Exploratory + Collaborative Learning in Programming: A Framework for the Design of Learning Activities

Agoritsa Gogoulou[1] Evangelia Gouli[1] Maria Grigoriadou[1] Maria Samarakou [2]
[1]Department of Informatics & Telecommunications, University of Athens, Greece
[2]Department of Energy Technology, Technological Education Institute of Athens, Greece
rgog@di.uoa.gr, lilag@di.uoa.gr, gregor@di.uoa.gr, marsam@teiath.gr

Abstract

In this paper, we propose a framework for the design of learning activities, which follows the key idea of the "Learning-for-Use" model and adopts characteristics from the exploratory and collaborative learning. We elaborated on the framework in the design of learning activities for Introductory Programming courses and we developed and evaluated in a real classroom environment, learning activities for the teaching of the infinite "while" loop. The experimental results are encouraging for the effectiveness of the activities in enhancing learning and indicate that the framework enables the students to gain deeper understanding of the programming constructs under consideration and to apply them more effectively.

1. ECLiP: The proposed design framework

Towards the direction of defining and using an integrated framework for the design of learning activities, we propose a framework, called ECLiP: Exploratory + Collaborative Learning in Programming. The framework follows the key idea of the "Learning-for-Use" model [1] and adopts characteristics from the exploratory and collaborative learning. ECLiP forms a basis for the design of an integrated set of learning activities, in the context of any subject matter, comprising a three-step process: (i) Motivation to acquire knowledge: can be achieved by creating demand and/or eliciting curiosity [1]. The set up of conditions that are likeable/meaningful to the students, and/or they are related to a goal that is challenging, and/or they provoke and give the opportunity to the students to express their beliefs/opinions, can contribute positively to the creation of motivation, (ii) Knowledge construction by Exploring+Collaborating: can be supported through observation and/or communication with others [1]. The design of learning activities may take advantage of the characteristics of the exploratory and collaborative learning and enable the students to explore on their own the meaning and the particular properties of the concepts/ constructs under consideration and to externalize/ negotiate/argumentate on their thoughts/actions, and (iii) Applying-Refining Knowledge: can be supported through the processes of reflection and application [1] that enable the students to reorganize their knowledge and connect the "new" knowledge to the existing and support knowledge retention, future retrieval and use.

The three steps of the ECLiP framework may be interlaced (e.g. the process of reflection may be partly or fully incorporated in the second step or the process of application may cause "new" motivation). The elaboration of the framework in the design of activities for Introductory Programming courses resulted into the following guidelines for each one of the three steps:

Becoming a motivated student: It is essential to engage the students in learning activities concerning simple authentic problems that are close to their experience, show the usefulness of the programming process beyond the specific course and make them to become curious and seek for new knowledge. Various forms of scaffolding (e.g. questions, case studies) may be included in order to (i) stimulate the students to acquire the new knowledge, and/or (ii) make them aware of their own difficulties.

Learning by Exploring+Collaborating: The learning activities may guide the students to activate their existing mental model and subsequently to revise/modify it. This can be achieved, for example, if the students are engaged in learning activities that ask them, at first place, to predict the results of a simple program, addressing specific programming constructs, and afterwards to examine and compare the predicted results to the actual ones [2]. Through suitably designed questions and additional scaffolding tasks, the students can understand the functional characteristics of the programming constructs and revise appropriately their mental model in case of preconceived misconceptions. Additionally, the students may run and study examples, try to understand their function and make changes to the programs or proceed with solving similar problems. Moreover, in order to enhance the learning and support the development of skills for self-monitoring/self-control/critical thinking, it is essential to incorporate characteristics from collaboration. The collaboration may take place at different stages and may have the form of groups where the students act equivalently or the form of "role playing" where they act according to specific roles (e.g. "pair-programming" [3]).

