AUTO-COLLEAGUE
A Collaborative Learning Environment for UML

Maria Virvou and Kalliopi Tourtoglou
Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou St., Piraeus, Greece

Keywords: Collaboration, learning, collaborative learning, CSCL, motivation, motivation management, collaboration management, group, team, role, stereotype, student modelling, user modelling, software engineering, UML.

Abstract: In this paper we present AUTO-COLLEAGUE, a collaborative learning environment for UML. AUTO-COLLEAGUE is a Computer-Supported Collaborative Learning (CSCL) system. It is based on a multi-dimensional User-Modeller component that describes user characteristics related to the UML domain knowledge, the performance types, the personality and the needs of the learner. The system constantly monitors, records and reasons about each learner's actions. As a result of this process, AUTO-COLLEAGUE provides adaptive advice and help to users so that they may use UML more efficiently and collaborate with other members of their team more constructively.

1 INTRODUCTION

Collaborative learning is a situation in which two or more people learn or attempt to learn something together (Dillenbourg, 1999). There are many theories to support the advantages of adapting collaborative learning. As such, collaborative learning has attracted a lot of researcher energy lately and has resulted in many computer-supported collaborative learning environments.

Collaborative learning has found its most effective application in software systems, as these tools can be appropriately implemented. Software systems that include mechanisms of collaborative learning are widely known as Computer-Supported Collaborative Learning (CSCL) systems. CSCL systems have been widely used over the last decade with the aim to encourage learners to collaborate simulating the traditional class courses. This simulation is oriented in two kinds of interaction: the peer-to-peer and the teacher-to-learner interactions.

In view of the above, we have developed AUTO-COLLEAGUE (AUTOmated COLLaborativE leArning Uml Environment). It is a UML training environment for software engineers. The Unified Modelling Language (UML) is an object-oriented visual modelling language that is used to specify, visualize, construct, and document the artifacts of a software system (Rumbaugh, 1999). The use of UML has grown enormously in many organizations that develop software during the last decade. UML is, also, very popular amongst education institutes, that use it as a tool for software engineering training. Being widely used in industry by now, proficiency in UML is certainly a valuable asset for every Computer Science student (Engels et al., 2006). For the purposes of AUTO-COLLEAGUE, collaborative learning seems appropriate for training users in UML. As such, AUTO-COLLEAGUE provides a multi-user, synchronous and remoted environment for software engineers to collaborate during the learning process.

The design and the architecture of AUTO-COLLEAGUE are based on the principles of the constructivism learning theory (Piaget, 1973). Constructivism is based on the principle that human learning is constructed. One of the main perspectives of this theory is that the learners should be autonomous and independent during the learning experience. The sense of autonomy reveals other aspects of the human personality, such as the motivation. Motivation is defined as the forces within an individual that account for the level, direction and persistence of effort expended at work (Schermerhorn et al., 2000). In the implementation of AUTO-COLLEAGUE, we have taken into account the impact of motivation of learners. Thereby, we have chosen to apply principles of a widely used motivation theory: the Maslow's Hierarchy of Needs (Maslow, 1954).

AUTO-COLLEAGUE bases its reasoning on a User Modelling component implemented with stereotypes. Learner modeling is the key concept

Virvou M. and Tourtoglou K. (2007).
AUTO-COLLEAGUE - A Collaborative Learning Environment for UML.
In *Proceedings of the Second International Conference on Software and Data Technologies - SE*, pages 35-39

concerning the application of artificial intelligence into educational software (Dillenbourg & Self, 1992). The stereotypes used in AUTO-COLLEAGUE encompass the personality, the performance type, the knowledge and the motivation sources of the learners. Stereotypes constitute a widely used technique of building user models. Stereotypes are simply collections of facet-value combinations that describe groups of system-users (Rich, 1979). Stereotypes have else been widely and efficiently applied to educational software (Beaumont, 1994). The User Modelling component will be explained in detail in a subsequent chapter.

