MULTI REPRESENTATION SYSTEMS IN THE DESIGN OF A MICROWORLD FOR THE LEARNING OF SORTING ALGORITHMS

Maria Kordaki
Department of Computer Engineering and Informatics,
University of Patras, 26500, Rion Patras, Greece
e-mail: kordaki@cti.gr

Micael Miatidis
Lehrstuhl Informatic V, RWTH, Aachen
e-mail : MIATIDIS@CS.RWTH-AACHEN.DE

George Kapsampelis
Department of Computer Engineering and Informatics,
University of Patras, 26500, Rion Patras, Greece
e-mail: kapsabel@ceid.upatras.gr

ABSTRACT

This paper presents the design and the features of a Web-based micro-world -the SORTING micro-world - constructed for the learning of sorting algorithms by secondary level education students. The design of this microworld has been based on modern constructivist and social theories of learning. SORTING is an interactive environment that can support learners: a) to express their own sorting procedures in multiple representation systems (MRS), b) to express their own approaches to typical sorting algorithms, c) to correct their sorting actions while actually sorting entities using typical sorting algorithms by receiving immediate feedback by the system, d) to explore typical sorting algorithms in MRS when these algorithms are automatically performed by the system, and e) to give different meanings to a sorting algorithm when it is represented in different RS so as to acquire a broader view of this algorithm. The typical sorting algorithms in focus are: Bubble-sort, Quick-sort and Selection-sort. The Representation Systems (RS) provided are: i) real object-based simulations: the learner can sort simulations of real objects by using hands-on experience, ii) free-text RS: the learner can reflect on their hands-on experience and then express their sorting approach using natural language in the form of free-text, iii) pseudo-code RS: the learner can express their sorting approach using pseudo-code, and c) flow-chart based RS. Animations are also provided to visualize the sorting of elements using the sorting algorithms mentioned above. These elements are simulations of objects taken from the real lives of learners. Moreover, interactive animations demonstrating all steps performed in sorting the said elements are provided. These animations are also interlinked with the corresponding pseudo-code and flow-charts. Thus, learners have the opportunity to explore the algorithm execution in a variety of RS. The system is also more adaptive to integrating sorting algorithms than those previously mentioned.

KEYWORDS: SORTING; EDUCATIONAL SOFTWARE; CONSTRUCTIVISM; SECONDARY EDUCATION

1. INTRODUCTION

Sorting is a significant and universal activity in all cultures; associated with financial and commercial life, it has strong social power. This power is due to the fact that sorting is based on comparisons performed on countable entities of interest and value (Bishop, 1988). Sorting is also an element of science and technology as well as of people's every day life (Knuth, 1973; Linderson & Vitter, 1985; Bishop, 1988). It is worth noting that 25% of the total active time of computers is dedicated to sorting procedures, while the time used for account-sorting in large financial institutions is about 2 hours per day (Knuth, 1975; Linderson & Vitter, 1985). The general value of sorting is based on the fact that sorted entities usually offer opportunities for optimistic solutions to a variety of problems, thus necessitating automation of the sorting process using computers and, subsequently, the invention of fast sorting algorithms such as Bubble-sort, Selection-sort, Quick sort, etc. (Knuth, 1975; Linderson & Vitter, 1985).

Despite the fact that the entities to be sorted can be in a variety of forms, such as numbers, files, objects, etc., in every case all these entities must include measurable information. The fundamental operation implied in the sorting process is the comparison between the entities to be sorted and their mutual transference.

Understanding sorting algorithms is difficult for students to grasp; they are procedures based on a number of different layers of abstraction and, therefore, their understanding is based on the use of students' higher mental functions (Matsaggouras, 1997). Moreover, when students are faced with computer-based sorting algorithms, a number of difficulties occur (Vlachogiannis, et all., 2001; Spohrer, & Soloway, 1989). Specifically, students face difficulties understanding how a specific sorting algorithm works and how this algorithm is described using basic algorithmic structures such as iteration structures, control structures and assignment statements. Regarding iteration structures, students face difficulties in understanding the iteration pattern used in the comparison procedure, how nesting loops work and what the meaning of both the variables and the counters used is. In particular, students seem to confuse the static meaning of a variable used in a mathematical equation with the dynamic meaning implied in a variable used in a computer-based algorithm. In addition, students lack the ability to initialize the counters used while usually paying no attention to initializing or terminating the whole sorting algorithm. Regarding assignment statements, students seem to ignore their dynamic character and usually use them as mathematical equations. Finally, when the use of control structures is necessary, students face difficulties in determining the kind, the value and the position of the conditionals needed.

