Qualitative Research Projects in Computing Education Research: An Overview

Anders Berglund, Mats Daniels, Arnold Pears

Uppsala University
Department of Information Technology
P.O. Box 337, 751 05 Uppsala, Sweden

{Anders.Berglund, matsd, Arnold.Pears}@it.uu.se

Abstract

Qualitative research approaches have much to offer computing education research (CER). Conducting studies which are theoretically anchored in pedagogy, as well as in computing, can help us to draw more solid and significant conclusions about how students learn computing. When studying teaching and learning situations it is important to take into explicit account what is meant by learning. We claim that this implies that research into student learning is strengthened by increased awareness of the role and relevance of qualitative research approaches in CER.

With this aim in mind we identify the area of "pedagogically anchored qualitative research" and place it within the broader CER landscape. We also provide an overview of research methods and resulting studies, which we feel exemplify the value this type of study has for the CER community.

Keywords: Computing education, Computing education research

1 Introduction

Computing education research (CER) is a cross-disciplinary field comprising computing – of course – as well as a wide range of other disciplines: pedagogy, psychology, cognitive science, learning technology, and sociology, to mention a few. The defining feature of CER (unifying this diversity) is the intent to improve learning and teaching within computing, and thereby to contribute to computing.

Many CER studies avail themselves of research methods common to the natural sciences, and are conducted in a quantitative research tradition. We argue that there are relevant computing education research questions where a qualitative methodological stance allows the researcher to obtain insights that are not accessible using research approaches commonly emloyed in the natural sciences. Thus, qualitative research extends the range of researchable questions in CER.

The aim of this paper is to highlight and exemplify qualitative research projects in CER anchored in a pedagogical tradition. That is, we discuss studies which take a theoretical standpoint combining aspects of computing (what is learned) with education (how does the learning take place) and is based on mainly qualitative research approaches.

Since pedagogically anchored research endeavours are explicitly tied to teaching and learning, they immediately make practical contributions to computing education. Furthermore, the fact that a paper has an explicit methodology places the researcher in at least two research communities: a computing education research community and a pedagogical community. Such work has a clear frame of reference, and takes a clear theoretical stance, giving others a context for evaluating the quality of the work. Projects of this type define an emerging sub-discipline of computing education research with considerable promise.

Our argument is that computing education will benefit from complementing the many positivistic research projects (for example statistical studies) with a larger number of pedagogically anchored qualitative research projects. The larger repertoire of research approach that are available, if both qualitative and quantitative methods are considered, enhances the number of researchable questions, and thus enables CER to address a wider range of research questions.

The remainder of the paper is structured as follows. In the next section, we describe the "landscape" of computing education research, placing pedagogically anchored qualitative CER into the broader field of CER. We also discuss issues associated with research communities and traditions. Section 3 discusses different approaches to performing pedagogically anchored qualitative research in computing education. An extended overview of qualitative research projects is given in Section 4, and different forms of results from such research are discussed. Section 5 discusses the significance of pedagogically anchored qualitative research for CER and draws some conclusions.

2 The landscape

Individual research endeavours are part of a tradition, and are, consequently, closely related to the community to which the researcher belongs. As a result we will discuss the nature of the CER community as well as sketching the CER landscape and characterizing the sub-fields of CER.

2.1 The computing education research community

Both the subject area and the community that "belongs" to the field are important when we discuss what computing education "is". Fincher discusses the need to identify a community of practice[1] in an article by Clancy, Stasko, Guzdial, Fincher & Dale (Clancy et al. 2001)[2].

She identifies particular difficulties in accomplishing this in emerging fields. Knowing who is "in" the community is a problem, since many of the leading researchers within the field are better known for their contributions to other sub-areas of computing. In emerging multi-disciplinary fields it is also hard to determine where the boundaries of the community are. If we consider CER as an "emerging field", how might we delineate its community? Venues for scholarly publication is one approach, professional bodies and associations is another.

There is a set of journals (for example Computer Science Education, Journal of Information Technology Education) and conferences (ACM Innovation and Technology in Computer Science Education, ACM Symposium of the Special Interest Group in Computer Science Education, Annual Finnish - Baltic Sea Conference on Computer Science Education, Australasian Computing Education Conference Workshop of the Psychology on Programming Interest Group, IEEE Frontiers in Education) where computing education research plays the leading role, or is one of the key topics.

