The pSKILLS experience: Using modern Educational Programming Languages to revitalise computer science teaching

Emanuela Ovcin[1], Chiara Cerato[2], David Smith[3], Nektarios Moumoutzis[4]

eovcin@corep.it, ccerato@corep.it, davidsmith@intuitivesolutions.com.au, nektar@ced.tuc.gr

[1-2]Consorzio per la Ricerca e l'Educazione Permanente (Italy), [3]Bundesministerium für Unterricht (Austria), [4]Technical University of Crete (Greece)

Abstract

Engaging learners has never been more challenging as ICT provides an ever-increasing array of hardware and applications designed to stimulate and motivate interest and absorption. Educationalists are faced with opportunities to select the platform and environment that address the needs of the lifelong learner, from student to administrator, that includes the curriculum moving from knowledge transmission to knowledge construction.

This paper presents the first tangible results by a two-years project (PSKILLS) aiming at enabling computer science teachers in secondary education to exploit modern educational programming languages. The project is set in the context of student-centered pedagogical settings, in order to contribute to making their courses effective, creative and attractive.

Several surveys at international level were conducted: to ascertain the current state of students and teachers training in computer sciences; to evaluate existing pedagogical approaches best suited to the computer science teaching; to identify, between available Educational Programming Languages, the most suitable one to be used for the training activities of the project.

On the basis of the identified training needs, a suggested common curriculum was created and recommendations for the adaptation of national computer science curricula were produced based on the use of the selected EPL, Scratch. The adapted curriculum would cover all core programming concepts with an intuitive user interface that can be easily understood by beginners to programming.

Training Scenario Templates focusing on the pedagogically-driven use of EPLs were conceived to provide training scenarios for teachers training to design, plan, and orchestrate learning activities that are intended to increase students' motivation and engagement. The templates are to be used with an array of tools including the Multimedia Open Learning Environment (MOLE), which is the learning support platform used for the establishment of the training infrastructure, managing courses, supporting learning processes and communities through the Web.

1. Introduction

pSkills (Programming Skills Development in Secondary Education by means of Modern Educational Programming Languages) [1] is a two-year project co-funded by the European Commission in the Life Long Learning Programme, Comenius sub-programme. The project is conducted by a Consortium of seven partners belonging to five European countries (Greece, Austria, Italy, Estonia and Hungary) with expertise across various levels education and training as well as in the development of technology platforms.

The objective of the project is to enhance the use of Educational Programming Languages (EPLs) by Computer Science teachers in Secondary school, in order to create computer science courses that reduce the abstract nature of computing as well as being more attractive and creative. The first outcomes, already delivered by the project, include the development of a common computer science curriculum and the identification and selection of the most appropriate EPLs for the target groups, as well as the design of a training framework for teachers and learning scenarios templates for students

aimed at giving teachers the opportunity to select, design and implement/model their own scenarios. A learning support platform has been developed to provide a common access point (repository) of learning/training scenarios and accompanying material as well as to provide an infrastructure supporting the in-service training of pSkills primary target group (computer science teachers in secondary education).

2. A Common computer science curriculum integrating Educational Programming Languages

The pSkills project proposed to examine the structure of the computer science curriculum in schools across Austria, Estonia, Greece and Italy and whether using modern educational programming languages in computer sciences would have a positive impact on both teachers and students to meet the needs of learners [2]. This approach is fundamental to allow the integration of educational programming languages and reduce the abstract nature of computer studies.

An examination of the education systems in the four countries found differences in school organisation, in academic content and the delivery of that content. To enable a comparison between the computer studies curriculum offered in the four countries, the IFIP curriculum [3] was selected because it provided a matrix to determine the stages of computer science delivery. A documentary analysis method was selected to collect data about the stages of computer science delivery from the official education ministry portals of the respective countries and compare those data with the IFIP stages. The analysis clearly showed that the core curriculum for computer sciences in most countries is centred on computer literacy. This type of approach is not only to ensure that students have the appropriate ICT skills across the curriculum, but that teachers too are trained to use computers [4], [5] and it may also reinforce traditional methods of teaching [6]. This type of pedagogical approach is not motivating for students [7] who are already proficient users of technology and it may be a reason why they do not elect to undertake further studies in this field.