Providing secondary level education students with opportunities to sort measurable entities using hands-on experience and then to reflect on their sorting actions is essential to their conceptualization of sorting algorithms, as in this way students can become aware of the sorting approach used (du Feu, 1999). Students also can be encouraged in their understanding of sorting algorithms by starting to express their individual sorting approaches and then gradually move to the use of typical sorting algorithms provided by the literature. The construction of each individual student sorting approach can become an essential background in supporting her/his construction of the typical sorting algorithms. Moreover, understanding of a sorting algorithm is a process of giving meaning to its representations in a variety of representation systems starting from sorting real life entities, progressing to explain it in natural language, next forming an appropriate flow-chart and finally translating the whole procedure into pseudo-code and, consequently, into a programming language (Vlachogiannis, et al. 2001).

However, sorting algorithms are rapidly introduced into the curricula of secondary level education. In fact, students are introduced directly to the Bubble-sort algorithm through its representation in pseudo-code using Pseudo Pascal. As a result, the sorting algorithm in focus becomes a meaningless procedure and, in most cases, students have no other choice but to memorize it.

Educational software can act as catalyst 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). The design of such software is usually inspired by modern constructivist and social theories of learning (von Glasersfeld, 1987; Vygotsky 1978; Noss & Hoyles, 1996; Jonassen, 2001). The said theories emphasize learning as an active, subjective and constructive activity placed within a rich context that is also meaningful for the learners, including: computer tools, authentic tasks, experimentation using hands-on experience and the use of Multiple and Linked Representation Systems (MLRS), (von Glasersfeld, 1987; Bishop, 1988; Jonassen, 2001). The essential role of educational software provision with MLRS in student learning has been acknowledged by many researchers (Kaput, 1994; Kordaki & Potari, 1998; Borba & Confrey, 1996; Kordaki, 2005). Using MLRS, students have the opportunity to observe how the variation in one representation system (RS) can affect the other RS provided. In this way, students can explore different views of the concepts in focus and have the opportunity to put them in a broader context. Using MLRS, students also can make connections between different kinds of knowledge they possess, such as intuitive, visual and symbolic knowledge (Kordaki & Potari, 1998; Borba & Confrey, 1996). The role of MR-based educational software in the learning of concepts regarding Computer Science and Engineering is also acknowledged (Vlachogiannis et all. 2001; Zikouli, Kordaki and Houstis, 2003; Kordaki, in press).

A number of animation materials for the learning of sorting algorithms, including educational software (ES), have been reported. Some well-known examples are: a) 'Sorting out Sorting' (Baecker, 1981). This is based on video presentations, introducing the idea of representing data elements in a sort by bars with length of the bar corresponding to the sort key value of the element. These bars are presented in a sort using 9 sorting algorithms, b) 'Balsa' ES (Brown, 1980). This is also based on visualization of sorting entities but a variety of sorting algorithms can be used simultaneously, c) Xtango ES (Stasko, 1990), which also supports visualization of sorting entities using a plethora of sorting algorithms. In addition, users can interact with these visualizations (and stop the procedure at any time) as well as create and run their own sorting algorithms, d) 'GAIGS', web-based ES (Naps, 1990), which supports algorithm visualization as well as text-based information about sorting, and e) Sort Animator (Dershem & Brumnund, 1998), which supports visual presentation of the sort animation which is demonstrated in relation to its code. Students are allowed to specify different colors to the sort animation, the number of elements being sorted as well as the speed of animation and its execution. In addition, the lines of code are highlighted as they are executed by the animation. All these aforementioned ES are based on presentations (in a visual or textual form) of an algorithm in execution and are mainly dedicated for higher education. Despite the above, ES for the learning of sorting algorithms by secondary level education students, allowing them to express their own intuitive sorting approaches in MRS as well as their own approaches, including their mistakes, to typical sorting algorithms, and helping them to smoothly pass from an intuitive knowledge to a more sophisticated knowledge regarding sorting algorithms, has not yet been reported.