Professional associations also exist, such as the IEEE Education Chapters and the ACM SIGCSE. CSERGI – Computer Science Education Research Groups International[3] is another type of initiative to enhance collaboration between the different computing education research groups on a world-wide scale. Another example of this type of action is the recent formation of the Australasian Computing Education Community[4]. Other more focused initiatives to promote research in computing education , e.g. the bootstrapping, scaffolding and brace projects (Petre, Fincher and Tenenberg, 2003), also serve to create an identity for the community.

These initiatives create a broad identity of a research community within computing education, enabling researchers to act in a context, sharing a language, questions, methodology and identity with other researchers. However, publication and research discussion related to pedagogically anchored research in CER is widely distributed and has not yet found its identity as a recognized area of CER.

[1] Fincher refers to Becher (1989) and Crane (1972) for research communities.

[2] The article consists of five different contributions, one by each author. The reference to Fincher in this paper refers to Fincher's contribution in the article by Clancy et al. (2001).

[3] http://www.docs.uu.se/csergi/

[4] http://cerg.csse.monash.edu.au/acec

2.2 Sub-fields of computing education research

Several characterizations of the sub-fields of computing education have been proposed (see for example Fincher & Petre, 2004; Greening, 1996; Holmboe, McIver & George, 2001). In our discussion we follow the four broad areas of computing education research identified by Fincher (Clancy et. al. 2001): (1) small scale investigations of a single aspect; (2) investigations motivated by the use of tools; (3) investigations of specific mental and conceptual skills in the psychological tradition; and (4) research anchored within the educational tradition.

We briefly sketch the first three areas and their contributions. The fourth, being the focus of our discussion, is the subject of the remainder of this paper.

2.2.1 Small scale investigations

A survey of the contributions to the ACM SIGCSE (Special Interest Group on Computer Science Education) symposium (Valentine, 2004), the ACM ITiCSE (Innovation and Technology in Computer Science Education), conference as well as the journal Computer Science Education indicates that most of the projects presented are case studies on specific courses. These studies are often performed, evaluated and reported about by the teacher of the course. The motivation for conducting them is usually the needs of the computing educators of the universities in question, and they typically address problems that have arisen in real teaching situations. Frequently the questions addressed concern the introduction of new methods or using new tools in the teaching.

Although valuable as a means of sharing experiences between computing educators, the results are, as also pointed out by Holmboe, McIver and George (Holmboe et al., 2001) and Carbone and Kaasbøll (Carbone and Kaasbøll, 1998), often hard to generalize, since they are not based on a theory of learning and are often carried out without a theoretically anchored methodological standpoint.

2.2.2 Investigations motivated by the use of tools

The development of new tools for teaching and learning of computing is in itself a broad field. It spans from tools aiming to highlight a particular issue, such as the transfer of packages in a particular network protocol (Mester and Krumm, 2000), over algorithm animation tools, such as Jeliot[5] to learning platforms or environments, for example BlueJ[6]. The point of departure can also be taken in different theories of learning (as for example cognitive psychology) with tools offering automated help to learners, such as the Virtual Approbatur (Torvinen, 2002), and in more general tools finding their roots in the research in educational technology. Here collaborative learning environments of different character and tools

[5] http://www.cs.joensuu.fi/jeliot/

[6] http://www.bluej.org/

aiming to facilitate collaboration can be placed, such as the Explanogram[7] (Pears and Erickson, 2003).

This field of research is of a growing importance, as it offers tools that could help to enhance learning, save teaching resources and promote collaboration. Since good forums within which to discuss this branch of computing education exist – both conferences and journals – we leave this area outside our discussion in this paper.

2.2.3 Investigations in the psychological tradition

Research in computing education at a university level has for a long time been dominated by cognitive psychological approaches. Here the aim is to explore the nature of knowledge structures, the acquisition of knowledge, and the different ways in which these can be made more efficient.

Such research often constructs and deploys models of human thinking in order to gain insights that then can be used to facilitate learning, for example Wiedenbeck's work on factors that affect success of non-majors in learning to program (Wiedenbeck, 2005), Holmboe's investigations on students' mental models (Holmboe, 2000), and the work of Robin, Rountree and Rountree (Robin et al., 2003) on novice programmers.

Research in this tradition is frequently based on experiments or quasi-experiments, testing hypotheses on the effects of educational methods or devices by classical methods of psychology, with statistic comparisons of the performance of a trial group and a control group before and after the former has been subjected to the treatment. As examples of the many good studies in this tradition, we mention Wu (Wu, 1993) concerning recursion; McIver's work concerning programming languages in introductory courses (McIver, 2000); and Almstrum's study of students' understanding of mathematical logic (Almstrum, 1996).