The project consortium used the analysis to recommend a direction for change within the compulsory computer science curriculum and adopt a methodology using Educational Programming Languages (EPLs) [8] to stimulate student interest in conjunction with different pedagogical approaches. In order to use EPLs effectively, the project further recommended the development of a common curriculum to incorporate a greater emphasis on programming.

The development of a common European curriculum was based on the premise that it be broad enough to extend from primary through to tertiary education and that the core components of computer science should be expressed as a function of an Educational Programming Language, requiring the identification of all aspects of the science with the appropriate level of complexity for the learner. The development of a common European curriculum was done simultaneously with the selection of an EPL as both factors impacted upon the other.

Eight areas from computer science theory to artificial intelligence were defined by the consortium as necessary to shape the framework needed for a computer science curriculum in school education and providing the required knowledge needed for the suite of courses developed for higher education. The identified core components were then mapped to the functionality of educational programming languages. Together these areas represent the basis of the European Common Curriculum

3. Modeling Learning Scenarios with Scratch

Structured templates of Learning Scenarios [9] have been created on the basis of pSkills survey on the current computing curricula and following the choice of the most suitable EPLs for the target users (teachers and their students at secondary education schools). The selection of EPLs was based on the comparative analysis of the most common EPLs, considering specific characteristics of the software and attitude to facilitate the learning process: basic understanding and usage of EPLs; distributing and sharing software; basic programming concepts; special functionality of EPLs; creating and managing movement; creating and using sound; creating and using graphics; quality of code; debugging. Shortlisted EPLs were surveyed and additional criteria were set, including supported languages and availability of an active development community.

Following this survey, Scratch [10] (Fig. 1) was found to be the most suitable tool for the scope of the project, since it covers all the basic logic of programming while having a very intuitive interface for both, students and teachers. Apart from being the most attractive and simple to use tool, it has a large community of users offering thousands of projects shared through the official web portal. The second

EPL selected, Squeak-Etoys [11], supports simulations and offers a more sophisticated environment than Scratch: it is therefore suggested for more advanced users.

