Impacts of interdisciplinary dialogue to computer science education

Veijo Meisalo
Department of Teacher Education, PO Box 38, FIN-00014
University of Helsinki, Finland, e-mail: meisalo@cc.helsinki.fi

Erkki Sutinen
Department of Computer Science, PO Box 26, FIN-00014
University of Helsinki, Finland, e-mail: sutinen@cs.helsinki.fi

Jorma Tarhio
Department of Computer Science, University of Joensuu
PO Box 111, FIN-80101 Joensuu, Finland
e-mail: tarhio@cs.joensuu.fi

Abstract

Based on our experiences in an interdisciplinary research group, we propose more flexibility in those areas of computer science education which are interesting also to other disciplines.

Keywords
Informatics, other disciplines, informatics majors, noninformatics majors, curriculum (core), role of CIT

Informatics in Higher Education F. Mulder & T. van Weerts (Eds.)

1 DIALOGUE EDUCATIONALISTS AND COMPUTER SCIENTISTS

The categorical organization of the university makes professionals working on the same topic from different backgrounds alien to each other. The area of computer-supported learning environments is an example of this. Educationalists and computer scientists do not know, and hence respect, each others' theoretical background or methods. Although the principles of designing computer-aided learning environments encourage teamwork and close cooperation,the implemented software is seldom balanced. Usually either a computer scientist or an educationalist is the dominating partner.

If an educationalist is the originator of the software, the result might be recognized by its inflexible implementation, possibly making the student a passive browser of a book-like organized file. The educationalist's poor programming and implementation skills may lead to a disastrous result, although he is an expert in activating teaching methods. The solution does not necessarily make use of the computer's potential: the role of the computer is obscure or taken as given. The code may be scattered, consisting of goto statements from one page to another.

The problems of the software, made by a computer scientist, are quite opposite but at least as serious. Often, a computer scientist does not think of his product from the learner's point of view: in educational software, this is reflected in vague pedagogical goals. Moreover, the computer scientist is seldom an expert in evaluating the software from the user's perspective.

2 INTERDISCIPLINARY RESEARCH

In 1995, we started an interdisciplinary project of animation-aided problem solving (AAPS). The participants came from Computer Science (CS) and Education. The aim of the project is to develop computer-supported methods for creative problem solving, similar to those developed earlier for automatic algorithm animation by the computer scientists (Lahtinen *et al.*, 1996). Creative problem solving is needed in various areas of education (Kuitunen and Meisalo, 1996).

Although the research topic was relevant to both the disciplines, it required several meetings to understand each others' vocabulary, methods, and perspectives, even the way of doing things together. The meetings were necessary to make each participant a genuine contributor of the group. The dialogue, with concurrent development of new computer-aided learning tools, gave us a clear view of the problems of computer-mediated education. In general, cooperation of computer scientists and educationalists is essential in making pedagogically sound, technically advanced, and innovative computer-aided learning environments.

3 UNCOVERING THE CORE OF COMPUTER SCIENCE

The dialogue has enforced each participant of the research group to evaluate his role in the whole project. From a computer scientist's perspective, the question is: 'What has computer science to offer to the interactive learning environments?'

This question has made us think of computer science as a problem solving science. Traditionally, CS education has concentrated on closed problem settings instead of open ones. However, any genuine research starts from a wonder, with no explicit and well-formulated questions.

Furthermore, a computer scientist is an expert in the area of discrete structures. These are an important element of any learning environment where the active user is allowed to manipulate graphical objects, representing his mental models, for example as concept maps. The fact that the learning environments do not give much to an active learner is due to the programmers with no expertise in efficient implementation of advanced structures, which belongs to the core of computer science.

4 SUGGESTIONS FOR RENEWING COMPUTER SCIENCE CURRICULA

A dialogue between computers scientists and other professionals, carried out together with a common project, provides new horizons in understanding computer science as a discipline. It also clarifies questions of the role of CS in the area of computer uses in education which has traditionally been the duty of educationalists. In addition, it offers practical consequences on how and what aspects of CS to teach to nonmajors.

The dialogue has made clear the importance of skills, like those of creative problem solving, to enhance the traditionally knowledge-centered CS curricula. For nonmajors, we emphasize the importance of programming skills.

Based on our experiences in using creative problem solving methods in CS research, we will teach a course on this topic. This course not only sketches various problem solving methods and encourages the students to apply them, but also challenges the students to design software to promote problem solving.

To employ their expertise in activating the learners with the help of computers, the educationalists should know the required technologies. Therefore, a course in computer-aided learning environments for educationalists should consist not only of applications and pedagogical theories, but also of programming and software implementation skills.

The dialogue has shown that it is also possible to bridge the gap between research and education. We organized a data structures laboratory for CS majors of the third year, utilizing group processes and creative problem solving methods (Meisalo *et al.*, 1997).

5 REFERENCES

Kuitunen, H. and Meisalo, V. (1996) The LOTTO project: Creative problem solving for schools. Group processes and creativity in action, in *Proc. XIII Research Symposium on Mathematics and Science Education* (ed. M. Ahtee *et al.*), Report 162. University of Helsinki, Department of Teacher Education, Helsinki, 61-70.

Lahtinen, S.-P., Lamminjoki, T., Sutinen, E., Tarhio, J. and Tuovinen, A.-P. (1996) Towards automated animation of algorithms, in *Proc. Fourth International Conference in Central Europe on Computer Graphics and Visualization 96* (eds. N. Thalmann and V. Skala), University of West Bohemia, Department of Computer Science, 150-161.

Meisalo, V., Sutinen, E. and Tarhio, J. (1997) CLAP: teaching data structures in a creative way, in: *Proc. ITiCSE '97, Integrating Technology into Computer Science Education*, ACM, Uppsala, 117-119.

6 BIOGRAPHY

Veijo Meisalo is an associate professor and Dean of the Faculty of Education at the University of Helsinki, Finland. His research interests cover group processes and creative problem solving in science education.

Erkki Sutinen is a researcher in the Department of Computer Science, University of Helsinki, Finland. His main interests are algorithms, especially approximate string matching and designing computer-based tools for active learning, like concept mapping and algorithm animation.

Jormo Tarhio is an associate professor in the Department of Computer Science at the University of Joensuu, Finland. His research interests are in the areas of string algorithms, animation and computer-supported concept mapping techniques.