Taking into account all the above, we constructed a web-based micro-world - the SORTING micro-world - aiming to provide students with opportunities to construct the concept of a sorting algorithm through: a) their active involvement in sorting entities using their own sorting approaches and expressing these in a variety of RS, b) their active involvement in sorting entities using typical sorting algorithms provided by the literature and expressing these algorithms in a number of RS, c) their exploration of dynamic representations (in the form of animations) of typical sorting algorithms in a number of linked RS. Such a computer learning environment has not yet been reported.

This paper is organized as follows: In the following section, the rationale of SORTING is presented. Next, the general architecture and the features of SORTING and a description of the activities that students can perform within this context are demonstrated. Finally, there is a discussion of the features of this microworld and proposals for future research work are given.

2. THE RATIONALE OF SORTING MICROWORLD

The design of SORTING was the result of a synthesis of three models; namely, the learning model, the subject matter model and the student model. In the following section, the aforementioned models are presented.

2.1. The learning model

The construction of the learning model was based on modern contructivist and social theories regarding knowledge construction (von Glasersfeld, 1990; Vygotsky, 1978). Constructivist learning theories emphasize the active, subjective and constructive character of knowledge, placing learners at the centre of the learning process (von Glasersfeld, 1990). In addition, social theories of learning stress the importance of computer tools, and more specifically the significance of computer-based learning environments in helping learners to express their own knowledge as well as to explore the knowledge of others integrated in these environments (Noss & Hoyles, 1996; Crawford, 1996a; Crawford, 1996b). Constructivist learning also stresses the need for learner experimentation by acquiring hands-on experience while using primary sources of data (www.ncrel.org/sdrs/areas/issues/content/cntareas/science/sc500.htm.). Moreover, the role of authentic real life problems is appreciated in motivating the learners to be actively involved in their learning. Furthermore, a number of researchers acknowledge the significant role of MLRS for learners in: a) the development of a broader view of the concepts in focus, as well as realization of the basic aspects and structure that constitute these concepts, by making translations and transformations of a concept in the different RS provided (Dettori & Lemut, 1995; Dyfour-Janvier, Bednarz, & Belanger, 1987; Kordaki, 2003), b) actively constructing their own knowledge by experimenting with cognitive transparent RS, c) exploring the knowledge of others by experimenting with 'cognitive opaque' dynamic RS, d) supporting their mental internal representations related to a specific learning concept (Sutherland, 1995), e) motivating them to be actively involved in their learning, especially when the RS provided are related to learner interests, and f) becoming their primary references regarding a learning concept (Dyfour-Janvier, Bednarz, & Belanger, 1987; Kaput, 1987). On the whole, understanding a concept is a process of giving meaning to its different representations as well as making connections between these different meanings.

Based on the above, an attempt has been made to interpret the aforesaid learning theories within the context of the computer, aiming at the formulation of a number of operational specifications necessary for the design of SORTING microworld. These specifications were used for the construction of the SORTING microworld and are presented below:

Table 1. The learning model: from theoretical considerations to design principles

The 'learning' model	
Theoretical considerations	**Design principles**
The active role of the learner	• Interactivity of the RS used in the environment • Availability of tools to sort a number of elements from the students' world
The subjective character of learning	Availability of tools to help students: • express their own knowledge by selecting among a variety of RS the most appropriate for their cognitive development • sort elements according to their own sorting procedures • sort elements according to their views (including mistakes) about typical sorting algorithms
The constructive character of learning	Availability of tools: • to explore the knowledge of others included in dynamic RS e.g. visual animations of a sorting algorithm, dynamic flow-chart and dynamic pseudo-code RS • to sort elements using different sorting algorithms • to sort elements using hands-on experience • to give appropriate feedback on learner actions for self-correction

2.2. The subject- matter model

This model has been constructed by taking into account the knowledge needed for the construction of a sorting algorithm and more specifically the basic sorting algorithms such as Bubble-sort, Selection-sort and Quick-sort. Bubble-sort is proposed by the secondary school curricula in Greece, while the Selection-sort and Quick-sort algorithms have been selected to give students opportunities to compare the speed of these algorithms as well as to develop a broader view regarding sorting algorithms. The basic parts of each specific algorithm in focus are: a) the initialization and the termination of the whole sorting procedure, b) the iteration pattern used in the comparison procedure, c) the kind of conditionals used, d) the assignment statements used, and d) the variables and the counters used. The operational specifications aim to provide different kinds of both information and actions needed to acquire all the aforementioned knowledge. The different forms of information are as follows:

- text-based information, explaining how each sorting algorithm in focus works
- visual information, in the form of an analytic simulation of a sorting procedure where visual objects are sorted automatically using a typical sorting algorithm
- pseudo-code, indicating how each typical sorting algorithm can be described in a language compatible with computers
- visual information, in the form of a flow-chart indicating how the sorting algorithm in focus runs
- inter-linked information. Linking visual analytic simulation of the algorithm used with the corresponded flow-chart and pseudo-code.