Research of this type serves, in many cases, as a model for how computing education research can, and ought to, be developed in the borderland between computing and other fields of research. It has contributed both through its results and by the rigor by which many research projects have been carried out. In this paper, however, we aim to discuss the kind of research that finds its methodological roots in qualitative research in pedagogy, leaving psychology and cognitive science for others to explore.

3 Research anchored within the educational tradition

Pedagogically anchored qualitative research projects in the computing education area are situated in the fourth category defined by Fincher (Clancy et. al. 2001), i.e. research anchored within the educational tradition. Before we give some examples of such projects, we will look into two issues. First the connection between research approach and outcome will be under investigation. This is done since we as users of our own research results might be tempted to overlook the issue of careful selection of research methods. Then we expand some on some qualitative aspect, since many computing researchers are unused with such research.

3.1 The relationship between the research and the outcome

When discussing a research project, having its theoretical basis within pedagogy, the possible outcome of is closely related to how the project is performed. Here there are several research approaches, or “starting points”, each with different empirical, epistemological and ontological[8] basis.

To consciously select an approach is crucial for a researcher, since different approaches offer varying perspectives on the research questions and serve to lead the researcher down different roads.

Explicit adoption of a research approach also facilitates communication with other researchers. A shared terminology becomes available and enables the researcher to learn from others and to judge and compare different projects. Furthermore the approach helps in guiding to what extent the results can be trusted and generalized to other groups of students or to other situations.

The relationship between an approach and an outcome is complex: Similar results, or at least results illuminating the same aspect of a research question, can be obtained in different ways. Neither does the selection of a particular way of performing the research necessarily lead to a certain type of result.

There is a growing awareness of the need to use relevant research approaches within computing education research. Initiatives promoting this type of research include the previously mentioned bootstrapping, scaffolding and brace projects (Petre, Fincher and Tenenberg, 2003), papers offering an overview of the current use of pedagogically anchored research approaches (Carbone and Kaasbøll, 1998; Holmboe McIver and George, 2001) and attempts to verbalize models of the research process (Daniels, Petre and Berglund, 1998; Pears, Daniels and Berglund, 2002), and panel discussions concerning computer science education research, for example (Booth, Berglund, Ben-Ari and Holmboe, 2004). The International Computing Education Research Workshop[9] (ICER) has its focus on research within computing education.

3.2 Qualitative research

When discussing research in education, the distinction between quantitative and qualitative research approaches is important. This distinction, and the resulting different selection of which methods to apply, is discussed in Gall, Borg and Gall (Gall, Borg & Gall 1996). They stress the different goals of these families of research approaches: while qualitative research aims to discover themes and

---

7 http://explanogram.it.uu.se

8 By *epistomology* we mean a theory of knowledge, and by *ontology* a theory about how reality is constituted.

9 http://icer2005.cs.washington.edu/

relationships, quantitative research projects is used to validate these ideas in particular cohorts.

Denzig and Lincoln (Denzig & Lincoln, 1994) describe qualitative research in the following terms:

> Qualitative research is multimethod in focus, involving an interpretive approach to its subject matter. this means that the qualitative researchers study things in their natural settings, attempting to make sense of, or interpret, phenomena in terms of the meanings people bring to them (ibid., p. 2)

Thus, while quantitative research is based on the assumption that the truth is objective and neutral, and corresponds in this way to research in science and technology, qualitative research challenges this assumption in varying degrees, for example by arguing that the question of objectivity is irrelevant, that it is impossible to determine if objective truth exists, or, simply by arguing that there is no objective truth[10].

With a qualitative research approach, the researcher aims to "reveal how the individual creates, modifies and interprets the world in which he or she finds himself" (Cohen and Manion, 1998, p. 8). Here the researcher's experience becomes a part of the research outcome, since a particular researcher brings his individual experience into the research situation. As a result, he or she obtains somewhat different insights than another researcher would. On these grounds, Fincher's efforts (Clancy et al., 2001) to find an identity for the community are important for the argument presented here, since a community of CER researchers using pedagogically anchored research approaches provides a forum in which to discuss and illuminate similarities and differences in the results of different projects.

4 An overview of research projects with pedagogically anchored qualitative research approach

The common identity of research projects in a pedagogically anchored qualitative tradition is that they discuss computing concepts, students and their relationships, with the aim of gaining insights in how to teach and learn computing. Most are based on consciously selected and deployed theories of learning. The different approaches and the different usages permit the researcher to explore and highlight different aspects of computing education.

