Research Agenda for Computer Science Education

Christian Holmboe
Dept. of Teacher training and School Development
University of Oslo, Norway
christian.holmboe@ils.uio.no

Linda McIver
School of Computer Science and Software Engineering
Monash University, Australia
linda.mciver@csse.monash.edu.au

Carlisle George
School of Computing Science
Middlesex University, UK
c.george@mdx.ac.uk

Keywords: POP-I.B. *programmer education* POP-V.B. *literature review* POP-VI.D. *Computer Science Education Research*

Abstract

This paper examines the nature and scope of computer science education (CSE) research. We first distinguish CSE research from other forms of educational research, outlining its aims and identity as a research discipline. In examining the state of the art of CSE research, we attempt to categorise past research studies into general themes, reflecting the diverse contributions to CSE made over the years. Further, we critique each category, highlighting possible benefits and limitations. We argue that there has been a lack of reference to pedagogical theory, underlying most past research studies. This has resulted in a failure to provide teachers with "pedagogical content knowledge", critical to gaining useful insights into cognitive and educational issues surrounding learning. We conclude by providing guidelines for CSE research, stressing the need for a stronger connection to the theoretical frameworks of education-related disciplines such as pedagogy, epistemology, curriculum studies and psychology.

Introduction

A review of existing computer science education (CSE) literature shows that the effort to date has largely been in a few important, but relatively limited areas (such as descriptions of courses, development of tools, and computer aided learning). In contrast, long-established scientific disciplines such as physics and chemistry possess a large body of literature concerned with education in those disciplines. Specific problems which arise in the teaching and learning of these subjects have been thoroughly researched, and there is a long tradition of pedagogical research specifically relevant to each discipline. Many computer science educators have no formal training in education. As a result, the field of research tends to be grounded in the technology, rather than in the pedagogy or didactics of computer science.

In this paper we attempt to categorise the CSE literature and highlight some areas whose expansion and consolidation would provide a solid foundation for both future research and the development of innovative teaching techniques and tools which support and enhance computer science education.

What is CSE?

Computer science is a rapidly changing and increasingly diverse academic discipline. Relevant issues for research concerning the teaching and learning of different parts of this discipline are even more diverse. This paper will not provide a unique definition of research in computer science education;

nevertheless it is necessary for the further discussion, to offer some general pointers to our understanding of what constitutes the discipline.

Computer science - a diverse and developing discipline

The term *computer science* itself might be a problematic one. Other possible terms could be *informatics* or *information science*. Different universities, research groups, practitioners, countries and languages have developed different connotations to these words. A possible and open-minded definition of computer science is that it is the collection of scientific disciplines oriented toward the electronic or digital storing and processing of information. Others may state that this definition is too wide and advocate a division into different disciplines. One way to do this is to say that computer science, or indeed informatics, is concerned with the technical aspects of designing computers and computer systems, whereas the more social scientific issues of computers in organisations and society or psychological issues of human computer interaction constitute scientific disciplines of their own. In addition to these aspects we have neighbouring research areas such as communication, media, mathematical modelling, information copyright law etc.

In this paper the more open-minded view is adopted, but there will be an emphasis on what constitutes computer science as a scientific or academic discipline as opposed to a craft.

What CSE is not

Internationally there is an exponentially growing body of work being done on *computers in education*, both from computer scientific as well as from pedagogical and sociological points of view (Boyd-Barrett & Scanlon, 1990; Oettinger, 1969). We wish to distinguish this kind of work from CSE in that this work does not usually focus on the subject matter of computer science, but on the implementation of technology as a pedagogical means in the teaching and learning of other subjects.

A rough division can be made between the teaching and learning of computer science as vocational training on the one hand and as a theoretical academic study on the other. The focus of this paper is on the teaching and learning of the academic discipline of computer science.

Subject specific educational research

There is a strong tradition for research focusing on the application of pedagogical principles along with several other areas of interest concerned with the teaching and learning of a scientific discipline as a school subject. In Germany as well as in the Nordic countries this research has been associated with the term *subject didactics* (Gundem, 1998), whereas it in France is denoted simply as *didactics*. The term didactic in this sense should not be confused with the connotation most widespread in English of a teacher-controlled lecturing mode of information delivery. We therefore choose here to denote this versatile area of research *subject specific educational research.*

Hence, the academic discipline, computer science education (CSE) would be the subject specific educational research for the subject computer science.

