Starting with Parallel Processing: A multi-representational learning environment for beginners

Maria Kordaki[1] and Yannis Solos[2]

[1]Dept of Computer Engineering and Informatics University of Patras
[2]Dept of Informatics, School of Sciences and Technology, Hellenic Open University
e-mail: kordaki@cti.gr, SolosJohn@yahoo.gr

Abstract: This paper presents a multi-representational computer-based environment, the Starting with Parallel Processing (SPAP) environment, designed for learning the concept of parallel processing to beginners through the possibility of their both performing and exploring serial and parallel sorting procedures. In designing this environment, social and constructivist learning theories have been taken into account. By interacting within SPAP, learners have opportunities to express their previous knowledge of serial and parallel sorting through a step-by-step procedure and receiving constructive feedback. Moreover, learners have the chance to actively explore step-by-step serial and parallel sorting automatically performed by the system. Students have the chance to observe these sorting procedures step-by-step in Multiple Representation Systems. SPAP was piloted with a class of 31 actual first grade students in secondary school. The analysis of the data shows that all these students recognized that sorting using parallel comparators is faster than when serial comparators are used.

Keywords: Educational software, multiple representations, parallel processing, secondary education

Introduction

Nowadays, the impact of Computer Science (CS) throughout our lives is obvious. Indeed, computers have already had an enormous impact on the way we live, think and act. To this end, there is a pressing need to give the rapid growth of computing and technology in the modern world academic coherence, alongside the need for an educated public that can utilize that technology most effectively for the benefit of humankind (ACM, 2003). That is to say, in order to function in society, the average citizen in the 21st century has to understand at least the principles of computer science. In fact, Computer Science is a mainstream discipline that can no longer be ignored by public schools in the 21st century (ACM, 2003). In particular, much evidence (National Research Council, 1999) confirms an urgent need to improve the level of public understanding of computer science as an academic and professional field, including its distinctions from the other sciences. It is acknowledged that both elementary and secondary schools have a unique opportunity and responsibility to address this need.

Bearing the above in mind, the conceptualization of CS concepts at all levels of education is clearly essential. To this end, a four-level model curriculum of CS for K-12 has been proposed (ACM, 2003). One of the four goals of this curriculum is to introduce the fundamental concepts of CS to all students beginning at primary level. As regards the learning of CS concepts, it is clear that whatever is achieved in high school depends upon the effectiveness of student access to technology and achievement of computer-related learning milestones at primary level. Level I of the afore-mentioned curriculum (recommended for grades K-8) is geared to provide primary and secondary school students with fundamental CS concepts by integrating basic technology skills with simple algorithmic thinking. Within this framework, understanding of the concept of parallel processing is viewed as being critical (Bell, Witten, and Fellows 2002). It is also worth noting that both students in secondary education and adults have difficulties in understanding how a computer works (Soloway & Spohrer, 1989).

The active role of the student in contexts emphasizing the understanding of algorithmic logic through simple ideas and pleasure activities has been acknowledged as being essential for the conceptualization of basic CS concepts and skills (Papert, 1991; Bell, Witten, and Fellows 2002). Within this context, the understanding of various CS concepts such as binary encoding, debugging, cartography and parallel processing, are viewed as being not only essential for primary and secondary level students to grasp but also achievable, especially through performing simple fun activities (Bell, Witten, and Fellows 2002).

It is possible for young students to grasp the concept of parallel processing as it has meaning and value, not only in reducing the time needed to solve a problem using computers but also in reducing the time necessary to do a variety of everyday real life activities. For example, a meal can be prepared faster when its parts are prepared simultaneously than when the preparation follows a serial process (naturally, this applies only when the completion of one part of the meal does not require the completion of another part of it); a building can be constructed faster when a number of constructors work simultaneously on both the same and different jobs than when the individual jobs are performed sequentially. Of course, a variety of jobs cannot be performed simultaneously, for example, taking a bath and dressing, reading and sleeping etc. For young students to be able to conceptualize parallel processing, some simple fun activities focusing on students' active participation in sorting using serial and parallel sorting networks have been proposed (Bell, Witten, and Fellows 2002). Such activities have been tested in the field using real students with positive results, in terms of both the development of strong motivation for their engagement in the activity and also their ability to discriminate between parallel and serial processing (Kordaki and Mastrogiannis, 2007).