The research approaches that are presented here are selected on a pragmatic basis: All of them have been used in research projects in computing education and have produced interesting results

4.1 Work in the Vygotskian tradition

The work of Vygotsky (Vygotsky, 1986) and his colleagues constitutes the basis of a multi-faceted research tradition in which learning is understood as related to, dependent on and being a part of the students' environments. Studies conducted in this tradition focus on the collective, of which the individual is a part, and the use of tools (intellectual as well as physical) as mediators of the learning. With this perspective, the students' actions: their talking, their use and development of tools, come to the fore, while their experience of their learning and situation in general become less significant[11].

Holland and Reeves (Holland and Reeves, 1996) have studied group work for student teams working with software development. They use activity theory (Engeström, 1987) and introduce the term perspective as a "view from somewhere" that is collective, historical and develops over time in the course. The teams took different perspectives, and as a consequence, they differed in their construal of the object of their work, the importance they gave to different sub-tasks and the way in which they carried out the work.

Chew, Beaumont, Seah and Westhead (Chew et al., 2004) also deploy activity theory to study students' actions in distributed teams in a computing course - in their case, an internationally distributed course, with team members both in Leeds, UK, and in Singapore. They focus on how students select their tools for communication, and identify, through the use of activity theory, how conflicts in the use of communication tools have arisen from the situation, and how they have been overcome by the students.

Ben-David Kolikant (Ben-David Kolikant, 2004) argues, based on the theories of situated learning (Lave & Wenger, 1991) that two cultures, namely that of academics and that of computer users, meet - and clash - when students learn computing. She illustrates her reasoning with a study of high-school students learning concurrent and distributed computing. The main thesis of her paper is that the study environment should create an opportunity for the students, and motivate them, to cross this cultural boundary.

The implications of situated learning and other socio-cultural approaches for teaching of computing are analyzed in a literature review by Ben-Ari (Ben-Ari, 2004). Based on his study he argues that teachers of computing should study the communities of practice of professional computer scientists and design educational activities which model the activities of those communities.

The projects summarized above all discuss how collectives, or teams, develop and how tools are deployed in specific situations. The significance of the results lies in the potential to "go outside" the question of how a particular student learns about a specific concept. The

[10] The terms *qualitative* and *quantitative* research are used in this paper as similar to *positivist* and *non-positivist* research (the latter term is sometimes labeled *anit-postivitist* or *post-positivist* research). In the literature the exact meanings of this terms and the distinctions between the dichotomies postivist/non-positivist and quantitative/qualitative are debated. For the pragmatic purpose of this paper, these distinctions are irrelevant.

[11] An introduction to this approach is given by Säljö (2000).

approach invites the researcher to study the complex picture that arises when students interact in a given environment.

4.2 Phenomenography

A phenomenographic research project aims to explore a phenomenon (for example the concepts of class and object or the network protocol TCP) from the students' perspective, and reveal qualitatively different ways in which this phenomenon can be experienced, understood or perceived (Marton and Booth, 1997).

Phenomenography has proved to be successful as a research approach in studies of learning of computing concepts. Computer networking protocols (Berglund, 2005), programming (Booth, 1992; Bruce et al., 2004), the concepts of object and class (Eckerdal, 2005) and the concepts of information systems (Cope, 2000) can be mentioned as examples. The approach has also been used to explore students' objectives in their studies in higher education (see Ramsden (Ramsden, 1992) for an overview of such studies).

4.3 Work drawing on constructivist theories of learning

Phenomenography and socio-cultural studies are relevant since they can provide insights into the students' own understanding to the object of their learning, and learning as integrated in the environment, respectively. In contrast, constructivism focuses on how a learner comes to learn something. The term constructivism serves as a label for a diverse family of perspectives, dealing with factors such as the role of other learners in the construction of knowledge and how the learning environment influences or interacts with the learner during the learning process. Despite variations in primary focus, these perspectives all share the claim that knowledge is actively constructed by a learner (see for example von Glasersfeld (von Glasersfeld, 1995)).

The influence of constructivism in computing education (and in education in general) is implicit in the frequent use of projects and labs in teaching computing. Through these learning events the students are expected to construct their own knowledge and gain new insights in the issues under scrutiny. However, Ben-Ari (Ben-Ari, 2001) points out that only a handful of papers in the ACM Digital Library explicitly mention the term "constructivism". A recent survey in the same library returns a slightly larger number of papers, however, in most of these the term is used in a general sense or as a label, indicating that the purpose of an assignment or an exercise offered to the students is that they should learn themselves or "construct" their own meaning or knowledge. Only a few of the authors consciously deploy constructivistic theories of learning as a part of their argument.