2.3 The student model

The construction of this model was based on the difficulties of secondary level education students (6th grade, 18 year old) regarding the Bubble sort algorithm that emerged from a field study and which were reported in the 'Introduction' section of this paper (Vlachogiannis, et all. 2001). In this study a primary version of this software regarding the Bubble sort algorithm was used. The data emerging from this study indicated that, in order to understand sorting algorithms, secondary level education students needed to: a) sort entities using hands-on experience and receive immediate feed-back, b) explore analytic animations of the sorting algorithms in focus automatically performed, and c) experience practical sorting in combination with visual animations of sorting entities, dynamic visual flow charts and the correspondent dynamic visual pseudo-code.

Table 2. The learner model of sorting: from theoretical considerations to design principles

The learner model in performing essential tasks	
The task of sorting using student-based sorting procedures	Tools to sort elements: • by using hands-on experience Tools to interpret the hands-on experience in: • free-text • pseudo-code
The task of sorting using typical sorting algorithms	Tools to sort elements: • automatically • by using hands-on experience Tools to interpret the hands-on experience in: • free-text • pseudo-code Tools to explore: • the pattern of iteration used in the comparison procedure, nested loops, conditionals in terms of their kind, value, and their position, variables, counters and assignment statements, the initialization and termination of the whole sorting procedure

3. THE FEATURES OF SORTING MICROWORLD

The SORTING microworld is organized in HTML pages. The features of SORTING are presented in its main page and are classified into three parts: a) information, b) work place, and c) animations.

a) Information: General information about sorting as well as specific information for the algorithms in focus is provided in the form of text to the learners.

b) Work place. Students are provided with four RS, in the context of which they are given the chance to express their own: a) sorting procedures and b) conceptions regarding typical sorting algorithms in focus and to receive immediate feedback from the system. The RS provided are:

- Simulated real objects (SRO).
- Text-based Representation Systems (TRS).
- Pseudo-code based Representation Systems (PSRS)
- Specific Expression based Pseudo-code Representation Systems (SEPCRS)

The above RS are presented on the same page (see Figure 1) and are described below:

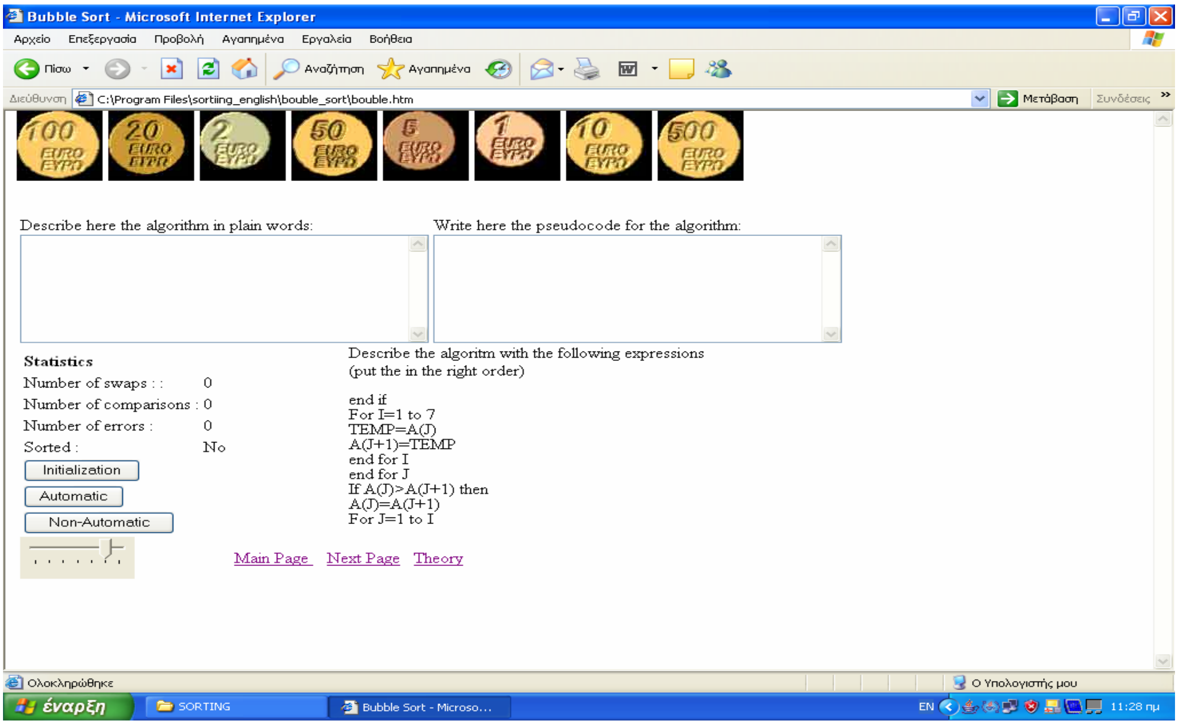
Simulated real objects (SRO). Students have the opportunity to experiment with a sorting procedure using hands-on experience. In fact, students can sort SRO familiar to them (coins in Euros), firstly by using their own intuitive sorting algorithms and, secondly, by using the sorting algorithms in focus. In this way, students have the opportunity to reflect on their sorting actions and to be aware of the sorting algorithms they used. The system can initiate the coins automatically by clicking the appropriate button. The system also counts the number of swaps, comparisons and all wrong sorting actions performed when a typical sorting algorithm is used. When a student performs a wrong sorting action, they receive an automatic response from the system. In addition, the system sends a message when the list of coins is sorted.

