ACTA UNIVERSITATIS UPSALIENSIS
Uppsala Dissertations from
the Faculty of Science and Technology
62

Anders Berglund

Learning computer systems in a distributed project course

The what, why, how and where

UPPSALA
UNIVERSITET

Dissertation for the Degree of Doctor of Philosophy in Computer Systems presented at Uppsala University, 2005.

Abstract

Berglund, A. 2005. Learning computer systems in a distributed project course: The what, why, how and where. *Acta Universitatis Upsaliensis. Uppsala Dissertations from the Faculty of Science and Technology 62.* 232 pp. ISBN 91-554-6187-5.

Senior university students taking an internationally distributed project course in computer systems find themselves in a complex learning situation. To understand how they experience computer systems and act in their learning situation, the what, the why, the how and the where of their learning have been studied from the students' perspective. The what aspect concerns the students' understanding of concepts within computer systems: network protocols. The why aspect concerns the students' objectives to learn computer systems. The how aspect concerns how the students go about learning. The where aspect concerns the students' experience of their learning environment. These metaphorical entities are then synthesised to form a whole.

The emphasis on the students' experience of their learning motivates a phenomenographic research approach as the core of a study that is extended with elements of activity theory. The methodological framework that is developed from these research approaches enables the researcher to retain focus on learning, and specifically the learning of computer systems, throughout.

By applying the framework, the complexity in the learning is unpacked and conclusions are drawn on the students' learning of computer systems. The results are structural, qualitative, and empirically derived from interview data. They depict the students' experience of their learning of computer systems in their experienced learning situation and highlight factors that facilitate learning.

The results comprise sets of qualitatively different categories that describe how the students relate to their learning in their experienced learning environment. The sets of categories, grouped after the four components (what, why, how and where), are synthesised to describe the whole of the students' experience of learning computer systems.

This study advances the discussion about learning computer systems and demonstrates how theoretically anchored research contributes to teaching and learning in the field. Its multi-faceted, multi-disciplinary character invites further debate, and thus, advances the field.

Keywords: Computer science education, computer science education research, computer networking, computer systems, phenomenography, activity theory, learning, higher education, team collaboration, remote collaboration.

Anders Berglund, Uppsala University, Department of Information Technology, DoCS, Box 337, SE-751 05 Uppsala, Sweden

ISBN 91-554-6187-5
ISSN 1104-2516

Printed in Sweden by Elanders Gotab, Stockholm 2005.
Distributor: Uppsala University, Box 510, SE-751 20 Uppsala, Sweden.
www.uu.se, acta@ub.uu.se

To my mother

CONTENT

I. SETTING THE SCENE

II. METHODOLOGICAL DEVELOPMENT

III. EMPIRICAL INVESTIGATIONS

Summary

How can students' learning in a complex learning environment be understood and studied using approaches founded in pedagogic research? This question is the focus of this thesis about learning computer systems in the context of an international distributed project course.

The students who followed the course, as a part of their education in information technology or computer science, worked in teams of six, where three of the members were in Sweden, while the remaining three were in USA. The team members jointly produced a software system which controlled a computerised wooden toy. Computer communication problems are vital to producing a working system. The selection of which network protocols to use and which rules for communication to apply are, thus, central elements of the project.

An analytic separation has been performed with regard to what, why, how and where the students learn. The what aspect describes that within computer systems that the students learn: network protocols. The why aspect describes the students' motives for taking the course. The how aspect describes how the students go about learning computer systems. The where aspect describes the learning environment, as it is experienced by the students.

Data has been collected through interviews with the students about their learning and their learning environment. The analysis reveals the qualitatively different ways in which the four aspects are experienced without categorising individuals.

What: Four different ways of understanding network protocols have been identified: as communication between two computers; as a connection over a network; as a set of rules; as a standard. The different ways of experiencing a protocol are relevant with different tasks at hand in a development project in computer systems.

Why: Three different motives for taking the course have been identified: academic achievement; capacity to work in projects; social aspects of learning. The three motives can in their turn be experienced in different ways. For example, four different ways to experience academic achievements have been discerned: to get a grade; to learn computer science for the project; to learn how to learn computer science; to learn something new.

How: Seven different ways to act in order to learn computer science have been identified. They differ in aspects such as if the learning concerns unrelated phenomena, whether the parts or the whole of a computer system is in focus, or if the learning relates to a personal development as well as to the subject area.

Where: The environment in which the students learn is complex and comprises many factors. Different ways to experience the control structure in a team, a decision taken in the team, the value of the meetings with the teachers, and the experience of being graded in this course have been analysed and described in categories. Tensions or contradictions have been identified between certain factors. For example, an ad hoc based group structure, where a social game is a key feature of the team, could lead to conflicts. On the other hand, a structure, where important decisions are taken jointly, normally leads to a stable situation.

The desire to focus on students' experiences motivates an empirical, qualitative, phenomenographic research approach. The different components of the students' experiences of their learning are integrated by extending the approach using elements of activity theory. The context of the learning has been examined from different perspectives. A differentiation is made on the basis of who experiences a certain context in a certain situation. A student experiences a context as a background to the phenomenon that is discussed in an interview situation. The collective context is visible during the phase of analysis and encompasses the individuals' contexts, but supersedes them, in that the holistic perspective offers new insights about the context. The researcher experiences a context, that encompasses his understanding for the students' contexts, both the individual and the collective, as well as his or her own experience of the situation, the subject area and the research. Finally, a methodological framework is developed to integrate the insights that are gained into a whole.

The synthesis highlights relationships between the four aspects described above. Two over-arching qualitatively different ways to experience the learning situation have been identified. The situation can be experienced in terms of meeting the formal course requirements, or as an opportunity to learn about computer systems. While the first is characterised by contradictions, the latter offers better opportunities for focused learning.

A model is developed which relates what the students learn to how they go about learning, their motives in learning something and how they experience their learning environment. For instance, imagine a student who experiences his or her study situation as focused on learning computer systems, and who acts to learn something new by gaining personal experiences. He or she is in a better situation to develop a nuanced understanding of network protocols than another student, who is dominated by the requirements of the environment and who tries to pass the course by learning certain concepts by heart.

This thesis is, as a inter-disciplinary study, intended for different readers with varying needs. It can be read against a background of computer science, computer science education research and pedagogical research.

Many opportunities for a teacher to improve his or her teaching can be found in this complex interactions uncovered by this study. To identify and develop those factors in the environment that serve to enhance learning is simultaneously powerful and hard to master.

An important contribution of this thesis is to outline the foremost questions in making efforts to improve student learning in computer systems. The thesis identifies complex relationships and results that go beyond that which normally is taken for granted by teachers and students. It has resulted in new insights into the complex nature of the relationships between the what, why, how and where of the students' experience of learning of computer systems in a distributed project. Fundamental questions concerning students' learning of computer systems and the relationship to their experienced learning situation emerge as a result.

I. SETTING THE SCENE

Chapter 1. Introduction

This thesis discusses how senior undergraduate students come to learn about computer science concepts, such as computer network protocols, within an course in computer systems, in which the students work in a distributed project teams.

1. Research questions

The starting point for this investigation is the research questions I formulated in the grant application to fund my doctoral studies to the Knowledge Foundation in Sweden:

> The question I want to address in my research is, in what ways students, who take part in international collaboration as part of courses in computer science, experience their learning and collaboration. The question can be analysed into a number of aspects which relate to
> - how the collaborative learning situation is experienced and tackled
> - how ICT is experienced as support for peer learning and peer teaching
> - how the situation is seen as a contribution to future professional life
> Further, these can be examined in the light of the outcomes of learning in the specific course context.

These research questions have endured as focus points of the research, although their meanings have inevitably evolved.

2. Contributions

The contributions of this thesis are four-fold:

Firstly, insights are gained into the learning of computer science with the aim of improving learning and teaching. Considering the need for a renewal of undergraduate education in computer science, results concerning learning of computer science are relevant.

Secondly, how students experience their collaboration in a distributed project-based course is studied. The conclusions contribute to the development of ICT-supported[1] courses that span different sites, universities and even countries, since insights into how the students experience their situation is a powerful tool for a course developer and teacher.

Thirdly, this study informs the discussion of how teaching should promote the future professional life of the students. The role of projects and distributed learning environments in the computer science curriculum can today hardly be underestimated. Professionals need to collaborate, and as a consequence, teaching methods intended to stimulate collaboration are highly relevant.

Fourthly, the research process as such contributes to teaching and learning of computer science. The project uses a theoretically anchored way of studying the students' learning of computer science, and relates the results concerning learning to the experienced learning environment. It furthers the discussion concerning learning and teaching by being grounded in both what the students learn, here computer science, and theoretically solid pedagogical research.

3. My rationale

My objectives in embarking on the journey which led to the current thesis can be found in my professional background. Working for a long time as a lecturer in computer science, I found that my interest was divided between computer science per se and the teaching of computer science. From here I came to reflect upon how students go about learning computer science, and how teaching could be improved based on insights in the students' learning of the subject area.

The students' study object is then computer science; computer science, as it is understood by the students, comes to be one of the cornerstones of this thesis. Although the ultimate aim is better teaching and learning in computer science, the focus in the thesis is on how students understand and learn about computer networking. The rationale for this focus on the learning of the subject area is expressed in the following way in Ramsden (1992):

> Teaching and learning in higher education are inextricably and elaborately linked. To teach is to make assumptions about what and how the student learns; therefore, to teach well implies learning about students' learning. (p. 6).

Thus, understanding how students understand concepts within the subject area, the conditions for learning and how the learning takes place, are powerful tools for teachers who want to improve their teaching. And my route to this understanding has been to carry out the research that is embodied in this thesis.

---

1. The abbreviation *ICT* stand for *Information and Communication Technology*

My research has been carried out in interaction with the computer science education community. Not only have I discussed my work with my supervisors and colleagues, but I have also published work in international journals and conference proceedings and presented my work at seminars. The reviewed articles are listed in Appendix A.

Part I of this thesis describes the setting for my work: the course the students take, computer networking and the background in phenomenography and activity theory. In part II, I describe the methodological tools that I have developed to use in my research. Part III describes the empirical investigations, while part IV is devoted to the results in the form of a full picture and a discussion of the significance of the outcome of the thesis.

Chapter 2. The Runestone initiative

The research presented in this thesis is situated within an internationally networked project: the Runestone initiative. The initiative consists of three components: (1) an internationally distributed project-centred course in computer systems (2) a pedagogical development project and (3) research into computer science education.

During its initial years, the project was funded by a grant from the Swedish Council for the Renewal of Higher Education, and has also been sponsored by NyIng (a Swedish initiative for the development of engineering education) and Uppsala University. The course is still (2005) running, now without any special funding. Since the initial years, staff have changed at both sites, and new hardware has been deployed. Over 500 students have taken the course over the years. Within the initiative over 25 reviewed conference publications and two journal contributions have been published[2]. Two PhD theses (Last, 2003; Hause, 2003) and one licentiate thesis (Berglund, 2002) are also based on data from this initiative. Thus the initiative is many ways a success story (Daniels, Berglund and Pears, 2003).

This chapter first describes the course that the students took and then briefly touches upon aspects of the pedagogical development. The final section discusses relevant work within the initiative.

1. The undergraduate course within the Runestone initiative

The learning that is investigated takes place in a course about distributed computer systems and real-time programming (Daniels, 1999). During the course, the students jointly, in internationally distributed teams, develop software systems that are intended to solve a technically advanced computer science task. The course, given jointly by two universities: Grand Valley State University, Allendale, MI, USA (GVSU) and Uppsala University, Uppsala, Sweden, (UU) has been offered since 1998.

2. `http://www.docs.uu.se/docs/runestone/index.html`

1.1. The institutional environment

The course participants major in computer science at the two universities that participate in this project. They take the course during their third (UU) or fourth year (GVSU). The universities are different in their respective foci: Uppsala University, with its 30 000 students is an internationally recognized research university, while Grand Valley State University, with over 20 000 students, mainly serves the local community and the region offering education at Bachelor's level, professional degrees, and Master's degrees in some subject areas, such as in computer science. Being in different countries, the universities are parts of different educational systems.

A difference between the university milieux can be found in the socio-economic conditions for the students. Education is free in Sweden, and the vast majority of Swedish students profit from governmental study loans and grants. American universities charge for tuition, and the students, or their parents, finance their living costs. Many students at GVSU work half-time or more, while only a few Swedish students have professional occupations in parallel with their studies. As a consequence, the working hours of the American students has been shown, in some teams, to put serious constraints on possible meeting hours, and to be an influential factor in the general planning of student time commitments.

1.2. The students' task

Data for this research project was collected during the spring term 2001. During this year, there were 14 teams of five or six students, each team comprising students in approximately equal numbers from both universities, collaborating mainly by e-mail and Internet Relay Chat, IRC[3]. The task assigned to the students was to write a program that offers to the end-user the ability to control a Brio labyrinth (see figure 1).

The labyrinth is a Swedish toy, the aim being to manoeuvre a steel ball from a starting point to a final point on the board, by tilting it so that the ball moves without falling into any of the holes. The original labyrinth has, as is shown in the left picture of figure 1, knobs that are used to control the angle of the board. The labyrinth was modified to have motors to control the board and a camera to give feedback to the controlling software system, as shown on the right of figure 1.

3. *Internet Rely Chat*, *IRC*, is a system for human communication over Internet. A computer running an IRC program can be used in a way similar to a text telephone and offers to the user a possibility to communicate with any other IRC user in the world.

 

Figure 1. A Brio labyrinth, and a modified version with a camera and motors added

On the Web-page related to the course[4] the students' project was described in the following way:

> This project involves designing and implementing a distributed, real-time system to navigate a steel ball through a board by tilting the surface of the board via positioning motors. The board and ball are a modified version of the well-known Brio Labyrinth game. A monochrome digital video camera focused on the board is available to aid in navigation. The user interface is presented through a web browser. Users who play the game specify a path for the ball to follow, then get feedback on the result of their run.

The project has elements of real-time control (the Brio game), low-level distributed systems (multiple CPUs to gather data and to drive the motors), and high-level distributed systems (web interface, network programming), in addition to some demanding requirements on the language used to implement portions of the project (dynamic code loading, security).

As should be clear from the description above, this is a rather large and complex project. Several smaller sub-problems had to be solved and later integrated, in order to create a working software system. The time for the full task was limited to approximately 8 weeks to fit the universities' requirements on semester lengths, exam periods etc.

During the spring term of 2001, the students were given a slightly modified task compared to the previous years. They were handed program packages that had been produced in the previous spring term and were asked to improve it by making three major changes of their own choice. This year one team managed to complete the task and produced a working software system that to a significant degree corresponded to the specifications, while the other teams presented prototypes that were not judged to be complete by the teachers. Still the result was not a disappointment for the universities.

4. http://www.csis.gvsu.edu/class/brio/BrioProject/ProjectDesc/BrioProjectOverview.html

They had anticipated that only a few or possibly none of teams would complete the task; prototypes at different levels of completeness were the expected outcome.

1.3. The learning objectives from the universities' perspective, the official “what”

Looking at the official documentation at the two universities, descriptions of the course content can be found. At GVSU the course is the senior project course for majors. The following course objectives are described[5]:

> Experience software maintenance and development phases. Integrate experience and knowledge from other courses and apply them to a project. Experience working in a distributed team.

At UU the Runestone project is the concluding part of a large course that spans three-quarters of the academic year. The project corresponds to one third of this course, and comes at the end of the full course. It is preceded by coursework on computer networking, real-time systems and distributed systems. The aim of the full course is described thus[6]:

> The course provides basic knowledge of the design of distributed systems and their underlying communication subsystems with special focus on real time and embedded applications and control systems.

When the project starts, the UU students have encountered theoretical aspects of the course content in the teaching, and have done several smaller practical labs. The course content for the full course is described in the following way:

> [...] Methods for achieving user transparency, e.g. synchronization, interprocess communication, distributed control and consistency primitives. Time handling, fault tolerance, language support and scheduling for real time control. Case studies.

The formal educational framework, into which the project should fit, is established by these descriptions. It should be a senior project for majors, where a software system should be developed. According to the GVSU specification, it requires application of experiences from earlier courses. UU is more explicit about the content of the project, which should encompass elements of computer networking, distributed systems and real time control.

---

5. `http://www.csis.gvsu.edu/class/brio/CourseSpecific/467Syllabus.html`

6. `http://dbteknat.its.uu.se/cgi-bin/dbteknat/dbteknat?crsECTS.hpg&0&eng&1IT070`

1.4. The collaboration from the universities' perspective, the official “how”

The course objectives, as they are presented by the two universities, do not specify the technical content of the course in detail, and are still more open when discussing how the international project should take place.

The web-site for the course[7], used jointly by the two universities, gives more detailed information about how the course was planned for the spring of 2001. It states:

> There are two major aspects of this project: Developing the software. Building a virtual work team.
>
> Software development involves splitting up the work and allocating it to members of the group, and making sure that your group understands what is happening in the project. Consequently one of the major features of this project is for each group to have a regular contact with one of the teaching staff to report on the progress they are making and to ask questions that might develop.

One of the teams had only Swedish participants. Since this team consisted of some very advanced students, they were assigned a different task and are not considered in this research project. Two teachers, one from each university, taught the course in collaboration. There was also specific technical support available for the students with issues like operating systems and practical questions concerning the functioning of the Brio board. At GVSU this service was offered by the technical staff of the department. In Sweden this job was given as a task to the team that was formed only of Swedish participants.

Each team of students was assigned one teacher, either in Sweden or in the US, who coached, supervised and graded the team. As a result approximately half the students in each team, were supervised by a teacher whom they had never met, and who did not recognize their faces. It was decided to hold weekly meetings with the teachers, called milestone meetings, where the teams should report the progress they had made, and discuss problems and other issues that had arisen during the week.

The first sub-task for the teams was to plan the rest of their activities. They had to choose a team-leader, select which code[8] to work with and decide which modifications to make on the code.

The grading was based both on the students’ work process, mainly evaluated through the weekly meetings, and the outcome of the work. There were both individual and team-based components in the grading. The

---

7. `http://www.csis.gvsu.edu/class/brio/BrioProject/`

8. The term *code* refers to the program packages, produced by the students during the previous year. The term is used throughout the thesis, to keep close to the terminology used during the course.

grading is further discussed in chapter 13, as well as in Pears, Daniels, Berglund and Erickson (2001), and Daniels, Berglund, Pears and Fincher (2004).

1.5. The tools for collaboration

For all interaction with the teachers, as well as with team members at the other university the students used different ICT-based tools. One initial physical meeting was arranged for the Swedish students, in the US a few meetings were organized, mainly to teach Java. The choice of which tool(s) to use for the collaboration was left to the students to decide. Except for a preference from the teachers to use IRC for the weekly meetings, no restrictions were put on the teams' choice of communication and collaboration tools. All teams used IRC and e-mail for their daily work, in some cases with password protected web-pages as an additional resource. Some teams also used CVS[9].

2. A technical description of the student project

On the web-site of the course, a technical description of the setting and the requirements for the desired results was available. Figure 2 shows the principal hardware and software components. The system as a whole consists of some inter-connected sub-systems that might run on the same computer or on separate computers. The hardware, operating systems, standard communication solutions etc. are supplied by the two universities, while the task of the students is integrate these parts.

The software should allow an end-user to draw a path that he or she wants the ball to follow in a web-browser. The ball should follow this path, through an automatically controlled tilting of the board. The resulting path, that always differs to some extent – smaller or larger – from the desired path, should then be shown on the screen. The movement of the board is controlled by stepper-motors are connected to a server, often called the game server by the students and the staff. It is marked ① in figure 2 and can be seen as the centre of the system.

The system needs to keep track of the movements of the ball in order to control the motors in a relevant way. This requires feedback, which is provided by a camera that constantly observes the board, as shown in right picture of figure 1. The purpose of the video server (marked as ② in figure 2) is to interpret the images from the camera and transform the information

9. The abbreviation *CVS* stands for Concurrent Versions System. CVS is a version control system that records the history of files with their changes.

the camera offers into information about the ball: its position, speed and direction of movement etc.

The game server acts as the coordinator of the system, getting information from the camera through the video server and information on user's demands from the applet[10]. From this information the game server should calculate how the motors should move, and send the required information to the motors for these movements to take place. This server should also provide the information about the movements of the ball and the status of the system to the applet.

Except for the cameras, where different brands with different characteristics were used, the hardware settings were basically the same at both places. The Brio boards and the physical equipment were getting old in 2001, and the variations between the different boards adds new problems for the students. The system ought to be written in a way that makes it work correctly on most of the boards.

As should be clear from this description, communication between computers, virtual machines[11] or hardware controllers is an important element of the system.

3. Pedagogical development within the Runestone initiative

In a project of this kind it is hard, not to say impossible, to draw a firm line between research and pedagogical development. The initiative has been the basis for a network, where the different actors have had different individual aims for participating (for example to find data for a thesis, to teach a course, to study the project as a testbed). The actors have jointly worked for the development of the initiative; everyone being dependent on both his or her own work and the activities of others. After the initial phase of the initiative, during which journeys and working time were paid for, the actors have been obliged to find their own funding.

---

10. *Applets* are Java programs that are intended to be run in a web browser, such as Netscape, or by a dedicated appletviewer. Applets are frequently used to implement graphical interfaces for web-based applications.

11. A *virtual machine* can be described as a simulated computer that runs on another computer. In other words, a virtual machine is a program, that, when executed, behaves as a computer with well-defined properties. Virtual machines are one of the underlying techniques for platform-independent programs. Java that can be run on different kinds of computers and in different environments has a virtual machine. Java's virtual machine is (at least in theory) the only program that has to be rewritten to run Java in a new environment.

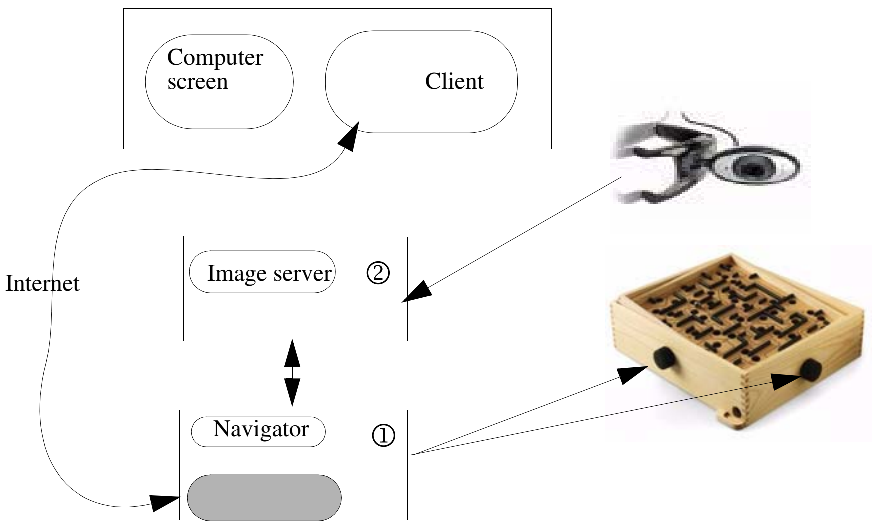

Figure 2. The architecture of the Brio system. The grey oval symbolises a web-server.

As an example of the mutual interaction between the pedagogical development and the research[12], the role of grading within the Runestone course can be mentioned. At a certain time, this was the theme for a vigorous debate. It was argued that it was unfair to have a course with different grading schemes for the participating students (Pears et al. 2001; Daniels et al., 2004). Had there not been research performed within the initiative, that could serve to mediate in the debate, it is not likely that the course would have continued (Mats Daniels, private communication).

Another result that overlaps the hypothetical border between research and development, can be found in the narrative papers, telling the "Runestone story" that have been produced within the initiative (such as Daniels, 1999; Daniels et al., 2003; Last, Almstrum, Daniels, Erickson & Klein, 2000; Last, 2002). Such papers both serve as an exchange of the ideas between the

12. Since this thesis is concerned with the students' learning of computer science, a clarification of the differences and similarities of the two terms *research* and *pedagogical development* falls outside the scope of this project. Instead, the two terms are used in an intuitive way.

practitioners, inside or outside the initiative, and as a source upon which theoretically illuminated research can draw.

4. Research within the Runestone initiative

Although the different research projects that stem from the initiative use the same student project as their object of research and as source for the empirical data, they have distinctively different research questions and theoretical backgrounds. The different research questions complement each other, and the results only occasionally overlap. Such overlaps function as confirmations of the results, since similar conclusions have been reached in different ways.

