METAPHOR-BASED EDUCATIONAL SOFTWARE FOR BEGINNERS' LEARNING OF OPERATING SYSTEMS

Maria Kordaki*
Adjunct Assistant Professor
**Department of Computer Engineering and Informatics, Patras University, 26500, Rion Patras, Greece*

George Strimpakos*
Prospective Computer Engineer

Zisis Charalampidis*
Prospective Computer Engineer

Thanasis Daradoumis**
Associate professor
***Department of Computer Sciences, Open University of Catalonia, Rambla Poblenou 156, 08018 Barcelona, Spain*

ABSTRACT

This paper presents the design of educational software for beginners' learning of basic concepts of Operating Systems (OS). The design of the proposed educational software is based on modern social and constructivist learning theories as well as on the essential role of metaphors and of computer tools in student learning. In particular, the architecture of the proposed software consists of two main parts: a) the information space where students are provided with opportunities to access appropriate information about OS and to study scheduling, using a simple and fundamental algorithm; the Priority Round-Robin (PRR) algorithm, and b) the experimentation space where learners can take control of their learning by forming their own set of tasks, assigning them priorities and durations and also scheduling them using the PRR algorithm. Finally, the proposed system has been designed so that to provide pupils with feedback so that they may make self corrections.

KEYWORDS

Educational software, Operating Systems, Secondary Education.

1. INTRODUCTION

Understanding and advocating of Computer Science (CS) as an essential component of a well-rounded education is a key factor in ensuring that young students have the skills needed, not just to survive, but to thrive in this increasingly technological and global society. To this end, a four-level model curriculum of CS for K-12 has been proposed (ACM, 2003). For the achievement of the goals of this curriculum, the introduction of young students to the concept of an OS is critical. At this point, it is worth noting that, various studies have reported the significance of the use of metaphors in the learning of concepts within the complex domain of CS (Carrol & Tomas, 1982; Carrol and Mack, 1999; Hsu, 2006; Borgman, 1999; Cameron, 2002). A metaphor is a kind of linguistic figure with the form "X is Y" consisting of two major parts (Carrol and Mack, 1999). These major parts include novel concepts (X = target domain), previously acquired knowledge (Y = base domain), and the relationship between the two (Wozny, 1989). The significant role of metaphors is based on the fact that - when they are used - the critical role of students' prior knowledge in acquiring new knowledge is acknowledged as well as the essential role of engaging learners in real life problems (Lakoff & Johnson, 1980; Ortony, 1979). In the context of metaphors, learners can also form more comprehensive mental models, which results in better performance when they carry out complex tasks (Hsu, 2006).

Appropriately designed computer-based learning environments can play a crucial role in student learning (Noss and Hoyles, 1996; Jonassen, 1999). In fact, computers are an ideal medium for providing unique opportunities for the design of learning situations within the framework of modern social and constructivist theories of learning (Jonassen, 1999). To this end, students could become actively engaged in their learning and also receive appropriate feedback (Hummel, 2006). In addition, computers can provide wide opportunities for the construction of appropriately designed metaphors for the learning of complex concepts. Here, it is worth noting that, the importance of using educational software in teaching OS courses is clearly depicted in surveys conducted by (Davoli & Goldweber, 2003; Anderson & Nguyen, 2005). Several attempts have also been made to build general-purpose computer system simulators that allow undergraduate students to undergo an innovative and pedagogically different experience of learning OS (Morsiani & Davoli, 1999; Goldweber, et. al, 2005; Holland, et. al., 2002; Hovemeyer, et. al., 2004). More recent work has contributed in creating instructional simulators with a high degree of performance realism and simplicity (DeRosa, et. al., 2006; Dobrilovic & Stojanov, 2006; Liu, et. al., 2007; Laverell, et. al., 2008). Despite the above, educational software for the learning of basic aspects of OS by secondary level education students through the use of metaphors has not yet been reported. In the following sections of this paper, the design and the features of the proposed software are described. Finally, conclusions and proposals for future research are drawn.

2. DESIGN OF THE PROPOSED SOFTWARE: THE RATIONALE

In the design of the proposed educational software, socio-cultural and constructivist perspectives on knowledge construction (Jonassen, 1999; Vygotsky, 1974) were taken into account. The role of tools and of digital media and learning environments as mediators for the development of students' higher mental functions has been also acknowledged (Vygotsky, 1974; Noss & Hoyles, 1996). In the context of constructivist learning, the role of exploiting a student's prior knowledge is very important in facilitating his or her learning of new concepts. To this end, the role of metaphors as mediators of students' previous knowledge in the construction of a knowledge representation for some knowledge domain has been acknowledged (Carrol and Mack, 1999). Metaphors can have the form of kernel comparison statements whose primary function in learning is to stimulate active learner-initiated thought processes. Metaphors are open-ended because open-ended comparisons stimulate these processes more than explicit and comprehensive comparisons do. Metaphors are also incomplete, but claiming to be powerful because of both; the similarities and differences between the target and the source domain (Dubinsky & Hazzan, 2003). The success of a metaphor depends on having a familiar domain to analogize from and on recognizing enough in the new domain for some correspondence to be established (Gentner, 1983). In fact, a metaphor 'X is Y' can facilitate active learning by providing them with opportunities to generate and test various hypotheses about the target novel domain (X) based on the similarities and dissimilarities it has with the base domain (Y). To this end, metaphors can be used as orienting frameworks, at the same time leaving many operational details for the learner to discover. Metaphors used cannot be too complex but have to be carefully crafted and presented in such a way as to help learners to form comprehensive and appropriate connections between the metaphor and critical points of the new knowledge. The use of more than one metaphor is also appropriate for the learning of a novel concept. It can also become a principal learning process, when students are encouraged to be involved in a process of generating their own metaphors regarding a novel learning subject (Carrol and Mack, 1999).