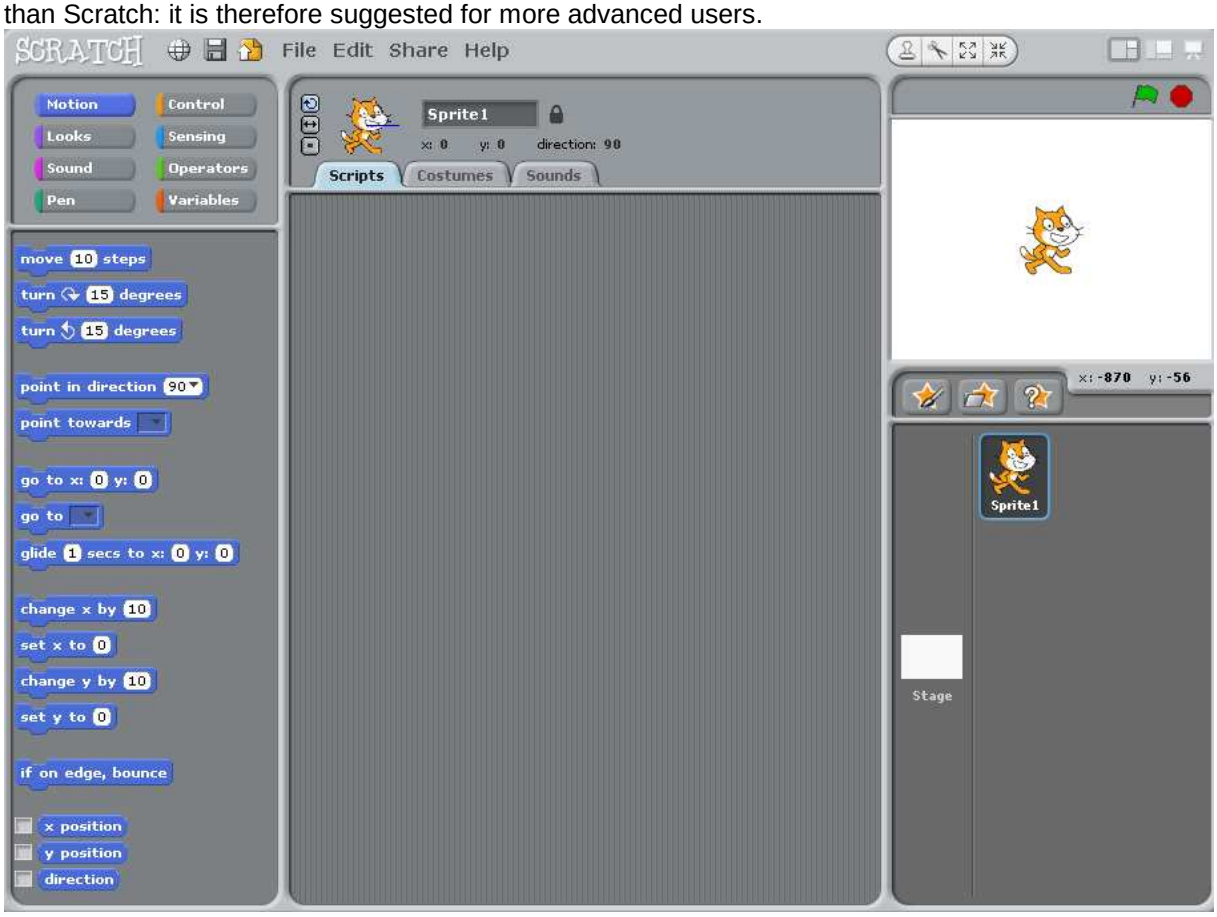

Fig.1. Scratch homepage

Learning Scenarios have been created on the basis of pedagogical approaches based on the constructivism and problem-solving approaches, to introduce the EPLs as a way of helping students to understand programming concepts. Training scenarios focus on the pedagogically-driven use of EPLs in the context of training teachers to design, plan and structure learning activities to increase students' motivation and engagement with programming. The concept of learning design has been used to describe a representation of the learning experience to which students are exposed. For example, a teacher may design a learning activity that requires students to use basic functions of Scratch for creating a simple program. Learning design is defined as "an application of a pedagogical model for a specific learning objective" [12], with the intention that with sufficient and effective support from the training material, the pre-structured templates and the training framework, computer science teachers will be able to design pedagogically-driven learning activities that will lead to the full exploitation of Scratch.

A scenario development methodology has been created to structure scenarios that may be used by computer science teachers, in order to design their teaching activities and courses that can encourage students to understand, reflect and solve problems. The methodology has been focused on processes and strategies to enable teachers to develop more explicit pedagogical structures while proposing strategies, tools and resources that would inspire them to teach programming courses in more creative ways, as well as giving them the opportunity of adapting and modelling scenarios to their needs. pSkills is aimed at giving teachers the opportunity to describe and share their own experiences of teaching programming with other teachers, with mentoring activities foreseen to provide teachers with some feedback for further strengthening the process of designing their own scenario. Templates are designed to be used initially for computer science teachers as a pedagogical guide with a teacher specifying a pedagogical approach that is aligned with the purpose and context of the learning activities and the intended learning outcomes.

The pSkills scenarios are structured with attributes referring to: the structure of the scenario (simple or composite), curriculum and intended learning outcomes, activity and environment for learning. These scenarios will be tested and validate in four countries (Greece, Austria, Italy and Estonia) and the structure may be modified after the trial period. Based on the templates the consortium produced four teachers' scenarios for helping computer science teachers to emulate pedagogical practices and learning activities in real-world contexts and these will be used in conjunction with the MOLE portal [13] and the previously mentioned training platform for validation.