Hause and Woodroffe (2001) and Hause (2003) study the communication and interaction within the student teams, in order to find characteristics in the interaction patterns within teams that perform well and those that performed poorly in software engineering. Data is collected from the e-mail conversations as well as IRC sessions between the team members. In her thesis (Hause, 2003), data is coded according to a set of categories developed within the research project using discourse analysis. She has demonstrated that communicating among team members is important, and that the timing of the interaction is crucial. High performing teams communicated less, and were more organised in the way they conducted their meetings and work. Furthermore, she argues that the control structure, and leadership style in the software development process as a whole was crucial.

The team development in the student teams and the role of conflicts form the focus of the work performed by Last (2003), who uses grounded theory[13] as a research approach. In her work, she investigates if team development models developed and validated with face-to-face teams require modification when applied to virtual teams and if certain types of conflict in a team result in a more productive team and a better product.

Pears, Daniels, Berglund and Erickson (2001) and Daniels, Berglund, Pears and Fincher (2004) address the issues of the impact of the different grading scales on the students' motivation to contribute to the work of the

---

13. Last (2003) describes her interpretation of grounded theory as an "interpretative methodology that has its own rigorous method for data collection and data analysis. The term grounded theory derives from the requirement that theory must emerge from data, that is, theory must be grounded in data. This approach is inductive rather than deductive. [...] Research questions are open-ended rather than stated hypotheses to be proved or disproved, and the emergent theory should explain a phenomenon that is relevant and problematic for those involved. The grounded theory approach 'involves deriving constructs and laws directly from the immediate data that one has collected rather than from prior research and theory' (Gall, Borg, & Gall, 1996, p. 10)." A classic reference is Glaser & Strauss, 1967.

team. There it is argued, based on a statistical analysis of the peer evaluations, that the different grading scales did not affect the students' level of input in the project.

Daniels, Faulkner and Newman (2002) discuss open-ended group projects (OEGP) in computer science education in a comparative study and create a framework for describing such projects as the Runestone initiative. They argue, based on student evaluations, discussions with employers and their own experiences, that OEGP projects, where the end-product is not well-defined, are valuable in preparing the students for their professional lives.

Chapter 3. Computer networking

Communication between computers can be analysed, designed, and described in different ways. Focus can be on different aspects of the communication, whether it is the physical transmission of raw data or a semantically rich communication between two application programs such as mail-programs, or at any level between these two. A strategy is needed to handle this complexity in order to make the design of a computer network and its components a feasible task. Layered design has historically been the dominating methodology to tackle this situation and to define tasks that can be further developed.

This chapter first presents the idea of peer communication and then describes layered models and the layers of the TCP/IP[14] stack (Stallings, 2004). RMI is described, before the final section where aspects of practical programming in the Runestone project are discussed.

1. Peer communication

An *internet* or an *internetwork* consists of a set of independent networks, that each has its own character. The Internet makes use of the layered structure of the TCP/IP protocol stack, in which each layer offers services to the level above it. The services offered by one layer are implemented with the help of services offered by lower layers.

An internet comes to "look" different for users at different levels. A user sending an electronic mail through a mail program experiences an internet differently from a student who works on the Runestone project with a routine that implements communication between the camera and the game server (see chapter 2). While the former uses a program that in its turn implements the SMTP protocol, the latter is most probably writing a C++ or Java program that uses the services of TCP.

Figure 3 illustrates this with an example. An end-user deploying a program that implements the FTP (File Transfer Protocol) protocol sees his program and the communication with another FTP program, as the FTP

14. *TCP/IP* stands for *Transmission Control Protocols* and *IP* for the *Internet Protocol*. Abbreviations are rarely spoken out within the field of computer communications. Acronyms are used as names of the protocols as well as the other entities discussed.

protocol prescribes. Thus for him there is *peer communication* (marked as *a* in the picture) between two FTP programs. The user of this program does not need to be aware of the underlying levels.

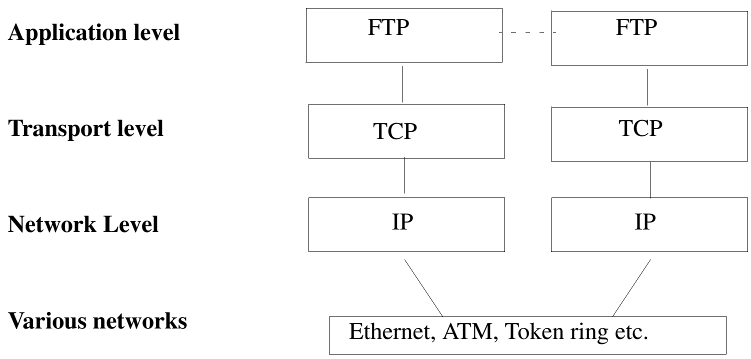

Figure 3. An example of peer communication in a layered model. The dotted line indicates peer communication between two FTP programs, as it is perceived by the user.

2. The TCP/IP protocol stack

Basically, there are two layered architectures that are discussed today: The Internet architecture, which is frequently referred to as the TCP/IP architecture (see figure 4) and the Open Systems Interconnection architecture, OSI. The former completely dominates practical applications and will be in focus in this presentation.

The TCP/IP protocol hierarchy allows communication over networks with different characteristics and makes it possible to create application programs for large numbers of purposes.

Application level

At the *application level*, a large number of protocols can be found that offer a rich variety of services to the user of a computer on an internet. As an example can the Hyper Text Transfer Protocol, HTTP, be mentioned. It defines the rules for the communication between a Web-browser (Netscape, Internet Explorer, Mosaic, Lynx etc.) and a web-server, which stores and organises web pages. This means that anyone who wants to create his or her own web-browser (or web-server) needs to write a program that follows the

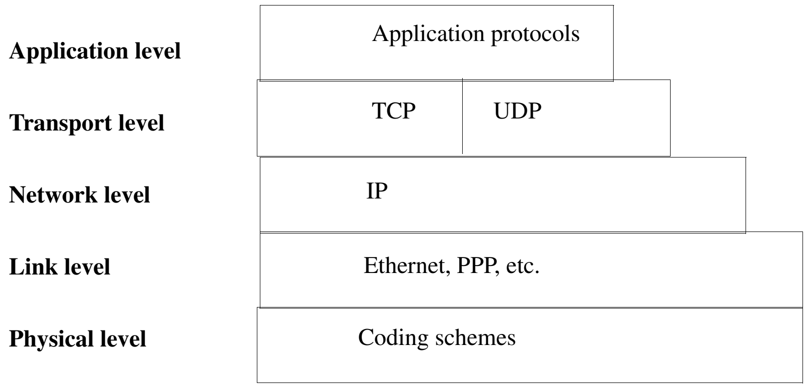

Figure 4. The TCP/IP layered model

rules of the HTTP[15]. Examples of other application level protocols are mentioned in table 1.

Table 1. Example application protocols

Abbreviation	Full name	Purpose
SMTP	Simple Mail Transfer Protocol	Transfer of mail
NNTP	Network News Transfer Protocol	Transfer of news and support for news reading in USENET news
FTP	File Transfer Protocol	Transfer of files between computers
SIP	Session Initiation Protocol (SIP)	Internet telephony
NV	Network Video	Video Applications

Transport level

The *transport level* offers the possibility for two computers to keep an end-to-end exchange of data. This service is organised in a layer of its own, since it is needed by many applications. The dominating protocol at this level is the *Transmission Control Protocol*, TCP.

TCP is a reliable, connection-oriented protocol. This means that the protocol allows communication where data is delivered to another machine on the internet without errors and in the order it was sent. A user of the TCP protocol needs to set up a connection in order to communicate with another

15. RFC 2616

computer, abstractions that to a large degree resembles the operational semantics of a telephone call.

UDP, User Datagram Protocol, is an unreliable, connectionless protocol. This means that a sending computer does not get any confirmation whether the data sent has reached its destination or not. Still, it is useful in applications where but where speed is important, such as video conferencing or telephony. A user of a video connection can more easily accept a loss in the quality of the image than delays of various lengths.

Network level

The *network leve*l handles the heterogeneous structure of the underlying networks, large size of an internetwork, and its continuously changing topology. It mainly offers two services to the level above: it handles addressing on an internet-wide level and offers tools for delivery of data to the destination. In other words, the protocol at the network level, the *Internet Protocol* (IP) accepts packages of data from a higher level, translates addresses so that they correspond to physical interfaces of destination computers and forwards the data to its destination using the services of the underlying levels.

Data are sent by the IP protocols as IP *datagrams*, packages of a limited length (less than approx. 65 000 bytes) that each contains a part of or the full message. The datagrams "travel" independently over the net; the task of the IP protocol is to try to deliver them. Frequently datagrams needs to take many steps and pass several routers to reach its destination. The IP protocol is a *best-effort* protocol; it tries its "best effort" to send data, but does not guarantee any qualities of the service.

The underlying networks

These networks are connected to each other through *gateways* or *routers*. The independent networks that form an internet can for example be Local Area Networks, LANs, or other internets. An internet is thus a logical network, that consists of a collection of physical or logical networks

3. Remote Method Invocation

Remote Method Invocation, RMI, provides programmers with a facility to access code or objects on a remote machine. It is implemented on top of

TCP/IP and is closely related to the object-orientation[16] in the programming language Java. The execution of an object-oriented program takes place in the objects and in the interaction between the objects. Java RMI offers the possibility to execute Java programs where the different objects are distributed on different machines over an internet. To a programmer this means that RMI provides access to routines (called *methods* within the terminology that normally is used when Java is discussed) that exist on remote machines as if they were available on the local computer.

When transferring unformatted text between computers, all characters are, in the general case, treated in the same way. There is no semantically complex information in the characters that influences how the transfer should be made, with the exception of well-defined situations as indications of the end of transfer. When communicating between objects the situation is different: Part of the data that is transferred is information about other parts of the data. A call to a method on an object contains information about addresses, permissions, data to be transferred etc. This extra complexity simplifies the construction of programs that are distributed over a set of computers, but demands that the programmer handles security issues correctly.

4. Computer networking in practice in the Runestone project

The term *protocol* (often referred to as *network protocol* or *computer network protocol*) is used both for the abstract entity of formally defined rules and for the implementation of the services. Protocols contain information about message formats, formats for control information, responses to messages, timing requirements as well as information about how errors or other unexpected events should be handled.

Network protocols are standardised. Software that offers the programmer the routines that are needed to handle the communication have been developed for important protocols. For example, TCP offers, among other routines, functions that listen for incoming messages, set up connections, send messages and close down connections.

There are several possible protocols that can be used within parts of, or the whole project. Three protocols have frequently been used in the Runestone project: TCP, UDP, and RMI. Some program packages that

16. *Object-orientation* is based on the idea that a program consists of a set of communicating entities. Java and C++ are programming languages that support this style of design and programming. There is a vast literature on object-oriented programming and object-oriented programming languages. Budd (1999) discusses the ideas behind object-orientation and Java.

technically would suit the Runestone project, such as CORBA (Common Object Request Broker Architecture), have never been used during the years of the Runestone project (Arnold Pears, private communication).

TCP sockets, or the *TCP socket programming interface* are used by a programmer who wants to access the services of TCP. Sockets offer to the programmer practical routines, or mechanisms, to transfer data from one computer to another. A socket is the endpoint of a communication connection between two computers. This endpoint can be manipulated by a programmer using library routines. While the concept of sockets, stemming from UNIX, is now generally accepted, details of the implementations vary to a certain degree with different programming languages and operating systems. Java offers an abstracted, standardised interface for sockets that is not dependent on the operating system.

An example of the how sockets are used for a normal life cycle of a TCP connection and how it is handled by a programmer is indicated below:

1. A new socket is created
2. The new socket tries to connect to a remote machine
3. The connection is established and its details are agreed upon
4. Now, data is transferred
5. The connection is closed by any (or both) of the computers.

Each of these steps (except number 4) normally corresponds to one or a few lines of code.

The connection offered is full duplex; that is, both computers can send and receive data simultaneously. Data does not have any predefined meaning. What data means depends on decisions taken by the programmer. He or she can, for example, decide to communicate through commands with an HTTP server (a Web server), transfer a file where the data does not have any meaning for the transfer itself (but that most probably has an interpretation for the users), or send data according to a custom designed protocol. This means that the code handling data transfer can range in size and complexity from a few lines to long complex code units.

Chapter 4. Computer science education research

This chapter aims at sketching the identity of the emerging field of computer science education research, and thereby depict the landscape of which the current work is a part.

The field of computer science education research has a cross-disciplinary structure, and encompasses computer science – of course – but also a wide range of other disciplines: pedagogy, psychology, cognitive science, learning technology, sociology to mention a few. What unifies this diversified field is the *aim to improve learning and teaching within computer science*, and thereby to contribute to computer science.

The field of computer science education research can be characterized in that it encompasses both that which is learned about and the ways in which the learning is approached. A research project within computer science education should thus include aspects of computer science, as well as of education (to analyse how the learning takes place), with education here being interpreted in a broad sense. Research in education, not taking the subject area into account, or not aiming to enhance teaching and learning within computer science, falls outside this delimitation of the field.

1. The community of computer science education researchers

In an article by Clancy, Stasko, Guzdial, Fincher & Dale (2001)[17], Fincher discusses the need to identify a community of practice of researchers[18] within computer science education. In emerging fields, she identifies particular difficulties, such as knowing who is "in" the community. As many of the leading researchers within the field are better known for their contribution to other sub-areas of computer science, it is also hard to determine where the edges of the community are. There are, however, as

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

18. Fincher refers to Becher (1989) and Crane (1972) for research communities.

she states, a set of journals (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, Workshop of the Psychology on Programming Interest Group, Frontiers in Education) where computer science education research plays the leading part, or is one of the key topics. CSERGI – Computer Science Education Research Groups International[19] is an initiative to enhance collaboration between the different computer science education research groups a world-wide scale. A recent initiative is taken in Australia, in the formation of Australasian Computing Education Community[20].

2. Areas of research in computer science education

Having now established the identity of computer science education research in terms of its content, its community and its applications, attention can be turned to the different types of research projects that are performed.

Fincher (in Clancy et al., 2001) identifies four broad areas of computer science education research (below presented in an abbreviated and slightly edited form):

1. Small scale investigations of a single aspect of discipline and practice.
2. Investigations motivated by the use of tools in computer science teaching and learning.
3. Investigations of specific mental and conceptual skills in the psychological traditions.
4. Research anchored within the educational traditions.

I will here elaborate on these areas and their contributions, and exemplify the work performed within the first three of these. The fourth area, of which the current work is a part, is discussed more in depth in section 2.4.

2.1. Small scale investigations

A survey of the proceedings of recent years of the ACM SIGCSE (Special Interest Group on Computer Science Education) conference, the ACM ITiCSE (Innovation and Technology in Computer Science Education), as well as the journal Computer Science Education (vol. 9 – 11) indicates that

---

19. `http://www.docs.uu.se/csergi/`

20. `http://cerg.csse.monash.edu.au/acec/`

most of the projects presented are case studies on specific courses, often performed, evaluated and reported about by the teacher of the course. These studies are normally driven by the needs of the computer science educators of the universities in question and address problems that have arisen in real teaching situations. Frequently the questions addressed concern the introduction of new methods or new tools to teaching.

Although valuable as a means of sharing experiences between computer science educators, the results are, as also pointed out by Holmboe, McIver and George (2001) and 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. Investigations motivated by the use of tools

The development of new tools for teaching and learning of computer science 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[21] to learning platforms or environments, for example for Java, such as BlueJ[22]. 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 (Sutinen and Torvinen, 2003), and in more general tools finding their roots in the research in educational technology. Here collaborative learning environments of different character and tools aiming to facilitate collaboration exist, such as the Explanogram[23] (Pears and Erickson, 2003).

2.3. Investigations in the psychological tradition

Research in computer science education at a university level has for a long time been dominated by cognitive psychology approaches, with an aim 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 (Johnson-Laird, 1993). An example is the work performed by Baffes (1994), who has created a system that automatically identifies and recognizes mistakes made by programming students. Based on the information that is collected across multiple students in a database, the program models the error and offers the student relevant feedback. Holmboe (2000) argues that a teacher in

21. http://www.cs.joensuu.fi/jeliot/

22. http://www.bluej.org/

23. http://www.handwritten.net/

computer science needs the means to evaluate students' mental models as well as guidelines for designing learning environments. He describes typical aspects of students' mental representations, built on an empirical study of students taking a course in system development.

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. Wu (1993) has shown, in a controlled experiment on the teaching of recursion, a significantly better results for students that attended lectures that were based on concrete models, compared to those that were taught using abstract conceptual models. McIver (2000) has in an experiment compared error rates for students who learned Logo or Grail1 as a first programming language. She compared syntax errors as well as logical errors and concluded that the design of the programming language has a substantial impact on error rates for novice programmers. Also Robin, Rountree and Rountree (2003) have focused on the needs of novice programmers, in their mainly theoretical study of the differences between effective and ineffective novice programmers. Almstrum (1996) suggests that pre-university teaching in mathematical logic as well as the content of university level courses in discrete mathematics needs to be scrutinised. Her study, looking for the reasons for learning difficulties in the field and based on a large body of material, shows that novice computer science students experience more difficulty with concepts involving mathematical logic than with other concepts in computer science.

Research of this kind has its roots within both computer science and psychology and thus naturally falls within the delimitations of computer science education research that were stated above. The focus is on the learner, studied in a model, or in a quantitative investigation, and issues relating to the whole learning situation or the object of the students' learning takes second place.

2.4. Research in the educational tradition

For a teacher who has an interest in improving her teaching or a curriculum in computer science, an understanding for how her students learn about computer science concepts, as well as the conditions for their learning, become relevant. The type of insights that can be obtained about learning from a research project is, of course, closely related to how the research is performed. Different *research approaches*, anchored in education, here serve as lenses with different foci and help the researcher to focus on certain aspects of the students' learning of computer science.

There is a growing awareness of the need to use relevant research approaches within computer science education research. Initiatives taken to promote research include the bootstrapping and scaffolding projects (Petre, Fincher and Tenenberg, 2003), papers offering an overview of the current use of pedagogically anchored research approaches (Carbone and Kaasbøll, 1998; Holmboe et al. 2001) and attempts to verbalize models of the research process (Daniels, Petre, and Berglund, 1998; Pears, Daniels and Berglund, 2002). An upcoming conference, 1st International Computing Education Research Workshop[24], has its focus on research within computer science education.

In discussions concerning research in education, the distinction between positivistic and non-positivistic research is important. This distinction, and the resulting different selection of which methods to apply, is discussed by Cohen and Manion (1998). The two authors deploy the term "anti-positivist approach" for what I denote "non-positivistic approach" in this thesis.

> Investigators adopting an objectivist (or positivist) approach to the social world and who treat it like the world of natural phenomena as being hard, real and external to the individual will choose from a range of traditional options - surveys, experiments, and the like. Others favouring the more subjectivist (or anti-positivist) approach and who view the social world as being much softer, personal and humanly created kind will select from a comparable range of recent and emerging techniques - accounts, participant observation and personal constructs, for example. (Cohen and Manion, 1998, p. 7)

Positivistic research is thus based on the assumption that the truth is objective, neutral, and thus independent of the researcher. Non-positivistic research challenges this assumption in varying degrees, for example by arguing that the questions of objectivity is uninteresting, that it is impossible to determine if objective truth exists, or, simply by arguing that there is no objective truth.

With a non-positivistic research approach, the researcher aims to "reveal how the individual creates, modifies and interprets the world in which he or she finds himself" (ibid., 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[25] obtains somewhat different insights than another researcher would. On these grounds, Fincher's (in Clancy et al., 2001) efforts to find an identity for the community is important for the argument presented here.

Non-positivistic research within computer science education

The common identity of research projects in a non-positivistic tradition lies in that they discuss computer science concepts, students and their

---

24. `http://icer2005.cs.washington.edu/`

25. A researcher is in this thesis referred to as "he", even when used in a general sense, since I - the author of this thesis - am a man. Certainly, the arguments would not change, if "he" was replaced by "she" For a further discussion of gender issues, see chapter 9.

relationships, with the aim of gaining insights in how to teach and learn computer science. Still the projects differ in many ways. Most of them are based on consciously selected and deployed theories of learning. The different approaches and the different deployment of them permit the researcher to explore and highlight different aspects of computer science education.

Phenomenography

The educational roots of the current project are found in phenomenographic, non-positivistic, empirically based research tradition. Phenomenography here serves to reveal how students understand, or experience, some computer science concepts. Since phenomenography plays such an important role for how the research in this project is carried out, the next chapter is devoted to this approach and its applications in Computer Science Education Research.

Work in the Vygotskian tradition

Learning is seen as related to, dependent on and a being a part of the students' environments in the multi-faceted research tradition that has its roots in the work of Vygotsky (1986) and his colleagues[26]. Focus is on the collective, of which the individual is a part, and the use of tools, intellectual as well as practical, 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 their situation in general become less significant.

Holland and Reeves (1996) have studied group work for student teams working with software development. They use activity systems[27] 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 (2004) also deploy activity theory for the study of students' actions in distributed teams in a computer science 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 (2004) argues, based on the theories of situated learning (Lave & Wenger, 1991) that two cultures, namely that of academics

26. An introduction to this approach is given by Säljö (2000)

27. Activity theory is further discussed in chapter 6.

and that of computer users, meet - and clash - when students learn computer science. She illustrates her reasoning by a study of high-school students learning concurrent and distributed computing. The study environment should create the possibility for the students and motivate them to cross this cultural boundary. The implications of situated learning and other socio-cultural approaches for teaching of computer science are analysed in a literature review by Ben-Ari (2004). Based on his study he argues that teachers of computer science should study the communities of practice of professional computer scientists and design educational activities to model the activities of those communities.

The projects presented here all discuss how collectives, or teams, develop and how tools are deployed in specific situations. The significance of the results lies in the possibility to "go outside" the question of how a particular student learns about a specific concept. Instead the approach invites the researcher to study the complex picture that arises when students interact in a certain environment.

Work based on constructivist theories of learning

While phenomenography and socio-cultural studies are relevant for the current work in that they focus on the students' relationship to the object of their learning, and learning as integrated in the environment, respectively, constructivism instead describes *how* a learner comes to learn something. The term constructivism serves as a label for a diverse family of perspectives, varying in factors such as the role of other learners in the construction of knowledge and how the learning environment influences or interacts with the learner in the learning process. They all share the claim that knowledge is actively constructed by a learner (see for example von Glasersfeld, 1995). The influence of constructivism in computer science education (and in education in general) can be traced through the frequent use of projects and open labs used for teaching computer science. During these learning events the students are expected to construct their own knowledge and gain new insights in the issues under scrutiny.

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. Hajderrouit (1998) discusses on theoretical grounds how constructivist theory can be used to enhance students' learning of Java. Aharoni (2000) studies the cognitive process of students' construction of mental models of data structures. His empirical 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 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 (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 (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 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 organised in a way that the students meet different aspects of the phenomena taught. Greening's (2000) paper "addresses constructivist ideals at the level of implementation" (p.96), in that it proposes a programming assignment, that is consciously based on the constructivist theory of learning. 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, generalises such aspects and presents general guidelines for teaching computer science. These guidelines will be further discussed in chapter 10 in relation to the results presented in this thesis.

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 tackled from this perspective, when the papers argue that the teaching and the teaching environment should be designed to encourage good learning and to minimize the risk of that the students misunderstands important computer science concepts.

The critical tradition in computer science 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 imbalances, concerning for example gender, cultural, environmental or political issues. Such research can not be defined in terms of its ontological[28] standpoints or in terms of how the research is performed. An overview of the few research project performed within this tradition, together with a discussion of their relevance for computer science education

28. An *ontology* is, in short, a theory about how reality is constituted.

research is offered by Clear (2004). The work of Björkman and Trojer (2002) can illustrate feminist research[29] within computer science. They investigate, building their argument on some of the core literature within computer science, whether there are factors *within computer science 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 computer science and its knowledge production would serve to enrich computer science as well as education within computer science[30].

Also Stein (1999) argues, but on different grounds than Björkman and Trojer, that the computational metaphor[31] 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 computer science is taught. The need to view computer science from a non-positivistic perspective is also advocated by Nwana (1997). But in contrast to Stein, who bases her argument on that computer science is in a crisis, Nwana states that this crisis is a chimera. By viewing computer science and computer science education from a constructivist[32], non-positivistic perspective, he finds computer science to be successful. The question is important, he argues, since the value that is laid on computer science has a strong influence on the subject area and its teaching.

---

29. The term *feminist research* can serve as a synonym for *gender research*, in the way the terms are used in this thesis.

30. Some instances of critical research can, as has been demonstrated, be regarded as computer science education research, as the field is described here. The general aim of computer science education research, to improve computer science education through research, is not value-laden in the sense as the aim of changing that is a key characteristic of critical research. While computer science education research builds on traditional research approaches and aims to apply its results to improve computer science education, the critical research questions values and the foundations of the research as well as the results.