The aim of CSE-research

The main motivation for educational research in computer science education, as in all subjects, is to improve the quality of the teaching and learning of the subject in schools and universities. To be able to measure the success of this work, one needs to agree upon what constitutes an improvement. In particular, what is good teaching practice? How do we best facilitate the students' construction of knowledge? The findings of existing studies attempting to answer these questions are just as diverse as would be expected, considering the diversity of the individuals (ie. students, teachers and researchers) involved in them.

Pedagogical content knowledge

The good teacher needs, of course, to master the subject matter of his or her discipline well, but this is not enough. It is essential that the teachers have what is referred to as *pedagogical content knowledge* (Gess-Newsome & Lederman, 1999).

> *"Teachers need to know more than just their subject. They need to know the ways it can come to be understood, the ways it can be misunderstood, what counts as understanding: they need to know how individuals experience the subject." (Laurillard, 1993)*

Educational research should not mainly aim at prescribing successful methods of teaching – although this can provide useful guidelines for fellow practitioners. The aim should rather be to describe the different ways in which students come to understand, or not to understand, the subject matter. These descriptions, accompanied by knowledge of general pedagogical theory, epistemology and solid subject knowledge will make the foundation for answering the traditional *didactical* questions of *why*, *what, how* and *for whom*. Why should this subject matter be taught as a school subject, which topics do we find appropriate for such a subject and how do we best facilitate the construction of the desired types of knowledge. The *for whom*-question forms an important background for answering the other three. It is essential to keep these questions and their answers in mind. This goes for each individual teacher as well as for policymakers and educational researchers.

The Identity of CSE as a Research Discipline

As Ferguson claimed about the history of technology in the seventies, computer science education has ”all of the appearances of an academic field, yet it is difficult to find in it a discipline or conceptual framework that guides the work being done in its name” (Lewis, 1999a).

What then might constitute a common denominator for CSE as a research discipline? We will see that both the various academic connections to other established disciplines and the wide variety of relevant research methods ensures that the traditional constraints constituting a research discipline do not apply here. A possibility is to say that it is the common goal that connects the corners of the field. We can claim that what makes research in CSE an autonomous academic discipline is the aim to provide the knowledge needed to help our computer science teachers and lecturers attain *pedagogical content knowledge*.

“State of the Art” in CSE

Studying the recent publications on computer science education alongside the ones from more than twenty years back, there is a striking lack of reference either to a pedagogical frame of theory or to prior work and findings on the topic. These observations support the need for a common identity and frame of reference for future research in computer science education in order to build a foundation for progress.

In this section we outline several categories of computer science education research publications, and discuss the contributions of each category to the field. This analysis leads us to identify categories that could usefully be expanded, which we feel would contribute substantially to the strength and progress of the field as a whole.

New, untested ideas

Several papers have outlined new ideas and methods for teaching and

learning, which serve as a starting point for discussion, but do not yet provide evidence (either empirical or otherwise) to substantiate the effectiveness of the idea or method. Several examples include: implementation-independent pedagogic approaches to programming (Ford, 1984), use of program plans in teaching programming (Soloway, 1985) and use of mathematical constructs to teach programming (Elenbogen & O'Kennon, 1988). Some of the new ideas outlined provide the basis for further research, especially by other researchers who perform comparative studies between existing and novel methods.

Also in this category, are some publications which describe sophisticated educational software for teaching and learning (especially those alleged to have human-like intelligence), which either exist as a very limited functioning prototype, or are in the design stage. These publications give us a view of how computer science education can be enhanced in the future when intractable implementation problems are successfully solved. This category of work, although not of immediate practical use to educators, can be very useful in providing directions and the impetus for further research.