The catalytic role of appropriately-designed educational software on student leaning has been acknowledged by many researchers (Noss, & Hoyles, 1996; Duffy and Cunningham, 1996; Jonassen, 1996; 1999). In particular, the role of constructivist educational software providing students with Multiple Representation Systems (MRS) to express their inter-and intra-individual learning differences, in the learning of all subjects but especially CS, has been acknowledged (Dyfour - Janvier, Bednarz, & Belanger,1987; Kaput, 1994; Ainsworth, 1999; Kordaki, 2005). MRS can be used to provide opportunities for the students to select those most appropriate to actively express their knowledge and also as scaffolding elements that provide opportunities for exploration of the knowledge of others, especially in the cases of 'cognitive opaque' concepts and procedures for students. Simulations and MRS-based educational software are acknowledged as appropriate for the learning of engineering and CS concepts in primary and secondary levels of education (Kordaki, Miatidis, and Kapsampelis, 2005; Davidovitch, Parush and Shtub, 2006).

Despite the above, there is a lack of educational software for beginners to learn about the concept of parallel processing through serial and parallel sorting. Taking into account all the above and in our attempt to support primary and secondary level education students in their understanding of the concept of parallel processing, we constructed a multi-representational, constructivist learning environment. This environment provides the students with: a) MRS supporting their exploration of automatically performed but self-controlled simulations of serial and parallel sorting procedures, b) MRS supporting the step-by-step expression of their intuitive knowledge of serial and parallel sorting, and c) step-by-step appropriate feedback. Such an environment has not yet been reported.
In the next section of this paper, the design and the specific features of the proposed educational software are presented. Next, a pilot evaluation study of this software with real students is reported. Subsequently, the data from this pilot study are discussed and finally, conclusions and proposals for future research are drawn.

The design and the features of SPAP

Both social and constructivist learning theories were taken into account in the design of SPAP (Jonassen, 1999; Hannafin & Land, 1997; Duffy and Cunningham, 1996; Vygotsky, 1978). Constructivist learning theories emphasize the active, subjective and constructive character of knowledge, placing students at the centre of the learning process. Social theories of learning stress the importance of psychological tools as mediators for the development of students' higher mental functions (Vygotsky, as cited in Wertch, 1995). In particular, computer tools and learning environments can help students to express their own knowledge, to exp lore the knowledge of others integrated in these environments and to scaffold their thinking and actions in order to deepen their understanding (Bereiter and Scardamalia, 1989; Noss & Hoyles, 1996) as well as providing them with appropriate feedback. Tools can act as mediators of sociocultural development to help students enhance their Zone of Proximal Development, which is essential in the development of their knowledge (Vygotsky, 1978; Salomon, Globerson & Guterman, 1989). The mediating role of technology is based on the ability computers have to 'communicate' with learners through the variety of representations which can be constructed and displayed on their screens (Noss and Hoyles, 1996). The role of representation systems in the gaining of knowledge is acknowledged as significant by many researchers (Dyfour-Janvier, Bednarz, & Belanger, 1987; Kaput, 1994; Borba & Confrey, 1996; Ainsworth, 1999). In fact, it is acknowledged that MLRS can play a crucial role in student development of a broad view of the concepts in focus, as well as realization of the basic aspects and structure that constitute these concepts, translating and transforming a concept into the various RS provided and observing how variation in one RS can affect the other (Kaput 1994; Schwartz, 1995). It has been also acknowledged that MR-based educational software plays a significant role in the learning of concepts in Computer Science and Engineering (Zikouli, Kordaki and Houstis, 2003; Kordaki, 2005). Finally, the role of simulations which can be both automatically run by the system and manually handled by the students is stressed as being essential (Davidovitch, Parush, Shtub, 2006).