31. 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 my interpretation, the term is based on a step-by-step image of computing.

32. In the interpretation of constructivism presented by Nwana, computer science 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.

The emerging research within the critical tradition in computer science education contributes to the whole by addressing new questions, related to power (im-)balances. In fields as computer science which is male-dominated and founded in Western values, these questions are highly relevant for future research.

Multi-faceted approaches

The value of deploying several approaches in the same project within computer science education research has been pointed out by Greening (1996). He takes a stand on theoretical grounds, arguing that "the output from a multiparadigm[33] [research] approach to computer science education can be expected to provide a more encompassing picture of teaching and learning within the discipline" (p. 51). A similar approach has been advocated by Meisalo, Sutinen and Torvinen (2003), who have formatively evaluated a distance course in programming in computer science 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[34]-based course, with the aim of distinguishing between efficient and inefficient working groups, is presented by Kinnunen and Malmi (2004). They focus on the interaction in the groups by coding different functions of their conversations, deploying a coding scheme based on Bales (1951) and Flanders (1965). The results of this analyses is then related to findings from interviews, two surveys and the grading of the course. The Runestone initiative, where the different sub-projects adopt different approaches, as well as this thesis, where phenomenography is enhanced by elements of activity theory can be seen as two further example of combined approaches.

3. Researching students' learning of computer networking

Few of the studies performed in computer science education focus on network protocols. However, the journal Computer Science Education, Vol. 10, number 3, is devoted to network protocols. While most articles focus on methods of teaching or tools for teaching or for practical exercises for the students, some also analyse students' learning. Jard and Jéron (2000) present a case study on students' learning about validation of the alternating

---

33. Greening's use of the term *paradigm* broadly corresponds to the word *approach* as it is used in this thesis.

34. The abbreviation PBL stands for *Problem-based learning*,

bit protocol. They argue, based on their own experiences of teaching the course, that students understand the need for automated tools for protocol verification by trying to use such tools. Mester and Kruhm (2000) argue, based on their own experiences as lecturers, that animations of formal methods to a certain degree can improve students' results on exams, and that the students' ability to find imaginative solutions to particular kinds of problems increased. In a recent study Chang (2004) has developed and evaluated a course in computer networking, in which the students, in project teams set up and analyse networks. The lack of research in teaching and learning of computer networking has been identified, and projects have been initiated to develop tools for network learning, and to disseminate the results from on-going projects. A note-worthy initiative in this context, is a book concerning teaching of computer networking (Sarkar, ed., in preparation).

Chapter 5. The phenomenographic background

This chapter explores phenomenography, which serves as the core educational research basis for the work presented in this thesis.

1. Researching students' learning of computer science

A research approach[35], such as phenomenography and activity theory, offers guidelines for the empirical work as well as a theoretically anchored perspective on learning. 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 on different roads. Before further examining phenomenography and its contribution to this project, a few words about research approaches in general might be useful.

The relationship between an approach and the outcome is complex: Similar results, or at least results illuminating the same 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. The researcher himself – his beliefs, interests, previous experience, network of discussions, even his personality – is essential for the outcome of a project and become, to a certain degree, a part of the results. Other factors that influence the outcome are the setting in which the study is performed, as well as previous work within the domain performed in a similar way.

An approach also serves to facilitate 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 and to other situations.

The deployment of a particular approach in a project does not normally define which methods to use, for example how to collect data. These are

35. I use the term *research approach*, or just *approach*, instead of the frequently deployed terms *method*, *methodology* or *framework*. The term approach stresses the non-algorithmic, non-prescriptive character of my research. The frequently used term *paradigm* is not suitable in this context, since paradigms, as defined by Kuhn (1962), are incompatible.

methodological decisions that have to be taken based on the current situation in which the approach is a core elements. Since a research approach has its history, there is a tradition, or a network of competence, that the researcher partly can lean on in his selection to use one method (for example interviews for data collection) over another (for example observation).

2. The background of phenomenography

Phenomenography originated in a set of empirical studies of learning among university students carried out at Göteborg University, Sweden, in the early 1970s. The questions in focus then were, what it means that some people are better learners than others and why this is the case. Since then phenomenography has spread to different corners of the world and has developed in different directions. The approach has also evolved so that it can be used to tackle a large set of research questions (for a rich set of examples, see Marton & Booth, 1997).

Phenomenography has proven helpful in computer science education since it allows a researcher to retain focus on computer science concepts and principles, while studying the learning of these concepts from the students' perspective. The outcome of a phenomenographic analysis, in terms of categories that summarise how different key concepts are understood by the learners, are normally further discussed with the aim of offering recommendations for teaching.

The first example of this kind is Booth's thesis (1992). She studied what it means and what it takes to learn to program. Learning to program is characterised as a process of a growing awareness of what it means to program in terms of three elements: technical constituents, framework constituents and writing programs to solve problems. Booth's conclusion is that teaching should offer a rich variation in different ways of coming to an understanding of what it means to program. Bruce et al. (2004) have reached similar implications for teaching in their study of how novice programmers learn to program. It is crucial, they argue, that different ways of seeing the object of learning are exposed to the students. Eckerdal (in preparation) has, through her study of how novice students understand the computer science concepts of object and class, found that students experience difficulties in learning the culture of computer science and its implicit rules and norms. In a phenomenographic study, Cope (2000) argues that the experience of learning information systems shows a multitude of educational aspects, as a reflection of the complexity of the desired way of understanding an information system. The students' learning experience becomes more effective if active collaborative learning tasks that focus on important aspects are designed.

The focus on how students understand computer science concepts unifies these phenomenographic projects. The aims of the projects extend beyond the outcome space. In addition the implications for teaching are drawn out by relating the outcome back to the environment from which it originally stems.

3. Phenomenography as a way to explore students' learning of the subject area

A phenomenographic research project aims to reveal the qualitatively different ways in which a phenomenon, such as the computer network protocol TCP, can be experienced, understood or perceived within a student cohort. An experience, or understanding, of something is, according to the phenomenographic theory relational, in the sense that it is shaped both by the learner and that which he learns about. For example, the two network protocols TCP and UDP are experienced in different ways (at least a computer scientist would hope so). Similarly, the learner changes when he learns something, and comes to experience something in a new way. A student is thus "not the same" after understanding TCP in a new or different way. This relationship is illustrated by arrow ① in figure 5 below.

The researcher stands in a similar relationship to his study object as does the learner to that which is learnt about, as indicated by arrow ② in figure 5[36]. Denzin and Lincoln (1994) argue that a qualitative research approach, such as phenomenography, is "multimethod in focus, involving an interpretive approach to its subject matter" (p. 2). This means that a qualitative researcher "studies things [...] attempting to make sense of, or interpret, phenomena in terms of the meanings people bring to them." (ibid., p. 2). The outcome of a phenomenographic research project is thus the researcher's interpretation of the students' understanding of what they learn about. The outcome is in this way shaped both by the researcher and the object of his research, as the researcher learns about his object of research.

Variation comes to the fore in phenomenographic research in both these relationships: Firstly, the students experience that which varies, for example the speed of a connection, since variation is a requirement for something to be experienced. No one would, for example discern a communication link as slow, if all links had the same speed. Secondly, the researcher discerns the qualitatively different ways in which the phenomenon under investigation is

---

36. This is frequently referred to as a *second-order perspective* in phenomenographic literature.

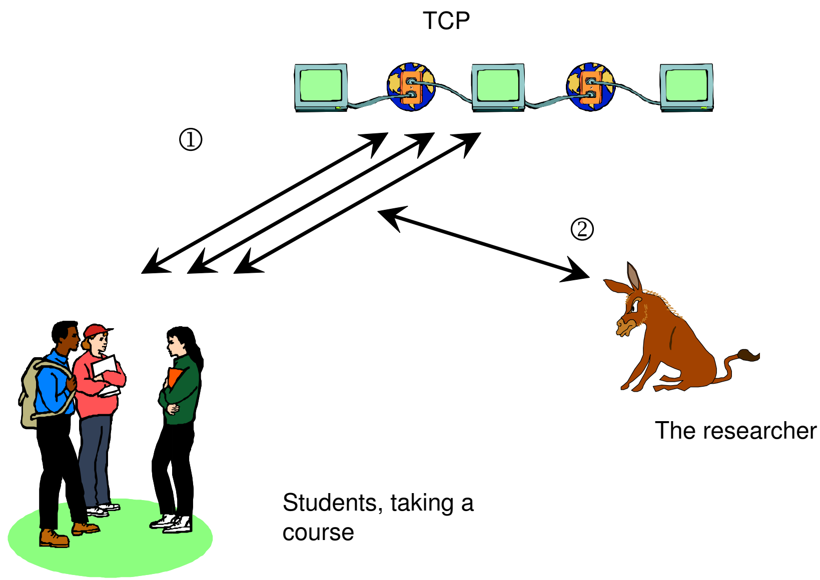

Figure 5. The perspective taken in phenomenographic research, adapted from Berglund (2002).

understood by the students. Here, the variation lies in the interpretations made by the researcher; his descriptions of the students' different perceptions vary.

Although a phenomenon can be experienced in countless ways, phenomenographic research on learning claims that a researcher can organize these different perceptions into a limited set of qualitatively different categories. Each category[37] then serves to summarize and describe a particular way in which a phenomenon is understood within a student cohort.

In this way the individual student comes to serve as a "carrier" of (fractions of) one or many ways of understanding something. The outcome of the project then describes the ways in which something is understood within a cohort. The individual voices have "disappeared" from the final result. What remains is the researcher's interpretation of what they have said. Thus, the result can only be interpreted for a collective.

To summarise, the results are descriptions of a set of phenomena from a subject area (for example computer science) as these are understood by students. A phenomenographic research approach makes the students'

37. The terms *category of description* (or *category*), *way of experiencing* and *conception* are used in phenomenographic literature to denote different facets of (the researcher's interpretation of) a relationship between an experiencer and that which is experienced.

perspective visible for others, and gives them the possibility to "talk" about their learning of the subject area and their experience of the learning environment.

3.1. What does learning mean?

Learning is always the issue of acquiring knowledge of something or capacities to do something. In phenomenographic research, focus is on how students come to experience something in a new or different way (see for example Marton & Booth, 1997; Bowden & Marton, 1998; Marton & Tsui, 2004). The learning outcome that is sought is that which is actually learnt from the point of view of what is meant to be learnt.

Learning something means to open an aspect which previously has been invisible or taken for granted. As an example, let us consider a student who understands the network protocol TCP as a way of communicating between computers. When he also comes to see the protocol as a standard, defined by human decisions, he has learned something about TCP.

To make learning possible, the learning environment – it could be books, lectures, web-pages, projects or combinations of resources – must give the student the opportunity to experience something in different ways. If the learning environment only emphasises one single aspect of TCP, it is not likely that a student could experience any other aspect of TCP than the one which is offered. Thus, learning requires that the learning environment offers a space of variation that makes learning of the desired aspects possible.

Still, there is no guarantee that learning takes place, even if a space of variation is present. Also the question about how a student goes about learning is a key issue. To continue the example from the previous paragraph, we can contrast a student who (at a certain moment) searches to learn about isolated concepts, with a student who searches to learn to make or transform something. The latter approach is certainly more powerful, in the search for a full and varied picture of what TCP means.

3.2. The object of learning

The term *object* is deployed within computer science as well as in education and has different meanings in the different contexts. In the programming language Java an object is an entity within a program. Activity theory offers still a different meaning, in that it proposes an object as the reason for existence of an activity (see chapter 6).

Two of the meanings for the word "object" presented in Webster (1979) serve as a sounding board for a discussion concerning the phenomenographic distinction between the *what*- and the *how*- aspects of the experience of learning.

These are presented in an abbreviated form[38]:

> 1. what is aimed at; that towards which the mind is directed in any of its states and or activities; goal; aim; ultimate purpose; aim.
> 2. a person or a thing to which action, thought or feeling is directed.

The *what*-aspect relates to the content that is learnt about, for example a network protocol. In other words, answers to the question "What did you learn?" are, at least from a teacher's point of view, supposed to discuss or describe something from within the course content or subject area. The *what*-aspect is normally referred to as the *direct object*, and shows many similarities with the grammatical concept of "accusative". This corresponds well with the second meaning from Webster.

In a situation like the Runestone course, much concern is related to *how* the students approach their task or *how* the learning takes place. This aspect, the *how*-aspect, can further be analysed by introducing a distinction between *the act of learning* and the *motive.* The *motive* is frequently referred to as the *indirect object of learning* in phenomenographic research. The act of learning refers to "the experience of the way in which the act of learning is carried out [...], the [motive] referring to the type of capabilities the learner is trying to master" (Marton & Booth, 1997, p. 84). The two are closely interrelated in that the motive towards which a student strives, influences, and is influenced by, how the learning takes place (the act of learning). The indirect object, or motive, in phenomenography is thus well described in the first definition in the quote from Webster above. The relationships between these entities are illustrated in figure 6 below adapted from Marton and Booth (1997). Here, it must be remembered that the students experience the learning as a whole. The distinction is entirely analytical – the two aspects can only be thought apart – and aims to be a tool for the researcher in his efforts to understand, analyse and describe the students' learning.

The "what"

Different ways of understanding something, for example the network protocol TCP, do not only differ in what TCP is understood to be (its meaning), but also in its structure (its parts and their relationships). The former aspect is normally referred to as the referential aspect, while the latter is called a structural aspect (Marton & Booth, 1997). The two aspects create a whole, where structure presupposes meaning and vice versa. A coin of one euro can serve as an illustration: To get a full picture of such a coin, both the meaning (a currency in many European countries; that is, a legal tender) and its shape (round, consisting of two different metals) must be

38. The remaining five meanings are not relevant in this discussion and refer to something tangible; appearance; a person that excites pity; grammatical entities; and in a philosophical sense, anything that can be known.

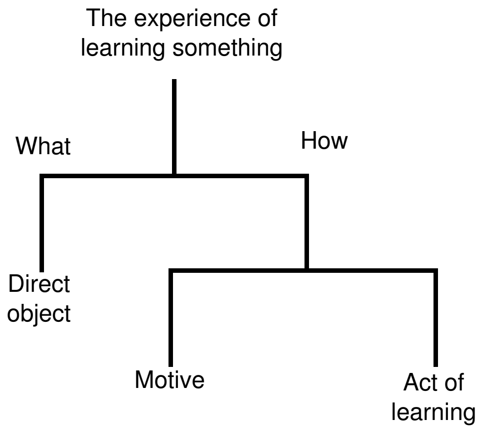

Figure 6. The experience of learning something can be analysed into a *what* aspect and a *how* aspect, where the latter is further described into the act of learning and the motive (adapted from Marton & Booth, 1997, p.85)

known. For someone who only knew its meaning or value without knowing its structure, it would be impossible to recognize it in a pile of round metal pieces. The situation is similar for someone who only knew its shape: He could find it, but would not realize that it was possible to buy something for it.

In the structural aspect of an experienced phenomenon a further distinction can be made: between the aspects that are in focus, often referred to as the *internal horizon*[39]; and that which surrounds the phenomenon and to which it is related and of which it is a part, often denoted the *external horizon.* For example, the euro coin would not be understood as a euro coin if not a specific structure (the bi-metal structure with its specific picture) stood out as an internal structure to be discerned against a background of other coins (external horizon). Figure 7 below summarises this example, with a picture adapted from Marton and Booth (1997). In this thesis, I normally prefer discussing a foreground-background relationship, since I find that the word *horizon* puts too much emphasis on the “border” or “contour” to be intuitively appealing for some of the phenomena studied.

39. To be more precise, the *internal horizon* denotes the phenomenon in focus, relationships between its parts, and its contours

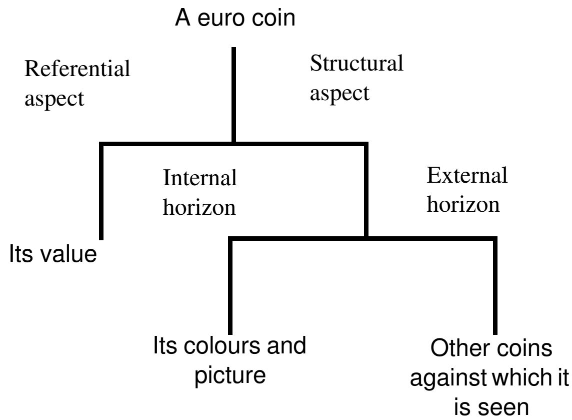

Figure 7. The different aspects of the experience of a euro coin.

The "how"

The quote from Marton and Booth (1997) presented earlier on page 40 indicates that it is the *students' experience* of their learning that forms the core of the act of learning. In other words, the researcher strives to describe the act of learning from the students perspective. The term "act" should here be interpreted in a broad sense, beyond the physical acts that a student performs in order to learn, such as reading a book, solving a problem and asking a friend. The term "act of learning" also includes abstract aspects, such as how students go about achieving their aims.

Not only how a student goes about learning, but also what he aims to learn, the motive, is crucial for the learning outcome. A student who, in the Runestone project, aims to develop technical skills is more likely to learn new programming language constructs than a student who aims to develop his social skills.

Many of the phenomenographic research projects which focus on the act of learning have investigated different aspects of the dichotomy of surface and deep approaches to learning. The two approaches of learning differ in whether focus is on the sign (for example a text to read), or the signed (the meaning of the text). The term approach, as it is used here, contain components of both the act of learning and the motive, as it describes both what the student aims to learn and how he goes about reaching this aim.

3.3. A way of experiencing something

Marton and Booth (1997) define *a way of experiencing something* in the following way:

> “A way of experiencing something” is experiencing something as something, experiencing a meaning that is dialectically intertwined with a structure. “A way of experiencing something” is a way of discerning something from, and relating it to, a context. (p. 112)

The different ways, in which something can be experienced, can be logically related to each other, since they represent different perspectives *of the same thing*. Marton and Booth (1997) further argue that each way of experiencing something, should represent a meaning, intertwined with a structure, *of the whole phenomenon*. Even if the experience of something is only partial, it is still experienced as a whole by the experiencer, and does not, from this perspective constitute a part. An example from chapter 8 can illustrate this distinction. An understanding of TCP as communication between two computers (category 1) must be seen as a partial understanding of TCP, since there are other, more advanced ways in which the protocol can be experienced. Still, it is a way to experience the whole protocol, since all parts of the protocol, which are available for the experiencer, are taken into account. It does not solely focus on a single part, such as for example only on the packets, while neglecting the whole.

4. A framework based on phenomenography

This chapter has described the foundations of phenomenography in general terms, without emphasizing the particular requirements that stem from the current research project. The issues related to the current project, as the methodological aspects of phenomenography and what the researcher’s role is are still to be discussed, and will be a theme of chapter 7.

To extend the use of phenomenography to also encompass the students’ experience of their learning environment is not uncomplicated. To tackle this issue, it is rewarding to investigate how the concept is treated in another approach. Some key elements of activity theory, an approach in the Vygotskian tradition, will for this reason be examined in chapter 6. The aim is to find tools that can be used to synthesise the results of phenomenographic analyses into a whole.

How the concept of context appears in phenomenographic research, how it can be handled, and how it can enhance the quality of the outcome rest to be settled. This certainly have methodological implications and is therefor a theme in chapter 7.

These discussions of phenomenography, activity theory and context in phenomenographic research lays the ground for the development of a methodological framework (in chapter 8), aimed at analysing and synthesising the students’ learning of computer science in a complex learning environment.

Chapter 6. The activity theory background

Elements of activity theory are used in my work, as a complement to the dominant research approach, phenomenography. My purpose of drawing on activity theory is to gain access to the intellectual tools aimed at synthesising the phenomenographic results and analysing context.

The theoretical aspects are based upon the interpretations of activity theory by Engeström, 1987, 1993; Kuutti, 1996; Bannon, 1997; Kaptelinin, Kuutti & Bannon, 1995). The interpretations of activity theory, as they are presented here, are grounded in the needs of my research and do not in themselves do justice to the richness and the possible deployment of activity theory.

1. The activity theory tradition

Activity theory is a research tradition, or research approach, that is developed from the work performed in the Soviet Union by Vygotsky (1978) and Leontev (1981). Kaptelinin et al. (1995) point out that activity theory, in spite of its "name", is not a theory in the strict interpretation of the term:

> First, while it is usually associated with Leontev's approach, there are also several other interpretations of Activity Theory. Second, and more important, it is not what people usually mean by a theory, but rather consists of a set of basic principles which constitute a general conceptual system which can be used as a foundation for more specific theories. (p.189).

Activity theory is not in itself a fixed and stable research tradition, and does not have a given interpretation. Instead, activity theory and its community can, as the phenomenographic community, be modelled as an activity system, that develops over time in the multi-voiced international network of researchers. Human-computer interaction (Nardi, 1996a, Kaptelinin, 1996, Kuutti, 1996), educational technology (Bellamy, 1996), health care (Engeström, 1993), projects in programming (Holland & Reeves, 1996), children learning graphs (Åberg-Bengtsson, 1998), learning astronomy in a technology rich setting (Barab, Barnett, Yamagata-Lynch, Squire & Keating, 2002), and distributed collaborative learning (Fjuk, 1998) can be mentioned as examples of areas to which activity theory has been applied during recent years. These diverse research projects all make use of activity theory as a tool to analyse complex systems where several individuals interact in a system rich in technology, rules of conduct etc.

Before going into a deeper discussion on activity theory, a short clarification on the terminology I have chosen might be useful: I let the term *activity* denote an “actual”, “concrete” activity in the “real” world, built up by its *constituents*. The model or description of an activity is an *activity system*, consisting of different *components*. A common graphical representation of an activity system is a *triangle*, showing *nodes* and the relation between nodes (see figure 8). These terms all reflect the same activity, but emphasise or express different aspects of it, as different sides of the same coin[40]. In most situations in the text, the terms could be interchangeable for each other with only slight change in the interpretations.

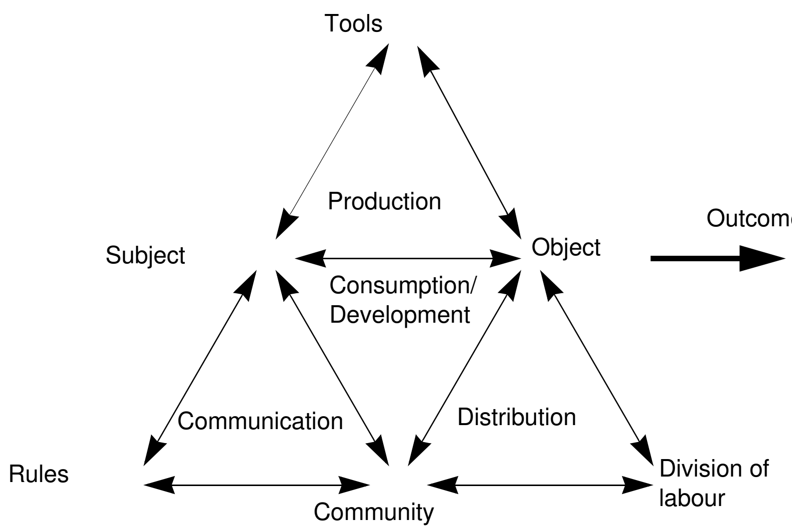

Figure 8. The components in an activity system and their relationships (developed from Engeström, 1987, p. 78)

1. Understanding an activity as a system

An activity is a theoretical entity that is discerned, described and used by the researcher. The activity theory model aims to capture the socially based nature of human activity. The constituents of an activity continuously interact and thereby develop into new forms that together shape an activity.

40. Although these are three aspects and a coin has two sides, the analogy is appropriate

In this way, activity theory serves as a “clarifying and descriptive tool rather than a strongly predictive theory” (Nardi, 1996a), and does not prescribe any particular way of understanding learning.

An activity system basically describes the interaction between a *subject*, an individual, and an *object*. The activity is directed towards the *object.* Through this interaction the object is transformed into an *outcome*. The relationship is not direct, in the sense that the subject interacts directly with the object, but is indirect through the use of *tools*[41], both physical and intellectual. The interaction is *mediated* by the tools. An activity is, however, larger than the tool-mediated relationship between a subject and an object. It is integrated in a collective activity. The *community* that shares the same object, for example of learners or practitioners, is a fundamental constituent. The relationship between the subject and the community is mediated by *rules*. The term *rules* must be understood in a broad sense, including aspects as diverse as the legal systems, cultural or social rules for people’s interaction as well as the habits that have developed within the current activity. The relationship between the community that together share an object and the object itself is mediated through a *division of labour* (Nardi, 1996b; Bellamy, 1996), that also is developing and changing over time. An activity can be modelled as an activity system, consisting of these components, as illustrated in figure 8, building on Engeström (1987).