Reports from the trenches

A large number of publications, especially at the annual conferences of SIGCSE and ITiCSE, are written by practitioners in computer science education who base their writing on their own experiences of teaching a certain course. Many of these have experienced problems with low-attaining students or high failure and drop-out rates in their introductory courses. Based either on intuition or on ideas picked up from colleagues, they have then implemented some kind of change in the delivery of the course. The effect of this change is then evaluated in a conference paper based on how the student responded, how the lecturer feels that it worked or sometimes on the results from the final exam (Carbone & Kaasbøll, 1998). These papers give a lot of insight and ideas on what might be done in teaching a particular course. The sharing of ideas and techniques for teaching is critical to the evolution and progression of computer science education.

Many such papers are based on sound computer science theory (Allen et al, 1996; Bauer, 1979; Berglund & Daniels, 1997; Dietrich & Urban, 1996; Koffman & Wolz, 1999; Smith & Rickman, 1976), but few also refer to pedagogical theory. Many of the problems described in 1996 had already been identified in 1976. In addition, there are obvious difficulties in empirically evaluating such courses – aside from the expense of running two concurrent courses and comparing results, such techniques would be ethically dubious, potentially disadvantaging students in one course or the other. Where comparisons can be done across different years, the number of changes between the courses makes it difficult, if not impossible, to evaluate the effect of individual changes.

Other papers go further, by asserting their proposed method as the optimum path to achieving student excellence. Sometimes attempts are made to justify such claims (Clark, 2000; Koelling & Rosenberg, 1996; Meertens, 1981), but they are often limited to subjective reflections and arguments which, while useful as bases for discussion, do not constitute strong evidence.

Discussion of theory

Some publications also offer references to epistemological theories like constructivism (Ben-Ari, 1998; Hadjerrouit, 1998), or theories of natural language acquisition (Murnane, 1993). Unfortunately, in many instances, these references are cited in the introductory sections of the publication, but in the later sections, the findings of the study are rarely discussed within the wider context of these supporting or related theories.

However, these papers are important for the discussions they initiate, and the conclusions they draw. They provide a basis for empirical research, to verify and explore the theories, and how they relate to computer science education.

Computer Aided Learning & Intelligent systems

Since the introduction of computer technology in schools in the late seventies, the prophecies have been running strong about how it will revolutionise schools and education. The impact and possibilities offered by educational software, the Internet and interactive multimedia have also been advocated for computer science education. The research done on artificial intelligence has given rise to the idea of developing intelligent tutoring systems where the computer “learns” about the student's way of thinking and working and gives feedback and instruction based on this (Chalk, 2000; Pirolli, 1986; Polson & Richardson, 1988; Sleeman & Brown, 1982). The rationale for this research has been the provision of dynamic individualisation of instruction and a sophisticated level of computer-mediated learning. Research studies in this area, however, have yet to fulfil many of their major aims (e.g. the ability for the system to diagnose errors, adapt to individual needs, draw inferences and solve problems in the domain), simply because computers are not capable of possessing human traits such as intuition and common sense.

This research also raises the issue of having automated standardised tests to save work and to avoid the highly biased results from a human teacher reviewing the students' work (MacNish, 2000; Seffah et al, 1999).

Expert/Novice differences

A large number of studies have been carried out that aim to describe the different ways in which experts and novices master problem solving situations or picture the tasks they meet in computer scientific areas (Batra & Antony, 1994; Kahney, 1983; Soloway & Spohrer, 1986; Wiedenbeck, 1985). These studies provide useful insights into the differences between novices and experts, with a view to setting the benchmark for novice achievement. Such information is of critical pedagogical importance to teachers and educators.

Empirical Studies

There is also a body of empirical work which focuses on specific programming phenomena, analysing students' code, or working from interviews with students who are attempting to solve a particular problem. Some papers study a specific group of learners, such as students with no programming experience (McIver, 2000), or students learning a particular language (Eisenstadt & Lewis, 1992; Putnam et al, 1986; Van Someren, 1985). Others target particular programming constructs such as conditional statements (Sime, Green, & Guest, 1973), teaching techniques for particular constructs (George, 1996; Good, 1999), or student programming errors (Pea, 1986; Soloway & Spohrer, 1986) There are also studies that aim to describe the mental models or individual understanding students seem to have concerning programming or systems design (Bhuiyan, 1992; Booth, 1992; Brooks, 1999; Holmboe, 2000; Petre & Blackwell, 1999).