SPAS was constructed to provide students with opportunities to understand the advantages of parallel processing by sorting whole numbers of their choice step-by-step using serial and parallel comparators. To perform these sorting activities, serial and parallel algorithms are used. SPAP provides visualization of sorting entities, which learners can control step-by-step. For example, in serial sorting, the system performs one comparison of the sorting entities at any one step and visualizes the appropriate exc hanges of these entities. In parallel sorting, the system simultaneously performs a group of comparisons of the sorting entities at any step and also exchanges these entities properly.
Learners also have the opportunity to actively express their own approaches to serial and parallel sorting by trying to sort numbers of their choice. During this activity, students can receive immediate feedback so as to be encouraged to try again and to correct their own sorting approaches by trial and error.
Students also have the chance to study each sorting procedure expressed in MRS, such as: a) diagrammatic visual representation of the sorting entities using the sorting procedure at hand, b) appropriate description in natural language, and c) sequences of the numbers to be sorted as they are in the current stage of each step of the sorting procedure. In fact, each of these sequences represents the current status of the previously mentioned numbers during the specific sorting algorithm realization.

Sorting using serial algorithms. An example of the visualization of a serial sorting procedure of 6 numbers is presented in Figure 1. The various numbers are presented in colored frames of different colors, thus giving students the opportunity easily to observe the exchanges of numbers which are performed after each comparison. Information about the internal logic of sorting is also provided in text format. In addition, the current sequence of numbers in each step of the sorting procedure is presented in another window. Students are also provided with the opportunity actively to sort numbers of their choice using the previously mentioned serial sorting procedure and receive immediate feedback step-by-step to correct their approaches.