An activity is constructed and reconstructed by the individuals within the system. The activity system as such is more than the sum of individual discrete actions. Their behaviour, or actions, are constantly influencing the activity and are parts of the activity itself. As the activity is evolving it thereby affords new actions by individuals, actions that in their turn become parts of the activity. Hence the individual and the activity are inseparably intertwined.

The sub-triangle in the centre in figure 8 is denominated *consumption* in Engeström (1987, p. 78), on the grounds that it has “subordinated the three dominant aspects of human activity - production, distribution, exchange (or communication)” (ibid., p. 78). Engeström argues that “consumption is also a production of human beings themselves” (p. 79). Here openings can be found for an experiential interpretation of the middle triangle, where humans (or subjects) mature, or develop, in their interaction with the other constituents of the activity of which they are parts. On these grounds, the middle triangle will be labelled *development* in this thesis. Such a

41. Some authors, notably Vygotsky (according to Engeström, 1987) distinguish between *signs* and *tools*. The former, that are described as a psychological generalization of the concept of physical tools, differ in that they imply and require reflection and consciousness. Since such distinctions are without interest for the way activity theory is deployed in my thesis, I will use the term *tools* in a generalized meaning, that also encompasses *signs.*

denomination is consistent with the two later quotes, and captures, in my meaning, the experiential nature of the current research.

2. Basic principles of an activity

As Kaptelinin et al. (1995) point out, activity theory can be described through a set of principles, or key concepts, that together describe what constitutes an activity. The principles below are based on the work of Engeström (1987), Kuutti (1996), and Bannon (1997), and reflects my use of activity theory as a tool for analysing learning in a complex situation:

Principle a. The object as a reason for existence of an activity

Principle b. The dynamic structure and historicity of an activity

Principle c. An activity as a context

Principle d. The role of mediation

Principle e. The role of inner contradictions

Principle f. Individual actions as parts of an activity

a. The object as a reason for existence of an activity

Engeström argues that an activity is defined by its object, and that the object is then the motive for the existence of the activity. What an individual does, might seem meaningless without considering the object of the activity. The role of a goalkeeper in a football team, cannot be understood without relating it to the object of a football game, to win by making more goals than the opposite team. Engeström (1987) makes a still stronger claim:

> Furthermore, what distinguishes one activity from another is its object. According to Leontev, the object of an activity is its true motive. Thus, the concept of activity is necessarily connected with the concept of motive. Under the conditions of division of labour, the individual participates in activities mostly without being fully conscious of their objects and motives. The total activity seems to control the individual, instead of the individual controlling the activity. (p.66)

According to Engeström the *object-focused*[42] character is thus a key principle of an activity. As an example of an object in an activity a medical institution (Engeström, 1993) can be mentioned. Everything that takes place in the medical institution aims at curing patients.

---

42. Engeström uses the term *object-oriented.* However, *object-oriented* has a different meaning within the field of computer science (see for example Budd, 1999). To avoid confusion, I will write *object-focused* as a synonym to *object-oriented.*

Problems arise, however, when the discussion turns into what an object "really" is, and how it is identified in a real setting. Engeström and Escalante (1996) discuss this issue, and why the object has such a "slippery and transitional nature":

> The object should not be confused with a conscious goal or aim. In activity theory, conscious goals are related to discrete, finite, and individual actions; objects are related to continuous, collective activity systems and their motives. [...] [T]his animate and transitional nature of objects is their necessary characteristic as objects of activity, that is, integral components of a system of human practice. [...] But objects do not exist for themselves, directly and without mediation. We relate to objects *by means* of other objects. [...] This means that objects have two fundamentally different roles: as objects [...] and as mediating artifacts or tools. (p. 360 – 361, italics in original)

The multiple roles of a compiler in a programming development project can serve as an illustration. Normally it functions as a tool, that transforms code from one form to another. Occasionally, however, when problems with the compiler are encountered, possibly because a bug within the compiler is suspected, the compiler becomes the object of the program developing activity. Then, when the problems are solved, the compiler regains its role as a tool. When working practically, it is clear that there are no sharp limits between these two understandings of what the compiler "is". Before starting to debug the compiler, in order to identify possible bugs, there has been a period – long or short – with evolving contradictions between the components in the system when problems have been suspected, and when the role of the compiler has been unclear.

b. The dynamic structure and historicity of an activity

An activity is a dynamic entity that is always under development. Its current state is thus related to its history, and thus a historical perspective is needed. Kaptelinin et al. (1995) argue that also tools must be seen in this way, since they are carriers of other people's experience:

> [...] tools usually reflect the experiences of other people who have tried to solve similar problems at an earlier time and invented/ modified the tool to make it more efficient. This experience is accumulated in the structural properties of tools (shape, material, etc.) as well as in the knowledge of how the tool should be used. Tools are created and transformed during the development of the activity itself and carry with them a particular culture — the historical remnants from that development. So, the use of tools is a means for the accumulation and transmission of social knowledge. (p.192)

The same reasoning holds, according to Kuutti (1996) for the activity as a whole. Since it is under continuous change, it has a history of its own. Parts of that history, he argues, stay embedded within the system as it develops.

Thus, the activity contains its own history and a historical analysis is often needed to understand its current state and development.

c. An activity as a context

The activity is what gives "meaning to seemingly random individual events" (Engeström, 1993, p.65), that is, an activity is a context, in which the participating individuals and the different events are integrated. With this approach, it is impossible to study a part of the activity as a separate unit without seeing it as belonging to and interacting with the whole system. The activity becomes a unit of analysis. This does not imply that the activity is a homogenous and harmonious entity. On the contrary, it has a multi-voiced character, a history embedded and contains different, often contradictory, expressions and events.

This argument has been put forward on many different occasions. As an example relevant for the current study, Fjuk and Ludvigsen (2001) can be mentioned. They argue, based on a number of empirically based studies, that collaborative ICT-based learning situations can only be understood when the whole system is considered, that is, when the "*unit of analysis* [is extended] from technology and pedagogy themselves to real-life experiences".

d. The role of mediation

The tool-mediation of the relationship between the subject and the object is fundamental in activity theory, as well as in other theories about learning that are based on Vygotsky's work. The tools, or signs, have mediating functions between the subject and the object and are "at the same time both enabling and limiting" to use the words of Kuutti (1996, p 27). A program, that can not be edited, does not exist for the programmer, as he cannot "see" or "touch" the program without such a tool. The editor and the compiler are the results of historical development, that contains much of what is known about a program, and comes to define a program for the programmer. A successful artefact, as here the editor, tends to become invisible, in that it allows the programmer not to focus on the editor, but to focus on, and interact with, the program itself (Säljö, 2000). Certainly, physical tools are of ultimate importance in computer science: Without the computer the program would not exist.

The key notion is that a human never interacts directly with an object. The "simple stimuli-response process is replaced by a complex, mediated act" (Vygotsky, 1978, from Engeström, 1987, p. 58). In this way, "the use of signs leads humans to a specific structure of behaviour that breaks away from biological development and creates new forms of a culturally-based psychological process" (Vygotsky, 1978, from Engeström, 1987, p 59).

e. The role of inner contradictions

The multi-voiced, continuously changing character of an activity makes inner tensions or contradictions between and within its constituents inevitable. Furthermore, an activity is not an isolated system, but rather a part of a network of interacting activities. Tensions between different activities, each with its particular object, are also sources of contradictions. The contradictions, or with the words of Engeström "double bind situations" indicate "misfits" within the system. Kuutti (1996) describes the contradictions in the following way:

> Contradictions manifest themselves as problems, ruptures, breakdowns and clashes. Activity theory sees contradictions as sources of development; activities are virtually always in the process of working though contradictions (p.34)

The role for contradictions in progressing the activity has been discussed by Engeström (1991), who writes:

> From the viewpoint of historicity, the key feature of expansive cycles is that they are definitely not predetermined courses of one-dimensional development. What is more advanced, 'which way is up', cannot be decided using externally given fixed yardsticks. [...] The internal contradictions of the given activity system in a given phase of its evolution can be more or less adequately identified, and any model for future which does not address and solve those contradictions will eventually turn out to be non-expansive. (ibid., p. 14-15)

His words do not support any simplistic interpretation of activity theory, according to which the contradictions, should make the situation impossible, and in this way provoke a revolutionary change. Instead, he argues that changes, in spite of how they are initiated, must resolve contradictions in a current situation in order to promote learning.

f. Individual actions as parts of an activity

The long and complex activity consists of smaller entities and can be described in three levels that express the "hierarchical nature of human activity" (Engeström, 1990, p. 172). Engeström (1990) articulates the three levels, distinguished by Leontev (1978, 1991), in the following words:

> *Activity* is the molar unit, collective in nature and driven by a complex motive of which the individual actors are seldom aware. Activity manifests itself in the form of goal-oriented individual *actions* in which the subject is consciously aware of what he or she is trying to accomplish. Actions, in turn rely on automatic *operations*, dependent on the conditions at hand (p. 172 - 173, italics in original)

As stressed above, an activity should be understood as a collective, changing context, that consists of, and gives meaning to, individual actions. Engeström points out that "we may very well speak of the activity *of the individual*, but never of *individual activity*; only actions are individual" (1987, p 66, italics in original).

As an activity is defined by its object, an action has its immediate, defined *goal,* of which the actors are consciously aware. The relatively short-lived actions have different reasons to be performed in different activities and can thus not be understood without its activity, or context. Someone can, for example, read a book in order to learn the programming language Java to use in a programming project. A book can also be read for pure enjoyment, as a moment of escape from a tiring daily routine.

Actions are carried out by chains of *operations*, which are routinely made and determined by their *conditions*. Normally they are performed without conscious decisions. Turning the page, when reading the book about Java, serves as an example.

3. Towards a methodological framework

The two research approaches that form the theoretical background for the empirical investigation in this thesis, phenomenography and activity theory, are now put on the map. Phenomenography offers the intellectual tools for the researcher to describe something as it is experienced from the learners' perspective, and focuses on the relationship between the learner and the object of his studies. In a phenomenographic research project, the issue of the context is normally relegated to the background. Activity theory on the other hand, takes an externalist perspective, and helps the researcher to analyse human activities in complex systems. The system as a whole is described as a context, and individual actions are described in terms of their function in the whole.

Clearly the two approaches show differences, but there are also similarities. For example: Both these approaches advocate that learning is studied in realistic, "naturalistic" settings. None of them put a dualism between matter and thought to the fore. While phenomenography is explicitly non-dualistic, activity theory, as deployed here, focuses on the external (or materialistic) aspects.

The ground is thus prepared for a further discussion of the concept of context in phenomenographic research in the next chapter. This is followed by a discussion on how phenomenographic study of the experience of learning in a learning environment can be further illuminated by deploying elements of activity theory in chapter 8.

II. METHODOLOGICAL DEVELOPMENT

Chapter 7. About context in phenomenographic research

The two key constituents in this study are the students' experiences of their learning of the subject area and the experienced context of this learning. These two entities are mutually constituted. Thus, to get a full picture of the learning in the Runestone course the two components as well as their constituents must be studied.

The focus on the learners' experience suggests the use of phenomenography, since this approach aims at revealing the learners' relationships with, or experiences of, different phenomena. The outcome of a phenomenographic study normally focuses on the variation in the relationship between the student and the object of his studies with the context being relegated to the background.

To study the context of the learning within a phenomenographically anchored project demands that I, as a researcher, go outside the "classic" use of phenomenography in my investigations. To do so, I will develop what I mean by context in phenomenographic research and illuminate the different nuances of the term from the point of view of my project.

Still, gaining an understanding of the different meanings of the term is not in itself sufficient for reaching my aims. I also need to elaborate on how to perform research in such a way that I can formulate results that are valid for the students' experience of their learning environment, and not only of the object of their learning.

1. What is context?

A corner-stone in phenomenographic research is that the researcher's attention is not directed to the ways in which the researcher experiences the phenomenon, but towards the different ways in which the research participants do so. An experience would be quite different if the experiencer was placed in some other grouping of people, or location, or epoch. By replacing "situation" with "context" in the quote below, a description of the context as something that gives meaning to a phenomenon is obtained.

> We cannot separate our understanding of the *situation* and our understanding of the *phenomena* that lend sense to the situation. Not only is the situation understood in terms of the phenomena involved, but we are aware of the phenomena from the point of view of the particular situation. And, further, not only is our experience of the situation moulded by the phenomena as we experience them, but our experience of the phenomena is modified, transformed and developed through the situations we experience them in (Marton & Booth, 1997, p. 83).

The issue of context appears in a number of guises in educational research. More general definitions of context with a scope outside that of education, are given in a dictionary. Mirriam-Webster[43] confirms the everyday understanding of *context* in its definition. It can mean either the text into which a particular passage is woven and which casts light on the passage, or the socio-spatial setting or situation in which an event occurs, and which is intimately related to the event. In either case, an *outsider,* whether a reader or an onlooker, can gain meaning of a text passage or an event, through consideration of the context, whether languaged or experienced. This can be contrasted with the phenomenographic project of describing phenomena (and by implication the context of the phenomena) as experienced by an *insider* or research participant. Such a distinction is important, since insiders and outsiders "see" or experience things in different ways. For example, the rules for grading in the Runestone course are interpreted by the insiders, based on their experienced situations of being students in the course. An outsider, as a future employer can be expected to "see" the grading in a different way, for example as a statement of some aspect(s) of the student as potential employees.

Ekeblad and Bond (1994) have made a similar distinction between the ways in which two different research forms treat the notion of context. On the one hand, in "an *experientialist* perspective […] the research question is designed to seek an understanding of what it is that is experienced" whereas, on the other hand "an explanatory or *externalist* approach to research assumes that we are looking at the impact of context *on* an individual" (p. 148; my italics). The first, experientialist, perspective is in line with the phenomenographic tradition. The externalist perspective leads researchers to observe events and to analyse them from within their theoretical and methodological frameworks, rather than seeking to see them through the experiences of the actors in the events. This is the most commonly applied approach to the issue in activity theory studies.

43. http://www.m-w.com/dictionary.htm

The work discussed in this chapter has its origin in the needs of three phenomenographic research projects, two in physics education (Adawi, 2002; Ingerman, 2002) and the one I propose in this thesis. While discussing our respective work within our[44] projects, we repeatedly returned to the meaning of the *context* for the participants and for ourselves as researchers. In this section I will present those of our insights that are relevant for the current study, drawing empirical examples from this project.

2. The roles of the actors in an interview

Since the analysis takes its starting point in the data that has been collected, the quality of the data is crucial for what might emerge from the analysis. The interviews are not an objective probe into the interviewee's experiences of the phenomenon of interest. On the contrary, interviews are a complex interaction between the interviewer and the interviewee. From this point of view, it is important to distinguish between the two actors in this situation: the *researcher*, or the interviewer, who has taken the initiative to the interview, sets the topic and has in various ways prepared the situation, and the *interviewee*, whose motives only can be seen through what he does or says. There is an imbalance of power in the nature of data collection. We have found it worthwhile to make a distinction between two different meanings of context, related to this imbalance of power:

1. The *prepared context*, as defined by the researcher. It defines what the researcher considers to be relevant for the interviewee to make sense of the situation at hand. In the current project, questions concerning a specific computer science concept were asked sometimes in the context of courses, at other occasions in relation to the project the students were working on and at still other occasions related to other computer science concepts.
2. The *experienced context,* as experienced by the participant. It is what the participant experiences as being relevant for making sense of the situation at hand. This is interwoven with the experience of the phenomenon under consideration.

We found that our understanding of the data was enhanced by switching between consideration of these two ways of seeing context – now regarding the experienced context, now the prepared context – and weighing the two against one another.

44. Since the results presented in this sub-section are a joint achievement of the four authors to Adawi, Berglund, Booth and Ingerman (2001), I will refer to the authors of that article as "we" in this section.

3. Who is experiencing the context?

By introducing the distinction between the prepared and the experienced context, we realized that the issue of who actually is experiencing the experienced context became crucial. This raised the question of how the researcher could work towards an awareness of the context during the stages of the phenomenographic study.

In the attempt to systematise the issue of context in phenomenographic research, we distinguished three distinct levels, referring to whose experience of context we are dealing with.

1. The researcher's context
2. The collective context
3. The individual's context

In the following I will further discuss these concepts, and exemplify by extracts from the current work.

The researcher's context

It has earlier been claimed, in chapter 5, that a researcher, when engaged in a phenomenographic study, stands in the same relationship to his object of research as the learner stands to the object of learning. The object of research is embedded in a context, and this context can be said to lend meaning to the object. We call this context the *experienced context of the researcher*, or simply, the *researcher's context*.

When performing the current study, I was informed by such factors as different research approaches, computer science knowledge, computer science education research, and problems associated with learning and teaching computer science, just to mention a few examples. The researcher's context is, unless explicitly stated, visible neither in the original data nor in the results, although it plays an important role for what kind of outcome a researcher can obtain.

To confuse the variation in ways of experiencing the context of a study with the variation in ways of experiencing the phenomenon of study is to risk losing fundamental insights. In my research this issue has come to the fore at some occasions. For example, when constructing the interview questions concerning the code the students had selected, I was expecting a certain type of answer, discussing features of the code, as well as the actual process of selecting the code. My assumption was that the project and the learning environment should serve as a background for the students in this situation. It turned out that my initial judgment, that the technical constituent of the code was important, was valid. But by deliberately switching between the different contexts in my analysis, and by avoiding consideration of my preunderstanding, I could discern a relationship between the students and the choice of a code. For the students, different features of the code stood out in relation to a background related to their situation, but for me as a researcher, having a different perspective, it was possible to distinguish features of this relationship seen in relation to the students' strategies in their selection of the code.

The collective context

When analysing the interviews, the researcher sometimes finds that light is shed on some utterance made by one interviewee by reading it against the background of the context deduced or assumed by the researcher from reading an interview extract by another interviewee. Switching between these two perspectives allows the researcher to let an aspect of a phenomenon as experienced by one participant interact with an expression of an experienced context that originates from another participant. This leads us to introduce the notion of *the experienced context of the collective*, or, in a shorter form, *the collective context*.

An example of how the concept of the collective context is used to further my understanding of an individual utterance is illustrated with the following extract of an interview with Axel[45]. The discussion concerns grading as something that is important for their studies:

Interviewer: Anything else you want to say about the group or the project?
Axel_1: Um, no, no aspect that I can think of.
Interviewer: If there will be anything later, you just say it.
Axel_1: Um, I guess, the one, the one.. it's not an issue really. I don't want to make it sound like an issue, the question that so many of the American students have had, is that the Swedish students get either a credit or no credit and the American students get graded on it. And it's not, I don't want anybody to misunderstand, it's not that we feel that the Swedish students are going to do less work because in our case, especially that is completely untrue. They've done as much work, or more, than us every step of the way. [...] I feel kind of handicapped on this project by the fact that we're getting a grade.. [...] And so we're spending a lot more time than is really useful getting together reports and making sure that we've got all the stuff that we need to get a good grade, instead of working to, to make sure that we do the project and get it up and running.

His initial hesitation, his concerns to be fair towards the Swedes, and his own doubt about the benefits of the fine-grained grading scale can be interpreted as ways to talk about something that is significant for him. But in what way, and why, this question is a concern cannot be read from his statements. Alec's statement can shed light on this issue:

45. The conventions for naming are discussed on page 85.

Interviewer:	Yes. You said something else, you said that, that your two classmates here in the US originally have planned just to go for passing. The Swedish students only have pass and fail. Do you see any differences between the two sides depending on this?
$Alec_1$	Um, here I know in computer science as far as graduating, um, a higher grade point is very important. Um, anything, I would say below a 3 is.. um, you're cutting out your job options.

According to Alec, a good grade is a requirement to get an interesting job, thus grades are important for the future. If the interviews had been studied one by one, this relationship would not have come to light. This is a concrete example of fragments that illustrate in what way the collective context is more than the sum of the individual experienced contexts and can serve as a tool when analysing data.

Glimpses of the cultural aspects of the context are occasionally visible to the researcher in the data because of the variation in cultural experience that the individuals express. As an example of this, an example related to grading is offered. Anthony, refers a discussion he had with Måns:

Interviewer:	Um, there are differences, um, a different grading system.
$Anthony_1$:	Yes, we've been running into that for the past week. [...] Whereas we really want that A, 'cause we know we can get that A, and we deserve that A, 'cause we've been working hard on it. And actually that just came up in our IRC meeting today, um, Måns, the Swedish group member, asked, 'are you guys really concerned about the grades'. And yes, it really, it reflects upon us, showing that we did our work, what we've accomplished.

Anthony points out that the final grade of the course has a different value in the two countries, and in this way, makes a certain cultural aspect of the context visible. For him, the importance of a grade normally is taken for granted. The explicit articulation of the cultural aspect gives a background against which all the other quotes can be seen. This insight concerning the collective context has been of great importance for the work concerning grading (Chapter 13, section 4).

The individual's context

When an individual experiences something or talks about a phenomenon, for example during a phenomenographic interview, some aspects of the phenomenon come into focus, while others remain in the background. The phenomenon is thus experienced against and interwoven with an experienced context, what we can refer to as the *experienced context of the individual*, or the *individual's context*. Examples of different individual's contexts can be seen in the following extracts concerning RMI.

Interviewer:	[...] What is RMI?
	[...]
$Sven_2$:	That is, it is a ... one moves files between, yes....for instance if I were to use RMI that was sort of ... I have the game server and a file that had marbleinfo and so the information on [...] speed so then I want to move over to mine...and then I should use RMI, hard to explain, but ...

Sven discusses how RMI is used in his project. Staffan offers a different answer to the researchers question:

Interviewer:	RMI?
$Staffan_1$:	Oh, that's Java's version of client server, it has a stub and a skeleton which one uses. You send you from your client ... you can fetch and allow to execute things from the server via. It feels as if they are local on your ... on your client, but you execute from the server actually.

Although I can only speculate when interpreting statements of an individual, informed guesses can be made. The context of Sven is based in his work with the Runestone-project, and includes the different components of the software system and their interaction. Sven takes the role of a student, and assumes, correctly, that the interviewer is aware of the structure of his project. Staffan also acts as a student, when he answers inspired by the type of academic explanation that can be found in books or that is given in lectures. The two situations mentioned, the project and the "school-setting", in which the phenomenon seems relevant, are part of the experienced context and do not represent the focus of attention.

4. Analysing context in the different stages of a phenomenographic study

The distinctions of the term context presented above owe their value to how they can help a researcher in his task of obtaining good results from his research endeavours. The examples given above point towards different applications. But still, a systematic way of taking context into account all through the research process is needed so that these distinctions can serve a purpose. I will walk through the different stages of a phenomenographic research project and discuss how different nuances of context come to be important and in what way an awareness about them contribute to the total outcome of this project.

The way of working that is presented here is not to be taken as an algorithm – phenomenography is not prescriptive and, consequently, does not demand that certain steps are taken. Instead "the actual methods used vary according to the specific question being addressed" (Booth, 2001, p. 172). The researcher must keep the subject area present throughout the research process, not as an isolated phenomenon in itself, but as a crucial part of the learning situation.

Formulating the research question

The research question, emerges from the researcher's background and immediate context, and is formulated within a field of research related by subject interest and by methodological approach.

In phenomenography, it is how the students experience some phenomenon, or a set of phenomena, that is in focus. These phenomena are known to the researcher, in his role of a computer scientist. The strategy is aimed at revealing how it is known or understood by the students. In formulating the research question, therefore, the researcher has to be open to ways of knowing the phenomenon other than his own, and has to be open to the research participants experiencing the context in unexpected ways. The researcher should thereby be avoiding two potential traps: excluding the participants' experienced context(s) from the context of the research question, and taking a common cultural ground for granted.

Data collection

As Booth (2001) points out, "phenomenographic research into learning is empirical in that the source of data are the learners themselves" (p. 172). The aim of the data collection is to collect *a pool of meaning,* that is, expressions of the varying ways in which a particular phenomenon is experienced within a group.

The predominant way to collect data in phenomenographic research projects is through interviews. A researcher normally starts an interview with a set of open questions, based on his ideas of what he wants to learn from the interviewee. The interview normally has an open form, where the interviewer encourages the interviewee to talk freely, and during which interesting statements from the interviewee are followed up.

What then are the relevant aspects of context in data collection, if one goal is to maximise the variation in the pool of meaning? The researcher acts in a context as he experiences it, in which a particular interview is seen against the background of earlier interviews and the anticipation of the interviews to come. That context can be seen as if the interviewee of a particular interview, through the mediation of the researcher, participates in an on-going discussion around a certain phenomenon both with the researcher and all the other interviewees, the latter being intellectually present while physically absent.