These studies all examine the behaviour and responses of students tackling real programming problems, to learn about the difficulties students have when learning to program. This type of research provides a firm basis for improving teaching techniques, and the creation of effective tools for teaching programming. This is a category which could usefully be expanded in order to strengthen the field of computer science education.

Learning from others

> *"We have to talk about research needs in a way that engenders ever more possibilities. Rather than boxing in the researchers, we must see ways to push the limits and explore new and different frontiers"* (Lewis, 1999b)

Educational research

Being strongly linked to the content and epistemology of the particular school subject in question, subject specific educational research will inevitably differ from one subject to another (Gundem, 1998). Even so, looking to the variety of work being published in more established fields (e.g. science education, mathematics education and teaching and learning of foreign languages) may give several useful pointers to researchers in computer science education when determining the focus of their future work.

Answering the first of the *didactical* key questions (i.e *why* computer science should be taught as a school subject?) in turn raises another interesting point for educational research; the formulation of the identity of the academic field to be handled in the school subject at hand. This is research concerned with, for example, what science is really about as an academic field, and how this influences the everyday lives of the average man or woman (Driver et al, 1996; Sjøberg & Kallerud, 1997)? This constitutes much of the rationale for including a particular subject in the school curriculum at a certain level. We briefly touched on this issue in a previous section concerned with the reduction of computer science to mere computer handling skills. Establishing the intrinsic value of computer science as a school or university subject, is therefore an important aspect of CSE research, in the same way that it has been so for example in recent science education research (Jenkins, 1996; Millar & Osborne, 1998).

Students' understanding, or indeed misunderstanding, of particular aspects of a subject matter has long constituted an essential part of subject specific educational research (Ryder et al, 1999; Sierpinska, 1994). To a certain extent this has also been the focus of CSE research (Booth, 1992; Navarro-Prieto & Catmas, 1999). Hopefully we will see even more of this in the future, since this kind of research plays an important role in helping teachers reach pedagogical content knowledge.

One further topic for educational research is concerned with the implementation or impact of certain epistemological theories on the teaching and learning of a given subject. It is important to establish a sound theoretical frame of reference to the observation and description of learning processes, be it in the classroom or elsewhere. The trends in this kind of research have recently shifted somewhat from constructivistic points of view (Driver et al, 1994; Glasersfeld, 1989) toward a stronger focus on situated learning (Anderson et al, 1996; Hennessy, 1993) and sociocultural aspects of learning (Säljö, 2000).

Other related research areas

Subject specific educational research will, as we have seen, have a lot in common with several other academic disciplines, both methodologically and in the research questions being raised.

> *"In France didactics is based on psychology, pedagogy and epistemology. Even so a specific frame of reference or theory of its own has been developed."* (Gundem, 1998)

Psychological research has been closely linked to computer science in at least two ways. The study of cognitive aspects of learning to program or computer system comprehension has been and should be given attention. The epistemologically based research mentioned in the previous section naturally draws upon more general pedagogical and psychological findings.

A quite different link between psychology and computer science has occurred in the vast body of research often denoted as cognitive science. Here facets of human understanding, knowledge construction and problem solving strategies have been studied, in order to simulate aspects of human intelligence in a computer program, creating what is referred to as artificial intelligence. Research in this area relevant to CSE concerns the development of "intelligent" educational systems, as discussed previously.

A common denominator for the impact that psychological work has had so far on CSE is that it seems to be driven by economic motives such as increasing productivity and minimising human errors in the computer industry. A wish for the future would be that computer science educators look more closely at the resources available both in psychology and general pedagogical theory in their quest to educate the computer scientists of tomorrow.

Work within areas of computer science itself can also be applied to research in CSE. There is for instance a large body of work being done on human computer interaction (HCI) and on the related field of computer supported collaborative learning (CSCL). These studies examine the implications of the computer interface on the mental processes of the user of computer systems (Littleton & Light, 1999). In a meta-perspective this provides essential input to computer science education as well, since most of the activities carried out by the learner in a computer science course will be done in front of a computer that will somehow influence the learning process. Parallel to this, but unfortunately not very often connected, is a lot of the research being done on computers in education that usually focuses on different ways of implementing computer technology in the teaching and learning of different subjects. The work in these two areas should of course be considered as influential to research in CSE as well as in other school subjects. Maybe even more so.