2 RELATED WORK

AUTO-COLLEAGUE is a CASE tool, but its main objective is to provide a collaborative environment, to model users and to offer intelligent advice based on stereotype-based reasoning. As such, it is not our aim to compete with well-established and efficient CASE tools for UML, such as Rational Rose.

There have already been developed quite many efficient CSCL systems to support distant users to work together (some of them into groups) and to provide them with an assistant on their learning process. However, after reviewing them, we have concluded that AUTO-COLLEAGUE introduces new aspects especially in the architecture and the variety of user characteristics it evaluates. Most of the CSCL systems base their inferences mainly on the participation of the users in a chat system and the outcomes of this participation. In particular, COLER (González et al, 2000), LECS (Rosatelli & Self, 2004), COLLECT-UML (Baghaei & Mitrovic, 2005), DEGREE (Barros & Verdejo, 2000), HABIPRO (Vizcaíno at al., 2000), FLE3 (Chen et al., 2003) and Learning Environment (Or-Bach & Van Joolingen, 2004) do not focus on including in their inference mechanisms important learner characteristics, such as personality. In contrast, they focus on qualitative evaluations of learners' participation in the system and the effects that this participation had afterwards in their progress. In contrast, AUTO-COLLEAGUE is is based on a User Modeller, which models the learner in several aspects. There have been other CSCL systems that perform user/learner modelling, such as GRACILE (Ayala & Yano, 1998) and HELP-DESK (Greer et al., 1998). However, these systems too have not taken into consideration human characteristics as AUTO-COLLEAGUE. In specific, they successfully use social user traits that are related mainly to the collaboration. On the other hand, with the aim to have an integrated approach to the human/learner model, AUTO-COLLEAGUE uses characteristics related to the performance type and the personality.

Another detected difference is the way that the system estimates what has been done wrong. For example, COLER (González et al, 2000), LECS (Rosatelli & Self, 2004), COLLECT-UML (Baghaei & Mitrovic, 2005) and HABIPRO (Vizcaíno at al., 2000) evaluate the user performance (on the domain knowledge and skills) based on submitted solutions to specific problems. On the contrary, AUTO-COLLEAGUE continuously evaluates the efficiency of the learners. That is achieved by tracking every action of the users even idleness.

We also remarked the absence of the precious contribution of a human trainer (COLER, COLLECT-UML, GRACILE, HABIPRO and Learning Environment). AUTO-COLLEAGUE provides the potential to a person to be assigned to the system as a trainer. This gives him/her a special role in the system. The trainer is responsible for defining crucial attributes/parameters of the learners, evaluating, applying, and contributing to the advice and conclusions of AUTO-COLLEAGUE.

3 OVERVIEW OF THE SYSTEM

At the front end, AUTO-COLLEAGUE is an environment in which software engineers can create UML diagrams. Additionally, software engineers can provide UML diagrams as input to the process of generation of an XML file (descriptive of the UML diagram), a database structure and a source code project written in Borland Delphi. A screen-shot of AUTO-COLLEAGUE is illustrated in Figure 1. The main form contains two toolbars, a status bar, the forum section, the UML workspace and a horizontal and a vertical ruler perimetric the workspace. The two toolbars are the standard and the project toolbar. The standard toolbar includes the buttons for creating a UML diagram, opening an existing one, saving, printing and clearing the current UML diagram and an exit button. The project toolbar has three buttons, which respectively have the functionality to export an XML file, a Delphi application and a database structure. The status bar on the lowest part of the form is used by the system to show additional and explicative information to the user, such as the cursor position, the log data and help messages. In the forum section, the user can send messages to another user of the system after selecting the purpose of message (to make a question, to request help, to propose an answer to a stated question, to state his/her agreement/disagreement, to simply communicate, to greet). The UML workspace provides the necessary user interface to design UML diagrams. The user can add UML classes (properties, attributes, methods, method parameters etc), notes, inheritances and compositions.