The researcher has a certain aim: he wants a particular phenomenon to become the focus of mutual attention in such a way that the participant can reveal the ways in which he experiences it. To achieve this, he *prepares* contexts for the participants to engage with, to experience, and to speak in. This can include choosing the environment where the interviews are held, choosing the theme of the interview, working out what questions to ask, planning specific follow-up questions that might be needed, and remaining open and flexible, patient and persistent throughout.

In this study, the different concepts under investigation were discussed within several contexts, prepared by the researcher. Two extracts where the network protocol RMI is discussed in different contexts can serve as examples. In the first excerpt, the interviewer introduces the theme concerning the changes the team has decided to make to their code.

Interviewer: Um, you mentioned three changes you plan to do on it. Can you tell me more a bit more about that?
[...]

Albert$_1$: Um, the client and server separation, um, is going to involve a little bit more. In fact it will probably involve quite a bit more. [...] And the way it receives the client object as it's passed from the server, the client calls the method and it passes itself to the server and then the server passes it to the navigation. And then the navigation class uses that object, the client object, to call a function to send the path that was run back directly to the, excuse me, client.

Interviewer: Back to the client?

Albert$_1$: Yeh. Back to the client, which, um, the way that we've been reading about RMI, is not the way that it should be done. [...]

In this context, where the research leads the discussion to relate to the project, the client-server separation is discussed. Later during the interview, the interviewer asks Albert about RMI:

Interviewer: You have mentioned some here. What is RMI, could you explain that to me please?

Albert$_1$: Um [...] But it stands for remote method invocation and what it is, is you have an interface that is, um, that a class, um, uses this interface and the client also uses this interface.[...]. And so when the client wants to use those methods, what it does is it does a look-up on that server. You pass up the server and you pass up the object that you want to look up. And what it returns is that it returns that object, and then, in this client, you use that object as you would a local object. You could call functions on it, stuff like that, um..
[...]

Albert$_1$: So you can call methods that are actually located on the server and it does things like that. And that's the basic concept of it. It gets pretty involved if you do call back, um, which I haven't been able to find too much on.

This time Albert explains RMI, without referring to any specific machines or computers. He does not mention explicitly that objects can be on any machine, but seems to take for granted that questions of this kind "What is RMI?" shall be answered with a "school-book" answer. These two discussions of RMI differ, in that the first is related to the project the student is working on, and the latter is a discussion in more general terms. Together they offer different aspects of how RMI is experienced by the students.

The ground for selecting a certain code have been discussed with the students. As an interviewer I expected answers that discussed computer science and/or the students' study situation. Sebastian, however, offered a different answer:

Interviewer:	Yeah, you said that you used code number 3. Why?
	[...]
Sebastian1:	No, I think it is number 1, because, somewhere, somone said that the codes were listed according some kind of rating: Ett, två, tre
Interviewer:	Who said that?
Sebastian1:	I dn't really know. I think of was one of the Americans [...]

This is an unexpected answer, more psychological or cultural in its nature, in that it interprets the teachers' behaviour, rather than relating to computer science or Sebastian's study situation.

I have argued that by being careful to maintain an awareness of the experienced context the researcher can exploit the prepared context during data collection to strengthen the variation of the pool of meaning.

Analysis of data

Albert's discussions of RMI, that were introduced above, can be further considered in the context of the data analysis. When the two data extracts are considered in the pool of meaning, stripped of the context, the statements can be analysed as representing qualitatively different ways of understanding RMI, in the light of two computers and the internet respectively. The discussion with Sebastian, also mentioned above, has been analysed with the whole pool of meaning as a background into a category labelled "Pragmatic or opportunistic aspects constitute the basis for the selection of code". In both these cases the original contexts, that concerning RMI and that concerning the selection of a code, that were prepared and introduced by the researcher (the project and a general university setting) or taken up by the student (interpretations of the teachers' aims) are no longer present in the categories. How did this happen?

Statements made by the students are taken out of the interviews in which they were originally uttered. They are then, by the researcher, put into another context, now of the growing categories. Thus de- and recontextualisations are made in an iterative process, where the researcher starts with a tentative understanding, and then, through reconsiderations and refinement, reaches an outcome space.

The *decontextualisation* is an essential aspect of the analysis stage of a study, in order to see the phenomenon under scrutiny. The very interviews are decontextualised from the situation in which they were made. And then, as the pool of meaning is formed, individual pieces of data are removed from the prepared context and lose contact with the experienced context.

During the *recontextualisation* the researcher has the freedom to take individual extracts of data and put them into juxtaposition with other pieces of data, or with prepared contexts, or with whole interviews, with his knowledge of the full situation that data is discussing. It aims at a recontextualisation of the pool of meaning into the set of logically and empirically related categories of description.

The researcher is, when analysing the data, in a learning situation (see page 37). As the researcher is a learner, it is then clear that the researcher's context is of fundamental importance in creating structure and meaning in the categories of description.

For example, the resulting categories are stripped of the contexts of the individual: no mention of projects or interpretations of the teachers work. The individuals' own words are lost (though individual fragments of data are used to illustrate the categories in a full description). The collective context, or spectrum of experienced contexts are hidden in the set of descriptions. The researcher's context is once again dominant.

Deploying the results in context

The very aim of the current research project is pragmatic: to improve certain aspects of teaching and learning in computer science. With such a focus in the research endeavour, the issue of the deployment of the results is of utmost importance. The goals are only met, when the results are brought back to the situation from which the research questions originally stem.

In many projects the three stages already considered make up the whole of the phenomenographic study. What happens after the arrival at and description of the outcome space depends on the formulation of the research question, and can shift from phenomenography's second-order perspective back to a first-order perspective. But the results that are then used are, in themselves, stripped of all context, unless the researcher makes a conscious effort to relocate the results in their original contexts. This can be carried out at any of three levels – the individual's, the collective's or the researcher's level.

At the *individual's level*, the researcher can reflect the outcome space back on the interview, to let the results illuminate the original interview. This is practised in chapter 10, where it is argued that a good understanding of a network protocol is described as a capacity to shift in relevant ways between different ways of experiencing the protocol and that such shifts are occurring in the data.

At the *level of the collective*, studies can be used as a foundation for conclusions about the variation in how a certain phenomenon is experienced in a larger group of students. An understanding of the variation in the ways a network protocol is experienced, is certainly a good tool for teachers both when planning teaching and in the classroom.

Finally, at the *level of the researcher*, the results become a part of the researcher's understanding of his field, and thus inform future research. This line of reasoning is visible in chapter 10, where the results from section 1, that concern variation in students' understanding of individual protocols, have informed me, as a researcher, in my analysis of the general concept of network protocols, that is presented in section 2.

5. Using the concept of context in phenomenographic research

The concept of context in phenomenographic research is analysed and developed in my work in order to make possible the development of a methodological framework to study the students' experience of learning within the learning environment. To reach this aim, this section distinguishes between the context, as it is experienced by its different "owners" and from the perspective of the prepared context for an interview. These distinctions provide direction, in that they offer the clarifications that are needed to design a methodology that takes context into account in phenomenographic research. In the next chapter, where a methodological framework serving to take context into account in a phenomenographic research project is developed, these insights will be applied.

Chapter 8. A methodological framework

A phenomenographic study of selected phenomena in the Runestone environment offers insights into particular constituents of the students' experience of both their learning and their learning environment. Although the students experience the learning environment as a whole, rather than as a set of isolated phenomena, there is a danger that the researcher may get a fragmented view of the situation.

To address this issue, I have developed a framework that uses concepts of activity theory to synthesise the phenomenographic outcome. By using the intellectual tools from activity theory, the researcher can better understand how the different constituents interact, and can thus provide a holistic account of the experience of learning in this particular environment.

In this way, the framework offers a tool for analysing and describing *the learning of computer networking in a complex course setting as experienced by the students*.

The research discussed in this chapter is thus related to a methodological development that enables the researcher to take a systematic perspective of the issue of context in my project. The discussion of what context means is taken further from the previous section, in that I here propose a methodological framework[46] for my project.

1. The points of departure for the methodological framework

Marton and Booth (1997) define the unit of research of a phenomenographic project in the following terms:

46. I use the term *framework* to denote a way to perform research. This framework is targeted towards certain types of research questions. The term *approach*, that I use for phenomenography or activity theory does not imply that certain procedures should be followed, and have a more generalized meaning than framework. I prefer not to use the term *method* to denote the framework, since method, in my understanding, *only* stresses the procedural component, and thus can be used in a similar way to the term *technique*. A *methodology* focuses on the study of methods and the philosophical assumptions that underlie the research process.

> The unit of phenomenographic research — a way of experiencing something — […] is an internal relationship between the experiencer and the experienced. (ibid., p. 113)

Taking this as the point of departure for the methodological framework, the issue of relating it to, or extending it with, the experienced context of the individual and/or that of the collective becomes crucial.

1.1. Focus on phenomena within the subject area

In the discussions concerning the framework, it is useful to distinguish between those phenomena that are parts of the learning object within the subject area, and those phenomena, that although not present as learning objectives, are parts of what constitutes the learning environment for the students. As examples of the first, that I will refer to as *phenomena within the subject area*, can be mentioned the different network protocols, both in practice and in theory, as well as the use of compilers and methodologies for programming design. Those phenomena that are parts of the learning environment, will be referred to as *phenomena that are contextual to learning of the subject area*, can be exemplified by the collaboration with other team members or the grading, issues that often are regarded as important by the students, and that sometimes play the role as goals in themselves. The distinction is important, since my aim is to study students' learning *of computer science* in the international distributed project course. Thus, for me as a researcher, the learning of computer science should stand out in the fore, with other phenomena constituting the background

However, the students' relationships to these phenomena change dynamically. A student can, for example, focus on some aspect of a concept within the course content, such as a networking protocol, at a particular moment, whilst another issue, such as concerns about decision-taking in the team, are relegated to the background. Later, focus might shift to decisions concerning the distribution of work, with the concepts of network protocols residing in the background.

Both learning concepts within the course content (for example network protocols) and experiencing other phenomena related to the course (for example a team structure) are important for a researcher to describe and analyse if he is to understand the students' experience of their studies in a particular setting. Both are needed to understand the complex relationship between the learning outcome and the experience of the learning environment.

1.2. The level of analysis

To structure such an analysis, it is useful to base the analysis on the three different levels of the concept of context in phenomenographic research: the experienced context of the individual, the experienced context of the collective, and the experienced context of the researcher. These distinctions are here extended to encompass also the concepts of an activity system:

(1) At the *individual level*, case studies are constructed, where individuals are studied in the light of the activity. Relations and contradictions in the different ways in which an individual experiences a learning environment, give the researcher a "feel" for the individual in the activity.

(2) At the *collective level*, the contradictions are described and analysed. As stated earlier, the contradictions within an activity system are its source of change. Analysed in this way, the contradictions between and within the *experienced* constituents are discriminated and discerned, offering a way to understand the dynamics of the learning processes in the team.

(3) At the *researcher's level*, a further analysis can be performed. The new analysis can be structured to reveal contradictions and aspects related to the interrelationship between components of the activity in question, and to those phenomena that arose originally, through the different categories. By performing this analysis, the researcher can gain insights into the learners' experience of a particular aspect of the activity.

Analyses at these three levels, taken together, offer to researchers and course organisers tools that can be used for understanding the learning in a real situation as experienced by the participants.

1.3. The methodological framework

Now, when the various roles of different phenomena are described, and the different levels of the analysis are set, an implementation[47] for the framework can be proposed. The structure of the implementation deployed in this thesis is shown in figure 9.

First, interview data is analysed in the phenomenographic tradition. The resulting categories and their relationships are then further analysed based on elements drawn from activity theory. The outcome of the 'pure' phenomenographic analysis is thus not the final outcome of the research project. In addition to its role as an outcome in itself, this phenomenographic account serves as material for analysing the students' experience of the learning environment in a particular situation as it is perceived or experienced by its participants. The final outcome of a research project is then multi-faceted and consists of several components.

---

47. Certainly, the framework can be implemented in several ways, as long as the aims are met.

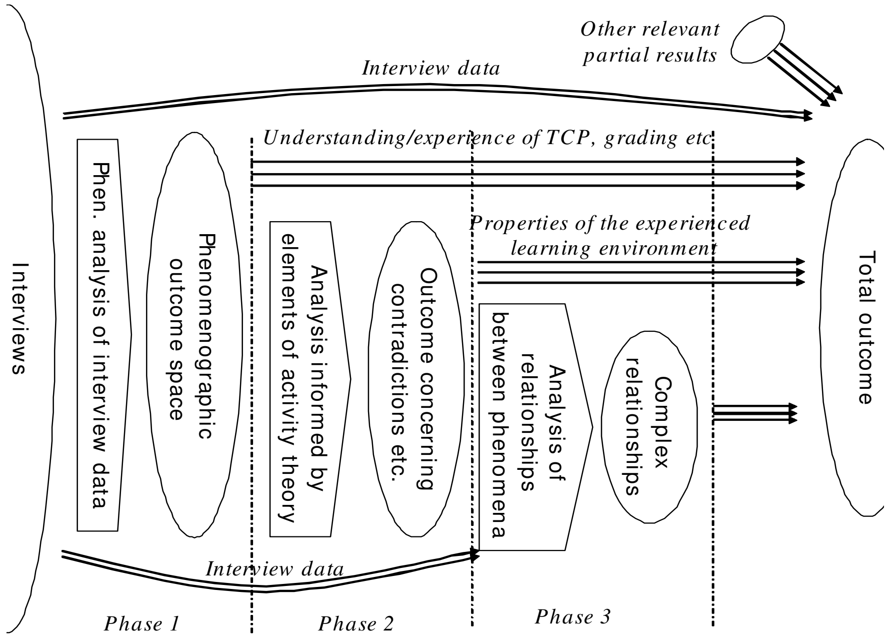

Figure 9. The flow of the analysis performed according to the framework. The figure illustrates the phases after the interviews are performed. The pentagons illustrate steps of the analysis, while each oval represents the outcome of a particular step in the analysis. A further description is given in the text.

1.4. The three phases of the flow of the analysis

Although the framework is not algorithmic, its overall structure can be described in three phases, or steps.

In the first phase, the left-most pentagon represents the phenomenographic analysis and the oval represents its outcome: the variations in the students' experience of different phenomena in their learning and their learning environment. The analysis must here be performed with the issues of context in mind, as proposed in the last chapter, to assure that the phenomenographic outcome space can be used in the later phases of the project.

During the second phase the analysis is advanced by means of some intellectual tools related to activity theory. Through the analysis represented by the second pentagram, the researcher gains the means to further explore his phenomenographic results by deploying concepts from activity theory. The aim is here to create the potential to capture, analyse and describe the contradictions that form a part of the participants' experience of the whole. In the second phase, each category of description is first associated with one or more component(s) of an activity system by the researcher. This association demands of the researcher that he "goes inside" the activity and explores it from the participants' perspective. Through such associations each component of the activity comes to have some phenomenographic categories related to it, that together indicate the variation in how this particular component is experienced by the subjects.

Finally, in the third phase (shown in the small pentagon and its small oval) a further analysis, inspired by activity theory, is conducted. Here, the partial results concerning categories and contradictions are synthesised in order to reveal different forms of relationships between categories. As will be demonstrated in the empirical part of this thesis, the various categories of a particular phenomenon can be associated with different components. Each component of the activity system can, as a consequence, contain aspects of descriptions of various phenomena. The triangle of the activity system, now describing the various ways in which the learning situation is experienced by its participants, is further analysed deploying some intellectual tools of activity theory. Particularly, contradictions will be sought for and analysed.

The total outcome, aiming to explore the full picture of the students' experienced learning environment, consists of the results from these different phases.

1.5. Levels of abstraction in the three phases

The flow of the analysis can be discussed from another perspective as well, focusing on the level of abstraction: The data for phase one consists of the interviews together with other data that has been collected. Data here is concrete in its form. Specific statements of named students make up the core part of the data. The outcome of the phenomenographic analysis (input to phase 2) is more abstract in that individuals are "lost" and the descriptions are now at the collective level. The output of phase 2, which, together with the interviews also serves as the input of phase 3, is still more abstract in the sense that the results at a collective level now are the object of the investigation, which aims to reveal relationships between categories. At the same time, data here also consists, through the interview excerpts, of concrete statements, aiming to "flesh out" the previous analysis. And finally, the total outcome is concretized, but now not on an individual level, but instead in issues of teaching, learning and course design.

1.6. Remarks concerning the framework

The framework as presented in this chapter is intended to be used in the *analysis* in a research project. Consequently, it does not offer any directions for the researcher either in the formulation of research questions, data collection nor in the deployment of the result. Still it is not independent of these steps in the research process: It is intended to be a tool for the analysis of a certain class of research problems, namely those where the experience of learning of a subject area occurs in situations where the students study in complex learning environments.

Data must be collected with this aim in mind, and must encompass students' statements about concepts in the learning (for example network protocols) and as well as issues that are contextual to the learning. Finally, to deploy the results and bring them back into teaching and learning situations, it is necessary to have an environment that is open to change.

The framework is here described in an idealized way and is analytically divided into three phases, each representing a specific step of the analysis. In a real setting, where the framework is deployed, several of the phases can be performed in parallel, for example the work with one phenomenon in the core of the learning object can be in the first phase, while another (as an issue related to the learning environment) may have reached the second phase. Working in this way offers opportunities for the researcher to use insights gained from the analysis in a later phase, in an earlier phase of his work with another phenomenon.

2. Examining the role of activity theory in the framework

In the previous chapter, a distinction between *experientialist* and *externalist* approaches to research, introduced by Ekeblad and Bond (1994), is discussed. Applying this distinction, a phenomenographer clearly takes an experientialist's perspective, while a research project in the activity theory tradition rather can be expressed as having an externalist's perspective.

However, looking deeper into this aspect of activity theory a more complex picture evolves. A participant in an activity, a subject, such as for example a student in a university course, is an integrated part of the activity. The activity is a context in and within which he or acts and interacts. In this way the student, through his or her actions, is integrated with the activity. It is thus meaningless, with this approach, to discuss the "impact of context *on* an individual" (compare the quote above). Still, I argue, the activity theory research approach is an externalist approach, since its research object is the *activity as such.* An activity is a theoretical construct, that is formed by the researcher to analyse a complex situation. When describing an activity as an evolving system, stressing the collective structure and the historical components, the perspectives from outside, and from the researcher, are needed.

My perspective is different, compared to that of activity theory, as I do not focus on the system as such, but on the *experiences* of the individuals who are parts of the activity. With my approach, I get “close to”, and even “inside”, the participants as I describe their experience, while I, as a consequence, have a lesser possibility to study the activity system as an entity in its own right.

Another aspect that makes firm statements on activity theory as a “pure” externalist approach questionable is related to the choice of empirical data. Data based on experience can, and according to Engeström (1987) should, be used by the researcher for gaining insights about the activity.

He also addresses the issue of capturing and using data collected from individuals, or, with his terminology, “personal views”, in Engeström (1990), and illustrates his arguments with an empirically based example from Finnish doctors’ meetings with patients. During the meetings the object of a doctor’s work changes from focusing on the patient’s external features, to a first impression, and then further to a meaningful pattern. Finally, the outcome from this pattern is a diagnosis and a treatment plan. As the object changes, the whole system evolves in an integrated process. The intellectual and physical tools used by the doctor vary as the activity evolves. While history taking is more important in the beginning of the meeting, examination and test findings are mainly used by the end. The interesting point with this example is that the evolving activity is seen from the doctor’s perspective. The object and the tools are described as the researcher interprets how the subject sees them.

In the same article Engeström also advocates that actors within the activity should take a “system view”. He expresses his purpose in the following way:

> [...], I will argue why it is vitally important for the actor to take the system view and for the researcher to take the personal view. This does not imply an attempt to merge or ‘bridge the gap’ between the two views [...]. I argue for switching between multiple views.

Nardi (1996b) follows a similar line of reason when discussing where data for the construction of the object should be collected from:

> [A]ctivity theory provides the more satisfying option [than situated action] of taking a definition of an activity directly from a subjectively defined object, rather than imposing a definition from the researcher’s point of view. (p. 83)

Here, Nardi argues that the model of the activity should be based on a subjective definition of an object. However, she does not say that the activity as a whole, or the analysis of it, shall be solely based on experiential data.

3. Examining phenomenography in the framework

A cornerstone in phenomenography is *variation,* as phenomenographic research aims at discerning the *variation* in the ways in which something can be experienced. Individuals experience something in differing ways, although the phenomenographic research can discern a limited number of qualitatively different ways in which a phenomenon is experienced. The description of these categories, each corresponding to a particular way of experiencing the phenomenon, discerned by the researcher, forms the basis for the outcome. With the aim of activity theory, namely to study an activity as an entity in its own right, the situation becomes different. A research project within this tradition normally focuses on the development of *one* activity, or *one* network of activities, and is analysed in an externalist perspective, that is, from a view-point from the outside. With this approach, where the study object is *one course of events*, the issue of variation becomes subordinate or disappears completely.

Thus, the question of by whom, or from which perspective, a phenomenon or activity is experienced or analysed has become a key issue. Since I study an activity, as it is experienced by its participants, deploying a predominantly phenomenographic approach, it is important for me to analyse and describe the *variation in the ways* in which the activity can be experienced.

When studying someone's experience of something, issues concerning the *awareness* and *changes in awareness* become crucial. Marton and Booth (1997) stress this issue by defining the individual way of experiencing a phenomenon in a particular way as "[t]he aspects of the phenomenon and the relationships between them that are discerned and simultaneously present in the individual's focal awareness" (p. 101). They also point out that the focal awareness changes, and that we, at a particular moment, only have one or a few aspects in focus. In this way the experience of a phenomenon dynamically changes with the change of focal awareness. This is illustrated in the chapter concerning students' understanding of network protocols (chapter 10), where it is shown that students' experience of a particular phenomenon in the learning situation changes, spontaneously or when being stimulated or triggered, for example by questions where the phenomenon has been presented in a new context.

When deploying the tools drawn from activity theory to study the students' experience of learning, care must be taken by the researcher not to confuse the dynamic changes in the students' experience with the changes of the activity as a whole. While the former takes place in the relationship between an individual student and aspects of phenomena within the activity, the latter is a change mainly outside the student, but in which the student through his or her actions is a part. An experience of a phenomenon is not delimited to the phenomenon as such. Instead, we experience "situations of which a particular phenomenon may be a part, transcending it, but still experienced against just that background" (Marton & Booth, 1997, p. 142).

A learner focuses his or her awareness on various aspects of the phenomenon at different moments, and thus, with the different focus, delimits the phenomenon differently over time from the background against which it is seen. As a result, the researcher can not presuppose that there exists a constant distinction between a phenomenon and its background, that is invariant over all the categories discerned. Thus, a one-to-one mapping between the components of an activity system and the categories of description of a phenomenon may only occasionally exist. Instead, the categories describing the different ways in which a phenomenon is experienced can be related to different components.

As is illustrated in the discussion in this section, care must be taken not to confuse the different perspectives in the two approaches: as experienced from the inside by its participants or as it is discerned by the researcher from the outside.

4. The terminology in this methodological framework

A research approach within pedagogy does not only carry with it the basic assumptions about learning, and how learning can be studied. It is also, in itself, a system with its own history and its own language to express ideas. When combining research perspectives, some clarifications about basic concepts as well as the terminology might be needed. I will, in this sub-section, discuss the terms *context*, *collective* and *experience* in relation to my approach.

Context

There have been lively discussions on the issue of *context* and its implications for learning and for research about learning.

Engeström (1993) summarises the notion of context within activity theory in the following way:

> For activity theory, contexts are neither containers nor situationally created experiential spaces. Contexts are activity systems. An activity system integrates the subject, object and the instrument [...] into a unified whole. (p. 67)

With this meaning of context, earlier discussed in chapter 6, a learner is a part of and is acting with and within the context. The course that this project is based on can, as an example, be regarded as an activity and thereby a context.

In phenomenographic research the word context has a different meaning, related to the awareness of the experiencer, and differentiating between that which is in focus, and that which is relegated to the background, or context, in a certain situation.

Clearly the two usages of the word are different. The phenomenographic use of the word is at the same time more narrow (since it is delimited to experience) and broader (since it is not limited by the activity, as it is discerned by the researcher). It would thus be incorrect to compare activity theory and phenomenography through the use of the term. Instead, the different meanings can be related to the different needs within the two traditions: Activity theory focusing on a system as a whole, and phenomenography taking the experience of individual learner as its point of departure.