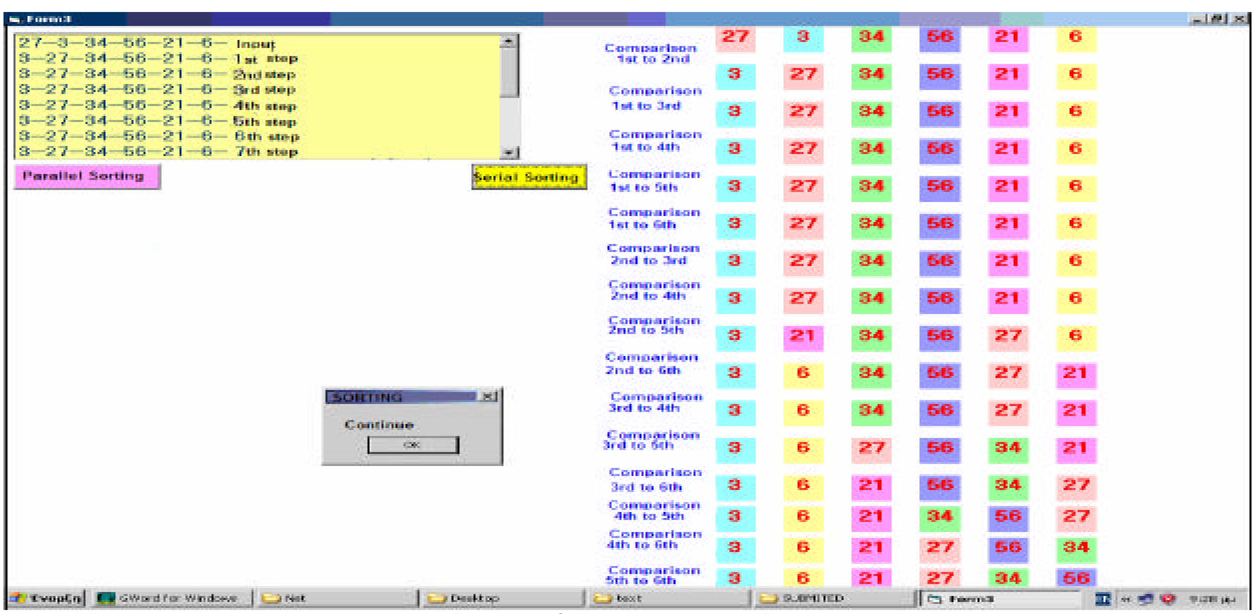

Figure 1. Visualization of sorting using serial comparators

Sorting using parallel algorithms. SPAP provides possibilities for sorting whole numbers using parallel comparators. The sorting procedure is automatically performed by the system but can be controlled by the student step–by-step. The whole sorting procedure is visualized. Figure 2 presents the sorting of 6 numbers using parallel comparators.

Students can observe the numbers compared (C= Comparison) in each node as well as all the comparisons performed in each step of the said sorting procedure. Arrows are also used to demonstrate the direction and position where each number is transferred after a specific comparison. In addition, to clarify the whole sorting procedure, the color of the frame of each number is fixed, thus giving students the opportunity easily to observe the exchanges of the numbers performed after each comparison. In the same figure, the sorting procedure used is also described, but in natural language. The current sequence of numbers in each step of the parallel sorting procedure is also presented in another window. Moreover, SPAP provides opportunities for the students to sort numbers of their choice using the previously mentioned parallel procedure and receive feedback, step-by-step, to correct their actions.