Text based Representation Systems (TRS). After experimentation and reflection on a specific sorting procedure or algorithm used in order to sort the provided coins, as described in the previous section, students are asked to interpret their experience using a text-based representation system. This RS has been selected to provide students with the opportunity to reflect on their sorting actions and to form an informal description about the sorting algorithm used. By trying to form this description, students have the opportunity to be more conscious about the sorting algorithm they are experimenting with.

Pseudo-code based Representation Systems (PCRS). Students are asked to translate into pseudo-code the free-text description of the sorting algorithm they used. By using pseudo-code, students have the opportunity to move from the 'I' situation of the procedure, where they describe how they performed a specific sorting procedure, to the 'You' situation, where they instruct the computer to perform this procedure using a language compatible with it. The transfer from 'I' to 'You' is difficult and not obvious for students (Zikouli, Kordaki and Houstis, 2003).

Specific Expression based Pseudo-code Representation Systems (SEPCRS). While students find difficulties in forming a typical sorting algorithm using pseudo code, the system provides them with the opportunity to correct their attempts by trying to put in correct order specific expressions provided. If the result is correct the system automatically sends a message.

Figure 1. The work place element of the SORTING microworld



c) Animation. Students are provided with extra RS:

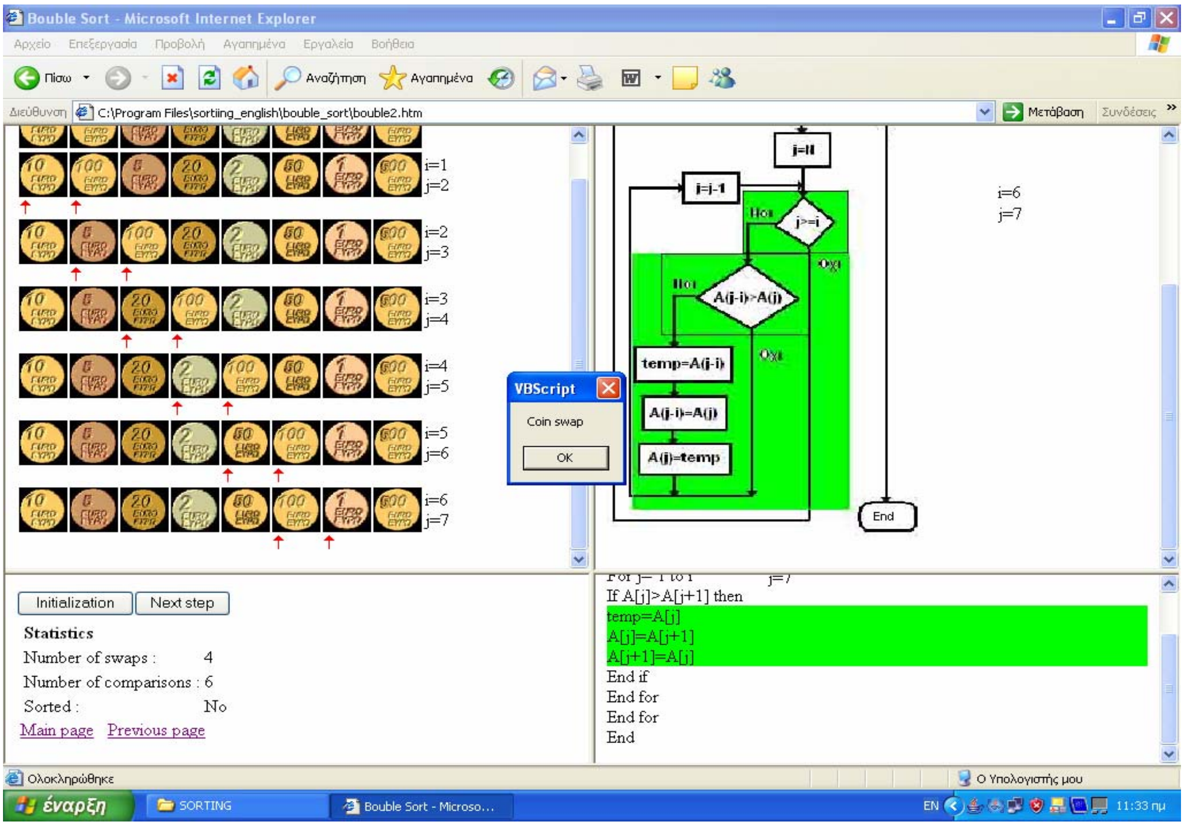
- Animation-based RS
- Flow-chart based dynamic RS
- Pseudo-code based dynamic RS
- Multiple-Linked Animation RS

Animation-based RS. Animations visualizing the sorting procedure of elements to be sorted using the sorting algorithms in focus are also provided. These elements are bars whose length corresponds to the sort key value of the element. By observing the process of sorting these bars using different sorting algorithms, learners can acquire a primary sense for each of them. Students also can observe the animation for sorting the Euro-coins - described in the previous section - using the aforementioned algorithms (see Figure 1). In this way, students can reflect on these animations and correct their own individual approaches to these algorithms.