Collective

Another word that has different interpretations within activity theory and phenomenography is *collective*, expressing a unit or an entity on its own within activity theory, while the word within phenomenography describes an analytic abstract construct, comprising the collection of people from whom data has been collected, stripped of the direct reference to the individuals and extending to similar people to whom one wants to apply the result.

Experience

The idea of *experience* and of experiencing something is a keystone in phenomenographic research and has thus a well-defined meaning related both to the individual that experiences something and to what he experiences. In activity theory, on the other hand, the word is used in a vague way and has no well-defined meaning.

Marton and Booth (1997) define *experience* in the phenomenographic perspective:

> An experience is an internal relationship between the person experiencing and the phenomenon experienced: It reflects the latter as much as the former. (p.108)

Clearly, this particular meaning of the word is not relevant in research in the activity theory approach, since it relates to an idea that is not present within that research approach. Instead, the word is used in different way: sometimes in a commonplace meaning, sometimes more specifically to contrast data collected through interviews or in similar ways such as *experiential* in the quote from Engeström on page 75. With this research being carried out with a phenomenographic approach, I use the word experience as is commonplace within the phenomenographic tradition.

5. Other projects encompassing phenomenography and activity theory

An extensive literature review has revealed that only a few research projects have been carried out where the use of phenomenography is extended to encompass aspects from activity theory. Also, the related approach has been searched for: to extend a research project in activity theory by considering aspects that are normally referred to as phenomenographic is also rare. The studies that have been found are different in their characteristics and their ways of encompassing the two approaches.

Åberg-Bengtsson (1998) studies young children who learn to draw and interpret graphs and charts in a basically phenomenographic project. The relationship between the learners, who collaborate with a researcher/mentor, and the object of their studies are studied in a phenomenographic tradition. She thus presents the learning that takes place as an outcome space consisting of sets of categories of description. The learning that she describes is however not only the children's relationship to what they study. Rather, since the children work in teams together with the researcher, the working group is a collective entity. In order to address this complex situation, as well as other contextual issues, Åberg-Bengtsson studies the whole learning situation in the light of an activity theory model.

In her work concerning the introduction of a new ICT-based tool into teaching and learning of mathematics at first year university level, Coupland (2004) aims at describing the different ways in which the students engage with the software, and how they appropriate it. The authors link phenomenography with activity theory at two "levels": On the surface level, phenomenography serves as a guide to the kind of data to be sought and a way to deal with the data collected. At a deeper level, she argues that the second-order perspective and non-dualistic views of the two approaches offer links.

Gordon (1998) has in her thesis investigated university students' orientations to learning statistics. She uses phenomenography combined with statistical analyses, to explore relationships and events for particular individuals in a setting that was bounded in space and in time. Activity theory served as a framework to transcend the particular context. In a later paper (Gordon, 2004), she deploys activity theory as a "filter", that serves to offer a focus and a framework for the mainly phenomenographic investigation.

In their study of distance learning for engineers Hultén and Booth (2002) have extended their phenomenographic approach through activity theory. They describe an extended relationship between an experiencer and the experienced, where this relationship is mediated through the tools as well as other components in the activity. Thus it is clear, that their approach in important ways resembles the approach presented in this thesis.

Kärkkäinen's research (1999) is predominantly within the activity theory approach, when she studies Finnish and American teachers who collaboratively produce a curriculum for elementary school. However, when analysing interview data concerning network building outside classrooms she is approaching a phenomenographic analysis.

A more theoretical approach, but still illustrated with empirical data, is taken by Gordon and Nicholas (2002). In their paper they argue that activity theory and phenomenography has a certain methodological fit. Despite their differences, they show similarities and are both clarifying and descriptive tools.

6. Towards the empirical study

This chapter offers a framework for taking the context into account in a phenomenographic research project. Together with the previous chapter, which highlighted the importance for the researcher to have a consciously developed perspective on context, the scene is set for the empirical study on how students come to learn computer systems in a international distributed project course

III. Empirical investigations

Chapter 9. The empirical study

This chapter describes aspects of the empirical study. In its first section, some strategic decisions are discussed, while the second highlights some methodological characteristics of the research project.

1. Objectives for selecting certain issues to research

When tackling the question about which issues to explore empirically, my point of departure has been taken from my research questions. With the complex environment that the students experience in the Runestone project, there are a number of phenomena that in different ways would contribute to the whole.

1.1. Which phenomena can be researched?

The following measures were used to determine which phenomena should be considered for research in the project

1. *Relevant for the field*. The field of computer science education research has an ultimate aim to improve teaching and learning in computer science. This aspect is then important for my selection of phenomena to investigate.
2. *Relevant for the students*. Since the purpose of this project is to investigate the students' experience of their learning, only those issues that are relevant to the students, are worthwhile exploring in the project.
3. *Researchability.* The researchability is constrained by the chosen research approach, since when the approach is selected some issues are no longer open for inspection.

The three criteria above delimit the type of questions I *could* investigate, but do not offer guidance in which questions I *ought* to tackle. Here, I have instead turned to phenomenography and activity theory for guidance.

1.2. Which phenomena is it relevant to study?

The distinction between the *what* and the *how* of learning, and the further refinement of the how aspect into the *act of learning* (see chapter 5) can, together with elements of activity theory, serve as an intellectual framework in my selection of which phenomena to explore.

The *what* aspect is primarily represented by the students' understanding and learning of computer network protocols. Network protocols were key components in the software systems that the teams were to construct and thus represented a key competence needed in the project. This investigation is discussed in chapter 10.

The *how* aspect is discussed in terms of the *motive* that underpinned the students efforts in the course (see chapter 11) and the act of learning indicating how they went about reaching their aims (see chapter 12).

The results foreshadowed above are not sufficient to describe the students' experience of learning *in a certain learning environmen*t. As a guidance in the process of selecting phenomena in the environment, I have deployed activity theory as a "magnifying glass", both when designing the interviews and when selecting phenomena for the analysis. Here, I have striven to "distribute" the issues analysed in such a way, that different constituents of an activity system have been touched upon. Phenomena that represent constituents of the learning environment, such as the experience of being a group member, have been blended with more "low-level" phenomena, such as the function of the milestone meetings (the teacher led team meetings) in this analysis (see chapter 13).

2. Methodological characteristics of this study

As a researcher, I have to take standpoints not only on which data I shall collect, analyse and report, but also consider methodological issues of importance to the study.

2.1. The interviews

Ten students were selected as candidates for interviews in the US and nine in Sweden during the spring of 2001, mainly based on background questionnaires and data concerning previous studies recorded by the two universities. With the aim of obtaining rich data for the analysis, variations in backgrounds, earlier study results, gender, age, motivation to take this course etc. were sought for. The students participated on a voluntary basis in the study and did not get any credit for participating.

Two interviews have been conducted with each student[48]. They have been carried out by me in Swedish with the students at UU and with a Swedish exchange student at GVSU and in English at GVSU and with an exchange student from a European country studying at UU. The first interview was made a few weeks after the course had started, and the second was carried out after the end of the course.

The semi-structured interviews began in similar ways. I introduced myself to those who did not know me, explained briefly about my research project, and about the privacy principles. There were many improvisations during the interviews in order to follow up statements by the students and to increase the variation in the ways that the topics were discussed.

2.2. Considerations concerning trustworthiness and ethics

The issues of trustworthiness and generalizability in phenomenographic research have been continuously debated. A key question in the debate is whether intellectual tools, normally related to positivistic research approaches, such as validity and reliability are relevant or if other arguments on trustworthiness of processes and results are needed. Cope (2004) argues that the analytic framework that is offered by the structure of awareness, as presented by Marton and Booth (1997) can serve as a basis for an analysis of validity and reliability. Different degrees of interjudge reliability in the creation of the categories have also been proposed as measures (Säljö, 1988; Marton, 1994). Sandberg (1995), on the other hand, argues that the interpretative process, which is in focus in the phenomenographic research project, would risk to be incorrectly judged by applying tools that originally were developed for positivistic research. Instead, he proposes interpretative awareness to judge the trustworthiness of a phenomenographic research project.

Taking the ideas behind Sandberg's (1995) standpoint as a guide, the issue of trustworthiness becomes a question for the reader. The researcher can offer transparency all through the process and in the results and describe the factors that influence his interpretations.

The research process in my study has, from the formulation of the research questions over the data collection and the data analysis to the presentation of the results, been a part of the departmental practice, integrated in pedagogical research traditions, and has been carried out by me in my role as a lecturer in computer science and doctoral student in the research school supported by the Knowledge Foundation, with focus on learning and information technology (LearnIT)[49]. All steps have been open

48. With a few exceptions, the students attended both interviews. Those who did not, offered explanations such as illness, exchange studies abroad, and shortage of time.

49. http://www.ped.gu.se/learnit

to inspection. I have discussed my work with colleagues both within computer science and within pedagogy, and both from within my department and from other institutions. The methodological considerations and growing results have been debated at conferences and seminars.

The phenomenographic research process offers insights at a collective level. This means that the researcher during the iterative research process transcends the statements of the individuals in forming the results. In this way the role of a statement of an individual becomes only a fragment of the full outcome space. Hence, the importance of possible misinterpretation of individual statements is reduced. The logical structure of the outcome space, illuminating the relationships between different aspects of the whole described in the categories, offers another tool for inspecting the results, now at the level of the researcher. Lack of satisfactory structure serves as a warning and demands of the researcher that he reconsiders all aspects of his study or that he further explains his results. In this study, logical structures have been identified in all phenomenographic outcome spaces.

The ethical aspects of this study are related to the considerations concerning trustworthiness. My honesty as a researcher, towards the research questions, in the process, in the results, and to all participants are here a key issue. Particularly, I find the honesty to the students important, in the process, in that my results reflects their original statements, and in my presentation. I have explicitly got their permission to analyse their statements in the context of the study, the outcome of my research at a collective level depicts their statements, and I respect their anonymity in my presentation.

2.3. Analysis of data and presentation of the data

The results concerning computer networking, presented in chapter 10, are based on the first set of interviews in both countries as well as the second set from Sweden. Two interviews from Sweden have been impossible to use, due to poor quality of the recording. The analysis presented in sections 1 and 4 of chapter 13 are also based on this restricted set of interviews. These results have later been confirmed, when the full set of interviews have been available. For the remaining analyses, all interviews have been used.

The interviews have been transcribed by native speakers of the two languages. I have, for the important parts of the interviews, checked the transcriptions. As a tool to index and sort my results, I have used a software system intended for analysis of qualitative data, Nudist from QSR Systems[50]. The translations of Swedish interview extracts presented in this thesis are made by a native English speaking computer scientist, who is fluent in Swedish.

50. http://www.qsr.com.au/

The use of Swedish and English

The interviews were analysed in their original language. Data thus consists of interviews in two languages, intertwined in the pool of meaning. Normally this way of working has been unproblematic, but occasionally language-dependent issues have arisen. An example of a situation when this issue comes to the fore is shown in the excerpts below, stemming from an interview that was carried out in Swedish. Sven has misunderstood, or does not remember, the correct meaning of the abbreviation RMI.

$Sven_2$:	Remote, and then there is Indication, but what is the other, in the middle then. I am not so ....
Interviewer:	Method
$Sven_2$:	Remote Method Indication
Interviewer:	And what's that?

This can have several explanations. Apart from indicating unfamiliarity with the concept, it can be a matter of the language. Since his mother tongue is Swedish, it is not obvious that the word "invocation" ("anrop" in Swedish) is a part of his English vocabulary. The other two words offer less difficulty: The Swedish equivalence of "method" is "metod". The word "remote" is frequently used within the field of computer science and is most known to an advanced undergraduate student in computer science.

Names of students and staff

In order to preserve the anonymity of the students, they have been assigned names that differ from their real names. The students, studying in Sweden, have been given names that start with an "S", while names that begins with an "A" indicate that the student is from USA. The Swedish exchange student at GVSU is assigned an "American" name. The excerpts in the thesis are preceded by their name and the suffix 1 or 2, indicating if the statement is from the first or the second interview. $Staffan_2$ thus indicates a statement made by the Swedish student Staffan during the second interview.

The interviewed students have on many occasions mentioned their team-mates by name. Their names are also replaced in this thesis. American students are given names that starts with a "P", while team mates from Sweden are given names that have an initial "M".

The names of the teachers are replaced. The teacher at UU is called Urban, while the teacher at GVSU is named Greg in this thesis. I am aware that this arrangement does not guarantee their anonymity. However, they have both agreed to participate in this project, knowing that their actions and discussions during the course would be documented. Other members of staff are referred to be names starting with a G for staff members at GVSU and with a U for those at UU.

Notes on some specific terms

The terms *team* and *group* were both used to denote a set of five or six students that worked together on a software system. Among the Runestone staff and the Swedish students the two terms were used in parallel (both words exist in Swedish, as *team* and *grupp* respectively), while the term *team* was most frequently used by the American students. In this thesis I deploy the term *team* except in quotes, when the English word *group* has been used or as a translation of the Swedish word "grupp".

The term *virtual team* is frequently used throughout this thesis. The term is used to denote the students' teams in the Runestone environment. Members in these teams never meet face-to-face. There are, as is pointed out by Last (2003), many definitions of the term, as well as many alternative terms with similar meanings. Lipnack and Stamps (1997), cited in Last p. 16, define a virtual team as "a group of people who work interdependently with a shared purpose across space, time, and organization boundaries using technology". This definition corresponds well with the usage in this thesis.

Another language related issue is the distinction between a generic *internet*, as a set interconnected network and the global *Internet*. Neither in Swedish nor in English is it possible to hear a difference between the two in the statements made by the students. The context in which the word is used only occasionally offers help for an interpretation. As a consequence, the distinction between *Internet* and *internet* has to a certain degree to be guessed. I prefer the word *internet* in cases of doubt.

Gender in the language used in this thesis

Although this thesis does not address gender issues, such questions arise in the use of the language. The Runestone project, as is the case with most situations in computer science education, is male dominated. Since there are considerably fewer females than males taking the course, I have chosen not to indicate if any particular quote is from an interview with a male or a female. The four female students are therefore referred to by "he"[51]. The females could, since they are few, easily be recognised by schoolmates or teachers if they were pointed out explicitly. The alternative, to denote all students by "she" and selecting female names, would give the reader less of a feeling of the "atmosphere" of this environment, and therefore, would give less justice to the situation.

A researcher is in this thesis referred to as "he", even when used in a general sense, since I, the author of this thesis, am a man. A teacher is, on the

51. Other issues that in different ways might influence the power balance, speaking English as a native language, weak competence in English, ethnicity, handicaps (such as dyslexia) are not taken into account in my research. With the way of making the students anonymous in the thesis, such issues are not visible for a reader, although partly available in the underlying data.

other hand, denoted by the pronoun “she”. As I refer to all students by male names, I also use the pronoun *he* for team members. Certainly the arguments would not change, if “he” was replaced by “she” and vice verse.

The research design and the way of presenting the data makes it hard, or maybe even impossible, to address gender issues in the current work and still respect the anonymity of the students. However, with the data that is collected, it is possible to address gender issues in future research.

Chapter 10. The what

This chapter focuses on students' understanding of the networking protocols TCP, UDP and RMI.

1. Students' understanding of specific network protocols

The students' experience of the three network protocols TCP, UDP and RMI show many similarities, but there are also differences, some of which are directly related to the different characteristics of and ideas behind the protocols.

The students were asked during the interviews to describe by TCP, UDP and RMI. When opening the subject of discussion, the three protocols were treated as three different topics by the interviewer. The discussion of each protocol started by the question "What is TCP?" and with similar questions for the other protocols. Later in the conversation about specific protocols, comparisons were frequently made, often at the initiative of the interviewee.

1.1. TCP

Three qualitatively different ways of experiencing TCP have been discerned within the student population. These are summarised in table 2.As will be shown in the descriptions of the categories, they differ not only in terms of "as what" TCP is understood to be (their meanings), but also in their structure (their parts and their relationships) that makes it possible to discern TCP from its surroundings, that is, to "see" TCP *as something*.

Table 2. The three categories of description of TCP

	Label	Description
1.	**Safe communication**	TCP is a protocol for safe transfer of packets between two specific computers
2.	**A connection**	TCP offers possibilities to create connections over an internet
3.	**A standard**	TCP is a formalized standard decided by a committee

Category 1. TCP as a safe[52] communication between two computers

Andy's statements during the first interview show the focus on two computers, which is a key characteristic of the first category:

> Interviewer: What is TCP?
> $Andy_2$: That is ... you communicate with .. between client and server with TCP packets.

Here, Andy describes TCP as communication between two specific computers, a server and a client. The concept of client-server implies that the issue of communication is integrated with the computers, since a server and a client could not be imagined without communication between the two; the communication is the basis for the existence of a server and a client.

In the continuation of this dialogue, the issue of safe communication is raised:

> Interviewer: What is a TCP packet?
> $Andy_2$: That's a type of packet, that one sends, that contains also ... so that one can get.... one must. It is a safe communication so that one knows ...[...] so that one always knows it arrived or not, in contrast to UDP.

Andy here points out that TCP provides safe communication and says that TCP informs whether data, in the form of a TCP packages, has arrived or not.

Sebastian explains during his second interview his understanding of TCP and in particular the use of acknowledgements to make sure that information arrives. The following excerpt starts with Sebastian's statement about where in the project his team has used TCP:

> $Sebastian_2$: No, down from the server and down to the hardware, the bits where we use TCP/IP.
> Interviewer: What is that?
> $Sebastian_2$: It is...it is a communication protocol which uses...ack?
> Interviewer: Acknowledgement?
> $Sebastian_2$: Yes, an acknowledgement, That is, that I know that the information I send has arrived correctly, and what comes back has also arrived. There is a bunch of other stuff that I have to look out for.

52. The term *safe* is used as synonymous to *reliable* in order to stay closer to the vocabulary used by the students.

By the words “I know that the information I send has arrived correctly”, he indicates the reason for the acknowledgement: To get a safe communication between the two specific, communicating computers.

A word about the hesitation in the third statement might be required. When Sebastian explains what a communication protocol is, he first has difficulties finding a Swedish word (“bekräftelse”), so he turns into English and starts saying “ack...”, for acknowledgement in a hesitant voice. As the interviewer, I then present the Swedish word to him, which he uses in the next statement. Since Sebastian has studied computer networking and TCP in a language other than Swedish or English before taking this course, I interpret his hesitation as a question of language and not as related to the concept as such.

This category discusses an understanding of the TCP, where the protocol is used for transferring data in packets between two specific computers. The focus is the packets and the two computers. TCP uses acknowledgements to verify that the information arrives safely at the destination.

Category 2. TCP as a connection over a network

This category expresses an understanding where TCP offers a possibility to create end-to-end connections over a network. When Albert was asked what TCP is during the first interview, he offered the following answer:

Interviewer:	Um, what is TCP?
Albert_1:	TCP, um, it’s um, part of the internet protocol. It’s used with part of the internet protocol typically. Um, it’s one of the methods of communications, I don’t know a whole lot about it, as far as the whole, um, design construction behind it.

Albert talks about TCP as an internet protocol and mentions that it is a part of an internet. Axel’s discussion follows the same line of thought.

Interviewer:	[...] Um, I want you to talk about TCP.
Axel_1:	TCP/IP?
Interviewer:	Ya.
Axel_1:	TCP/IP is how almost everything on the Internet communicates. IP addresses and everything, and that’s um, one of the fundamentals behind RMI also. One could give it the address where the object is [...] the IP address [...]

Beginning by saying that it “is how almost everything on the Internet communicates”, he indicates that he regards the protocol as a part of Internet as a whole. The importance of the protocol is emphasised by his reference to IP-addresses[53] and to RMI. Another student, Allan, also stresses that TCP is part of an internet:

Interviewer: Um, you've talked about TCP. What is TCP?
Allan_1: Basic concepts.. it's a protocol language, I guess you can call it, that you just put your data in and it's sent across the network using the different protocols you want to use, like IP or.. I can't think of any other protocols off my head. But it is more or less a packet that you put your data in and you send across and it has some features such as, keeps things in order when you, um, when you get to the, um, when it gets to the server you want to go to.

TCP is a protocol language[54] that is used for sending data across a network. In this way, he clearly indicates his view that TCP is an integrated part of the network. He then explains its main feature, as he sees it: The order of data is kept when sent to the application program through the TCP sockets, although data physically might have arrived at the server in any order. Clearly, Allan has a focus on the end-to-end communication. The connection becomes safe, while the timing, routes etc. of the individual packets vary.

In this section, a category of TCP has been described in which the protocol is understood as an end-to-end connection over an internet, and at the same time, as an integrated part of the network. The protocol is built on a technology with acknowledgements with packets that can differ in rate, individual reliability etc.

Category 3. TCP as a standard communication tool

This category of description expresses an understanding where TCP serves as a communication tool. Adam expresses this in the following way:

Interviewer: So, what is TCP then?
Adam_1: Well that I have studied in some networking classes um, Transfer Control Protocol, something along those lines. Um, that is just a protocol for computers to communicate with each other. That's a standard that was created by a committee somewhere, sometime, and it's just a, it's a protocol, meaning that it's, it specifies um, the layout and the size and what's in the header and footer of packets being sent across networks and things like that. So it's, it's a standard communication tool

53. An *IP address* is a unique 32-bit number that is assigned to computers on an internet. This address is used for all communication with the host. IP addresses are written as four decimal numbers with dots between. As an example 130.238.8.89 is the address of the computer used by the author.

54. The term *protocol language* refers to a formal language, in contrast to a natural ("spoken" or "human") language, within the field of computer science. A formal language is used to express statements about calculations in a general sense, such as for example when giving instructions to a computer.

TCP is a standard that is created by a committee. The form of the packages sent is the result of conscious decisions, taken by the committee. Later, when the choice of TCP instead of RMI as the principal protocol for their project is discussed, he continues:

> Interviewer: Yes, but can you tell why you have chosen TCP?
> $Adam_1$: Right, it's for one thing it doesn't require this registry running in the background. It's sort of a universal standard so that, you know, our applet can be run on any computer anywhere and still communicate with the game server running on Linux or whatever. Um, so I guess just being a standard and being more flexible than RMI.

TCP has two advantages over RMI, according to Adam. One is technical: TCP is simpler since it does not require a complicated background program to be run. The other advantage is that TCP as a well-defined standard increases the flexibility. Adam compares the use of RMI and TCP on several occasions during the whole interview. From his remarks above and his comments in general, it is clear that he takes for granted that TCP offers safe communication. It is never spoken aloud, rather it can be seen as a condition for the rest of his conclusions. Adam reasons about TCP without making direct references to the technical structure or the entities in the communication process. Instead, he talks about standards, flexibility, and tells the interviewer that size and design of packages are decided, without mentioning what the packages look like. In this way, he refers to properties of the protocol in an indirect way, from a position outside the two protocols.

In this category, TCP is described as a standard As such it is not only related to computers, but to human decisions as well. The discussion is mainly focused on how decisions are taken and the consequences of the design.

Structure and meaning in the students' understanding of the TCP

A further analysis of the categories of TCP is founded on the distinctions between the structural and referential aspects, as described in chapter 5. Applying this reasoning on the three categories for the TCP protocols, a difference in "as what" TCP is understood to be (meaning, or referential aspect), as well as in its structural aspect (parts and their relationships), become recognizable. TCP is, in all the three categories, experienced as an inseparable part of the framework to which it belongs. The protocols are experienced as integrated with specific computers, the network, or the world outside the network, respectively, and can not be separated from its surroundings. The protocols would not exist without the environments in which they are, and the environments would not exist without network protocols. This makes the analytic separation of the protocol from its background precarious, and care must be taken. In discussion I will therefor use the terms *in focus* or *in the fore* for that, which stands in the foreground.

The background in this particular context will be denoted *framework*[55], since it resides in the background, but offers a structure to t*hat which is in the fore.*

Focus in category 1 (“TCP as a safe communication between two computers”) is on the two communicating computers and the specific packets that they transfer. Chapter 5 stated that variation is needed to make something possible to discern. Here variation is brought about by different packets, containing different (parts of) messages, that are transferred at a varying pace. The communication between the two computers is understood in the framework of a set of other computers, constituting a network.

Category 2 (“TCP as a connection over a network”) describes TCP as a connection. A connection can, with an obvious risk of oversimplifying, be compared to a telephone call that is set up between two computers. Such an understanding of TCP has an end-to-end protocol in the fore and presupposes a network consisting of several computers that provides a media, or carrier, for the connection. In such a bigger network particular computers, or the other protocols that exist, can not be discerned. Here variation is introduced in the different routes the packets take, in which order they arrive, while the end-to-end connection is stable. The packets, that are seen as “individual” entities in the first category, are here part of a “collective”. The behaviour of individual packets is not an issue.