Among those who apply constructivism, Hajderrouit (Hajderrouit, 1998) can be mentioned. He discusses on theoretical grounds how constructivist theory can be used to enhance students' learning of Java. Aharoni (Aharoni, 2000) studies the cognitive process of students' construction of mental models of data structures. His empirically based work indicates that the students develop a relatively low level of abstraction, despite the fact that a main goal of data structures courses is to obtain a high level of abstraction. His conclusion is that this abstraction barrier must be overcome before implementation of data structures can be taught. This could be achieved by giving the students assignments, in which they can "play" with data structures, much as they would do with concrete objects.

One of Fleury's (Fleury, 2000) recommendations to instructors on how to teach Java as a first programming language is similar: Playing with programs is a means for asking questions and getting answers that are relevant to their current level. Recursion is in focus in the paper of Levy and Lapidot (Levy and Lapidot, 2000). They have discerned, through analysis of a learning situation where different aspects of recursion are highlighted, that there are important differences in the language spoken by students, on one hand, and that used by teachers and in books, on the other hand. It is, they argue, important that teachers are aware of these differences, so that they can aim to close the gap. They also propose that teaching should be organized in a way that the students meet different aspects of the phenomena taught.

Greening "addresses constructivist ideals at the level of implementation" (Greening, 2000, p.96), proposing a programming assignment, that is consciously based on the constructivist theory of learning. Ben-David Kolikant (Ben-David Kolikant, 2001) proposes ways to change a course in concurrency, based on an analysis of how students solve concurrency problems. Her findings indicate that the students invent their own models of concurrency, as they work through their assignments. Ben-Ari's work on constructivism, discussed above, generalizes such aspects and presents general guidelines for teaching computing.

The research projects discussed in this section take their point of departure in how the students themselves construct models of that which they learn. Teaching is investigated from this perspective and it is argued that the teaching and the teaching environment should be designed to encourage good learning and to minimize the risk of that students develop misconceptions in regard to important computing concepts.

4.4 Critical enquiry in computing education research

The critical research tradition is characterized by its foundation in explicitly stated values, and its attempts to reveal and address problems related to power (im-) balances, concerning for example gender, culture or environment, as well as political issues.

Emerging research within critical enquiry in computing education contributes to the whole by addressing new questions, related to power (im-) balances. In fields such as computing, which is male-dominated and founded on Western values, these questions are highly relevant for future research.

This field of research can not be defined in terms of its ontological standpoints or in terms of how the research is performed. An overview of the few research projects performed within this tradition, together with a discussion of their relevance for computing education research is offered by Clear (Clear, 2004). The work of Björkman and Trojer (Björkman and Trojer, 2002) provides a good illustration of feminist research[12] within computing. They investigate, building their argument on some of the core literature within computing, whether there are factors within computing itself that serve to preserve to the current dominant gender structure within the field, and in that case, how this can be addressed. Their conclusion is clearly stated:

> We consider it of vital and decisive importance that gender research is done from within computer science. We argue that focus should be within the discipline itself, and that research should go inwards in search of the core issues in CS. Since the focus is very clearly within the discipline, the work has to be done from within. (ibid., p. 91)

Such research, they argue, on the core of computing and its knowledge production would serve to enrich computing as well as education within computing.[13]

Stein (Stein, 1999) argues, but on different grounds than Björkman and Trojer, that the computational metaphor[14] needs to be replaced with the notion of interacting entities. She argues, based on her view of computation being in crisis, that the metaphor of "computation as interaction", among other effects, would largely influence and change how computing is taught. The need to view computing from a perspective, where the researcher is a part of the results, is also advocated by Nwana (Nwana, 1997). But in contrast to Stein, who bases her argument on a view of computing in a crisis, Nwana states that this crisis is a chimera. By viewing computing and computing education from a constructivist[15] perspective, he finds computing to be successful. The question is important, he argues, since the value that is placed on computing has a strong influence on the subject area and its teaching.

4.5 Multi-faceted approaches

The value of deploying several approaches in the same project within computing education research has been pointed out by Greening. He argues that "the output from a multiparadigm[16] [research] approach to computing education can be expected to provide a more encompassing picture of teaching and learning within the discipline" (Greening, 1996, p. 51). A similar stand has been advocated by Meisalo, Sutinen and Torvinen (Meisalo et al., 2003), who have formatively evaluated a distance course in programming in computing with quantitative and qualitative methods. They use quantitative techniques to analyse issues such as logs and examination rates, while qualitative techniques are used to offer insights in trends and tendencies. An approach to studying learning of programming in a beginner's PBL-based[17] course, with the aim of distinguishing between efficient and inefficient working groups, is presented by Kinnunen and Malmi (Kinnunen and Malmi, 2004). They focus on the interaction in the groups by coding different functions of their conversations, using a coding scheme based on Bales (Bales, 1951) and Flanders (Flanders, 1965). The results of this analysis are then related to findings from interviews, surveys and the grading of the course. The Runestone initiative (Daniels, 1999), where research studies in the different sub-projects adopt different approaches, can be seen as another example of a combined approach.

