Collaborative tools in learning a programming language

Maria Virvou
Department of Informatics
University of Piraeus
Piraeus, Greece
Email: mvirvou@unipi.gr

Sotirios Christos Sidiropoulos
Department of Informatics
University of Piraeus
Piraeus, Greece
Email: csidir@unipi.gr

***Abstract*—It is generally acknowledged that students can benefit from the use of a collaborative learning environment typically resulting in higher achievement and greater productivity. However, there has not been sufficient progress in personalizing e-learning systems, especially in the case of collaborative systems. In this paper we describe a novel e-learning system for the teaching of programming languages through the use of collaborative tools. The collaborative tools support student interaction and group formation based on each student model. In conjunction with collaboration our system offers error diagnosis and user modeling based on individual models. Our research has been applied on an e-learning system for the Python programming language.**

I. Introduction

Computer Supported Collaborative Learning (CSCL) aims at supporting groups of learners in acquiring content knowledge in a specific domain by means of computers. Computer-supported learning creates new possibilities for learning among students, a fact that may benefit the language tutoring use [5]. CSCL environments promote collaboration among students and enhance their skills triggered either by learning within a group or by learning in a technology-rich environment [3].

Learning a programming language has always been a tedious task, especially for beginners with no programming or algorithmic background. Traditional ways of learning include reading of theory and then completion of assignments that the teacher has defined. In these process though, students may have some trouble understanding parts of theory or accomplishing a task on their own. Through collaborative learning students can work together towards a common goal, in this way students are learning together and can increase their own and their peers awareness on a subject [4], [7] . Students are able to join actively in knowledge construction by assisting creation, exchange, and analysis of information during learning group interactions. Some particular benefits of collaborative problem-solving include [1]:

- encouraging students to verbalize their thinking
- encouraging students to work together
- ask questions
- explain and justify their opinions
- increasing students responsibility for their own learning and increasing the possibility of students solving or examining problems in a variety of ways.

Collaborative learning has an obvious potential to improve critical thinking, creative thinking, elaborative thinking, social communication, and social skills like leadership, decision-making, trust- building, conflict-management, etc. [2]. In recent years, increasing research effort has been put into the development of personalized systems and has influenced the area of educational software [9]. Intelligent Tutoring Systems (ITSs) are computer programs that aim at providing cost effective one-on-one tutoring . They provide personalized instruction to students, because they are designed to know exactly the person being taught, the curriculum taught and the way of tutoring. ITSs offer sophistication, interactivity and adaptivity to students. Sophistication and interactivity render the students capable of reciprocating to the difficulties of the educational process. Moreover, adaptivity is a very crucial matter in educational systems that aim at reaching a much more heterogeneous group of learners in settings where no teacher is available to help users during their learning process [9]. To a large extent in ITSs, intelligence and adaptivity are achieved by the incorporation of a student modeling component. Student modeling attempts to model students skills and declarative knowledge and can adapt instruction to his/her individual needs . The classical architecture of an intelligent tutoring system is composed of four elements that are closely related. Namely, there are the Domain model (Domain Knowledge), the Student Model, the Teaching Model (Tutoring Component) and the User Interface [7]. The domain model consists of a representation of the domain to be taught. The student model stores some information about the students performance or behaviors, including his/her knowledge level, his/her preferences, his/her learning history and helps the ITS to personalize the teaching strategy. The tutoring component contains a representation of the teaching strategies of the system and provides adaptive instructions to the students. Finally, the user interface provides the means for the student to interact with the ITS, usually through a graphical environment and sometimes through a rich simulation of the task domain the student is learning . Our system aims to make the learning curve easier with the help of social interactions. Although there have been attempts of making a platform to support social coding and development (www.github.com) there hasnt been any similar action towards a learning environment.

978-1-4673-1678-1/12/$31.00 ©2012 IEEE

II. Related Work

Reference [1] has introduced a constraint-based ITS, that supports both problem solving and collaborative learning, for the learning of object-oriented design using Unified Modeling Languages. Reference [2] has introduced a Collaborative Learning Tool Applying to C Programming Language which consists of a server / client tool which embedded voice and text chat communication to support collaborative learning via internet. Learner can study from group's collaborative learning, find and solve the problem of C programming language designing through communicating and discussing. Reference [8] has introduced a mobile computer supported collaborative learning environment with Handhelds interconnected by a wireless network. The description of two face-to-face collaborative learning activities for children without technological support, and their evaluation results between the design of two face-to-face MCSCL activities, using Handhelds wirelessly interconnected. Real Time Collaborative Programming allows geographically distributed students to work concurrently and collaboratively in the same programming task in order to design, code, debug, test and document [6]. COLLEGE (COLLaborative Edition, compilinG and Execution of programs) facilitates the collaborative learning of Programming. The collaboration between students, besides offering cognitive benefits, is a motivating aspect since the students use tools that are familiar to them, such as the chat or the electronic mail [2].

