Challenging Prospective Computer Engineers to Design Educational Software by Engaging them in a Constructivist Learning Environment

MARIA KORDAKI

Department of Computer Engineering & Informatics, University of Patras, Rio, 26500 Patras, Greece,

e-mail: kordaki@cti.gr

Challenging Prospective Computer Engineers to Design Educational Software by Engaging them in a Constructivist Learning Environment

This study focuses on the development of prospective computer engineers' knowledge of educational software design through their involvement in a constructivist learning environment emphasizing project based learning. Within this environment prospective computer engineers (PCE) adopted a variety of roles namely: learners, teachers, users, designers, implementers and evaluators of educational software dealing with concepts of Computer Science while also taking into account theoretical educational considerations regarding constructivism and social views of knowledge construction. The analysis of the data shows that PCE frequently start by considering traditional behavioristic views regarding teaching and learning as well as regarding the design of educational software. The PCE progressed to accept more constructivist views regarding teaching and learning as well as to designing educational software by communicating their knowledge with their colleagues and the teacher in the project based context. PCE also progressed through the evaluation of the educational software using real classrooms. By exploiting the given feedback, PCE improved the quality of software specifications.

Keywords: constructivism; design of educational software; computer science;secondary education.

Introduction

In today's digital world, computer and networking technologies play a crucial role in the culture of all societes, of all scientific disciplines and of people's every day lives. Computers also have a significant impact on our financial and commercial life, so they have social power. Moreover, computers can act as catalysts in the learning and teaching of all subjects providing promising opportunities for the construction of new meanings regarding the whole learning process (Papert, 1980; Noss & Hoyles, 1996; ACM, 2001).

Based on the above, Computer Science and Engineering (CS&E) concepts are appreciated as essential in the curricula of all levels of education (ACM, 1997a; ACM, 1999; ACM, 1997b). Moreover, the integration of educational software and especially of open problem solving computer environments (OPSCE) in the curriculla of all learning subjects is aknowledged. In some countries, CS&E concepts are part of the education of all secondary level education pupils while emphasis is put on specific concepts in classes preparing pupils to acquire certain skills regarding computers. In other countries, there are also some important attempts being made regarding the use of the computer as a cognitive tool in learning concepts of all scientific disciplines. Therefore, computer engineers can play a crucial role in K-12 education, both as teachers of CS&E, as well as users, designers, developers and evaluators of educational software and especially of OPSCE. The need to prepare these engineers to play this significant role is obvious.

Constructivist learning contexts can become powerful for learners. In these contexts learning is viewed as an active, subjective and constructive activity as the learner is at the center of the learning process (von Glasersfeld, 1987). Examples of such contexts include project-based authentic tasks with real-world relevance (Bishop, 1988). Learners are also encouraged to develop many higher-order thinking skills when they become involved in projects regarding the design of educational software. Moreover, a number of studies have already well documented the promising results of engaging learners in the role of designer of educational software (Harel, 1991; Kafai, 1996; Neo & Neo, 2002; Liu & Hsiao, 2002). Despite this fact, constructivist project-based environments engaging prospective computer engineers in the role of the designer and of evaluator of OPSCE have not yet been reported.

The aim of this study is to investigate the effect of a constructivist learning (CLE) on the knowledge of prospective computer engineers (PCE) regarding the design and the evaluation of OPSCE. This environment was designed to present PCE with opportunities to actively learn how to design and evaluate OPSCE for the learning of Computer Science and Engineering concepts by primary and secondary level education students. The environment emphasizes project based learning. In this environment PCE were put in the position both of designers and of evaluators of OPSCE. The rationale of the designed CLE is presented in the following section of the paper. Next, the context of the study, as well as the design of the CLE in which PCE acted as learners, is described. Then, the progress of these engineers through the design, the implementation and the evaluation of the educational software they produced in this constructivist framework is also presented. Finally, the effects of the designed environment on prospective computer engineers' knowledge, regarding both the design and the evaluation of educational software, are discussed. This study ends with the conclusions.