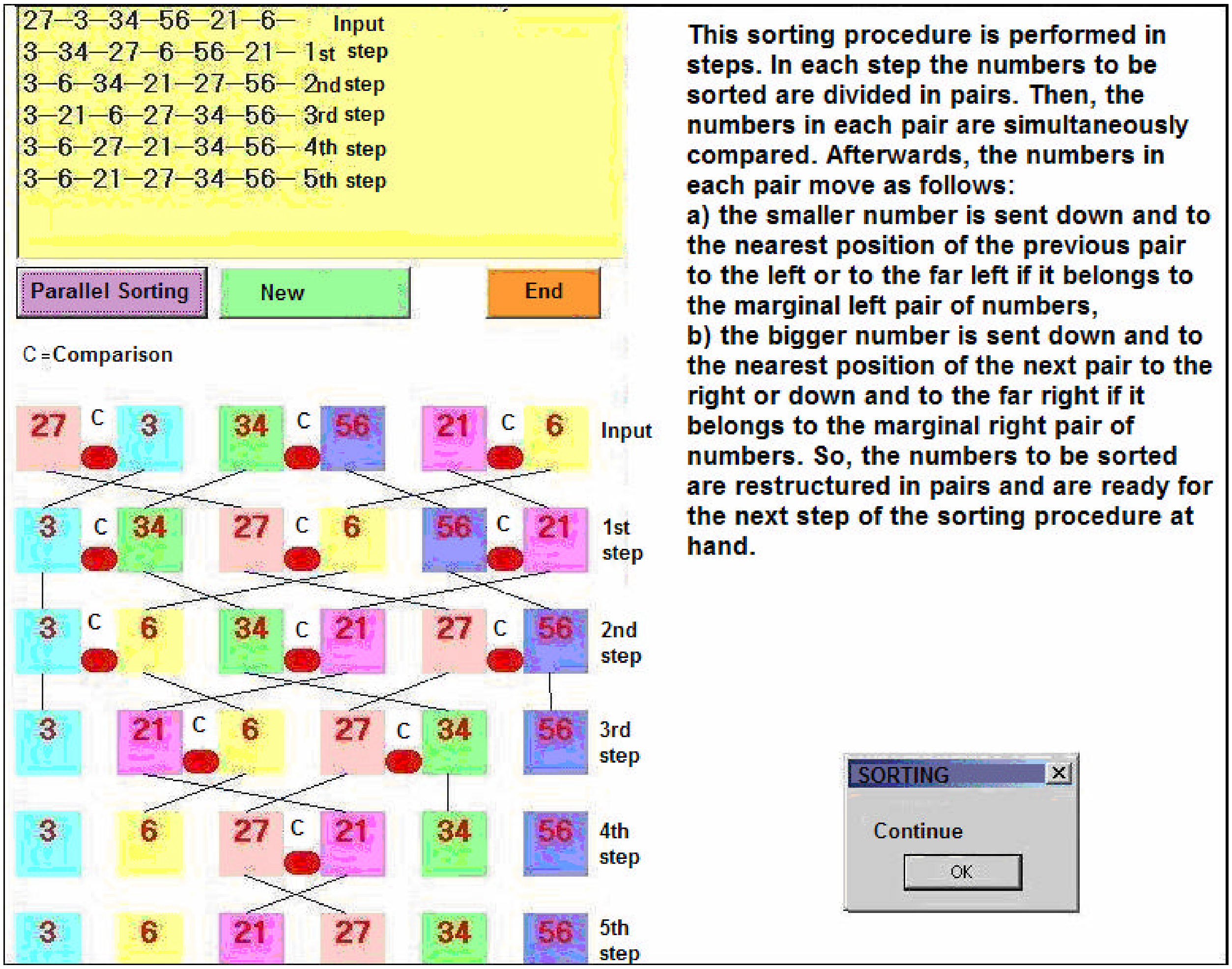

Figure2. Visualization of sorting using parallel comparators

Finally, it is worth mentioning that serial and parallel sorting are displayed in different windows; consequently, students can simultaneously study both of these procedures.

Technical characteristics of the software. Microsoft Visual Basic V.6.0 was used.

The pilot evaluation study of SPAS

SPAP was piloted in the field using actual students. From a methodological point of view, this pilot evaluation study is a qualitative research (Cohen and Manion, 1989) investigating: a) students' previous knowledge of serial and parallel processing, b) students' conceptions of serial and parallel processing as a result of their interaction within

SPAP and c) usability issues regarding this software. Specifically, a class of 31 first grade students (13-years-old) participated in a learning experiment performed in a secondary school (Gymnasium of Saint Constantine) in the region of Aitoloakarnania, Greece. The students experimented within SPAS for about 2 hours and followed their needs. The researchers undertook the role of observers in order to record everything that students said and did during this experiment and did not affect their actions. The data resources consisted of the students' written answers to the questions posed and the field notes of the researchers.

The learning experiment. Each student entering SPAP was asked to give six numbers and to automatically perform their sorting step-by-step, firstly using the serial sorting procedure (A) and secondly using the parallel sorting one (B). Next, each student was asked to write down their answers to the following questions on a sheet of paper: a) which sorting method did you like and why?, b) which sorting method do you think is faster and why? The terms "serial" and 'parallel' processing were not used in this phase of the experiment as the students were not familiar with them and so as not to affect their responses in any way. Here, it is worth noting that these students had never thought about the differences between parallel and serial realization of a procedure. Students' responses and justifications were classified in the following four categories:

? 1. *Sorting procedure (A) is faster than procedure (B) because (A) is more understandable.* This opinion was expressed by 3 students. In our view, these students provided this justification by using the sorting procedure that was faster for them to realize as a criterion.

? 2. *Sorting procedure (B) is faster than procedure (A) because in (B) many comparisons are performed simultaneously*. Here, it is worth mentioning that different justifications for this argument were expressed by the students. Some students (6 students) maintained that: 'sorting procedure (B) is faster than procedure (A) because, at the same time, in the former procedure many comparisons are performed simultaneously while in sorting procedure (A) only one comparison is performed'. Other students (9 students) contended that: 'sorting procedure (B) is faster because, at the same time, 'pairs of comparisons are performed while in sorting procedure (A) only one comparison is performed'. In addition, some students realized that: 'the number of comparisons performed during sorting procedure (B) is smaller than the number of comparisons performed during sorting procedure (A)'. In our view, students who expressed opinions that fall into this category appeared to understand that sorting realized by using parallel comparison procedures is performed faster than sorting realized by using serial ones.