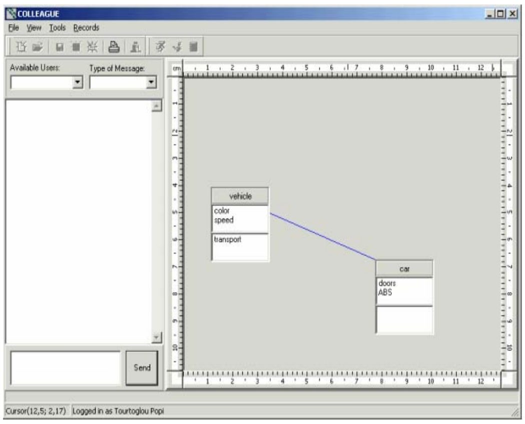

Figure 1: Screen-shot of AUTO-COLLEAGUE.

There are two human actors in the AUTO-COLLEAGUE environment: the learner and the trainer. The learners are the users who intend to learn UML and the trainer is the coordinator/coach of this process of learning. The trainer is the partner of the system as his/her responsibility is to define the roles, teams and groups of the learners and to evaluate and apply appropriately the information given by the Collaboration Module. The roles, teams and groups are explained later in detail. In brief, these concepts concern the structured organization of the learners in the system.

The aim of AUTO-COLLEAGUE is to provide a collaborative environment. To achieve this, it includes the Collaboration Module. The Collaboration Module constantly traces the attributes, the operations and the progress of the users in order to draw inferences on which may be the most effective collaboration patterns between the colleagues. The user attributes are maintained in a stereotype-based structure by the User Modeller. The User Modeller includes attributes that refer to the UML domain knowledge, the performance type, the personality and the motivation sources of the users. The User Modeller will be described in detail in a subsequent chapter.

AUTO-COLLEAGUE includes five main components: the User Modeller, the Collaboration Module, the Tracker, the Feedback Module, and the Advisor. We have already mentioned the User Modeller and the Collaboration Module. The Tracker is the module that stores every action, every message and every error of the user. The Feedback Module is responsible for the interactive messages of the system to the user. The answers to these interactive messages are stored and used by the User Modeller for further inferences. The Advisor is the module that informs the trainer and the learners about the conclusions of the Collaboration Module.

4 USER MODELLER

The User Modeller is of vital importance in AUTO-COLLEAGUE. It is the module that performs the essential reasoning for the users. This outcome often is passed to another component, the Advisor. The architecture of the User Modeller is based on stereotype reasoning.

The user features that we decided to record are the personality, the performance types, the levels of expertise and the motivation sources. The User Modeller is composed by four stereotypes: the personality, the motivation sources, the performance types and the levels of expertise.

Personality refers to some typical characteristics of the software engineers that influence their behavior in relation with the acquisition, the usage and the sharing of knowledge. The personality features used by the system are the self-confidence, the leading potential, the diligence and the willingness to help others, conformity, participative and communicative. The self-confidence feature is what the user believes about him/herself as compared with the reality that AUTO-COLLEAGUE has assumed. The leading potential refers to the ability of the user to manage a group. For example, a person who has leading potential should be able to be influential within the group and, at the same time, preserve a harmonious relationship between them. The diligence measures the amount of work that the user is willing to take. The willingness to help others is the eagerness of the software engineer to cooperate with others and help them even if they do not belong in the same team. Conformity expresses the ability of the user to work in accordance with the established group in which s/he belongs. For example, a conformist would not have the tendency to disagree (without having a substantial reason) with some particular colleagues. A user who is participative is the one who seems to have a share in the message forum. The same definition applies to the communicative characteristic. The difference between these two is that the "participative" refers to messages of knowledge content, in contrast with the "communicative" that refers to message of general content, such as greeting a peer.

Performance types are the main typical manners of software engineers when using a program. The defined performance types are: sceptical, hurried, slip-prone, unconcentrated and efficient. Sceptical refers to the user who frequently lets the program idle for a noticeable amount of time, in his/her effort to complete a task. Hurried is a user who completes tasks quite quickly but not successfully. A user may be characterized as slip-prone when although s/he has the essential UML knowledge to complete a task successfully, s/he makes mistakes due to carelessness or imprudence. Unconcentrated is a user who seems to be sceptical (delays to act), but actually cannot be productive in UML probably because of lack of ability to concentrate. Efficient is the user who seems to accomplish successfully a task at a reasonable speed.