Taking into account all the above, we propose the design of an interactive metaphor-based computer environment for beginners' learning of basic concepts related to an OS. Specifically, the emphasis is put on the understanding by students of the most important missions of an OS, which are the management of all the resources that constitute the configuration of a modern computer system and which coordinate the access any user activity will have on them (Tanenbaum, 2001; http://en.wikipedia.org/wiki/Operating_system). Scheduling is a key concept in computer multitasking and multiprocessing operating system design, as well as in real-time OS design (Tanenbaum, 2001; http://en.wikipedia.org/wiki/Scheduling_(computing)). It refers to the way processes are assigned priorities in a priority queue. This assignment is carried out by software known as a scheduler. Among the plethora of common scheduling practices and disciplines, the '*Priority Round-Robin Scheduling'* algorithm is one of the oldest, simplest, fairest and most widely used scheduling algorithms designed especially for time-sharing systems (Faisstnauer, Schmalstieg, Purgathofer, 2000). Thus, we decided to design an emulation of *Priority Round-Robin (*PRR) algorithm (Faisstnauer, Schmalstieg,

Purgathofer, 2000) in order for students to understand how scheduling works. The PRR algorithm was chosen because of its simplicity but it was preferred to the classic RR algorithm because the latter was too simple.

3. DESCRIPTION OF THE PROPOSED EDUCATIONAL SOFTWARE

The proposed software will be organized in HTML pages, with its features being presented on its main page, and classified into two parts: i) information, and ii) experimentation space.

3.1 Information

The information space has been designed to provide three type of information: a) general information - in the form of text - about basic aspects related to OS, such as task management, scheduling and the notion of time quantum as well as a simple description of the PRR algorithm, b) appropriate URLs for further study, and c) metaphoric ready examples of task scheduling using a specific time quantum and the PRR algorithm. The metaphor examples used have to be selected in order to be simple and familiar to the students. To this end, the set of tasks to be scheduled have to stem from students' everyday life; thus, they will have the opportunity to easily grasp the concept of task scheduling using priorities. In addition, students will have the opportunity to give meaning to the concept of time quantum, as it will be far greater than those actually used by a real OS. By exploring these metaphor examples, students also will have the chance to estimate similarities and dissimilarities between these metaphoric examples and real OS. In this way, students can acquire a deeper understanding of the basic concepts of OS and of PRR algorithm.

3.2 Experimentation Space

Within the experimentation space, students can learn about the previously mentioned concepts by taking an active role designing their own metaphor examples. In fact, students have to be provided with the following possibilities: a) to define a time quantum that they can select from a list and b) to define their own scenario of tasks (for example 10 tasks) and to assign them with the priorities they prefer. Students also will have the chance to define the duration of each task in terms of the time quantum they selected in the previously mentioned step, c) to schedule the set of tasks according to their own conception of PRR algorithm, and d) to receive feedback from the system in terms of automatically scheduling the said set of tasks using the PRR algorithm. Appropriate help have to be also provided for the students to easily perform all the related operations provided by the software. By forming their own metaphors, students can be active and also take control of their learning. Students can also focus on the previously mentioned critical concepts related to OS and the PRR algorithm. Students can also generate more than one metaphorical context of tasks so that they can clarify these critical concepts. Then, students can further discuss and hypothesize about similarities and dissimilarities of the task scheduling in this metaphorical open-ended context and the context of the real OS and the PRR algorithm. In this sense, the metaphorical contexts formed can act as orienting frameworks leaving many other details for the students to explore with their teachers in the discussion sections. Moreover, by forming their own familiar domain to analogize from and on recognizing enough in the new domain of OS using PRR, students are expected to be able to establish some correspondences.

4. EPILOGUE AND FUTURE PLANS

The design of interactive metaphor-based educational software for the learning of basic concepts of an Operating System by beginners has been presented in this paper. In the context of this software, students will have the chance to learn about some critical concepts related to OS, such as task scheduling, task priorities and time quantum, using a simulation of the *Priority Round-Robin algorithm.* In the context of this proposed educational software, students can experience enjoyment and pleasure by forming their own task scenarios and assigning them both priorities and durations as well as scheduling them and receiving feedback by the system. It is also expected that students will try interacting with the proposed software a number of times to

reflect and make clarifications and abstractions about basic aspects of OS and the PRR algorithm. To verify our expectations, the implementation of the proposed software well as the performance of field studies to assess its impact on the learning of real students are in our future plans.