The rationale of the constructivist learning environment

Constructivism emphasizes learning as an active, subjective and constructive activity placed within a rich and meaningful context for the learners (von Glasersfeld, 1987). Authentic tasks based on project work relevant to real world constitute such activities (Bishop, 1988; Liu & Hsiao, 2002). Project-based learning encourages learners to set their own learning goals and to be responsible for their learning. Learners also have the opportunity to be cognitively involved and use their higher-order thinking skills in project-based learning settings. These skills can be developed as a result of social interaction between both the members of each learning group as well as the members of the entire class including the teacher. Typically, project-based learning has five characteristics: a) centrality, b) a driving question, c) authenticity, d) constructive investigation and e) student autonomy (Thomas, 2000). The project is central to the learning environment and the driving question compels students to learn about the essential concepts of the learning subject. The driving question usually requires students to be involved in authentic real-life tasks as well as to play authentic roles. Performing these tasks, students are engaged in problem solving processes, including: investigations, decision-making, design, implementation, reflection and hypothesis testing. Throughout these processes, students are inspired to construct their own knowledge as well as to be autonomous in their decision-making and to take control over their learning. Moreover, successful engineers have to be experts in project-based work aiming at the solution of real problems as well as in group collaboration (ACM, 1991; Hagan & Sheard, 1998).

Engaging prospective engineers in the role of designers of OPSCE is a type of project based learning. A number of major thinking skills have been identified and are required for a designer. These skills have been placed in five categories: a) project management, b) research, c) organization and representation, d) presentation and e) reflection (Carver, Lehrer, Connel & Ericson, 1992). A number of studies have already well documented the promising results of engaging learners in the role of designer. Students developed problem-finding skills and improved their knowledge of fractions and Logo programming in their attempts to design educational software for the learning of the concept of fractions using Logo language (Harel, 1991). Students also reached a high level of reflection beyond traditional school thinking while they attempting to design games using Logo language (Kafai, 1996). In addition, students acquired certain design skills such as mental effort and involvement, interest, planning, collaboration and individualization through the design of hypermedia projects (Spoehr, 1993; Leahrer, et all, 1994). Apart from all the facts mentioned above, the design of OPSCE is not an easy task.

Designers of OPSCE are influenced by constructivist and social views of learning (Papert, 1980; CTGV, 1991; Laborde & Strasser, 1990, Noss & Hoyles, 1996; Duffy, Lowyck & Jonassen, 1989). Constructivist design emphasizes learning as an active subjective and constructive activity. At the same time, social considerations of knowledge, stress the role of computer tools as cognitive tools in the learning process (Vygotsky, 1978; Leahrer, et all, 1994; Jonassen, Carr & Yueh, 1998). Computers are an exceptional medium for constructivist learning (Noss & Hoyles, 1996; Jonassen, Carr & Yueh, 1998; ACM, 2001). They are expressive, giving pupils the challenge of intrinsic and visual feedback of their actions, helping them to take control of their learning. They also provide pupils with the opportunity to construct and to explore diverse and linked representations of a concept, which allows them to have both different cognitive starting-points and destinations. Computational objects can act as 'transitional' objects between the natural objects of the real world and the abstract world (Noss & Hoyles, 1996). By experimenting with these computational objects, pupils can smoothly surpass the gap between the concrete and the abstract. The above render evident the value of OPSCE.

In particular, OPSCE can play a crucial role in pupils' thinking, encouraging them to express their knowledge as well as to explore the knowledge of others (Papert, 1980; Laborde & Strasser, 1990; Noss & Hoyles, 1996). In such environments a variety of tools can be provided supporting the performance of complex, authentic real life activities by the pupils motivating them in their learning process (Jonassen, 1991; CTGV, 1992). Pupils can also express their inter- and intra-individual learning differences by selecting among the provided tools those most appropriate for their cognitive development (Kordaki & Potari, 2002).

A number of well known examples of OPSCE have been reported e.g. Cabri-Geometry II for the learning of concepts of Geometry (Laborde & Strasser, 1990), Interactive Physics for the learning of concepts of Science, the programming language Logo for the learning of concepts both of Geometry and of programming. Moreover, a number of problem solving environments regarding the learning of CS&E by higher level education students is also reported (Stasko, 1990; Naps, Hundhausen & Swander, 1993; Grigoriadou, Toula & Kanidis, 2003). However, OPSCE for the learning of CS&E concepts by primary and secondary level education are rarely encountered. Based on this fact the design of such environments it is a very interesting issue.

The context of the study

This study aims to illuminate how a constructivist learning environment that puts prospective computer engineers' in a designer's position, affected their knowledge of the design and of evaluation of educational software. This environment was designed for PCE and is offered to them as an elective course on their curriculum at the department of Computer Engineering and Informatics, University of Patras, Greece.

This innovative, two semester course is named ‘Educational Technology & Computer Science Education’. Each semester lasts for 12 weeks, with 2 lecture hours per week. The aim of this course is to help PCE to design appropriate OPSCE for the learning of specific computer science concepts for pupils at primary and secondary level education.