A level of expertise represents the extent of the software engineer's knowledge on UML and ability to construct a correct UML diagram. The UML knowledge is stored in AUTO-COLLEAGUE database, which means that the trainer can update it. This reflects flexibility in the maintenance of such a crucial element. The UML knowledge is stored in a hierarchical structure of UML concepts. The UML concepts are the principles of constructing UML diagrams, such as Class Definition, Attributes Definition and Inheritance. The levels of expertise used by the system are No Knowledge, Basics, Junior, Senior and Expert. Each level of expertise is linked to some specific UML concepts. This link represents the prerequisites on UML knowledge for a software engineer of level of expertise classifications. Additionally, there is a degree of knowledge upon each UML concept. For example, the Basics Level of Expertise can be linked to the Attributes Definition with degree 1. The fact that the structure of UML concepts is hierarchical contributes to the inferences rules of AUTO-COLLEAGUE stereotype-based user modelling.

The motivation sources can be defined as the forces that lead the learners to strive for the possession of knowledge and the development of skills and talents in the framework of satisfying their needs. The stereotypes of User Modeller include these motivation sources, which are structured on the basis of the Maslow's Hierarchy of Needs (Maslow, 1954). The levels of needs that our system uses are the social, the esteem and the self-actualization needs. The social needs refer to the need of the learners to feel members of a group, to have a common target and a pleasant communication, to be accepted by colleagues and to develop friendship relationships. The esteem needs are linked to the satisfaction of the self-ego that is the need to be a distinguished personality/entity. In particular, the esteem needs are the needs of a learner to be recognized, respected, admired or even envied by others. The self-actualization needs concern the achievement of the maximum of everything one considers as admirable. The motivation sources are implemented in the User Modeller structure with two attributes: the need and the strength of need. The motivation sources are included in the stereotype structure as their values can influence largely the inferences of the overall system. For example, the satisfaction of the esteem needs can trigger the user trait of personality by defining the user as self-confident. On the other hand, the dissatisfaction of the social needs may influence the behavior of the user and prevent him/her from being participative.

For the Inference Mechanism and the Triggering/Retraction Conditions of the stereotype-based User Modeller, we need to define certain variables that we have used. These variables are descriptive of all of the aforementioned characteristics that constitute the user stereotypes. These variables are: user actions, participation frequency, help requests frequency, help offered to others frequency, error frequency, average idle time between actions, speed of completing a task, relevancy of actions.

5 SUPPLEMENTARY MODULES

AUTO-COLLEAGUE includes six supplementary modules that support the main functions of the system. These are the Group-Role Module, the Tracker, the Historian, the Feedback Module, the Communicator, the Advisor and the Collaboration Module.

The function of the Group-Role Module is to allow the trainer to categorize the users into groups based on certain roles. This categorization is necessary for AUTO-COLLEAGUE to draw inferences about the collaboration schemes between software engineers. The Tracker is responsible for tracking the sessions, the log data, the actions, the errors and the messages of the users. The Historian is the module that provides the users with a supplementary functionality: the user can playback the exact movements s/he did in the past. The Feedback Module is responsible for requesting the user opinion on the assumptions of the system. The Communicator includes the mechanism of the message forum. The message forum is a subsystem that offers a chat utility. It enables the users to send messages to each other and, simultaneously, observe them. The Advisor is the module that undertakes the task of notifying the trainer of the system about the conclusions of the Collaboration Module. This is done by means of window messages and report forms. The Collaboration Module is a vital module as it serves the aim of AUTO-COLLEAGUE to be a CSCL system that is to offer an efficient collaborative learning environment. In other words, the goal of the Collaboration Module is to make the necessary inferences to draw conclusions about the most effective collaboration teams between the learners. The effectiveness is relative to the performance of all the learners as individuals and as a whole.

