PROSPECTIVE COMPUTER ENGINEERS AS USERS, DESIGNERS AND EVALUATORS OF EDUCATIONAL SOFTWARE

Maria Kordaki

ABSTRACT
This study focuses on the development of prospective computer engineers' conceptions about the design of educational software. More specifically, we investigate how the experiences of such engineers in the design, implementation and evaluation of educational software helped them to enhance their knowledge. These engineers were involved in constructing educational software dealing with concepts of Computer Science by also taking into account theoretical educational considerations regarding constructivism and social views about knowledge construction. The analysis of the data shows that prospective computer engineers frequently start by considering the design of educational software as an "easy" or "soft" task. This attitude is typically based both on these engineers' expertise regarding software design as well as the specific subject matter. Despite these initial perceptions, they have difficulties interpreting the theoretical educational framework in design specifications. Communication between the members of each design team and between the members of all the teams as well as the reflection on the early versions of the software were significant factors that helped these engineers improve the quality of software specifications. Finally, the evaluation process helped these engineers to realize that the learners' demands are crucial in the design of educational software. They also realized that in contrast to the general principle of software design that 'customers' specifications must not be violated', these specifications were frequently ignored in the case of educational software where learners are viewed (by the engineers) as possessing the knowledge of the designer.

KEY WORDS
Design, open problem solving environments, computer science, prospective engineers, constructivism

INTRODUCTION

Open problem solving computer environments (OPSCE) can play a crucial role in pupils' thinking, encouraging them to express their knowledge as well as to explore the knowledge of others [26], [23], [25]. Pupils can also express their inter- and intra-individual learning differences by selecting among the provided tools the ones that are most appropriate for their cognitive development [22]. Apart from all the facts mentioned above, the design of OPSCE is not an easy task.
Designers of OPSCE are inspired by constructivist and social views of learning [26], [9], [25], [12]. Interdisciplinary groups have to be formed to perform this design consisting of educational specialists, computer professionals as well as experts of the specific learning subject matter.

Constructivist design emphasizes learning as an active subjective and constructive activity. Moreover, social considerations of knowledge, stress the role of computer tools as cognitive tools in the learning process [30], [24], [15]. The role of complex, authentic real life activities is also acknowledged in motivating pupils in their learning process [14], [10], [20]. Computers are an exceptional and excellent medium for constructivist learning [25], [15], [1]. More specifically, computers are the only medium that can be expressive, giving pupils the challenge of intrinsic and visual feedback of their actions, helping them to take control of their learning. Computers 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 [25]. By experimenting with computational objects, pupils can smoothly surpass the gap between the concrete and the abstract. The above render evident the value of OPSCE.

A number of well known examples of such environments have been reported e.g. Cabri-Geometry II for the learning of concepts of Geometry, Interactive Physics for the learning of concepts of Science, the programming language Logo for the learning of concepts both of Geometry and of programming. A number of problem solving environments regarding the learning of Computer Science and Engineering concepts by higher education students is also reported. Despite this fact, OPSCE for the learning of Computer Science Education concepts by secondary or primary level education pupils are rarely encountered.

In today’s digital world, Computer Science concepts are appreciated because they are essential in the curricula of both secondary and primary levels of education [2], [3], [5]. Computer Science concepts are part of the education of all secondary level pupils in Greece while more emphasis is put on dedicated concepts in specific classes preparing pupils to acquire specific skills regarding computers. There are also some important attempts being developed 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 Computer Science concepts, as well as users, designers, developers and evaluators of OPSCE. The need to prepare these engineers to play this significant role is obvious.

In this study a constructivist learning environment has been designed to present prospective computer engineers (PCE) with opportunities to actively learn how to design and evaluate constructivist OPSCE. In this environment PCE are put in the position of designers of a learning environment. This environment is presented in the following section of the paper. Despite the fact that a number of such environments have been reported, these are not specifically designed to educate PCE. In the next section of this paper, the progress of these engineers through the use, 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 the design of educational software are discussed. Conclusions are also presented.

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 [29]. Examples of such contexts include project-based authentic tasks with real-world relevance [7]. 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 [28]. 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. In facing 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 become autonomous for their decisions 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 [4], [16].

Engaging prospective engineers in the design of OPSCE is a type of project based learning. Learners are encouraged to develop many higher-order thinking skills when they become involved in projects regarding the design of educational software. 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 [8]. 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 for fractions and Logo programming in their attempts to design educational software for the learning of the concept of fractions using Logo language [17]. Students also reached a high level of reflection beyond the traditional school thinking while they tried to design games using Logo language [18]. In addition, students acquired some design skills such as mental effort and involvement, interest, planning, collaboration and individualization through the design of hypermedia projects [27], [24].