This experimental study took place during the spring and the fall semester of 2001-02 and 2002-03. Two groups of fifteen and forty PCE participated in the course for these two years respectively. The researcher participated as a teacher of this course working closely with these PCE throughout the experiment. The context of the CLE mentioned above was modified after the first year of this study in order to better fulfil the PCE needs.

Part of this study, regarding the first year of the experiment, has been presented in Kordaki (2003) while the whole experiment is presented in this paper.

Data sources and analysis

Qualitative methodology, (Cohen & Manion, 1989) was used to illuminate the effect of the previously mentioned constructivist course on the knowledge of PCE regarding the design and the evaluation of educational software. More specifically, observations and reflections by the researcher were made during all phases that took place during the two semester course. Moreover, unstructured interviews were conducted with all the PCE who participated in this course. During these interviews, questions were addressed to the subjects at the end of this study, such as: what were your main problems during the course, what do you think is the more significant thing that you have learned, what were the main factors that positively affected your work?. The collected data consisted of a) the field notes of the researcher’s observations during the 2 year course, b) the audio tapes including the conducted interviews. The data were transcribed, collated and coded using the themes that emerged. Patterns from the data were extracted and the relationships among the coded segments were compared and contrasted. Using the research question as a guide, the data were sorted into categories according to their common themes and relationships (Babbie, 1989).

The constructivist learning environment

At the beginning of each semester PCE were asked about their expectations regarding this course and were explained its proposed outline and the learning approach to be used. This data was then used to form the final outline of the course. Unlike a traditional teacher-telling classroom, this class exploited the PCE previous knowledge about learning and teaching as well as their knowledge regarding the design and evaluation of educational software. Hence, in each two-hour teaching period, a main topic was presented as a driving question to be answered in each class. Then, prospective engineers worked in groups to express their own knowledge regarding each main topic and then presented this knowledge to the whole class. This was followed by a class discussion. The map of the opinions expressed by the members of the whole class, in the form of a hierarchical network was constructed on the blackboard by the researcher.

The organization of the CLE during the first year of the study (2001-02). It is worth mentioning that the organization of the course during this first year of the study (Kordaki, 2003) differed from the second year (2002-03). Each semester was divided in two phases of approximately six weeks each. The first six weeks of the spring semester was devoted to theoretical discussions and negotiations regarding teaching and learning.

The first six weeks of the fall semester was devoted to the same discussions focusing on the design of OPSCE using basic aspects of consructivism. During all these discussions PCE showed little interest. In addition, they were unable to put these theoretical issues into real practice during the second phase of each semester.

Based on these facts the organization of both semesters was modified, for the second year of the study where it was decided to give more time to PCE to acquire real experience in teaching as well as in designing OPSCE using basic aspects of consructivism.

The full description of the course during the second year of the study is presented in the following section.

The organization of the CLE during the second year (2002-03) of the study. The organization of the spring semester consisted of two phases. Phase I, (modified to approximately two weeks), was dedicated to discussions and negotiations about its outline as well as for the assignment of project work

Phase II, (modified to approximately ten weeks), focused on working in groups (3 – 4 PCE per group) to create a typical classroom environment for the learning of computer science concepts suitable for secondary level education pupils. Each group selected a topic relating to a specific CS&E concept. PCE followed a three stage model: designing the learning environment, experimenting within this environment as teachers in the prospective engineers' class and revising the constructed environment by reflecting on the experience they aquired in the previous stages.

During the design stage PCE were engaged in the construction of three models: firstly, the model of learning, secondly, the model of the specific learning subject matter, and thirdly, the model of possible pupils' actions in order to perform essential learning tasks regarding this specific learning subject.

To construct the first model, each group of PCE had to make explicit, basic aspects of constructivism and social views of learning in the form of specific teaching-strategies. At this stage discussions and negotiations were performed regarding the driving question namely: '*What is effective learning for the concepts of CS&E?*'. The goal of this question was to clarify traditional, constructivist and social views of learning. Here, discussions took place regarding the roles of: the teacher as facilitator in the learning process, the learner as an active and central actor in this process, problem solving activities such as helping learners develop higher cognitive skills, computer tools as cognitive tools, and collaboration among learners in any specific learning community. The need for the learning of computer science concepts by primary and secondary level education students was also disscused (ACM, 1999) and simultaneously, the discipline of computer science and engineering as: theory, science and engineering emphasizing the role of solving real problems by teams (ACM, 1991; Ellis, 1998; Hagan & Sheard, 1998; Kordaki, 2001). The role of finding out the most optimal solution by taking into account time, cost and efficiency in real-world engineering problems was also acknowledged (ACM, 1991).