5 Discussion

Being cross-disciplinary in nature, a pedagogically anchored qualitative CER project is likely to be "read" differently by audiences in the different disciplines. The significance of contributions and results will differ between these disciplines and over the readers, but should have a definite value for each of them, perhaps especially by illuminating how results in one discipline are connected to other disciplines.

Much current educational research does not consider the subject matter (that which is learned). There are, of course, good reasons for pure educational research to make such a delimitation. By adding the role of the subject area, that is, the object of the students learning, as happens in pedagogically anchored computing education research, we argue that the results will have greater impact on the community of teachers of computing. Research projects in computing education, of the type that are advocated in this paper, can also contribute to better teaching and learning of computing, and the theoretical anchoring extends this value beyond the scope of the individual study. This type of general result is thus

---

[12] The term feminist research can serve as a synonym for gender research in the way the terms are used in this paper.

[13] Some instances of critical enquiry can, as has been demonstrated, be understood as computing education research, as described in this paper. The general aim of computing education research, to improve computing education through research, is not value-laden in the same way as the aim of change is important to critical research. While computing education research normally builds on traditional research approaches and aims to apply its results to improve computing education, critical enquiry questions values and the foundations of the research as well as the results.

[14] She briefly explains this term as "an image of how computing works [...] that serves as the foundation for our understanding of all things computational" (ibid., p. 473). In our interpretation, the term is based on a step-by-step image of computing.

[15] In the interpretation of constructivism presented by Nwana, computing is only a value-, time- and context-bound social construct, with the knower and the known inseparable. However, the article is valuable even for a reader who negates the interpretation of constructivism he proposes.

[16] Greening's use of the term *paradigm* broadly corresponds to the word *approach* as it is used in this paper.

[17] The abbreviation PBL stands for Problem-based learning

also of potential value to the education research community.

Despite the potentially high value of pedagogically anchored qualitative research projects our literature survey identified only a few projects where the students and their learning computing are in focus. We thus see a need for a greater number of projects within these research traditions in computing education research.

Establishing a stronger identity and community for pedagogically anchored qualitative research in CER is important. By the pure existence of a community, the sub-field of CER is placed on the map. Additionally, the methodological considerations encountered when conducting a project in a broader context develop the theoretical framework of the sub-field. Finally, a common language for discussing pedagogically anchored qualitative research and research results develops.

6 References

Aharoni, D. 2000). *Cogito, Ergo sum! Cognitive processes of students dealing with data structures*. In Proceedings of the thirty-first SIGCSE technical symposium on Computer science education, Austin, TX, USA, 26 - 30.

Almstrum, V. (1996). *Investigating student difficulties with mathematical logic*. In N. Dean, & M. Hinchey (eds.), Teaching and Learning Formal Methods. London, UK: Academic Press.

Bales, R. F. (1951). *Interaction Process Analysis. A Method for the Study of Small Groups.* Cambridge, MA, USA: Addison-Wesley.

Becher, T. (1989). *Academic tribes and territories: Intellectual enquiry and the culture of disciplines*. Milton Keynes, UK: Open University Press

Ben-Ari, M. (2001). Constructivism in Computer Science Education. *Journal of Computers in Mathematics and Science Teaching.* 20(1). 45 - 73.

Ben-Ari, M. (2004). Situated Learning in Computer Science Education. *Computer Science Education*. 14(2). 85 - 100.

Ben-David Kolikant, Y. (2001). Garneners and Cinema Tickets: High School Students' Preconceptions of Concurrency. *Computer Science Education*, 11(3), 221 - 245 .

Ben-David Kolikant, Y. (2004). Learning Concurrency as an Entry Point to the Community of CS Practitioners. *Journal of Computers in Mathematics and Science Teaching,* (23)1, 21 - 46.

Berglund, A. (2005). *Learning computer systems in a distributed project course: The what, why, how and where*. Uppsala, Sweden: Acta Upsaliensis Universitatis.