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 of educational software. This environment was designed for PCE and is offered to them as an elective course on their typical 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 is lasts for at least 12 weeks, while the course consists of 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 secondary and primary levels education pupils.

This study took place during two semesters (the spring and the fall) of 2001. Fifteen prospective engineers participated in the course mentioned above. The researcher participated as a teacher of this course working closely with these PCE throughout the experiment.

THE CONSTRUCTIVIST LEARNING ENVIRONMENT

Unlike a traditional teacher-telling classroom, this class exploited PCE previous knowledge about learning and teaching as well as about the design and evaluation of educational software. More specifically 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 to present this knowledge to the whole class. This was followed by a class discussion; furthermore, the map of the opinions expressed by the whole class members, in the form of a hierarchical network was constructed on the blackboard by the researcher. At the beginning of each semester PCE were asked about their expectations regarding each course and were explained its outline and the learning approach used.

The organization of the spring semester consisted of two phases. Phase I (approximately six weeks) was devoted to discussions and negotiations regarding the driving question namely: what is effective learning for computer science and engineering concepts?. The goal of this phase 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 the learning process, problem solving activities as helping learners develop higher cognitive skills, computer tools as cognitive tools, and collaboration among all participants in each specific learning community. The need for the learning of computer science concepts by primary and secondary level education students was also disscused [3]. The discipline of computer science and engineering was also discussed simultaneously as: a theory, a science and engineering, emphasizing the role of solving real problems by teams [4], [13], [16], [19]. The role of finding out the most optimal solutions by taking into account the time, the cost and the efficacy of the products was also discussed [4]. Phase II (approximately six weeks) focused on working in groups to create an environment for the learning of computer science concepts for secondary level education pupils. Each group selected a topic regarding computer science concepts. Prospective engineers 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 stage.

During the design stage PCE were engaged in the construction of three models: firstly, the model of the specific learning subject matter, secondly, the model of learning and thirdly, the model of possible pupils' actions in order to learn this specific learning subject. To construct the first model, each group of PCE had to clarify which were the essential concepts of the specific learning topic as well as to analyze these concepts in elementary concepts. To realize the second model, they had to make explicit, basic aspects of constructivism and social views of learning in the form of specific teaching-strategies. To form the third model, they had to take into account possible pupils' behavior regarding the learning of the specific topic.

During the experimenting 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 consisted also of two phases. Phase I (approximately six weeks) was devoted to discussions and negotiations regarding the driving question: what kind of knowledge is implied in the design of constructivist OPSCE?. The goal of this phase was to clarify basic design principles of traditional, and of constructivist computer learning environments. During this phase PCE were asked to use and to criticise reported examles of constructivist learning environments namely Cabri-Geometry II, the Logo programming language, Interactive Physics, and the C.AR.ME microworld [20]. Here, the design of educational software was discussed as a process of modeling. More specifically, this process was analyzed in sub-processes including the construction of three models: firstly, the model of the specific subject mater to be learned by the pupils, secondly the model of learning, and thirdly the model of pupils' possible behavior in their learning of the subject matter. Apart from the fact that the construction of these models had already been disscussed in the spring semester, here these models were disscused in terms of specifications of computer tools. Moreover, the differentiation of the whole learning context as a result of introducing the computer as a cognitive tool in 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. Phase II (approximately six 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 and the same topic of the subject matter was assigned. Prospective engineers followed a five stage developed model: a) designing the learning environment, b) imlementing it using a specific language, c) evaluating this environment by presenting it in 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 the construction of the three previously mentioned models which were presented then to the class of their colleagues. Receiving feedback from the members of this class as well as the researcher, they had the opportunity to revise these initial models. The phase of evaluation 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. During this phase 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 phase 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 real pupils to try out the produced software. The collected data was classified into categories to give a picture of the learning that occurred within the context of each piece of these educational software.

DATA SOURCES AND ANALYSIS

Qualitative methodology [11] was used to illuminate the effect of the previously mentioned constructivism course on the knowledge of prospective computer science engineers' regarding the design of educational software. More specifically, observations and reflections by the researcher were made during all phases that took place during the two semester course. In addition, 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 referring to all phases that took place during the two semester course, b) the audio tapes including the conducted interviews. The data were transcribed, chunked and coded using themes that emerged from them. 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 [6].

RESULTS

The most important data findings are summarized as following:
Prospective engineers' previous knowledge about teaching and learning. Here, PCE emphasized the role of quality presentation of content by the teacher as well as the role of assessing pupils through exercises or drill and practice activities.