To realize the second model, each group of PCE had to clarify the essential concepts of the specific learning topic as well as to analyze and reduce these concepts to a more elementary level. At this point PCE also had to construct appropriate learning activities for their students. To do this, the PCE had to exploit the knowledge they acquired during the construction of the previously mentioned model.

To form the third model, PCE had to anticipate pupils' behavior regarding the learning activities to be performed for the learning of the specific topic. The construction of the three models above was based on the related literature. In order to make the third model more valid, specific investigations were performed by PCE in classrooms with pupils.

During the experimental stage, PCE acted as teachers and received feedback from their students. As these students were also PCE the feedback was essential and constructive. Feedback was also given by the researcher who also participated as a teacher of this class.

The organization of the fall semester also consisted of two phases. Phase I, (modified to approximately two weeks), where discussions and negotiations about the outline of the course took place as well as the assignment of project work.

Phase II, (modified to approximately ten weeks), focused on working in groups to create an OPSCE for the learning of computer science concepts by secondary level education pupils. Each group consisted of the same members as in the spring semester, each selecting the same CS&E concept. PCE followed a five stage model: a) designing an OPSCE regarding the assigned learning subject, b) implementing the designed OPSCE using a specific programming language, c) evaluating this environment by presenting it to the class of PCE, d) revising the constructed environment by reflecting on the experience they aquired in the previous stage and e) evaluating this environment using real pupils.

During the design stage PCE were engaged in discussions and negotiations regarding the driving question: '*What kind of knowledge is implied in the design of*

OPSCE?'. The goal of this question was to clarify basic design principles of traditional, and constructivist computer learning environments. During this stage PCE were asked to use and to criticise well known examples of such environments. Here, the design of educational software was also discussed as a process of modeling. More specifically, this process was analyzed in sub-processes including the construction of the three models mentioned in the spring semester. However, in this semester emphasis was put on the interpretation of these models in terms of specifications of computer tools. Moreover, the effect on the whole learning context as a result of introducing the computer as a cognitive tool into a typical classroom setting was discussed. More specifically, the effect of this tool on the role of : the teacher, the learners, the learning activities, and the collaboration involved was examined. In addition the role of OPSCE in the learning of computer science and engineering concepts was mentioned as highly valuable.

Based on the above PCE constructed the three previously mentioned models and used the results to implement the specifically designed OPSCE which they presented then to the class of their colleagues.

The stage of evaluation of the implemented OPSCE by the prospective engineers' class was designed to exploit their expertise regarding technical and educational issues as well as the variety of their views regarding the learning subject. Receiving feedback from the members of this class as well as the researcher, PCE had the opportunity to revise the models and OPSCE they constructed. During this stage each piece of the implemented educational software was presented and revised about three times as the feedback given by the participants of this class was appreciated as essential and constructive.

At the beginning of the evaluation stage with real pupils, the objectives, the possible research questions and the mtethodology of a qualitative evaluation study were disscused. Next, each prospective engineers' group selected a secondary school class of pupils to try out the produced software. The collected data were classified into categories to give a picture of the learning that occurred within the context of each piece of this educational software.

Results

The most important data findings are summarized as following:

Prospective engineers' previous knowledge regarding teaching and learning. Here, PCE focused on the role of quality verbal presentation of content by the teacher as well as the role of assessing pupils through drill and practice activities.

Prospective engineers' views regarding the teaching of learning theories. As, the first teaching attempt was to engage PCE in theoretical discussions and negotiations regarding constructivism and traditional theories of learning, these learners found this tedious and meaningless. They expressed the need for specific teaching examples so they could understand the differences between traditional and constructivist learning. What was needed was a vital constructivist teaching example. As, the atmosphere of the class was static and boring and the learners became tired and disinterested, the modification of the teacher –telling/asking- approach became obvious. Therefore, it was decided to teach constructivism by putting its implications immediately into practice. As a result, PCE were put in groups and were asked to co-operate to answer essential questions regarding teaching and learning as well as to give appropriate examples. Next, they were called on to present their work in front of their class and to negotiate their knowledge with the knowledge of their colleagues. The researcher tried to facilitate this negotiation by asking questions as well as by expressing her own knowledge.

The development of prospective engineers' conceptions about teaching and learning. PCE first teaching attempts emphasized: a) the presentation of the content of the subject matter, b) the construction of a question set to explore pupils' previous knowledge, c) school book-like activities, d) individualistic communication, e) the assessment of the pupils by using questionnaires. Throughout this stage, the intervention of the researcher helped PCE to locate the weak points of their teaching experiments. Apart from the weaknesses of these first teaching experiments, PCE stated that 'despite the fact that it is difficult to put basic aspects of constructivism into a real teaching practice we accepted these aspects as correct, but we need more experience and constructive feedback'.