? 3. *Sorting procedure (B) is faster than (A) because the visual representation of (B) on the screen of the computer is smaller than the respective representation of procedure (A)*. This approach was expressed by 4 students, who also wrote that: 'during sorting procedure (A), one comparison is performed each time'. However, these students were unable to understand the sorting procedure using parallel comparators. Therefore, these students compared these sorting approaches using the size of their visual representations on the computer screen as a criterion. This indicates that appropriate visual representations can help students to form intuitive hypotheses about the speed of the sorting procedures used.

? 4. *Sorting procedure (B) is faster than (A).* Students who expressed this opinion (9 students) were unable to form any verbal justifications, relying solely on their intuition.

On the whole, the majority of students (28 students) realized that parallel sorting (B) is faster than the serial one. About half of the students (15 students) also recognized that parallel sorting (B) is faster because, during this kind of procedure, many comparisons are performed simultaneously. In addition, the majority of students (22 students) were of the opinion that they preferred sorting procedure (B) because it is smarter, more sophisticated and more efficient than sorting procedure (A). However, these students found the parallel sorting procedure to be more difficult and more complicated than the serial sorting procedure. Despite the fact that this procedure was acknowledged by students as being easier and more understandable, they emphasized that this is a time-consuming and tedious procedure. In general, SPAP software is characterized as being attractive material by the students, who also liked its colorful and meaningful visual representations, including all the symbols used, such as nodes, arrows, number-frames etc. All students also expressed interest throughout all the steps of the proposed activity.

Usability issues. SPAP was easily used by all students during this experiment.

Concluding remarks and plans for future work

A multi-representational, simulation based, interactive educational software for beginners to learn the concept of parallel processing is presented in this paper. The design of this environment took into account social and constructivist perspectives regarding knowledge construction. This environment - Starting with Parallel Processing (SPAP) - provides learners with opportunities to conceptualize the fact that parallel processing is more advanced than serial processing through comparisons of whole numbers using serial and parallel sorting. In fact, SPAP provides learners with multiple representation systems to interact with, such as: a) diagrammatic visual representation of the sorting entities using the sorting procedure at hand, b) appropriate description of this procedure in natural language, and c) sequences of the numbers to be sorted as they are in the current stage of each step of the sorting procedure. SPAP provides students with opportunities to observe and to control serial and parallel sorting procedures in the said RS step-by-step as well as to actively express their existing knowledge of serial and parallel sorting and modify it by being given immediate feedback on their actions. SPAP was piloted with actual students. Analysis of the data provided evidence that SPAP helped all students participating in this experiment to acknowledge that parallel processing is faster than serial processing, especially in the process of sorting of whole numbers using serial and parallel comparators. It is worth mentioning that a large number of students also recognized that parallel sorting is faster than serial sorting, due to the fact that, in the former, a number of comparisons can be performed simultaneously in each step of the sorting procedure while, in the latter, only one comparison can be performed in each step. Students also found SPAP interesting and pleasant and the method of parallel sorting to be fast and smart. On the whole, SPAP seemed to be able to support secondary level education students in reaching an understanding of the significance of parallel processing in comparison to serial processing. Our future plans consist of evaluating this software with a larger sample of students and enhancing it by providing opportunities to sort a variety of entities.

References

ACM (2003). A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee.

Ainsworth, S.E. (1999). The functions of multiple representations. *Computers and Education*, 33(2-3), 131-152.

Bell, T., Witten, I. & Fellows, M. (2002). Computer Science Unplugged. http://www.unplugged.canterbury.az.nz

Bereiter, C. and Scardamalia, M. (1989). Intentional learning as a goal of instruction. In L.B. Resnick's (Ed.), *Knowing, learning and Instruction*. Hillsdale, NJ: Lawrence Erlbaum Associates, 361-391.