6 CONCLUSIONS

The collaboration management is a very complicated task as the aspects of the human nature are too complicated. With the aim to contribute to the scientific effort to find effective ways of building CSCL systems, we designed and presented in this paper AUTO-COLLEAGUE. It is a collaborative learning environment for software engineers to be trained on UML. The most crucial modules of AUTO-COLLEAGUE are the User Modeller, the Group-Role Module and the Collaboration Module. The stereotype-based User Modeller includes important descriptive user characteristics that have effects on the learning process. These characteristics are the knowledge, the performance types, the personality and the motivation sources. The Group-Role Module involves the structural classification of the learners in the system. Using the data of the User Modeller and the Group-Role Module, the Collaboration Module draws inferences based on the stereotypes of the learners and proposes the most effective collaboration arrangements. One very important functionality of AUTO-COLLEAGUE is also the provision of assistance to the human trainer.

REFERENCES

Ayala, G., Yano, Y., 1998. A collaborative learning environment based on intelligent agents. *Expert Systems with Applications, Elsevier Science, 14*, 129-137.

Baghaei, N., Mitrovic, A., 2005. Collect-UML: Supporting Individual and Collaborative Learning of UML Class Diagrams in a Constrain-Based Intelligent Tutoring System. *Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 3684*, 458-464.

Barros, B., Felisa Verdejo, M., 2000. Analysing student interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education, 11.*

Beaumont, I.H., 1994. User Modelling in the Interactive Anatomy Tutoring System ANATOM-TUTOR. *User Modeling and User-Adapted Interaction, 4*, 21-45.

Chen, W., Dolonen, J., Wasson, B., 2003. Supporting Collaborative Knowledge Building with Intelligent Agents. *In Palade, V., Howlett R., and Jain, L. (Eds.): Knowledge-Based Intelligent Information and Engineering Systems, Advances in Artificial Intelligence, Lecture Notes in Computer Science, Springer, 2773*, 238-243.

Dillenbourg, P., 1999. What do you mean by "collaborative learning"? *In P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches. Oxford Elsevier,* 1-19.

Dillenbourg, P., Self, J., 1992. A framework for learner modelling. *Interactive Learning Environments, 2*, 111-137.

Engels, G., Hausmann, J.H., Lohmann, M., Sauer, S., 2006. Teaching UML Is Teaching Software Engineering Is Teaching Abstraction. *Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 3844*, 306-319.

Greer J., McCalla G., Cooke J., Collins J., Kumar V., Bishop A., Vassileva J., 1998. The Intelligent Helpdesk: Supporting Peer Help in a University Course. *In Goettl BP, Halff HM, Redfield CL, Shute VJ, editors. Intelligent Tutoring Systems, Berlin Heidelberg Springer-Verlag,* 494-503.

Maslow, 1954. *Motivation and personality*. New York: Harper & Row.

Or-Bach, R., Van Joolingen, W. R., 2004. Designing Adaptive Interventions for Online Collaborative Modeling. *Education and Information Technologies, Kluwer Academic Publishers, 9:4*, 355-375.

Piaget, J., 1973. *To understand is to invent.* New York: Grossman.

Rich, E., 1979. User Modeling via Stereotypes. *International Journal of Cognitive Science, 3*, 329-354.

Rosatelli, Marta C., Self, J., 2004. A Collaborative Case Study System For Distance Learning. *International Journal of Artificial Intelligence in Education, IOS Press, 14,* 1-29.

Rumbaugh, J., Jacobson, I., Booch, G., 1999. *The Unified Modelling Language Reference Manual*, Addison-Wesley.

Schermerhorn, John R., Jr., Hunt, J. G., Osborn, R. N., 2000. *Organizational Behavior*.

Vizcaíno, A., Contreras, J., Favela, J., Prieto, M., 2000. An Adaptive, Collaborative Environment to Develop Good Habits in Programming. *In ITS 2000, LNCS, Springer, 1839*, 262-271.