Björkman, C., & Trojer L. (2002) *Computer Science and its Paradigmatic Basis - Broadening Understandings through Gender Research from Within.* In C. Björkman, Challenging Canon: The Gender Question in Computer Science. Licentiate thesis. Blekinge Institute of Technology, Karlskrona, Sweden.

Booth, S. (1992). *Learning to program: A phenomenographic perspective.* Gothenburg, Sweden: Acta Universitatis Gothoburgensis.

Booth, S. Berglund, A., Ben-Ari, M. & Holmboe, C. (2004). *What do we mean by theoretically sound research in computer science education?* Panel at the Conference Integrating Technology into Computer Science Education (ITiCSE'04), pp. 230 - 231

Bruce, C., Buckingham L., Hynd, J., McMahon, C., Roggenkamp, M., & Stoodly, I. (2004) Ways of experiencing the act of learning to program: A phenomenographic study of introductory programming students at university. *Journal of Information Technology Education*, 3, 143 - 160.

Carbone, A. & Kaasbøll, J. (1998). *A Survey of Methods Used to Evaluate Computer Science Teaching.* ACM SIGCSE Bulletin, 30(3),41 - 45.

Chew, S. C.., Beaumont, C., Seah, C. P., & Westhead, G. (2004). *Supporting Globally Distributed PBL Teams Using A Rich ICT Environment: How Do Participants Use Different Mediation Tools?* In the electronic proceedings of the Networked Learning Conference 2004, Lancaster, UK. Available at http://www.shef.ac.uk/nlc2004/Proceedings/Individual_Papers/Cheng_et_al.htm

Clancy, M., Stasko, J., Guzdial, M., Fincher, S., & Dale, N. (2001). Models and Areas for CS Education Research. *Computer Science Education*, 11(4), 323 - 341

Clear, T. (2004) Critical Enquiry on CS Education. In Fincher, S., & Petre, M. (Eds) *Computer Science Education Research*, London, UK: Taylor & Francis, 101 - 125.

Cohen, L., & Manion, L. (1998). *Research methods in education* (4th ed.). London, UK: Routledge.

Cope, C. (2000). *Educationally critical aspects of the experience of learning about the concept of an information system.* PhD thesis. La Trobe University, Bundoora, Victoria, Australia.

Crane, D. (1972). *Invisible Colleges: Diffusion of knowledge in scientific communities.* Chicago, IL, USA: University of Chicago Press.

Daniels, M. (1999). *Runestone, an International Student Collaboration Project*. NyIng report No 11. Linköping, Sweden: Linköping University. Also available at http://www.docs.uu.se/~matsd/NyIng.html

Daniels, M., Petre, M., & Berglund, A. (1998). *Building a Rigorous Research Agenda into Changes to Teaching. Proceedings* ACM Australasian conference of Computer Science Education, Brisbane, Australia.

Denzin, K., & Lincoln, Y. S. (Eds.). (1994) *Handbook of Qualitative Research.* Thousand Oaks, Ca, USA: SAGE .

Eckerdal, A. (2005) *Novice Java Programmers' Conceptions of "Object" and "Class", and Variation Theory*. Proceedings of the 10th annual SIGCSE ITiCSE conference on innovation and technology in computer science education, Lisbon, Portugal.

Engeström, Y. (1987). *Learning by expanding. An activity-theoretical approach to developmental research*. Helsinki, Finland: Orienta-konsultit.

Fleury, A. E. (2000). *Programming in Java: student-constructed rules*. In Proceedings of the thirty-first SIGCSE technical symposium on Computer science education, Austin, TX, USA, 197 - 201.

Fincher, S. & Petre, M. (2004). *Computer Science Education Research,* Routledge Falmer, January 2004.

Flanders, N. A. (1965). *Teacher Influence, Pupil Attitudes, and Achievement*. Cooperative Research Monograph No.12. U.S Department of Health, Education, and Welfare. Office of Education. Washington: U.S Goverment printing office.

Gall, M.D., Borg, W.R., & Gall, J.P. (1996). *Educational research: An introduction*. White Plains, NY, USA: Longman Publishers.

Greening, T. (1996). *Paradigms for educational research in computer science.* Proceedings of second Australasian conference on Computer Science Education, Melbourne, Australia.

Greening, T. (2000). *Students seen flocking in programming assignments*. Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education, Helsinki, Finland. 93-96.

Hajderrouit, S. (1998). A constructivist framework for integrating the Java paradigm into undergraduate curriculum. *SIGCSE Bulletin*, 30(3), 105 – 107.