Finally, the third category describes TCP as a standard, determined by a committee. Here TCP is the result of human efforts. Specific computers and packets cannot be discerned. Variation is introduced by the different decisions taken that define TCP in various ways. TCP is seen against the background of other communication protocols, with which it interacts and from which it is discerned.

When talking about TCP, as well as the other network protocols, the students frequently referred to the technical characterisation, or technical properties, of the protocol, telling the interviewer “how the protocol works”. No variation in the understanding of this technical characterisation for TCP has been found in data. TCP is experienced as a protocol with acknowledgement in the three categories. Rather, the technical characterisation is thus what gives a specific protocol its character that makes it possible to recognise TCP as TCP or UDP as UDP etc.

The outcome space that is sketched above is summarized in table 3 below. The second, third and fourth columns are the most important, since they highlight “as what” TCP is understood in the respective categories (second column), the aspects that are in focus, and in the background respectively (third and fourth) in the different categories. A hierarchy structure is identified in the different foci, where the focus of category 1 is a subset of

55. In the term *methodological framework*, the word *framework* has a different meaning.

Table 3. Understanding TCP, empirical results

	Label	As what is TCP experienced?	Focus in on	Framework	Important aspect of TCP
1.	**Safe communication between two computers**	Safe communication	Packets, and two computers	Other computers in a network	TCP as a software entity
2.	**A connection over an internet**	A connection	End-to-end-communication	An internet	
3.	**A standard for communication**	The results of decision	Definitions and decision	A world reaching outside the particular network	TCP as a set of rules

that of 2, as the focus in category 2 is broader. Category 3 is still broader than category 2, since the focus spans over both humans and networks.

While category 1 and 2 are similar in the sense that they both describe communicating computers, but are seen from different perspectives (or with different granularity), category 3 is characterised by the idea of the human influence on what the protocol "is". The term TCP, as used by professionals, denotes both the set of rules that governs the communication and the program packages used to implement them. While the former meaning is related to practical work with the protocol, a programmer's or a software developer's work, the latter is crucial for theoretical considerations, as research into networking and future development and design of TCP or other protocols. Categories 1 and 2 are then related to the concrete software and the work with it, while category 3 views TCP as a set of rules and thus as a result of human decisions. Thus, since a fundamentally new factor becomes a part of category 3, the qualitative difference between category 2 and 3 is more important than that between 1 and 2. The last column in table 3 highlights this difference.

1.2. UDP

UDP, *User Datagram Protocol*, is a connectionless protocol, as has previously been described in chapter 3. Also in the case of UDP, three qualitatively different ways of experiencing the protocol have been discerned.

The categories that are discerned for UDP, share their structure and important properties with the results for TCP, and are discussed in great detail in Berglund (2002). In fact, a large number of students spontaneously compared the two protocols and pointed out difference(s) between them. The key difference, that has been discerned, is that UDP is recognized by the students as an unsafe or connectionless protocol without an acknowledgement. The issue of being safe, or unsafe, is visible as a technical characterisation in all categories for both protocols and it is that which makes the students distinguish one protocol from another.

1.3. RMI

RMI, *Remote Method Invocation*, is closely related to the object-orientation in Java, since its purpose is to allow Java objects, that can be distributed over a network, to communicate. To a programmer, RMI provides access to routines (called methods with the terminology used when discussing Java) on remote machines as if they were available on the local computer.

Empirical evidence

Different ways of experiencing RMI have been discerned in the team of students. In many important ways the identified understanding resembles the structure that was described for TCP and UDP. However, the picture of the students' experience of RMI is somewhat more complex than the pictures given of the previously discussed protocols. A possible reason for the increased complexity lies in the different purpose, design and function of RMI. RMI gives a programmer the ability to create a program which, in its turn, can start other programs on other computers or machines. This means that RMI, from the programmer's perspective, offers more possibilities than the other two protocols, but at the same time becomes more complicated to handle. Table 4 summarises the categories of RMI that have been discerned and also shows the sub-categories of the first.

Table 4. How do the students understand RMI?

	Label	Description	
1a	**Interaction between two computers**	Data transfer	... on two computers
1b		Something more than data transfer	
1c		Using resources on specific computers	
2	**Sharing resources over an internet**	Sharing resources over an internet	
3	**Standard communication tool**	Standard tool in a framework that includes and goes beyond a computer network	

Category 1. RMI is for interaction between two computers

In this category, we meet an understanding where two communicating computers form a framework of which RMI is an integral part. The different roles or functions of the two computers in the interaction form the basis for the distinction between the three sub-categories

1a File transfer between two computers with undefined roles

In this sub-category RMI is a method for file transfer. Sven explains for us:

> Interviewer: [...] what is RMI?
> [...]
> $Sven_2$: That is, it is a ... one moves files between, yes....for instance if I were to use RMI that was sort of ... I have the game server and a file that had marbleinfo and so the information on [...] speed so then I want to move over to mine...and then I should use RMI, hard to explain, but ...
> Interviewer: But there are lots of ways to move information what is the thing that is typical for RMI?
> $Sven_2$: Now I am stuck....

Sven refers in a concrete way to the project he is working on and gives an example referring to a specific file that had to be transferred. The term "game server" refers to the program module that controls the whole software system in the project, while "marble info" refers to some specific information about the ball, possibly its speed and position. Sven states that he should use RMI to move a file, containing "marble info" to "mine", most probably referring to the module that he was working on. There are two specific computers present in his argument, those that the file is moved between. Other computers, or a network, are not mentioned.

1b Something more than file transfer between two computers with different roles

Here, we meet a sub-category that differs from the previous one, in that the two computers now have different, but yet not clearly defined roles. Samuel's statements can serve as a reference for this category:

> Interviewer: Can you explain to me what Java RMI is?
> $Samuel_1$: Yes, exactly, but also wants to have some some ... one wants ... what does one say ... one will order or order ... one wants to make a request so to speak, they, that is pretty good, that is sort of, I don't really know ... I now now sit here and speculate here ... I ... I... I don't know so much Java either and I am totally new to this Javathingumy all the time really, but I never worked with Java so that, um, I believe that it is like like some type also, um, ah, protocol to communicate with with servers and such.

He talks about "requests"[56] and communication with a "server".

56. A *request* or a demand for information, or actions to be performed, that can be sent from one machine to another. The term is then used with basically its normal English meaning within the field of computer networks.

Since he mentions a server, it can be deduced that he considers the other of the two communicating computers as a client, and in this way assigns them different roles[57]. He does not mention file or data transfer, nor does he have a well-articulated advanced understanding of the protocol. His expressions “order” and “to make a request” as well as his discussion of a server (and implicitly about a client) are relevant for the normal use of RMI as a tool for computer communication. At the same time, the explanation he offers of what RMI “is”: “... type of [...] protocol to communicate with servers and such” is unspecific and does not indicate a clear view of how RMI is intended to be used.

During the second interview, when the interviewer returns to the subject of RMI, he expresses a similar position:

Interviewer:	[...] What is RMI?
$Samuel_2$:	Remote Method Invocation
Interviewer:	Yaa
$Samuel_2$:	It’s something one uses if one wants to find some sort of address which doesn’t exist in its own own frame for it for this code which one makes. It it ... it is a concept that understand, but here its used in Java., eh...and Java I don’t know anything about actually.
Interviewer:	You have not used that?
$Samuel_2$:	No [...]

RMI is used to find an address, which is not within the code currently being executed. The two computers have different, but undefined roles. The interaction between the computers goes beyond a pure file transfer.

1c Using resources on two computers with well-defined roles

In this sub-category RMI is presented as a tool for executing programs on another machine.

Staffan offers a description of his view of RMI. In an interview excerpt, already discussed in relation to the concept of *individual context* on page 61, Staffan says that RMI is used on two computers, a client and a server. The client can execute a program on the server. In the beginning of his explanation, he talks about a stub and a skeleton[58]. This, together with the fact that he discusses the role of the client, indicates that he understands RMI as a tool for interaction, integrated with a context of two computers. He expresses a similar understanding during the second interview:

---

57. As mentioned earlier, the active part in the communication between two computers is often called a *client* while the *server* is the passive part

58. A *stub* that resides on the client offers the same interface as the server object to a program on the client. It takes the call and passes it to its corresponding server object. The *skeleton* that resides on the server side takes the call of the stub, and forwards it to the server object, waits for an answer, and sends this answer back to the stub. Stubs and skeleton together form a layer in the architecture of RMI.

Interviewer:	[...] What is RMI?
$Staffan_2$:	It's a sort of Java client server model..you can execute. It has something to do with stubs and skeletons. You execute ... I'm not sure how it works ... we had nothing of that kind in our project as it is right now, but ...
Interviewer:	What is stub? What is skeleton?
$Staffan_2$:	Ah, it it generates something ... you have the production on the server anyway and I think that is the skeleton and then this is generated in a way that I don't understand how it works but anyway you can execute those methods, those functions, which ... which are on this or that computer over there even though it seems like they are on your own computer. It something like that.

After stating that he is not sure how it works, he discusses how it can be used for executing methods or functions on another computer as if it were on "your own".

In the following excerpt of the discussion the interviewer starts by referring to an earlier statement made by Stig, where he says that the team was reading about RMI to learn more:

Interviewer:	[...] We start by RMI, which you had read about. What is that?
$Stig_1$:	It is ... it is Remote Method Indication means that that is with Java in order to...sort of as server client they will be able to communicate with each other. One should be able to use a client, should be able to use code that is on another computer or machine by setting up a connection just ... sort of ... like ...a shell so that one should be able to. It looks like as though one can use, that one like has all the information there, but... communication fetches somewhere else.
Interviewer:	You said as client server: What does client server mean in that case?
$Stig_1$:	Oh, it is that hard to explain ...such things.... you know what it is, but um...it's like a a client mostly program wants to get information from a server.... then they have to communicate with each other in some way and then one can decide...like sort of connect to each other in some way ....it is a special port or something...and that is, yes, that is used ...so we have one of these to the server in ours...in this project.
Interviewer:	And RMI is kind of special case, or?
$Stig_1$:	Yes, I think so. Used when it has to do with Java ... a Java client

Stig explains that the purpose of RMI is to offer one machine the possibility to use code on another machine. He also indicates that this has applications, namely when a program needs to access information from a server. He sees RMI as a way of using resources that are available on a computer other than the one you are currently executing your program on. In his explanation he talks about "communicate with each other" and "setting up a connection" on a "special port", clearly focusing on the two communicating machines. He refers to two computers or machines, not to any network as a whole.

Stig incorrectly states that RMI is an abbreviation for Remote Method Indication, instead of Remote Methods Invocation. Care should be taken not to draw any far-leading conclusions from this mistake. Firstly, as was pointed out on page 17, abbreviations are frequently used in daily conversations within the computer science community. Secondly, the term *invocation* is not necessarily known to the Swedish students (see page 85).

A category (with its sub-categories) where RMI is experienced as a tool for communication between two computers has been described. The actual communication that takes place between the two computers is discussed in three different ways, as transfer, as something more than transfer, and as using resources on another computer.

Category 2. RMI is for sharing resources on an internet

In this category, the perspective is broadening, as is for example explained by Abraham:

Interviewer:	[...] Um, what is RMI? Java/RMI?
Abraham$_1$:	Ah, Remote Method Invocation.
Interviewer:	Ya, OK
Abraham$_1$:	Very nice. It allows two Java virtual machines to talk to each other. They, an object on one machine could instantiate an object that lives on another machine and use that one's methods. That's how RMI is useful.

Abraham talks about RMI as a tool that offers the possibility for two Java virtual machines to communicate, and for objects to "live" on another machine. Although he talks about two machines, he does not give any reference to specific machines. Instead, he focuses on the object, and describes the machines as places where the object "lives". In this rather abstract perspective the focus is clearly not on the physical computers, and not even on particular virtual machines. Instead, they constitute the space where the objects live. Thus, the framework is an internet, a broader framework than the one presented in category 1.

Axel, during his first interview, expresses a similar understanding, but is more explicit on the usage of RMI:

Interviewer:	We have talked about RMI? OK what is RMI?
Axel$_1$:	RMI is Remote Method Invocation which is basically, you have a Java object on one machine somewhere, it doesn't matter where, and then you have a Java object on another machine somewhere, it doesn't matter where. And then you can, either one can call the other, or they can each call each other um. It's, basically, you have to register the object in the RMI registry and then essentially it works just like the other object on the same machine. It is a little bit slower than maybe a socket would be, but it's fairly stable if you can get the security issue right.

Axel says the objects, that call each other, may be on any machines. It is not important to him where they are. Having one object call another implies that they use each other's methods, which means that they use or share resources.

Sebastian opens the discussion about RMI by a general description

Interviewer: [...] What is RMI?
Sebastian$_1$: Yes, that combination of letters stands for Remote Method Invocation, which means that one can call a command from one virtual Java machine on another.
Interviewer: Can you please say a bit more on that?
Sebastian$_1$: Yes ... No ... but that is roughly the picture I have of it actually, I don't know exactly what happens then what ... where the command went.... where it gets executed somewhere, which processor it is that will work on it, which of the two virtual machines it is that...
Interviewer: ... that executes.
[...]
Interviewer: Hmmm, what, what could one use RMI for both in the project and in general. What is the point of the concept?
Sebastian$_1$: No, no, but it feels like I can win something by that ... that ... if I for example if I have a server and a client, so if I should execute a command that is on the server if I can execute it here without having the thing itself so that it executes here so the server gains from that.
[...]

Sebastian starts by giving a "school book" explanation of RMI. However, immediately after giving this explanation, he talks about an aspect of the concept that he does not grasp: He does not understand on which machine the code is executed. The question he raises is, seen in a technical perspective, relevant and can be taken as an indication that the first explanation, although in a "school book style" had a meaning for him and was not only memorised from a book. He later gives an argument for using RMI ("it feels like I can win [...] if I can execute it here"), that is consistent with his explanation.

Category 3. RMI as a standard communication tool

In the sub-section that discusses the empirical data for TCP, evidence was presented that TCP could be understood in a framework that went beyond a computer network, and that also considered human decisions. TCP was discussed as a standard communication tool, that was viewed "from the outside". A similar way of experiencing RMI can also be identified.

In the excerpt below, Adam discusses the choice of TCP, instead of RMI, for all communication throughout the code of the project:

Adam$_1$: Between, like the game server and the video and motor, you mean?
[...]
Interviewer: And you will just accept that they are TCP. So what you do is that you go for overall a TCP solution. OK Ya.
Adam$_1$: Right. And it's my impression that it doesn't matter what one part communicates in, because if it is communicating with RMI to the client, but with TCP to the motor, I mean it's just different ways of formatting the information, in a sense, so..
Interviewer: Ya, ya.
Adam$_1$: If it isn't TCP, you know, it doesn't really affect..

RMI and TCP can be seen as different ways of formatting the information, where different protocols can be used to solve different sub-problems. He mentions communication with the motor and the technical communication between the server and the client. The selection between the protocols, seen in this perspective, is not important, his argument continues. To make this comparison, he must be capable of taking a stand outside the protocols.

Alec also adapts a similar perspective when expressing his understanding of RMI:

$Alec_1$:	[...] But it's very lengthy and verbose, as far as a lot of work, and this RMI is quick and concise, but it seems to take away some of the flexibility.
Interviewer:	Uhum.
$Alec_1$:	There are probably ways to do things that I'm not talking to.., that I'm unable to do now, that I'm not aware of but, um, as of now it seems to take away some of the flexibility. I've also discovered that there's, like you were discussing, the security, which has to do with a..
Interviewer:	Uhum
$Alec_1$:	[...] a policy file that, um, that I have little or no knowledge of, just discovering it, but that I've begun some research on it and, um, as far as how that works. [...]

Alec says that there might be solutions that he is not aware of, at the moment of the interview. His position, that the solutions he has found are inflexible, and that there ought to be other solutions, demands that he takes a position outside RMI. Also, this argument requires that he is consciously aware of the fact there are decisions taken on the design of RMI.

In this category, RMI is understood as a standard tool and is experienced as a part of a framework that goes beyond a computer network and that is described a meta-level.

Different ways of experiencing RMI

With the many similarities between the findings for TCP and UDP and those of RMI, the presentation of the analysis for RMI will be briefer.

Although the three protocols are recognised by the students as a part of, and integrated with, the same environment, differences in their internal structure are perceived. While TCP and UDP are described in similar ways, the descriptions of RMI differ from this in important ways. In computer science this makes sense: While UDP and TCP are mainly used for data transfer, RMI is used for transferring and executing both code and data, and is thus considerably more complex.

In categories 1a and 1b, where RMI is described as a tool for data transfer or as a communication tool that goes beyond data transfer, the internal technical structure is not clearly articulated by the students. In the second category, the technical structure of RMI is described as an interaction between objects on virtual machines, which is clearly a more advanced view of the protocol. For the third category there is not enough empirical material to draw a full picture. Although it is possible to identify a category for RMI as a standard, and thereby to create the category of description, there is not enough data to completely inspect the technical characteristic. The aspects of the understandings of RMI are summarised in table 5. The hierarchical structure that is formed of categories 1, 2 and 3 is similar to that, which has been identified for TCP.

Table 5. Aspects of the categories of RMI

	Label	**As what is RMI experienced?**	**Focus in on**	**Background Framework**
1a		RMI is for data transfer		Two computers with undefined roles
1b	Interaction between two computer	RMI is for more than transfer	Two specific computers	Two computers with different roles
1c		RMI is for using resources		Two computers with well-defined roles
2	Sharing resources on an internet	RMI is for sharing resources on an internet	The interacting objects and their machines	An internet
3	A standard tool	RMI is a standard tool	Definitions and decisions	A world inside and outside

2. The students' experience of the concept of a network protocol

In the previous section an analysis of how students understand TCP, UDP and RMI as individual network protocols has been conducted. A question that naturally arises is what could be said about students' ways of experiencing the general concept of a "network protocol"?

An analysis of the students' understanding of the concept of network protocols as a whole can be made in several ways. An obvious alternative would be to ask a question such as "What is a network protocol?" during the interview. However, this question was never raised.

Another possibility would be to re-analyse the interview extracts concerning the individual network protocols, now in the light of the analysis made for the individual protocols and the categories of description, that were created. Another possible attempt is to go directly to the interviews to look for statements concerning network protocols in general that have been uttered during the interviews about the specific protocols.

In this section, I combine the two latter approaches. The previous analysis forms a background to those interview extracts, that in different ways address the general concept of a network protocol.

An important aspect of the categories of description that were created for TCP, UDP and RMI are the frameworks of which the protocols are experienced as parts. These frameworks are:

1. Two communicating computers
2. An internet
3. A world beyond computer networks

The frameworks can be seen as particular aspects of the ways in which the protocols are experienced. Since similar frameworks emerge from the analysis of all three protocols, it can be assumed that they are relevant for the experience of the general idea of a network protocol. I use this as a starting point, and I will explore this question further by considering these categories alongside some interview extracts. The findings for the general concept of a network protocol are presented in table 6.

Table 6. Categories of the general concept of a network protocol

	Label	What is the general concept of a network protocol experienced as?	Which framework is the concept experienced as integrated with?
1.	**A way of communication between two machines**	A protocol is a way of talking/communicating between two machines	One (or more) specific computers
2.	**Method of communication over an internet**	A protocol is a method of communication on an internet	An internet
3.	**A set of rules**	A protocol is a set of rules that are used on an internet	An internet
4.	**A standard**	A protocol is a standard	A world that goes beyond computer networks

Empirical findings

Four categories have been identified. The critical difference between the four lies in the qualitative ways in which the general concept of a network protocol is experienced.

Category 1. Network protocol as a way of communicating between two computers

In this category of description an understanding is voiced where network protocols are experienced as methods of communication between two computers. Anthony articulates such an understanding:

Interviewer: You talked about Java RMI. What is RMI?
Anthony_1: I don't even know. I know it's a type of protocol used between, um, talking between two machines.

The discussion continues, and UDP and TCP are discussed:

Interviewer: Uhum. What is TCP?
Anthony_1: TCP is another type of protocol .. used between two machines. There is TCP and there's UDP that's one of the things that I actually do remember from ah, networking class. And I believe TCP sends packets to one machine and then there is some sort of response saying that they got the packets or not.

While talking about TCP as another type of protocol, also used for communication between two machines, he spontaneously mentions UDP. By referring to the three protocols in this way, Anthony makes clear that there are properties that are shared between protocols: The three protocols mentioned are for communication or talking, and are experienced in a framework of two computers or machines.

Category 2. Network protocol as a method of communication over an internet

When prompted to reflect on the differences between the protocols TCP and UDP, Albert stresses the similarities between them:

Interviewer: O.K What is difference?
Albert_1: I don't know [laughter].
Interviewer: That's fine, that's fine.
Albert_1: I know, I know that it's part of it and it's separate. But it's just a different type of protocol that you use to communicate. I know that, but...

Later during the interview, Albert mentions RMI when answering a question about sockets:

Interviewer: There is another word you mentioned there, and that's socket. What is a socket?
Albert_1: A socket is pretty much like a, a port that is opened up on the server, or that is requested by the client and, it's assigned a number. And it's just sitting there and listening and um, it's just an open port and that port is um, designed to use a specific type of protocol, you know whether it be TCP, um, or the RMI. And it's opened up to listen on that and once it receives that connection you know, it connects on that port. So it's like an outlet socket, you know, you connect it in, you communicate and then when it's down it gets turned off and then that port is either closed or it stays open if it's required by the server.

It is clear that Albert sees UDP, TCP and RMI as protocols that share important properties, that are similar in many way, and that are parts of the same framework. He also describes the individual protocols in such a way that it can be assumed that he experiences a relationship between the protocols. There are differences between the three, but they are all closely related. Protocols are methods of communication. Axel discuss TCP and RMI as methods of communication in a framework of an internet as well:

Interviewer:	OK That's fine, that's fine. Um, I want you to talk about TCP.
$Axel_1$:	TCP/IP?
Interviewer:	Ya.
$Axel_1$:	TCP/IP is how almost everything on the Internet communicates. IP addresses and everything, and that's um, one of the fundamentals behind RMI also. One could give it the address where the object is [...] the IP address [...]

Axel also presents UDP as a protocol similar to TCP. The three protocols are integrated parts of the Internet. He talks about the protocols as "how almost everything [...] communicates".

Category 3. Network protocol as a set of rules

Allan discusses TCP as a protocol language used for sending data across a network, in an excerpt that was previously discussed on page 92. His statements that TCP is "a protocol language", that you "put your data in", and that different protocols might be used for the actual transfer, indicate that protocols are sets of rules. A protocol language is, in a computer science context, formal language or a set of rules. He does not regard his answer as only valid for TCP, since he talks about different protocols, without wanting to, or without being capable of, mentioning others by name. By mentioning the term IP, Internet Protocol, Allan relates to an internet.

Adrian tells the interviewer during the first interview that his team plans to remove RMI:

Interviewer:	Um, what is RMI? What is Java/RMI? The thing that you're removing?
Adrian1:	I don't know, and that's why we're removing it.
Interviewer:	OK
$Adrian_1$:	Cause we don't know enough about it. I, it's.. I've read briefly whole paragraphs about it. It's basically enabling it to get around security features that TCP/IP wouldn't allow. Um, or standard HTTP protocols. Um, like RMI, I guess allows complete access to certain files. Whereas if you go to HTTP, it's going to be a little bit slower, and you, there you have to worry more about the security issues, what you want to have access to.

The team plans to remove RMI to get around certain security features, and profit from, as they understand, the less strict rules of TCP. To get around security features is to avoid certain rules, since the security features mainly consist of rules that govern, or hinder, certain operations in order to guarantee security. This argument paired with his discussion about “allowing access” points towards the view of protocols as sets of rules; rules that are somewhat different for different protocols.

This category of description discusses an understanding of network protocols as a set of formal rules experienced in a framework of an internet.

Category 4. Network protocol as a standard

The fourth category describes the most abstract understanding of the concept of a network protocol. Here the idea of a standard is abstracted as well as the background against which it is seen, namely formal decisions. Adam explains during the first interview what a network protocol is, in a statement that has earlier been discussed in page 92. He starts by saying that the purpose of a protocol is to get computers to communicate. He then points out that TCP is a standard for a protocol, which was created by a committee. A protocol, in its turn, specifies the format for data sent across the network. TCP is, seen from this perspective, one of many protocols. A standard is, according to Adam, a set of rules that are created by a committee and are a result of human decisions.

The concept of a network protocol

The new analysis has revealed four categories of the general concept of a network protocol, instead of three, that emerged in the analysis of the individual protocols. This difference at first surprised me, but can be explained by phenomenographic theory.