Thoughts for the future

What are/should be the main areas and methods of research in CSE?

To establish research in computer science education as an academic discipline, it is imperative that the wide range of relevant issues is covered. At the same time, it is important to keep some common ground in order to achieve a feeling of identity and belonging within the field. The common ground outlined in the present paper is the common aim of most subject specific educational research – i.e. the facilitation of pedagogical content knowledge for practitioners.

Research methods relevant for carrying out this research will vary considerably, depending on the focus of the individual project. We have established that subject specific educational research is linked with several originally very different research traditions. A researcher in this field must be capable of border-crossing and be able to utilise the advantages that lie in the enormous selection of resources available.

Still it is important to keep in mind what Feyerabend (1975) implied in stating that "anything goes" for scientific research. Anything is possible or applicable as long as it is well founded in empirical results or in theoretical argumentation. The future work of CSE must have a stronger connection to the theoretical frameworks of education-related disciplines such as pedagogy, epistemology, curriculum studies and psychology.

Who do we expect to pursue these issues?

> *"... research on learning is usually conducted in departments of educational psychology, whereas research on teaching is usually conducted in departments of curriculum and instruction. Unfortunately, there is often far too little contact between researchers concerned with the two topics."* (Shuell, 1993)

Close collaboration between computer scientists and researchers in educational science, psychology, epistemology and related fields is imperative. With such collaboration, we can adapt what is already known to our own subject specific concerns, and build on existing educational research in meaningful and productive ways.

There is a need for more dedicated researchers in CSE, since the majority of work done in the past has been done by computer scientists reflecting on their own teaching practice. In more established educational research, like science education research, there is also an overwhelming majority of scientists doing the work. The difference is that these have usually converted into educational researchers with a solid knowledge of their subject matter. The studies carried out are not based on their own teaching practices so much as on other teachers' practices. The field has grown into an academic discipline of its own with its own faculty positions – sometimes located in a school of teaching and learning and sometimes in their old department. They represent – one might say – the state of the art in pedagogical content knowledge.

Conclusion

To date there has been a productive climate of educators sharing ideas, techniques and tools in order to improve computer science education throughout the world. However, a change in the focus of the field of computer science education research seems desirable at this point. More empirical research and comparative evaluation would build a stronger foundation for future research. A higher proportion of this sort of work would also strengthen the case for Computer Science Education Research to be taken seriously as an academic discipline, and counter the criticism often levelled at it, that it is merely a way for "teachers to write papers".

Acknowledgements

The authors would like to thank Clayton Lewis and the rest of the participants of PPIG2000 for encouraging and stimulating the writing of this paper.

References

Allen, R. K., Grant, D. D., & Smith, R. (1996). Using Ada as the first Programming Language: A retrospective. Paper presented at the *Software Engeneering: Education & Practice*.

Anderson, J. R., Reder, L. M., & Simon, H. A. (1996). Situated Learning and Education. *Educational researcher, 25*(4), 5-11.

Batra, D., & Antony, S. R. (1994). Novice errors in conceptual database design. *European Journal of Information Systems, 3*(1), 57-69.

Bauer, M. A. (1979). Experiences with PASCAL in an Introductory Course. Paper presented at the *SIGCSE 10*, Dayton, Ohio.

Ben-Ari, M. (1998). Constructivism in Computer Science Education. Paper presented at the *SIGCSE*.

Berglund, A., & Daniels, M. (1997). Improving Education Quality, a Full Scale Study. Paper presented at the *SIGCE Technical Symposium.*

Bhuiyan, S. H. (1992). *Identifying and Supporting Mental Methods fo Recursion in a Learning Environment.* Unpublished PhD, University of Saskatchewan, Saskatoon, Canada.

Booth, S. (1992). *Learning to Program: A phenomenographic perspective.* Unpublished Ph.D, University of Göteborg, Gothenbourg.

Boyd-Barrett, O., & Scanlon, E. (Eds.). (1990). *Computers and Learning*. Wokingham, UK: Addison-Wesley.

Brooks, R. (1999). Towards a theory of the cognitive processes in computer programming. *International Journal Human-Computer Studies, 51*, 197-211.