4. The training infrastructure: the MOLE Platform

The MOLE (Multimedia Open Learning Environment) platform is a multimedia information system for managing courses, supporting learning processes and learning communities through the Web. The platform includes features for self-learning and reuse of educational material in different platforms compatible with the international SCORM standard.

MOLE was initially developed to support education and organize laboratory processes and learning communities for the needs of the Electronic and Computer Engineering Department of the Technical University of Crete. In the context of the pSkills project, MOLE is used to provide the core infrastructure of supporting the training and learning activities of the project. At the first phase of the project the platform was adapted to support multiple languages and to ensure alignment with the requirements of the project.

A special installation was made for the pSkills project that can be accessed at http://pskils.moleportal.eu.

The MOLE offers a set of services for:

- the organization and management of digital educational content
- informing learners
- learning communities
- supporting educational activities
- monitoring services

The methodology followed to integrated MOLE services within the pSkills framework is based on the project's learning scenarios. Each learning scenario consists of activities that are mapped onto specific MOLE services to be used for the implementation of each training activity. The material that accompanies each learning scenario is integrated in the course digital repository.

5. Concluding remarks

The scenario-based approach adopted by the pSkills project supports the design and adaptation of pedagogy and EPLs for a teaching and learning process in computer science. However, acquisition of professional knowledge and learning come from a mixture of experience, research and mentoring and as such one cannot focus exclusively on representing knowledge and practice within a specific instance of training/learning scenarios. The pSkills scenarios are going to be used as a basis for creating relevant training material and content (e.g. Scratch cards, tutorials, videos, presentations) and together with the templates are going to be used during the pilot and implementation phases of the project for helping computer science teachers to create, re-use and adapt their own learning scenarios. A computer science teacher will study the available scenarios for re-using/adapting or creating new scenarios from the available templates, and determine the resources needed to apply the scenario (i.e. Scratch cards, video tutorial presentations etc). The teachers will utilise the resources provided by the project (e.g. scripts for training material development) and from other computer science teachers that are stored in the MOLE platform, to develop a rich and motivating computer science course which engages students and provides a basis for collaborative working with other computer science teachers.

References

[1] pSkills Website: http://pskills.ced.tuc.gr/.

[2] Anderson, T.R. & Rogan, J.M. (2011). Bridging the educational research-teaching practice gap: Curriculum development part1: Components of the curriculum and influences on the process of curriculum design. Retrieved February 11, 2011 from http://onlinelibrary.wiley.com/doi/10.1002/bmb.20470/full

[3] van Weert T. (Ed.). (2000). Information and Communication Technology in Secondary Education. Available at: http://wwwedu.ge.ch/cptic/prospective/projets/unesco/en/teachera.html (accessed 11December, 2009).
[4] Sharp, D.B. 2002. Optional strategies for developing teacher competencies. Proceedings on Computers in Education. 2: 1087-1091.
[5] Angus, L., Sutherland-Smith, W. & Snyder, I. (2003). ICTs and educational (dis)advantage: Cultural resources and the digital divide. In G. Walford & R.A. Jeffrey(eds.), Educational inequality and ethnography. London: JAI Press.
[6] Kirkland, A.B. (2009). Bridging the learning divide. Feliciter v.55(6), pp. 236-237.
[7] Piccianno, A.G. (2002). Beyond student perceptions: Issues of interaction, presence and performance in an online course. Journal of Asynchronous Learning Networks. v. 6.
[8] Rai, S., Wong, K.W. and Cole P. 2006. Game Construction as a learning tool. Proceedings of the 2006 international conference on Game research and development. Perth, Australia: Murdoch University.
[9] D2.2 Training Scenarios, 2010. http://pskills.ced.tuc.gr/Outcomes/Deliverables.
[10] Scratch Website: http://scratch.mit.edu/.
[11] Squeak-Etoys Website: http://www.squeakland.org/.
[12] Conole, G., M, Oliver. (2006). Contemporary perspectives in e-learning research. London: Routledge.
[13] MOLE for pSkills: http://pskills.moleportal.eu.