The role of appropriate examples and questions in understanding constructivist and social views of learning. Despite the fact that university teaching usually emphasizes lectures or presentations of the subject matter, PCE found this tedious and meaningless as regards the theories of teaching and learning. 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. Unfortunately, my first teaching attempt was to engage these learners in theoretical discussions and negotiations regarding constructivism and traditional theories of learning. As, I felt that the atmosphere of the class was static and boring as well as the fact that the learners became tired and non interested, I changed my teacher –telling/asking- approach. Therefore, I decided to teach constructivism by putting its implications immediately into my own teaching practice. As a result, PCE were put in groups and were asked to cooperate 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. I tried to facilitate this negotiation by asking questions as well as by expressing my own knowledge.

The development of prospective engineers' conceptions about teaching and learning. During the first semester five projects were assigned focusing on the following topics: a) multimedia b) bubble sorting algorithm c) basic algorithmic structures, d) files and peripheral storage devices e) I/O devices. To realize these projects, PCE engineers worked in groups of three. Each group designed a learning environment regarding the topics above and tried it out on their colleagues in the prospective engineers' classroom. Their 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'. Therefore, more time was needed to put these engineers into an iteration cycle of teaching and receiving feedback in order to put basic aspects of constructivism into real practice.

Motivation. Prospective engineers' were strongly engaged in the design, implementation and evaluation of the educational software that they were responsible for constructing. Three main factors motivated them : 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 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'.

Prospective engineers' as users of educational software. Here, as well, specific pieces of educational software (Cabri-Geometry II, the Logo programming language, Interactive Physics and the C.AR.ME microworld) were demonstrated in front of the prospective engineers' class. Then, they were called on to use these software and to make comments. These comments mainly emphasized the technical knowledge needed for the implementation of these software characterizing it as an 'easy' or 'soft' task. Questions such as 'which are the possible basic design principles implied in these software?' and 'what can pupils learn by interacting in these software environments?' were opaque for these engineers and difficult to address.

Prospective engineers' as designers of educational software. During the fall semester PCE had the task of extending the learning environments that they designed in the spring semester in the form of educational software. Their first design attempts emphasized the role of: a) Questions 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 and pupils have the task to learn it'. Moreover, the implied 'learning model' ignored the confrontation with pupils' inter- and intra-individual differences as well as the need for active and constructive participation of pupils in their learning. In 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 in actively constructing their knowledge and c) the role of appropriate tools to provide 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 that can help pupils be in cognitive conflict and correct their mistakes. By taking into account all these interventions mentioned above, prospective engineers' tried to create new versions of their piece of software. Then, they rejoined this 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.

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 among their colleagues the associated responsibilities b) in constructing 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. 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 role of presenting their pieces of educational software in front of the class of their colleagues including the researcher. As PCE stated 'we helped to clarify the design of our work while we were 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 emphasize 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.

Prospective computer engineers' as evaluators of educational software. Prospective engineers tried to evaluate pilotically with real pupils the pieces of software they 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'.

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, and Flash to implement the OPSCE problem- solving environments they designed. The knowledge of these software packages is also very useful for the career of a computer engineer in industry.

The learning outcomes. Five pieces of educational software were produced during this two-semester experiment. Three of them were accepted, presented and published in proceedings of typical conferences regarding Information and Communication Technologies in Education. These publications are : a) 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. b) 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. c) Vlahogiannis, G., Kekatos, V., Miatidis, M., Misedakis, J., Kordaki, M. & Houstis, E. (2001). 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, 2001.

DISCUSSION AND CONCLUSIONS

Prospective computer engineers participated in a specifically designed constructivist learning environment from the position of designer of OPSCE. In this 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 about teaching and learning as well as about the design of computer learning environments emphasized traditional behavioristic views. Prospective engineers relied on their expertise about the learning subject as well as about the typical software design. Moreover, they viewed the cognitive gap between their knowledge and the knowledge of learners as a gap that had to be exclusively filled by the learners attempts. Prospective engineers started to shift from this authoritarian position by having the opportunity to participate actively as learners in the constructivist environment that is presented in this study. By having a real-world 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 too. 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 prospective engineers had already been taught specific knowledge about Human computer interaction as well as about typical software design, they couldn't 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 have 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.

REFERENCES

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

ACM (1997a). 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 theACM/IEEE-CS Joint Curriculum Task Force. 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.

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, (269) Ireland.

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, (108-111). Ireland.

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 cration 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.

Kordaki, M., & Potari, D. (1998a). Children's Approaches to Area Measurement through Different Contexts. Journal of Mathematical Behavior, 17(3), 303-316.

Kordaki, M., & Potari, D. (1998b). A learning environment for the conservation of area and its measurement: a computer microworld. Computers and Education, 31, 405-422.

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.

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.

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.

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

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.

Maria Kordaki
Department of Computer Engineering and Informatics
University of Patras & CTI
26500, Rion, Patras, Greece
Email : kordaki@cti.gr