Prospective engineers' views regarding the most essential aspects of teaching and learning. PCE focused on a limited number of constructivist and social aspects of learning among the plethora of those which were discussed and negotiated within the CLE. More specifically, PCE emphasized as significant in pupils' learning: the role of cooperation, the need to take into account pupils' previous knowledge as well as their personalities, the need to not criticize the pupils' mistakes, the role of the teacher in a friendly relation with his/her pupils. In my view, PCE found these aspects meaningful as they are connected to some painful learning experiences which were recalled by the PCE from traditional learning settings in which had once participated.

Prospective engineers' as users of educational software. Here, as well, PCE's, mainly focused on the technical knowledge needed for the implementation of the specific pieces of software they used. Based on their expertise as developers of software, PCE characterized the design of OPCE as an 'easy' and 'soft' task. Questions such as 'what are the possible basic design principles implied in this software?' and 'what can pupils learn by interacting in this software environments?' were opaque for these engineers and difficult to address.

Prospective engineers' as designers of educational software. PCE first design attempts emphasized the role of: a) multiple choice questionnaires to investigate pupils' previous knowledge regarding the learning subject, b) quality presentation of contents using hypermedia or multimedia, c) drill and practice learning activities and d) pupils' assessment and/or self-assessment using interactive quizzes and/or multiple-choice questions. Some opinions also implied that educational software can entirely substitute the teacher by integrating specifically designed artificial intelligent parts. Despite the fact that careful design of the 'user model' and obedience to the users' needs are fundamental principles of typical software design, here, these principles were totally ignored. As these engineers stated : 'we have a good knowledge of the learning subject, it is the pupils' task to learn it'. Moreover, the implied 'learning model' ignored the confrontation with pupils' inter- and intra-individual differences. It also ignored the need for active and constructive participation of pupils in their learning. During this stage interventions by the researcher were realized. These interventions emphasized: a) the role of computer tools in constructing multiple representations to give pupils the chance to express their learning differences, b) the role of an interactive environment providing a set of appropriate tools to stimulate pupils to actively construct their knowledge and c) the role of appropriate tools in providing pupils with the opportunity of solving essential real-life problems. Interventions were also performed by the other members of this class offering ideas regarding a variety of topics such as: the learning activities, new representations of the subject matter, ways to help pupils deal with cognitive conflict and correct their mistakes. By taking into account all these interventions mentioned above, PCE tried to create new versions of their piece of software. Then, they rejoined the class to demonstrate these new versions of software and to receive feedback. This process was repeated about three times resulting in the final version of each piece of software.

Prospective computer engineers' as evaluators of educational software. Prospective engineers tried to pilotically evaluate in real classrooms the pieces of software they had constructed. They used qualitative methodology; observing what pupils can learn by interacting in these software environments, and collected the appropriate data. Then, they classified these data into categories describing pupils' learning. They also recorded pupils' difficulties regarding the specific operations of the software and recognized the differences between their own and the pupils' knowledge. They were also surprised at pupils' inter-individual learning differences. As they stated ' the design of educational software radically differs from the design of a typical system of software because the process of construction of the user model is different. In the case of educational software it is difficult to describe the needs of the users as they can develop different and unforeseen learning behaviors. Research with real pupils is needed to describe their learning needs more accurately'.

Team work. Prospective computer engineers stressed the advantages of working in a group. They stated that team work helped them to a) make the work of the project easier by splitting and sharing responsibilities among their colleagues b) in to construct richer design ideas c) to learn from the knowledge of others d) to perform the specific job that they individually preferred by sharing the entire project-work according to the strengths and preferences of each member of the group. Furthermore, when they worked as a group in the same room they used a brainstorming approach to express their ideas, then they tried to interpret these ideas into software specifications, next they coded the appropriate program and finally, they reflected and tried to correct and to enrich their initial design ideas. Friendship among the members of each team also played an essential role in shaping smooth cooperation.

Whole class discussions. All members of the prospective computer engineers' class emphasized the importance of the role of presenting their pieces of educational software in front of the class of their colleagues including the researcher. As PCE stated 'it helped to clarify the design of our work while preparing to present and defend it in front of the class of our colleagues'. Typical questions posed by the researcher included: 'how does this educational software meet the basic aspects of constructivism?', 'what can pupils learn by interacting with this software?' 'how can a pupil construct his/her knowledge in this environment?'. These questions helped these engineers to reflect on their work and to transform their pieces of software to highlight the active characteristic of learning. Moreover, all prospective computer engineers agreed that the given feedback was constructive as many new design ideas and corrections were proposed by their colleagues.