III. General Architecture

In the literature of Intelligent Tutoring Systems (ITSs) there is a classical architecture of systems which is composed of four elements that are related: the Domain Model, the Student Model, the Teaching Model and the User Interface. In our system we follow this architecture as described below. The Domain Model represents the theory for Python programming language, the Student Model models the user based on his/her knowledge and skills upon the theory being taught, the Teaching Model consists of the rules and methods that are used for displaying the appropriate content upon each user and finally the User Interface consists of the presentation layer for direct interaction with the user.

Our system is an online platform that can be accessed anywhere at anytime. Students can access the website and get personalized help from the system and from other expert users. The system also supports synchronous and asynchronous online discussion in order to exchange ideas on the curriculum being taught.

Our system promotes the teaching of the Python programming language. The learning pattern includes two branches. The first one is teaching mostly driven by the professor while the second branch offers the learner a more relaxed approach on learning based on his personal evaluation and interaction with other users in the system. The choice of what branch a student should follow is made by the teacher. The teacher has the possibility of authoring significant parts of the system such as the theory and exercises and sets the initial parameters for the system. When the initial set up is conducted, based on the educational approach the teacher wants to induce, students can follow the traditional course, where there is a structured follow-up of courses, or follow the more relaxed structure where, based on auto evaluation and social interaction, students can skip classes and eventually minimize the learning curve.

A. Student Model

Our system holds a database concerning the overall and specific knowledge of the student. We use the PostgreSQL database system to hold the student data. The database holds these items:

1) Overall Performance, it's the overall grade of the user.
2) Grade per Lesson, the grade the user holds in each lesson completed.
3) Social Interaction Level, it is computed by the user activity using the collaborative tools. This level is used to evaluate the system and compare the progress of students that make use of the collaborative tools with those who don't.

Initial data are given through a preliminary evaluation of each student upon entering the system. The student is called to fill a preliminary test with a fixed set of questions that help the system identify the level of knowledge the student holds. According to the initial data and the data received through the educational process we are able to construct a user specific student model. Our system holds three distinct categories of user categorization according to his/her cognitive level. These are:

1) Beginner, the user has no knowledge of programming.
2) Average, the user has some basic knowledge of programming and algorithmic problem solving.
3) Advanced, the user has enough knowledge of programming and is able to achieve complex programming tasks.

Depending on the progress of the student the system can change the initial user model. Based on these three categories and individualized data of each user the system can give specific suggestions upon the start and the end of a lesson section. In view of the above, the system adapts to each user and dynamically changes its content.

Another crucial task that is performed by the system is building of a history model of the students weakness and progress. The student model influences the operation of error diagnosis. Students may benefit from viewing their own student models, as they can review their performance each time they want in the educational procedure. Therefore, this kind of information is used for refining the error diagnosis component and for being presented to the users.

B. Error Diagnosis

The system holds three of lessons, those are:

- *Theoretical Lessons*: The theory is presented to the student. Goal is to understand the background theory for creating a program mainly algorithmic. Evaluation is done mainly with simple multiple choices when the student feels that is ready to take the test.

- *Algorithmic Lessons*: The student is presented with the theory but this time the evaluation is done by writing a simple program in pseudo-code.
- *Code Lessons*: In this type of lesson the student has to create his own program based on the instructions or fill a missing part of the code.

Depending on the type of the lesson error diagnosis is conducted using two different methods. The first method, which is used in theoretical lessons, includes a check on a predefined error database. The database is created by the teacher upon the creation of each exercise. The second method includes a syntax/error highlighter which highlights common syntax errors in Python language

Error diagnosis is conducted using two different methods. The first method includes a check on a predefined error database. The database is created by the teacher upon the creation of each exercise. The second method is based on an error diagnosis algorithm which can identify common errors on mathematical operations. The algorithm can identify misinterpreted operations and suggest the right answer and the appropriate theory. Which method is used depends on what the teacher has pre-selected.

C. Collaborative Tools

Our system supports several collaborative tools that help students cooperate on learning and problem solving. As shown in Figure 1 the student can collaborate with other peers in problem solving.