Multiple-Linked Animation RS. Interactive animations demonstrating step by step the visual history of sorting the Euro-coins using each of the algorithms in focus are also provided (see Figuere 2). Students can specify the speed of animation In addition, these animations are interlinked with the *pseudo-code RS* as well as with *flow-chart based RS*. As the sorting of coins progresses, the corresponding part of the flow-chart and pseudo-code is highlighted. When a coin swap is performed, two pointers indicate the coins exchanged and a message is also displayed. The number of swaps and comparisons as well as the value of counters regarding each algorithm execution step are also presented. Thus, learners have the opportunity to explore the algorithm execution in all these RS, correct their own individual approaches to these algorithms and acquire a broader view of each.

Technical information: *The microworld has been implemented using Java applets and Visual C++.*

Figure 2. Multiple ad Linked Animation RS integrated within the SORTING microworld



4. CONCLUDING REMARKS AND FUTURE WORK

The design and specific features of a computer microworld for the learning of sorting algorithms by secondary level education students has been presented in this paper. The design of this microworld has been based on a modelling process consisting of the construction of three models: a) the learning model, based on modern constructivist and social views of learning, b) the subject matter model, based on the information provided by the literature regarding sorting algorithms and their conceptualisation by students and c) the learner model, based on a field study regarding learner needs and difficulties while performing sorting procedures in order to grasp concepts related to sorting algorithms. In the construction of the aforementioned models, an attempt has been made to interpret theoretical principles in design specifications. An attempt has also been made to exploit the advantages of computer technology in terms of the use of computational objects and multiple and linked representation systems in the design of this microworld. The specific features provided by this microworld have not yet been reported by other researchers.

In the context of SORTING, students can study three sorting algorithms: Bubble-sort, Quick-sort and Selection sort. Students can use the basic features of SORTING to: a) sort entities using hands-on experience and receive immediate feed-back, b) explore analytic animations of the sorting algorithms in focus automatically performed, and c) experience practical sorting in combination with visual animations of sorting entities, dynamic visual flow charts and the corresponding dynamic visual pseudo-code.