Carbone, A., & Kaasbøll, J. (1998). A Survey of Methods Used to Evaluate Computer Science Teaching. Paper presented at the *ITiCSE*, Dublin, Ireland.

Chalk, P. (2000). Webworlds-Web-Based Modeling Environments for Learning Software Engineering. *Computer Science Education, 10*(1), 39-56.

Clark, D. (2000). Fast Tracking the Learning of How to Use Neural Networks as Classifiers. *Computer Science Education, 10*(2), 129-146.

Dietrich, S. W., & Urban, S. D. (1996). Database theory in practice: Learning from cooperative group projects. *SIGCE Bulletin*, 112-116.

Driver, R., Asoko, H., Leach, J., Mortimer, E., & Scott, P. (1994). Constructing Scientific Knowledge in the Classroom. *Educational Researcher, 23*(7), 5-12.

Driver, R., Leach, J., Millar, R., & Scott, P. (1996). Perspectives on the nature of science. In R. Driver, J. Leach, R. Millar, & P. Scott (Eds.), *Young people's images of science* (pp. 24-44). Buckingham: Open University Press.

Eisenstadt, M., & Lewis, M. W. (1992). Errors in an Interactive Programming Environment: Causes and Cures. In M. Eisenstadt, M. T. Keane, & T. Rajan (Eds.), *Novice Programming Environments: Explorations in Human-Computer Interaction and Artificial Intelligence* . Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

Elenbogen, B. S., & O'Kennon, M. R. (1988, Feb). Teaching recursion using fractals in PROLOG. Paper presented at the *SIGCSE*.

Feyerabend, P. (1975). *Against Method*. London: NLB.

Ford, G. (1984). An implementation-independent approach to teaching recursion. Paper presented at the *SIGCSE*.

George, C. (1996). *Investigating the Effectiveness of a Software-Reinforced Approach to Understanding Recursion.* Unpublished PhD, University of London, London.

Gess-Newsome, J., & Lederman, N. G. (Eds.). (1999). *Examining Pedagogical Content Knowledge*: Kluwer Academic Publishers.

Glasersfeld, E. v. (1989). Cognition, Construction of Knowledge and Teaching. *Synthese, 80*(1), 121-140.

Good, J. (1999). *Programming Paradigms, Information Types and Graphical Representations: Empirical Investigations of Novice Program Comprehension.* Unpublished PhD, The University of Edinburgh, Edinburgh.

Gundem, B. B. (1998). *Understanding European Didactics - an Overview*. (Vol. report 4 - 1998): University of Oslo, Dept. of Pedagogy.

Hadjerrouit, S. (1998). A Constructicist Framework for Integrating the Java Paradigm into the Undergraduate Curriculum. Paper presented at the *ITiCSE*, Dublin, Ireland.

Hennessy, S. (1993). Situated Cognition and Cognitive Apprenticeship: Implications for Classroom Learning. *Studies in Science Education, 22*, 1-41.

Holmboe, C. (2000). *A Framework for Knowledge:* Analysing High School Students' Understanding of Data Modelling. Paper presented at the *Twelfth Annual Meeting of the Psychology of Programming Interest Group*, Corigliano Calabro, Italy.

Jenkins, E. W. (1996). The 'nature of science' as a curriculum component. *Journal of Curriculum Studies, 28*(2), 137-150.

Kahney, H. (1983). What do novice programmers know about recursion? Paper presented at the *CHI '83 Conference on Human Factors in Computer Systems*, Boston, MA.

Koelling, M., & Rosenberg, J. (1996). Blue - A Language for Teaching Object-Oriented Programming. Paper presented at *the SIGCSE Technical Symposium*, Philadelphia, Pensylvania.

Koffman, E., & Wolz, U. (1999). CS1 Using Java Language Features Gently. Paper presented at the *ITiCSE*, Cracow, Poland.

Laurillard, D. (1993). *Rethinking University Teaching: a framework for the effective use of educational technology*. London: Routledge.

Lewis, T. (1999a). Content or Process as Approaches to Technology Curriculum: Does It Matter Come Monday Morning? *Journal of Technology Education, 11*(1).