Motivation. PCE were strongly engaged in the design, implementation and evaluation of the educational software that they were responsible for constructing. Three main factors motivated PCE : a) the reflection on their own learning experience and the recognition of the promises of contructivist and social views of learning. As they claimed : ‘we reflected on our own previous experience as learners and realized that all of our teachers viewed us as empty vessels never asking or exploiting our own opinions’, ‘we felt a vindication of our complaints regarding our past learning experience’, ‘we changed our views about teaching and learning’, b) the challenge to put into practice their newly acquired considerations of learning: ‘because our opinions about learning and teaching were transformed from traditional ones to those of a more contructivist nature, we would like to attempt to experience these new considerations in the design of learning tools and to try out these tools with real pupils to see the results’, ‘we saw that there are alternative working ways to teaching and learning’, c) to be competitive in their class: ‘I would like to be satisfactory in front of a class of my own colleagues’, ‘I would like to obtain a very good grade for this course’, d) to be appreciated by the scientific community: ‘my aim is to construct educational software that could be presented in a conference’.

Acquiring technical skills. Prospective engineers were helped to aquire specific technical skills by participating in this experiment. They studied and put into practice authoring tools and programming languages such as Delphi, ToolBook, Visual Basic, Visual C++, PHP, Jawa and Flash to implement the OPSCE they designed. The knowledge of these software packages is also very useful for the career of a computer engineer in industry.

The learning outcomes. Seventeen pieces of educational software were produced during this two-year experiment. Three of them were accepted, presented and published in proceedings of typical conferences regarding Information and Communication Technologies in Education (Venakis, Giannakopoulos, Pirli & Kordaki, 2002; Tsonis, Katis, Palianopoulos & Kordaki, 2002; Vlahogiannis, Kekatos, Miatidis, Misedakis, Kordaki & Houstis, 2002).

Discussion

Prospective computer engineers participated in a specifically designed constructivist learning environment from the position of teacher as well as of designer and evaluator of OPSCE. In this constructivist environment, learning was stressed through a) project-based work b) collaboration c) the expression of the learner's previous knowledge d) decision making e) presentation f) experimentation g) feedback and h) reflection.

Prospective computer engineers' previous knowledge regarding teaching and learning as well as regarding the design of computer learning environments rested on traditional behavioristic views. They relied on their own expertise regarding CS&E concepts and viewed the cognitive gap between their knowledge and the knowledge of their learners as a gap that had to be exclusively filled by the learners' attempts. These prospective engineers started to shift from this authoritarian position through having the opportunity to participate actively as learners in the constructivist environment that is presented in this study. By having a real-world working example of constructivist learning they reflected on their own past learning experience and simultaneously compared this experience with the experience they acquired as learners within this constructivist environment. By participating as learners in this environment they expressed their previous knowledge and realized that the learners' knowledge should not to be ignored. They also acknowledged the role of collaboration in small groups in order to perform actively meaningful activities. As these activities involved the design of learning environments that they then tried out in front of the class of their colleagues, they recognized the role of presentation and of receiving feedback from a group of experts. Here they also experienced the role of engaging in real teaching as opposed to being engaged in theoretical discussions and negotiations. Through this design and teaching experiment, prospective computer engineers accepted that constructivism is a viable learning and teaching perspective. Their efforts to be successful as constructivist designers of learning environments and as teachers, led them to acquire some confidence in this role. However, more time was needed to move from the role of 'learning to teach' to developing a constructivist teaching expertise. Even though prospective engineers felt comfortable in the role of the active learner they had difficulties as teachers in putting their learners in an active role also. They also had difficulties in focusing on the essential aspects of the learning subject and in designing meaningful activities for their learners as well as in firmly moving from the teacher-telling approach to one of a more constructivist nature.