?orba, M., & Confrey, G. (1996). A student's construction of transformations of functions in a multi-representational environment. *Educational Studies in Mathematics,* 31, 319-337.

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

Davidovitch, L., Parush, A. and Shtub, A. (2006). Simulation-based learning in engineering education: Performance and transfer in learning project management. *Journal of Engineering Education*, 95(4), 289-300.

Dyfour-Janvier, B., Bednarz, N., & Belanger, M. (1987). Pedagogical considerations concerning the problem of representation. In C. Janvier (Eds), Problems of representation in teaching and learning of mathematics (pp. 109-122). London: Lawrence Erlbaum Associates.

Duffy, T. and Cunningham, D. (1996). Constructivism: Implications for the design and delivery of instruction. *Handbook of Research for Educational Communications and Technology*, 170-198.

Jonassen, H. D. (1996). *Computers in the Classroom*. Colombus, OH: Prentice Hall.

Jonassen, D. ? . (1999). Designing constructivist learning environments. *Instructional design theories and models*, 2, 215-239.

Hannafin, M.J & Land, S. (1997). The foundations and assumptions of technology-enhanced, student-centered learning environments. *Instructional Science*, 25, 167-202.

Kaput, J.J. (1994). ?he Representational Roles of Technology in Connecting Mathematics with Authentic Experience. In R. Biehler, R. W. Scholz, R. Strasser, B., Winkelman (Eds), *Didactics of Mathematics as a Scientific Discipline: The state of the art*. Dordrecht: Kluwer Academic Publishers, 379 - 397.

Kordaki, M. (2005). A Special Purpose E-Learning Environment: Background, Design and Evaluation. In Zongmin Ma (Ed.), *Web-Based Intelligent e-Learning Systems: Technologies and Applications*. Idea Group Publishing, pp. 348-375, (Peer Review).

Kordaki, M. Miatidis, M. and Kapsampelis, G. (2005). Multi Representation Systems in the Design of a Microworld for the Learning of Sorting Algorithms. In Proceedings of *IADIS International Conference Cognition and Exploratory Learning in Digital Age* (CELDA 2005), Porto, Portugal, 14 -16 December 2005, pp. 363-366.

Kordaki, M. and Mastrogiannis, A. (2007). The concept of parallel processing in primary education: the role of playing games. 4th Pan-Hellenic Conference of teachers for ICT: "Exploiting the features of ICT in Educational Practices". Syros, Greece, May, 4-6, 2007.

Nardi, B.A. (1996). Studying context: A comparison of activity theory, situated action models, and distributed cognition. In B.A. Nardi (Ed.), *Context and consciousness: Activity theory and human-computer interaction*, Cambridge, MA: MIT Press.

National Research Council Committee on Information Technology Literacy, *Being Fluent with Information Technology*, National Academy Press, Washington, DC, May 1999. http://www.nap.edu/catalog/6482.html

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

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

Salomon, G., Globerson, T. and Guterman, E. (1989). The computer as a zone of proximal development: Internalizing reading-related metacognitions from a reading partner. *Journal of Educational Psychology*, 81(4), 620-627.

Soloway E. Spohrer J.C. (1989). *Studying the novice programmer*. Hillside, N.J. Erlbaum.

Schwartz, D. L. (1995). The emergence of abstract representations in dyad problem solving. *The Journal of the Learning Sciences*, 4(3), 321-354.

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

Wertch, J. V. (1995). Introduction. In J. V. Wertsch (Eds), *Culture, communication, and cognition : Vygotskian perspectives*. Cambridge: Cambridge University Press, 1-20.

Zikouli, K., Kordaki, M. & Houstis, E. (2003). A Multi-representational Environment for the Learning of Programming and C. *Proceedings of 3rd International Conference on Advanced Learning Technologies 2003*. Athens, Greece, 459.