Applying-Refining knowledge: Reflection may be achieved (i) by asking the students to check their thinking, and/or to reason their decisions and/or (ii) by engaging the students in collaborative activities in which they examine and discuss their ideas with others and/or evaluate others' statements/solutions. The learning activities concerning the application of knowledge, may ask the students to (i) develop/modify a simple program, and/or (ii) check the correctness of a program and modify it appropriately according to the problem definition, and/or (iii) act as evaluators of other students' work. The students may collaborate at different stages of the learning activities (e.g. at the beginning for exchanging ideas in the design of a solution or at the end for implementing a program).

2. ECLiP in practice

The experimental evaluation of the ECLiP took place during the "Introduction to Informatics" course at the Faculty of "Technological Applications" of the Technological Education Institute of Athens, in the beginning of the academic year 2002-03 and it was focused on the teaching of the "while" loop. Initially, we conducted a pre-test in order to examine whether the students encounter difficulties in case of an infinite "while" loop and to detect their performance level and any preconceived misconceptions. The students were divided into a control group consisting of 15 students taught according to the traditional teaching approach and an experimental group consisting of 20 students.

On the basis of the ECLiP and taking into account the difficulties that were revealed from the pre-test, we designed a set of three learning activities, which were applied to the experimental group: (i) Design of a problem solution, (ii) Exploration of the functional characteristics of the "while" loop, and (iii) Implementation of a program. The analysis of the students' answers to the activities, indicated that (i) the first activity motivated the students and the adopted form of collaboration ("pair-programming") contributed positively to the improvement of their performance, (ii) the second activity enabled the students to explore the characteristics of the "while" loop and understand the role of the "update" statement, and (iii) the third activity gave the students the opportunity to apply what they had learned by implementing the solution of the problem given in the first activity. The students' performance in implementing the program was significantly improved (50% of them provided a completely correct solution) in comparison to the results of the pre-test (20% of them gave a correct solution).

A post-test was conducted, to both groups, in order to investigate the effectiveness of the learning activities in enhancing learning. The post-test included two activities, slightly different, aiming to investigate whether the students were able to (i) identify an infinite "while" loop, (ii) reason the correctness/incorrectness of the solution, given a problem, and (iii) determine and implement the appropriate changes. The results, depicted in Figure 1, indicate that the majority of the students of the experimental group seemed to have gain deeper understanding while both groups improved their performance in identifying an "infinite" while loop and implemented successfully the appropriate modifications.

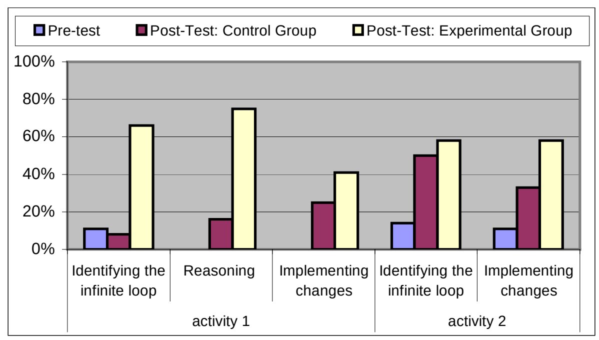

Figure 1. Results concerning the pre-test and the post-test

3. Conclusions

The ECLiP framework, proposed in this paper, supports the design of learning activities for Introductory Programming courses, addressing both the comprehension and the application level of the learning goals. The first experimental results revealed the effectiveness of the designed learning activities in enhancing learning. In the near future, we plan (i) to apply and evaluate ECLiP in other cases of programming constructs, (ii) to support/ adopt ECLiP in the context of a computer-based learning environment, and (iii) to apply ECLiP in the context of other subject matters such as Computer Architecture.

4. References

[1] Edelson, D., "Learning-for-Use: A Framework for the Design of Technology-Supported Inquiry Activities", *Journal of Research in Science Teaching*, 38(3), 2001, pp. 355-385.

[2] Lischner, R., "Explorations: Structured Labs for First-Time Programmers", *In Proceedings of SIGCSE '01 Conference*, ACM Press, Charlotte, USA, 2001, pp. 154-158.

[3] Williams, L. & Upchurch, R.L., "In Support of Student Pair-Programming", *In Proceedings of SIGCSE '01*

Conference, ACM Press, Charlotte, USA, 2001, pp. 327-331.