Despite the fact that these PCE had already been taught specific topics regarding Human Computer Interaction as well as typical software design, they were unable to transfer this knowledge to the design of the educational software that was assigned to them. They used a trial and error approach and through discussions and reflections they realized that learning is strongly dependent on the situation within which it occurs. Moreover, as regards specifications of computer tools, they couldn't interpret the models they had constructed in the previous semester, namely: the model of subject matter, the model of learning as well as the learner model. The behavior of these PCE was modified as their roles changed. In particular, when these PCE acted as learner-users of a complete piece of educational software they focused on its technical characteristics and regarded its design as an 'easy' or 'soft' task. When they acted as designers in specifications of computer tools they experienced difficulties in interpreting the theoretical aspects of constructivism and of social views of learning. As designers they also faced difficulties in focusing on the fundamental aspects of the learning subject and in constructing meaningful learning activities as well as in designing appropriate tools to support learners in performing them. From the designer's position PCE were also careless about the possible behavior of the learners acting within the context of a specific educational software. Group collaboration and constructive feedback from the entire class of their colleagues including the teacher helped these engineers to face these difficulties and to transform the pieces of software they had produced. Finally, when PCE acted as evaluators of the educational software with real pupils they acknowledged the difficulties mentioned above. From this experience PCE saw the need to design specific tools that a) support the performance of meaningful activities regarding the essential aspects of the learning subject b) support learners in expressing their individual learning differences. To do this, they realized that they had to move from their own to the learners' position by performing specific research to investigate the learners individual learning behavior.

Conclusions

This study demonstrates that the knowledge of prospective computer engineers regarding learning and teaching as well as regarding the design of OPSCE was transformed through their participation in a specifically designed constructivist learning environment. This knowledge was modified through experiencing different roles such as: learner and teacher as well as user, designer and evaluator of educational software. By examining their previous knowledge, which implied traditional behavioristic views of learning, they moved gradually to more constructivist considerations. By relying on their expertise as computer engineers they initially viewed the teaching of computer science concepts as well as the design of computer learning environments as 'easy' tasks. When they acted as learner-users of educational software they were unable to find its design principles. By putting them in a designers position they were faced with the problem of how to interpret constructivist and social views of knowledge in specifications of computer tools. Despite the fact that typical software design principles put emphasis on the fulfillment of the users' needs, in this case of educational software design these principles were violated. Collaboration between the members of each group as well as constructive feedback from the class of their colleagues including the teacher helped them to improve their design. Moreover, evaluation of this software with real pupils gave them a clearer picture of its learning effectiveness.

Acknowledgements : Many thanks to Sally McKevitt for proof reading and assisting with the English.

References

ACM (2001) *ACM Computing Curricula*, Final Draft. Retreived from http://www.computer.org/education/cc2001/final/index.htm.

ACM (1997) *ACM Model High School CS Curriculum*. Edited by Merritt, S., Bruen, J., C., Philip East, J., Grautham, D., Rice, C., Poulx, K., V., Segal, G & Wolf, C. Retreived from http://www.acm.org/education/hscur/index.html.

ACM (1999) *ACM / K-12 Task Force – Issues*. Retreived from http://www.acm.org/education/k12/

ACM (1991) *ACM Curricula Recommendations*, Volume 1: Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force. Retreived from http://www.acm.org/education/curr91/homepage.html.

ACM (1997b) *IS' 97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information systems*. Retreived from http://www.acm.org/education/curricula.htmlIS97.

Babbie, E. (1989) *The Practice of Social Research*. CA: Wadsworth Publishing Company.

Bishop, A. J. (1988) *Mathematical Enculturation*. Dordrecht: Kluwer Academic Publishers.

Carver, S. M., Lehrer, R., Connell, T. & Ericson, J. (1992) Learning by hypermedia design:Issues of assesment and implementation. *Educational Psychologist*, **27**(3), 385-404.

Cognition and Technology Group at Vanderbilt (CTGV) (1991) Some Thoughts About Constructivism and Instructional Design. *Educational Technology*, Sept. 1991, 16-17.

Cognition and Technology Group at Vanderbilt (CTGV) (1992) Emerging Technologies, ISD, and Learning Environments: critical Perspectives. *Educational Technology Research and Development*, **40**(1), 65-80.

Cohen, L. & Manion, L. (1989) *Research Methods in Education*. London: Routledge.

Duffy, M. T., Lowyck, J. & Jonassen, H. D. (1993). *Designing environments for constructive learning,* Berlin: Springer-Verlag.

Grigoriadou, M., Toula, M. & Kanidis, E. (2003) Design and Evaluation of a Cache Memory Simulation Program, *3rd IEEE International Conference on Advanced Learning Technologies*, Athens, July,9-11, pp. 170-174.

Ellis, A. (1998) Development and Use of Multimedia and Internet Resources for a Problem Based Environment. Proceedings of the *3rd Conference on Integrating Technology into Computer Science Education and at 6th Annual Conference on the Teaching of Computing*, Ireland. pp. 269.