It is clear that more research is needed, using the SORTING microworld in the field with real students, in order to test both its teachability and its usability. The findings emerging from such a study would be of great assistance in the re-designing of this environment so as to meet more effectively students' cognitive needs in terms of their understanding of the sorting algorithms in focus.

REFERENCES

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

Baecker, R.M. *1981*. *Sorting out Sorting*. Film, University of Toronto. M. Kaufmann, Publishers, SF, CA, USA.

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

Borba, M., & Confrey, G. 1996. A student's construction of transformations of functions in a multirepresentational environment. *Educational Studies in Mathematics,* 31, pp. 319-337.

Brown, M.H. 1988. Exploring algorithms using BALSA-II. *IEEE Computer*, Vol. 21, No. 5, pp.14-36.

Crawford, K. 1996a. Vygotskian Approaches in Human Development in the Information Era. *Educational Studies in Mathematics*, Vol. 31, pp. 43-62.

Crawford, K. 1996b. Distributed Cognition, Technology and Change. *Proceedings of the 20th Conference of PME*. Valencia, Spain Vol.1, pp.81-112.

Dershem, H.L. & Brummund, P., 1998. Tools for Web-based Sorting Animation. *SIGCSE Bulletin*, Vol. 30, No,1, pp. 222-226.

Dettori, G. & Lemut, E., 1995. External Representations in Arithmetic Problem Solving. In R. Sutherland & J. Mason (Eds), *Exploiting Mental imagery with Computers in Mathematics Education*. Springer-Verlang, Berlin, pp. 20-33.

Du Feau, C. 1999. A Sort for Statistics Lesson. *Teaching Statistics*, Vol. 21, No. 1, pp. 8-10.

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*. Lawrence erlbaum associates, London, pp. 109-122.

Jonassen, D. H., 2000. Revisiting Activity Theory as a Framework for Designing Student-Centered Learning Environments. In D.H. Jonassen & S.M. Land (Eds), *Theoretical Foundations of Learning Environments*, Lawrence Erlbaum Associates, London, pp. 89-121.

Kaput, J.J. 1994. The 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*. Kluwer Academic Publishers, Dordrecht, pp. 379 - 397.

Knuth, D.E., (1973). The Art of the Computer Programming*, Vol.3, Sorting and Search*: Addison-Wesley, Reading, Mass.

Kordaki, M. & Potari, D., 1998. A learning environment for the conservation of area and its measurement: a computer microworld. *Computers and Education*, Vol. 31, No.4, pp. 405-422.

Kordaki, M., 2005. The role of multiple representation systems in the enhancement of the learner model. *Proccedings of 3rd International Conference on Multimedia and Information and Communication Technologies in Education*. Caceres, Spain, pp.253-258.

Kordaki, M. (2005, in press). 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 Publishers.

Matsaggouras, E. 1997. *Teaching Strategies*. Gutenberg, Athens.

Naps, T.L., 1990. Algorithm visualization in Computer Science Laboratories. Proccedins of 21st SIGSE Technical Symposium on Computer Science Education. *SIGSE Bulletin.* Vol. 22, No. 1, pp.105-110.

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

Linderstrom, E.E. & Vitter, J.S., 1985. The design and analysis of bucket sort for bubble memory secondary storage. *IEEE Transactions on Computers*, Vol. C-34, pp. 218-233.

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

Spohrer, J. & Soloway, E., 1989. Novice Mistakes: Are the Folk Wisdoms Correct?. In E. Soloway & J. Spohrer eds), *Studying the Novice Programmer*. Lawrence erlbaum associates, Hillsdale, NJ.

Stasko, J.T., 1990. Tango: A framework and system for algorithm animation. *IEEE Computer*, Vol. 23, No. 9, pp. 39-44.

Vlachogiannis, G., Kekatos, V., Miatides, M., Kordaki, M. & Houstis, E., 2001. A multi-representational environment for the learning of Bubble sort. *In proccedings of Panhellenic Conference with International Participation 'New Technologies in Education and in Distance Learning'.* Rethymnon, Greece, pp. 481-495.

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

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

www.ncrel.org/sdrs/areas/issues/content/cntareas/science/sc500.htm. *Critical Issue: Providing Hands - On, Minds-On, and Authentic Learning Experiences in Science*.

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, p. 459.