Lewis, T. (1999b). Research in technology education - some areas of need. *Journal of technology education, 10*(2), 41-56.

Littleton, K., & Light, P. (Eds.). (1999). *Learning with Computers; Analysing productive interaction*. London: Routledge.

MacNish, C. (2000). Java Facilities for Automating Analysis, Feedback and Assessment of Laboratory Work. *Computer Science Education, 10*(2), 147-163.

McIver, L. (2000, April). The Effect of Programming Language on Error Rates of Novice Programmers. Paper presented at the *Twelfth Annual Meeting of the Psychology of Programming Interest Group*, Corigliano Calabro, Italy.

Meertens, L. (1981). Issues in the Design of a Beginners' Programming Language. In d. Bakker & v. Vliet (Eds.), *Algorithmic Languages* (pp. 167-184): IFIP North Holland Publishing Company.

Millar, R., & Osborne, J. (1998). *Beyond 2000: Science education for the future* . London: King's College, School of Education.

Murnane, J. (1993). The Psychology of Computer Languages For Introductory Programming Courses. *New Ideas in Psychology, 11*(2), 213-228.

Navarro-Prieto, R., & Catmas, J.-J. (1999). Mental representation and imagery in program comprehension. Paper presented at *the PPIG'99*.

Oettinger, A. G. (1969). *Run, Computer, Run*. Cambridge, MA: Harvard University Press.

Pea, R. D. (1986). Language-independent conceptual "bugs" in novice programming. *Journal of Educational Computing Research, 2*(1), 25-36.

Petre, M., & Blackwell, A. F. (1999). Mental imagery in program design and visual programming. *International Journal of Human-Computer Studies, 51*, 7-30.

Pirolli, P. (1986). A Cognitive model and a computer tutot for programming recursion. *Human-Computer Interaction, 2*, 319-355.

Polson, M. C., & Richardson, J. J. (Eds.). (1988). *Foundations of Intelligent Tutoring Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

Putnam, R. T., Sleeman, D., Baxter, J. A., & Kuspa, L. K. (1986). A Summary of Misconceptions of High School Basic Programmers. *Journal of Educational Computing Research, 2*(4).

Ryder, J., Leach, J., & Driver, R. (1999). Undergraduate science Students' Images of Science. *Journal of Research in Science Teaching, 36*(2), 201-219.

Seffah, A., Bari, M., & Desmarais, M. (1999). Assessing Object-Oriented Technology Skills Using an Internet-based System. Paper presented at the *ITiCSE*, Cracow, Poland.

Shuell, T. J. (1993). Toward an Integrated Theory of Teaching and Learning. *Educational Psychologist, 28*(4), 291-311.

Sierpinska, A. (1994). *Understanding in Mathematics*. London: The Falmer Press.

Sime, M. E., Green, T. R. G., & Guest, D. J. (1973). Psychological Evaluation of Two Conditional Constructions Used in Computer Languages. *International Journal of Man-Machine Studies, 5*, 105-113.

Sjøberg, S., & Kallerud, E. (Eds.). (1997). Science, Technology and Citicenship. *The Public Understanding of Science and Technology in Science Education and Science Education Policy*. (Vol. 7/97). Oslo: NIFU Report.

Sleeman, D. H., & Brown, J. S. (1982). *Intelligent Tutoring Systems*. NY: Academic Press.

Smith, C., & Rickman, J. (1976). Selecting Languages for Pedagogical Tools in the Computer Science Curriculum*.* Paper presented at the *SIGCSE 6*.

Soloway, E. (1985). From problems to programs via plans: the content and structure of knowledge for introductory LISP programming. *Journal of Educational Computing Research, 1*, 157-172.

Soloway, E., & Spohrer, J. (1986). Novice Mistakes: Are the folk wisdoms correct? *Communications of the ACM, 29*(7), 624-632.

Säljö, R. (2000). *Lärande i Praktiken: Ett sosiokulturellt perspektiv*. Stockholm: Prisma.

Van Someren, M. W. (1985). What's Wrong? Understanding beginners' problems with Prolog. *Instructional Science*, *19*, 257-282.

Wiedenbeck, S. (1985). Novice/expert differences in programming skills. *International Journal of Man-Machine Studies, 23*, 383-390.