examination of Separable User Interface Management Systems: Constructs for Individualization,” *SIGCHI Bulletin* (29:3), July 1997.

Wilkinson, N. M. *Using CRC Cards: an Informal Approach to Object-Oriented Development*, SIGS Books, New York, NY, 1995.