REFERENCES

Association for Computing Machinery, 2003. *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12* Task Force Curriculum Committee. Last retrieved on 20/05/08 from http://www.acm.org/education/curric_vols/k12final1022.pdf?searchterm=K-12

Borgman, C.L., 1999. The user's mental model of an information retrieval system: an experiment on a prototype online catalog. *In International Journal of Human-Computer Studies*, 51, pp. 435–452.

Cameron, L., 2002. Metaphors in the learning of science: A discourse focus. *In British Educational Research Journal,* 28 (5), pp. 673–688.

Carroll, J.M., Thomas, J.C., 1982. Metaphor and the cognitive representation of computing systems. *In IEEE Transactions on Systems, Man, and Cybernetics*, SMC 12 (2), pp. 107–116.

Carroll, J.M. and Mack, R.L., 1999. Metaphor, computing systems, and active learning. *In International Journal of Human-Computer Studies,* **51**, pp. 385–403.

Anderson, C. & Nguyen, M., 2005. A survey of contemporary instructional operating systems for use in undergraduate courses, *Journal of Computing Sciences in Colleges*, v.21 n.1, p.183-190.

Davoli, R. & Goldweber, M., 2003. New Directions in Operating Systems Courses Using Hardware Simulators, *ICSEE'03*, SCS 2003.

DeRosa, P., Shen, K., Stewart, C., and Pearson, P., 2006. Realism and simplicity: disk simulation for instructional OS performance evaluation, *ACM SIGCSE Bulletin*, v.38 n.1, March 2006.

Dobrilovic, D & Stojanov, Z., 2006. Using Virtualization Software in Operating Systems Course. In Proceedings of the International Conference on Information Technology: Research and Education, 16-19 Oct. 2006, pp. 222 – 226.

Dubinsky, Y. & Hazzan, O., 2003. Using Metaphors in eXtreme Programming Projects. In M. Marchesi & G. Succi (eds.) *XP 2003*. Springer-Verlag, Berlin Heidelberg, pp. 420-421.

Faisstnauer C., Schmalstieg D., Purgathofer W., 2000. "Priority round-robin scheduling for very large virtual environments". *Proceedings of the IEEE Virtual Reality 2000 Conference*, Washington, DC, USA, pp. 135.

Gentner, D., 1983. Structure-mapping: A theoretical framework for analogy. *In Cognitive Science*. 7 (2), pp.155-170.

Goldweber, M., Davoli, R., and Morsiani, M., 2005. The Kaya OS project and the μMPS hardware emulator. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, Caparica, Portugal, pp. 49-53.

Jonassen, D. H., 1999. Designing constructivist learning environments. *In Instructional design theories and models*, 2, pp. 215-239.

Holland, D., Lim, A., and Seltzer M., 2002. A new instructional operating system. In Proceedings of the A*CM 33rd Technical Symposium on Computer Science Education SIGCSE 2002*, pp. 111-115.

Hovemeyer, D., Hollingsworth, J. and Bobby Bhattacharjee, B., 2004. Running on the bare metal with GeekOS, *Proceedings of the 35th SIGCSE technical symposium on CS education*, March 03-07, 2004, Norfolk, Virginia, USA.

Hsu, Y.-C., 2006. The effects of metaphors on novice and expert learners' performance and mental-model development, *In Interacting with Computers,* 18 (2006), pp. 770–792.

Hummel, H., G-K., 2006. Feedback Model to Support Designers of Blended-Learning Courses. *In International Review of Research in Open and Distance Learning*, 7(3), pp. 1-16.

Lakoff, and Johnson, M. 1980. *Metaphors we live by*. Chicago University Press, Chicago, USA.

Laverell, W., Fei, Z., and Griffioen , J., 2008. Isn't it time you had an emulab? In Proceedings of the *39th SIGCSE technical symposium on Computer Science Education*, Portland, OR, USA, pp. 246-250.

Liu, H., Chen, X. and Gong, Y., 2007. BabyOS: a fresh start, *ACM SIGCSE Bulletin*, v.39 n.1, March 2007.

Morsiani, M. & Davoli R., 1999. Learning operating systems structure and implementation through the MPS computer system simulator. *ACM SIGCSE Bulletin archive*, Vol. 31, Issue 1, pp. 63 – 67.

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

Ortony, A., 1979. *Metaphor and Thought*. Cambridge University Press, Cambridge.

Tanenbaum, A.S., 2001. *Modern Operating Systems (2nd Edition).* Prentice Hall, NJ, USA.

Vygotsky, L., 1974. *Mind in society*. Cambridge, MA: Harvard University Press.

Wozny, L.A., 1989. The application of metaphor, analogy and conceptual models in computer systems. *Interacting with Computers*, 1 (3), pp. 273–283.