Jonassen, D. H. (1991) Objectivism versus constructivism: do we need a new philosophical paradigm? *Journal of Educational Research*, 39 (3), 5-14.

Jonassen, D. H., Carr, C. & Yueh, H-P. (1998). Computers as Mindtools for Engaging Learners in Critical Thinking. *Tech Trends*, **43**(2), 24-32.

Hagan, D. & Sheard, J. (1998) The value of Discussion classes for Teaching Introductory Programming. Proceedings of the *3rd Conference on Integrating Technology into Computer Science Education and at 6th Annual Conference on the Teaching of Computing*. Ireland. pp. 108-111

Harel, I. (1991) *Children designers: Inder-disciplinary constructions for learning and knowing mathematics in a computer rich school*. Norwood, NJ: Ablex Publishing.

Kafai, Y., B. (1996) Learning design by making games: Children's development of design strategies in the creation of a complex computational artifact. In Y. B. Kafai & M. Resnick (Eds), *Constructionism in practice*. Mahwah, NJ:Lawrence Erlbaum.

Kordaki, M. (2001) Special characteristics of Computer Science; effects on Teaching and Learning; Views of Teachers. *8th Panellenic Conference of Greek Computer Society*, Nicosia, Cyprus. pp. 492 – 499

Kordaki, M., & Potari, D. (2002) The effect of area measurement tools on children's strategies : the role of a computer microworld. *International Journal of Computers in Mathematical Learning*, **7**(1), 1-36.

Kordaki, M. (2003) Prospective Computer Engineers as Users, Designers and Evaluators of Educational Software. In Proceedings of *6th International Conference of Computer Based Learning in Science*. Cyprus, Nicosia, July, pp. 553 – 563.

Laborde, J-M., and Strasser, R. (1990) Cabri-Geometre: A microworld of geometry for guided discovery learning. *ZDM*, **5**, 171-177.

Leahrer, R., Ericson, J., & Connell, T. (1994) Learning by designing Hypermedia documents. *Computers in the Schools*, **10**(1/2), 227-254.

Liu, M, & Hsiao, Y-P. (2002) Middle School Students as Multimedia Designers: A Project Based Learning Approach. *Journal of Interactive Learning Research*, **13**(4), 311-337.

Naps, T., Hundhausen & Swander, B. (1993) *GAIGS*, User Manual, v.3.0. Lawrence Center Publications, Appleton,WI.

Neo, M. & Neo, K. (2002) Building a Constructivist Learning Environment Using a Multimedia Design Project – a Malaysian Experience. *Journal of Educational Multimedia and Hypermedia*, **11**(2), 141-153.

Noss, R., & Hoyles, C. (1996) *Windows on mathematical meanings: Learning Cultures and Computers*. Dordrecht : Kluwer Academic Publishers.

Papert, S. (1980) *Mindstorms: Pupils, Computers, and Powerful Ideas*. New York: Basic Books.

Spoehr, K., T. (1993) Profiles of hypermedia authors: How students learn by doing. Paper presented at the *annual meeting of the American Educational Research Association*, Atlanta,GA.

Stasko, J. T (1990) Tango: A framework and System for Algorithm Animation. *IEEE Computer*, **23**(9), pp. 27-39.

Thomas, J., W. (2000) *A Review of research on project-based learning*. Retrieved from http:www3.autodesk.com/adsk/index/o,327082-123112,00.html

Tsonis, G., Katis, A., Palianopoulos, Y. & Kordaki, M., (2002) A multi-representational environment for the learning of basic algorithmic structures. Proceedings of Panhellenic conference with international participation '*Informatics in Education'* (pp. 259-266) Rhodos, Greece, September, 2002.

Venakis, P., Giannakopoulos, Y., Pirli, M. & Kordaki, M. (2002) A web-based multi-representational environment for the learning of files and of peripheral storage devices. Proceedings of Panhellenic conference with international participation '*Informatics in Education'* (pp. 624-631) Rhodos, Greece, September, 2002.

Vlahogiannis, G., Kekatos, V., Miatidis, M., Misedakis, J., Kordaki, M. & Houstis, E. (2002) A multi-representational environment for the learning of the bubble sorting algorithm. Proceedings of Panhellenic conference with international participation "*New Technologies in Education and in Distance Education'*, (pp. 481-495) Rehymnon, Greece, June, 2002.

von Glasersfeld, E. (1987) Learning as a constructive activity. In C. Janvier (Eds), *Problems of representation in teaching and learning of mathematics* (pp. 3-18). London: Lawrence Erlbaum associates.

Vygotsky, L. (1978) *Mind in Society*. Cambridge: Harvard University Press.