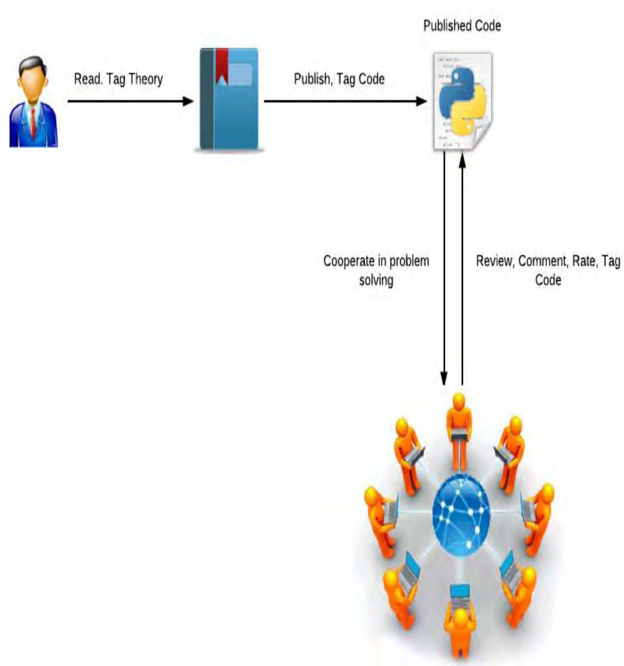

Fig. 1. Collaborative Process

- Publishing Code: When a student finishes an evaluation exam he can publish his code so that his/her classmates can observe it and comment on it. In this way students can share code and ideas for more efficient coding. Also students are able to rate a solution so that the best solution can be illustrated as the most effective and recommended.
- Social Tagging: Users have the option to tag lessons, exercises and solutions. This method has a positive impact on both the student and the teacher. The students can use these tags to find similar theory or exercises and the teacher can use the tags to understand what the students think about a certain lesson. For example a lesson that the teacher has assigned to novice users and is being tagged as very difficult by the students apparently means that it needs to be reconsidered or assigned to users that are more advanced.
- Group Formation: There are three methods of creating groups. The first one is the most common and widely used where the teacher assigns a work group. The second one is also common among students where they have the ability to create their own group mainly based on friendship. The third one, which the system uses by default, is conducted automatically by the system based on the data that have been collected about the student. The system collects data about the students' cognitive level, social activity and assigns them to groups using a clustering approach.

IV. Conclusion

In this paper we have described a computer supported collaborative learning system for tutoring the Python programming language. Our system incorporates a student modeling component which holds information about each student and promotes the individualized learning. Furthermore collaboration is succeded by code publishing and reviewing among students, tags demonstrating the difficulty of a lesson or exercise and group formation among students.

Through collaboration students can benefit from their peers and achieve better understanding of the learning concepts. Another crucial benefit is that the students may improve their knowledge level by interacting with others in a competitive way.

The system is currently in alpha stage and is being beta tested by a closed group of students. Based on the fact that we use an online platform it is easier for the students to reach the learning enviroment and also practice. It is in our future plans to expand the user database by making the platform available to the students of the "Software Engineering" class to evaluate our system in order to examine the degree of usefulness of the collaboration modules in computer assisted learning system. Moreover, we are going to evaluate the usefulness of the student modeling in the Python programming language learning .

References

[1] Baghaei, N., Wade, V., Ashman, H., and Smyth, B. (2006). *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 4018 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[2] Chang, W.-c. and Chen, K.-c. (2008). Collaborative Learning Tool Applying to C Programming Language. pages 178–186.

[3] Engelmann, T., Dehler, J., Bodemer, D., and Buder, J. (2009). Knowledge awareness in CSCL: A psychological perspective. *Computers in Human Behavior*, 25(4):949–960.

[4] Fu-Chien, K., Tien-Hsin, F., and Chia-Liang, K. (2006). *The Design of Internet Collaborative Learning System Structure with the Integration of 3D Virtual Instruments*. Ieee.

[5] Kessler, G. and Bikowski, D. (2010). Developing collaborative autonomous learning abilities in computer mediated language learning: Attention to meaning among students in wiki space. *Computer Assisted Language Learning*, 23(1):41–58.

[6] Shen, H. and Sun, C. (2000). Recipe: a prototype for internet-based real-time collaborative programming. In *In Proceedings of the Second International Workshop on Collaborative Editing Systems*, pages 3–4.

[7] Shijian Luo, Shouqian Sun, and Yunhe Pan (2005). An object-oriented integrated knowledge approach to Internet-based product collaborative conceptual design. *International Conference on Computer Supported Cooperative Work in Design*, 2:1129–1134 Vol. 2.

[8] Zurita, G. and Nussbaum, M. (2004). Computer supported collaborative learning using wirelessly interconnected handheld computers. *Computers & Education*, 42(3):289–314.

[9] V. Tsiriga and M. Virvou (2004). A framework for the initialization of student models in web-based intelligent tutoring systems. *User Modelling and User-Adapted Interaction, vol. 14, no. 4, pp. 289-316, 2004*, 42(3):289–316.