Holland, D., & J. R. Reeves (1996). Activity theory and the view from somewhere: Team perspectives on the intellectual work of programming. In A. Nardi (Ed.) *Context and consciousness: Activity theory and human-computer interaction*. Cambridge, MA, USA: MIT Press. 257-281

Holmboe, C. (2000). *A framework for knowledge: Analysing high school students' understanding of data modelling.* Paper presented at 12th Annual Workshop of Psychology of Programmers Interest Group (PPIG), Corigliano Calabro, Cosenza, Italy

Holmboe, C., McIver, L., & George, C. (2001). *Research Agenda for Computer Science Education.* In Kadoda, G. Proceeding of the 13th annual workshop of the Psychology of Programming Interest Group, 207 - 223.

Kinnunen, P. & Malmi, L. (2004). *Do Students Work Efficiently in a Group? - Problem-Based Learning Groups in Basic Programming Course*. In A. Korhonen & L. Malmi (Eds.) Kolin kolistelut - Koli Calling 2004. Proceedings of the Fourth Finnish/Baltic Sea Conference of Computer Science Education, Helsinki University of Technology, Helsinki, Finland, 57 - 66.

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. New Jersey: Cambridge University Press.

Levy, D. & Lapidot, T. (2000). *Recursively speaking: analyzing students' discourse of recursive phenomena.* In Proceedings of the thirty-first SIGCSE technical symposium on Computer science education, Austin, TX, USA, 315 - 319.

Marton, F., & Booth, S. (1997). *Learning and Awareness*. Mahwah, NJ, USA: Lawrence Erlbaum Associates.

McIver, L. (2000). *The Effect of Programming Language on Error Rates of Novice Programmers*. Paper presented at 12th Annual Workshop of Psychology of Programmers Interest Group (PPIG), Corigliano Calabro, Cosenza, Italy.

Meisalo, V., Sutinen, E. & Torvinen, S. (2003) *Choosing Appropriate Methods for Evaluating and Improving the Learning Process in Distance Programming Courses.* In the proceedings of IEEE Frontiers in Education, Denver, CO, USA.

Mester, A., & Krumm, H. (2000). Animation of Protocols and Distributed Algorithms. *Computer Science Education,* 10(3), 243 - 266.

Nwana, H. (1997). The computer science education crisis: fact or illusion? *Interacting with Computers*, 9(1), 27 - 45

Pears, A. & Erickson, C. (2003). *Enriching On-line Learning Resources with Explanograms*. In Proceedings of the 1st international symposium on information and communication technologies, Dublin, Ireland, 261 - 266.

Pears, A., Daniels, M. & Berglund, A. (2002). *Describing Computer Science Education Research: An Academic Process View*. SCS 2002 Western Multiconference, San Antonio, USA.

Petre, M., Fincher, S. & Tenenberg, J. (2003). *"My criterion is: Is it a Boolean?": A cardsort elicitation of students' knowledge of programming constructs.* Technical Report, University of Kent at Canterbury, U.K.

Ramsden, P. (1992) *Learning to Teach in Higher Education.* London, U.K.: Routledge

Robin, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137 - 172.

Säljö, R. (2000). *Lärande i praktiken: Ett sociokulturellt perspektiv*. [Learning in practice: A socio-cultural perspective] (in Swedish). Stockholm, Sweden: Prisma.

Torvinen, S.: (2002) *An overview of Virtual Approbatur*. Proceedings of the Second Annual Finnish/Baltic Sea Conference on Computer Science Education, University of Joensuu, Report A-2002-7, Koli, Finland, 83-85.

Stein, L. A. (1999). Challenging the Computational Metaphor: Implications for How We Think. *Cybernetics and Systems* 30(6). 1 - 35.

Valentine, D. (2004). *CS Education Research: A Meta-Analysis of SIGCSE Technical Symposium* Proceedings. In Proceedings of the thirty-fifth SIGCSE technical symposium on Computer science education, Norfolk, Va, USA.

von Glasersfeld, E. (1985). *Radical Constructivism: A Way of Knowing and Learning*. London, UK: Falmer

Vygotsky, L. (1986). *Thought and Language* (A. Kozulin, translation.). Cambridge, MA, USA: MIT Press..

Wiedenbeck, S. (2005) *Factors Affecting the Success of Non-Majors in Learning to Program* In the Proceedings of the First International Computing Education Research Workshop (ICER), Seattle, WA, USA

Wu, C.-C. (1993). *Conceptual models and individual cognitive learning styles in teaching recursion to novices*. PhD thesis, University of Texas at Austin, Austin, TX, USA.