A phenomenographic research project is an exploratory project or a learning situation for the researcher (see page 37). The second time I follow a similar line of thought, when investigating students’ ways of experiencing protocols, it can be expected that I will get new insights. Booth (1992) provides the metaphor of two travellers, exploring the same new land. If the first draws a map, which the second can use, the second traveller is, of course, in a different situation than the first. A similar reasoning holds for the same traveller, or researcher, returning a second time to the same place, or data.

In my analysis of the general concept, I had access to a richer set of data, than when studying the protocols individually. Not only could I utilise all those excerpts of the interviews, when computer networks were discussed without particular protocols being mentioned, I had also, during the latter of the analysis, access to all statements concerning the three protocols. In this way, I had access to more fine-grained data.

Another argument is based on the theoretical distinction between different contexts that is introduced in chapter 7. In the situation when I analyse the general concept, each statement could be read against a different context, than during the first three analyses. There were a larger number of statements, but more importantly, each single statement could be read against a more complex background, containing all statements concerning computer networking in the data. In other words, the experienced context of the collective, with which a particular interview excerpt was to interact, was different. In a similar way, the researcher's context had changed. In this situation, I could consider the general concept of a protocol and use a broader knowledge about the students' projects and learning in computer science. Consequently, the new analysis was performed in a qualitatively different environment, and builds both on data and prior analysis.

3. Discussing a good understanding of network protocols

Insights concerning the value, or use, of different ways of understanding the protocols pave the way for a discussion concerning a desirable learning outcome. In the following, focus will be kept on the network protocol TCP, but without putting the other protocols completely aside. Comparisons will be made, and interview excerpts concerning the other protocols will sometimes be presented. The focus on TCP can be explained in its dominant role in programming network applications. Furthermore, the data collected are richer for TCP than for any other protocol, possibly because of the role the protocol plays on internets. The rich data invites investigation and makes the analysis easier to perform.

The results of the further analysis is based in a further analysis of the empirical study. For such an analysis, I study the categories from *a researcher's perspective*. The categories then come to stand in the foreground, with the researcher's context (see page 58) serving as a background.

The value of such an analysis lies in that the categories are related to the research questions and their context: Applications in industry and academia, understanding of the students' situation, a possibility to "see" the categories from the outside, are just a few of the benefits that can be added to the previous analysis.

3.1. What is important to know about TCP and other protocols?

The examples below illustrate that different ways of experiencing a network protocol are relevant with different tasks at hand. That is, a student needs to be capable of experiencing a phenomenon in context-dependent ways, in order to have the capacity to solve the types of problems that arise during different parts of a project.

Relevance for a programmer

Describing TCP as communication between two computers (category 1), closely resembles discussions about programming. The descriptions of TCP made by the students can be directly linked to programming situations with two communicating computers in focus. It can thus be assumed that this perspective is fruitful when solving concrete programming issues. A quote from Sebastian serves as an illustration. When questioned by the interviewer about UDP, he compares UDP and TCP:

Interviewer: UDP?
Sebastian$_1$: UDP ... but that is another form of communication. TCP/IP is set up ... like TCP, in contrast to UDP, TCP sets up communication between two points, and they talk to each other and make sure that they don't drop anything sort of.

As has previously been noted (on page 95), TCP and UDP offer procedures, or operations, to a programmer who writes application programs. The procedures for TCP offer services like setting up a connection or sending data. The statements by Sebastian above can directly be related to programming issues when using TCP in an application program. Similarities between his statements and some basic operations on sockets are shown in table 7.

Table 7. Similarities between statements concerning TCP and basic TCP operations.

Words used by a student	**Basic TCP operation**
Set up	Connect to a remote machine
Talk	Send/Receive data
Implicit, a connection that is set up, also has to be closed	Close a connection

In the continuation, Sebastian returns to UDP. His way of talking is still close to the issues of programming:

Sebastian$_1$: UDP is [...] that the client asks what does this mean. Or what is this, or any question, whatever, and, so the server answers. And the server doesn't care in the end if the answer gets there or not. It is only a question and an answer, and then it is up to the client. If it feels that I didn't get any answer, it gets to ask again.

Here he talks about what a client that uses UDP has to do: If no data has arrived, the client repeats the question.

This line of reasoning is close to the steps that are taken by a program that uses the protocols, and is for this reason relevant when programming.

Relevance for a program designer

Experiencing a network protocol as a connection over a network (category 2), is useful when discussing the properties of a certain protocol or which protocol to apply in a given situation. Issues like in what situations and in what way to use the protocol come into focus here. It can thus be assumed that this category is fruitful for designing project solutions and selecting between protocols.

The excerpt of the first interview with Abraham, presented on page 100, can serve as an example. Abraham explicitly discusses the advantages of RMI when asked what it is. He clearly has an understanding of the purpose of RMI (execution of code on a remote machine). This understanding is useful for deciding when to use RMI in a particular system, and when to choose another protocol during the design phase of a project.

Relevance for theoretical development

Category 3 describes a way to understand network protocols, that to a large degree resembles that of category 2. The key difference lies in whether the protocol is seen as an application of a set of rules (category 2), or the rules in themselves (category 3). Certainly this difference has implications for practical work. A focus on a theoretical development is certainly fruitful when developing purposely made protocols.

Relevance to policy issues

The discussion about TCP as a standard concerns how a protocol is developed, what possible protocols there could be, and what properties a protocol could have (category 4). This understanding is useful for policy discussions and for theoretical development, where the rules governing a protocol have to be thoroughly analysed and discussed.

This position is clear in the excerpt presented on page 102, where Alec argues that he is not aware of all features of RMI. RMI is, as he understand it, quick and concise, but it is not as flexible as he thinks it ought to be. His conclusion is that he does not know the features of the protocol well enough. This line of argument is relevant when considering policy questions, such as how to design network protocols.

Adam’s answer to a question about what TCP is (earlier quoted on page 92) illustrates the role of the human decisions. He argues that TCP is a standard that is created by a committee. The rules, e. g. the form of the packets, are the results of conscious committee decisions.

The kind of reasoning, presented here is thus relevant when developing protocols and considering policy questions.

Summarising the usefulness of understanding TCP in different ways

The results concerning the relevance of understanding TCP in different ways are summarised in table 8.

Table 8. The relevance of different ways of understanding TCP

	Label	What is the general concept of a network protocol experienced as?	What is in focus?	What is it relevant for?
1.	**Way of communication between two machines**	A protocol is a way of talking/ communicating between two machines	Packets, and two computers	Programming
2.	**Method of communication over an internet**	A protocol is a method of communication on an internet	End-to-end-communication	Project design
	A set of rules	A protocol is a set of rules that are used on an internet	The rules governing the communication	Theoretical development
3.	**A standard**	A protocol is a standard	Definitions and decision	Development of new communication solutions

3.2. Levels of abstraction in understanding TCP

The different ways of understanding a protocol are closely associated with different levels of abstraction in the discussions concerning the protocol during the interview.

In interview excerpts that illustrate the first category ("Safe communication"), TCP is described in a concrete way. A nice example can be found in the dialogue with Andy on page 90. He discusses two communicating computers, whether the packets arrive or not, as well as the need for acknowledgements to be sent, all in very concrete terms.

To the second category ("A connection"), this aspect of the language has changed to a more abstract terminology. The quote of Albert, earlier presented on page 91, can serve as an example. Here the individual machines are no longer present. Instead, a network forms the background for the communication. The machines, that constitute the network are not seen, since the level of abstraction is different.

In the third category, the way in which the language used, when discussing TCP is different. Here TCP is discussed "from the outside". Adam, quoted on page 92, takes the "outsiders" perspective, when discussing TCP. His argument goes beyond the properties of the individual protocol. Instead, he takes the protocol as a whole as an object of discussion.

4. Learning computer networking

During the analysis, that has been presented in this chapter, the backgrounds against which the students experience a protocol have been stripped away; in other words, the statements made by individuals were decontextualised. Now, when the aim is to advance the discussion to explore the learning that takes place and to draw implications for teaching, it is useful to place the individual statements back into the context from where they came.

4.1. Shifts between different ways of experiencing network protocols

As has been pointed out throughout this thesis, the categories are constructs of the researcher, valid at a collective level. As such, they are stable. Individuals, on the other hand, experience a particular phenomenon differently at different moments; that is, they can shift between categories. With a distinction articulated by Pong (1999), shifts in focus can occur when the context in a dialogue shifts, that is when a new subject is discussed (labelled *inter-contextual shifts* by Pong), but also as intra-contextual shifts within the same context, either spontaneously by the student or as a part of a conversation.

Many intra-contextual shifts have been identified in the data. This can be due to the fact that the students are advanced computer science majors in their third or fourth year. As such they have had the opportunity to meet different views on the subject from their teachers, books etc.

These shifts have been identified in the interview transcripts. Still it can be assumed that shifts are provoked during the discussions as well as during the interviews.

4.2. Case studies on shifts

An example of such an intra-contextual conceptual shift from experiencing TCP as communication between two computers to experiencing TCP as related to an internet can be found in the first interview with Anthony. In the first part of the discussion of TCP, earlier quoted on page 105, Anthony tells the interviewer that he understands that TCP is used between two machines, for the purpose of sending packages. TCP has, according to him, a kind of response that indicates whether a package has arrived or not, that is, TCP has an acknowledgement. The dialogue continues:

Interviewer: So what's the implications of this?
Anthony_1: Um, it, it all depends on how you're coding it. It depends on how secure the network you're on. And if you actually trust just sending it out and just assuming that it gets there.

When the discussion continues Anthony gets a question about the implications. He argues that the implications depend on how "you are coding it", that is what your program actually does, and your understanding of the quality of the network. His focus changes from experiencing packages sent between two machines to experiencing TCP as a part of a network that he discusses in abstract terms and to which he assigns properties like trust. In this case, the shift was triggered by the interviewer asking a question that encouraged the student to reflect further on the subject.

Another example of an intra-contextual shift that a student spontaneously made during the interview can be found in the continuation of the extract of Sebastian from section TCP as a safe communication between two computers:

> Sebastian_2: Yes, an acknowledgement, That is, that I know that the information I send has arrived correctly, and what comes back has also arrived. There is a bunch of other stuff that I have to look out for. That the communication really works as it should, yes, between two software-created gadgets, that are sockets and ports.

By the end he mentions sockets and ports as "software-created gadgets". Here he shifts his focus and talks about abstract items, and thus expresses another way of experiencing the protocol.

Similarly several shifts between different ways of experiencing RMI have been identified. An example of an inter-contextual shift can be found in two excerpts of the first interview with Albert. In the first excerpt, the discussion is about the changes the team has decided to make to their project (see page 9). In an interview excerpt, earlier discussed on page 63, the interviewer introduces the question of the changes, but the concrete change that gives the direction to the continuation of this part of the interview, comes from the student. In the context of the client-server separation, one of the changes that the team has decided to make, Albert discusses in detail the interaction between the server and the client. He mentions data that is sent, and discusses which methods are called and on which objects they can be found. He clearly expresses an understanding where RMI is used as a tool for using resources and is seen in relation to two specific machines: the client and the server. Later during the interview, the interviewer asks him about RMI (see the second interview extract with Albert on page 63). This time he explains the function of RMI, without referring to any specific machines or computers. He does not mention explicitly that objects can be on any machine. However, his use of the word "remote" and the general attitude in his explanations clearly indicate that he experiences RMI in the framework of an internet.

Another case of shifts can be found in the first interview with Alec:

Interviewer: You are going to Java RMI, what is RMI?

Alec$_1$: It's um, a remote method communication. Um, Java sets up interfaces between two, let's see, um, classes, objects, and in the interface are methods that are available to the other class. And nothing else within the class. [...]

Here, he expresses an understanding where RMI is related to an internet (category 2), especially by mentioning "remote method communication". He articulates his understanding in an abstract way using words like "class" and "objects" and talks about the methods in the interface. He continues:

[...] RMI starts a connection on the port. It's not really a port, it's a registry number, and between, on that registry number they can communicate but only in the interface between the two. Um, I found, right off the bat, that you can't just compile these classes regularly. There is a RMI compiler. The RMI compiler creates two classes, a stub-class and a skeleton-class and these are needed for the communication between the interfaces. These are set up, um, separately to the communication. Um, I found that particularly interesting because it takes a lot of the work out. The hard coding I know and C++ I've seen the coding, I've never actually coded it. [...]

The discussion here moves towards coding, how to make a particular connection between two specific machines or computers work, and expresses a way of experiencing RMI that is described by category 1c. The continuation is interesting:

[...] But it's very lengthy and verbose, as far as a lot of work, and this RMI is quick and concise, but it seems to take away some of the flexibility.

Interviewer: Uhum.

Alec$_1$: There are probably ways to do things that I'm not talking to.., that I'm unable to do now, that I'm not aware of but, um, as of now it seems to take away some of the flexibility. I've also discovered that there's, like you were discussing, the security, which has to do with a..

Interviewer: Uhum

Alec$_1$: .. a policy file that, um, that I have little or no knowledge of, just discovering it, but that I've begun some research on it and, um, as far as how that works. [...]

This excerpt shows a dialogue that goes beyond the properties of an individual protocol. Alec says that there might be solutions he does not know about at the moment of the interview, since the solutions he has found are inflexible. To make this judgement, that the solutions are inflexible, and that there, as a consequence, ought to be other solutions, demands that he takes a position "outside" RMI, where he can talk about what properties he expects the protocol to have. This is an indication of a shift to experiencing RMI as related to a world that goes beyond computer networks (category 3).

Alec has made spontaneous inter-contextual shifts from 2 to 1c and further to 3.

4.3. Implications of shifts in ways of experiencing a protocol

In the discussion about shifts between different ways of experiencing network protocols three qualitatively different types of shifts have been identified: spontaneous intra-contextual shifts, triggered intra-contextual shifts, and inter-contextual shifts. These results harmonise well with the results about inter-contextual and intra-contextual conceptual shifts articulated by Pong (1999). The shifts discussed are summarised in table 9.

Table 9. Cases of shifts between different ways of experiencing network protocols

Name of student	Type of shift(s)	Categories of descriptions for the shifts
Anthony	triggered intra-contextual	$1 \rightarrow 2$
Sebastian	spontaneous intra-contextual	$1 \rightarrow 2$
Albert	inter-contextual	1, 2
Alec	spontaneous intra-contextual	$2 \rightarrow 1 \rightarrow 3$

For the intra-contextual shifts, the table shows the order in which the students expressed a certain way of experiencing the protocol, since the shift happened during a single episode of the interview. In the case of inter-contextual shifts, the order is not relevant, since the different ways of experiencing the protocol were expressed during different parts of the conversation.

Although there are many cases of shifts within the data, this does not imply that all students shift between all understandings. For each individual, it is possible to identify the most advanced understanding shown during the interviews. With some rare exceptions all shifts found in the data are between categories 1 and 2. There are some students who, despite provocation by the interviewer, only express one way of experiencing the protocols.

4.4. What is desirable learning of network protocols?

Marton and Booth (1997) discuss good learning and argue that the ways in which learning is experienced "differ in richness (different aspects of learning that are discerned and held in focus simultaneously) and situational appropriateness (which particular aspects are held in focus under the prevailing conditions)" (ibid. p. 55).

Richness and situational appropriateness in ways of experiencing network protocols

An argument for the value of a *richness* in of understanding of network protocols can be formulated in the following way: An understanding expressed in a more inclusive category offers the broader perspective needed to inspect and evaluate an understanding expressed in a less inclusive category. For example: To evaluate the solution to a programming problem concerning two interacting computers, as the code of an end-user program using TCP, it is necessary to shift to an understanding where the program is experienced in the framework of a network. By "stepping outside" the original reasoning to look at the problem as an issue of design instead of as an issue of coding, questions about the efficiency, usefulness and relevance of the solution can be discussed. Discussions based on a more inclusive category are in the field of computer networking also, as has been shown, more abstract, a feature that makes such judgements easier.

For solving complex or new problems it is thus necessary to shift between different ways of experiencing a protocol, since problem-solving involves different sub-tasks. To shift perspective, whether a shift is intra- or inter-contextual, triggered in a discussion or spontaneous, is not alone sufficient for problem-solving. Shifts have to be made in a relevant way, that is, they have to be *situationally appropriate*, in order for the student to be capable of evaluating when and why a specific way of understanding a protocol is fruitful.

Professional aspects on the learning outcome

A student, who can deploy and shift between different categories, has a mastership of the concept of a network protocol that is a meaningful subset of the professional view, here based on Feit (1998). Such an understanding encompasses the protocol both as a set of rules and as a software system. Furthermore, conscious shifts between the categories permit a student to work with the protocol in different ways. Still, when comparing the students' categories to a professional view, differences can be found. Professionals tend to refer to the format of packets, modes of addressing and other protocol features in a manner that is not evident in the students' statements. Whether these differences between how professionals and students talk about protocols are due to the context of this study remains an open question for future investigation.

The results concerning the students' understanding of network protocols indicate that the hierarchical structure between the categories does not indicate that a certain category is "more functional" than another. In this respect, the results presented here differ from those of most phenomenographic research, where the highest category frequently is judged as the generally most desirable (many examples of such studies are available in Marton & Booth, 1997). However, Booth (1992) presents a similar result when discussing three categories of programming (product orientation, problem orientation, computer orientation), identified within a cohort of novice engineering students. The three categories, she argues, are all relevant for professional programming during the phases of design, prototyping and coding, respectively.

It is thus tempting to search for an explanation for these differences in the character of computer science itself. The structure of computer science, particularly with respect to education, but also to professional life, is discussed in the IEEE/ACM Computing Curriculum 2001 in Computer Science[59]. The report states that the foundations of computer science are drawn "from a wide variety of disciplines" (p. 12) and that "all computer science students must learn to integrate theory and practice, to recognize the importance of abstraction, and to appreciate the value of good engineering design". Particularly for the area of "net-centric computing" it is argued that "mastery of this subject area involves both theory and practice". A professional in computer science must then be familiar with these different aspects of the field.

This discussion can be applied to the results concerning the network protocols in this study. For the sake of simplicity, TCP is deployed as an example in this discussion The first and second categories of understanding TCP correspond well to the role of the computer scientist in the phases of implementation and engineering design, respectively, while the third category is important for theoretical considerations and the advancement of the area.

The statement from the ACM curriculum committee thus supports the conclusion that a computer networking concept ought to be understood in several ways. In particular the report stresses the need to understand a computer system in a more advanced way than merely as a program:

> Graduates of a computer science program must develop a high-level understanding of systems as a whole. This understanding must transcend the implementation details of the various components to encompass an appreciation for the structure of computer systems and the processes involved in their construction and analysis. (p. 62)

On the other hand, the inclusive structure of the of the categories indicate that, from a learning perspective, the more advanced categories are the most desirable, since such an advanced category by necessity includes a less advanced.

59. http://computer.org/education/cc2001/final/cc2001.pdf

Chapter 11. The why

Until now, the focus of this thesis has been on how the students understand different phenomena, and how they go about learning. Only glimpses have been offered of the aims towards which the students strive, or in phenomenographic terms, what their motive for learning is. This chapter will offer a different perspective and will concentrate on the empirical results concerning what the students are striving for when they take the Runestone course.

First, it is clear from the accounts of most students that learning has taken place during the course. The learning here is multi-faceted, and is, in many situations, related to other issues than core concepts of computer science. It also includes for example project work, or insights concerning personal development. Possibly the words of Axel can serve as a summary not only for his and many more students reactions, but also of my personal reflection on the results presented in this chapter:

> Axel_2: I guess I learned a lot, but what I learned wasn't what I expected to learn.

This chapter concentrates on the different motives and their relationship to work in neighboring fields of research.

1. The students' motive for their studies in Runestone

Three different motives, which in turn can be experienced in different ways, have been identified.

A. Academic achievement
B. Project and team working capacity
C. Social competence

The first motive describes a concern with and direction towards academic results, the second has its focus on the project, and that which can be learned from it, while the third is turned towards the social or collaborative aspect of learning. Since the last of these motives is not in the core of the problems addressed in this thesis, a detailed analysis concerning social competence as a motive will be presented elsewhere. It is worth pointing out, though, that the findings concerning this motive focus on the pleasure of learning together as well as intentions to take responsibility for the team as a whole.

The three identified motives are separate entities and consequently do not constitute a phenomenographic outcome space. Thus logical relationships between them, in the way that is normal in phenomenographic research, are not offered here. On the other hand, relationships of different kinds, having their origin in their common roots in the Runestone environment, have been identified between the categories of two of the three motives.

The analysis also shows that the three can be experienced in different ways by the students. The outcome space obtained for each of the phenomena have a hierarchical structure. Furthermore, the analysis indicates that an individual student can experience one or more motive(s) in taking this course, and that he can simultaneously experience motives in various ways.

The data on which these findings are based stems from extracts of different passages of the interviews. Most of the findings are based on answers to the question "What have you learned from this?", where "from this" is associated with situation the students experience when taking the Runestone course. The initial question has been followed up by the interviewer, when he has asked for clarifications, or has summarised the student's words in order to induce a student to give a more elaborated answer. Other valuable contributions have been collected from those interview situations where a student has talked freely and has associated across different themes.

1.1. Academic achievement

This motive is directed towards, and framed by, the academic world. The categories identified all, from different perspectives, illuminate what it means to learn, or to be a student, at a university. The world outside the university is only touched upon in the corresponding interview excerpts, and then as a side comment aiming to illustrate a contrast between university related issues and other aspects of the question, for example future employment. The four categories of this motive are presented in table 10 below.

Category 1. The motive is to get a grade

The importance of grading is strongly worded by Alec, in a discussion that mainly concerns the role of the team leader, and what a team leader could do to make the team members take their own initiatives. He says:

Table 10. Categories of the motive Academic achievement

	Label	Description
1.	**The motive is to get a grade**	To comply to the formal requirements by getting a grade is what is striven for
2.	**The motive is to learn computer science for the project**	The project defines what computer science concepts need to be learned about
3.	**The motive is to learn how to learn computer science**	By doing the project, a student learns how to search for information and how to apply it
4.	**The motive is to learn something new**	The learning within computer science here encompasses new concepts and ideas.

$Alec_1$: Um, in all honesty I think um, some of my other group members here, um, they just want to do their job good enough to pass.

The purpose of the work in Runestone, has previously been discussed in relation to the grading. Both Anthony, cited on page 60 and Adam, page 182, express similar ideas.

Getting a grade, pass or higher, is the focus of this category. Only that, which contributes to a formal recognition within the academic system, is understood as being possible to do and worth doing. This perspective leads to a situation where the student is dependent on the requirements of the university system and the formal rules, as they are experienced by him.

Category 2. The motive is to learn computer science for the project

The next category discusses learning of computer science. The project offers a guideline about what to learn.

Stig points out during the first interview that the project offers possibilities to learn computer science:

Interviewer: [...] If you look at the knowledge of the subject in the whole group, or the Swedish section, whichever you like, do you together know enough computer science?

$Stig_1$: Umm, I think so. If not, we can learn that, I believe that.

Alec comments on learning computer science during the following episode of the second interview:

Interviewer: If I may go back to another question you talked quite a lot about and what you learned from this. You mentioned RMI, you mentioned Java coding. What else, some technical skills?

$Alec_2$: Client server applications, this was my first. I learned just basic set up. How to manage it and things of that nature. Learned a lot about how to comment and manage your code. Even though I was the only one working on it we did have it set up that you could track where last updates were done at so you would know where problems were. Little bit about running applications and learning some language.

	Most of our projects up until now had just been programs here. You make programs, you hit output on your screen. We never have interface with motors, cameras.
Interviewer:	Did you learn anything from that?
Alec_2:	Uh, a little bit. Most of the time when you do stuff like that it makes it more feasible. It is within grasp now, you can actually do this, ya know, you don't just think about it and go "wow I wonder how they do that". So a lot about that. The main point of it was reading somebody else's code and making sense of it. That's where most of my time went.

Alec both offers a list of what he has learnt (Java coding, RMI, client-server applications etc.), and tells that he has learnt to analyse code. He discusses his learning in terms of the project, and what he has learnt. From his last statement, it can be deduced that the learning is not a side effect, but something he has been striving for, by using words as "wow, I wonder how they did that".

In this category, the project serves both as a tool (by offering examples) and a catalyst (by giving a spark and directions) for the learning. The motive is to learn computer science, but the decisions about what to learn are dependent on the requirements of the project. As a summary, the second category presents a situation where both the setting and the student's own interest to learn constitute the key factors in the students' efforts to find a direction in his studies. The educational framework dominates the situation since it sets the limits for that which is possible to do.

Category 3. The motive is to learn how to learn computer science

Samuel spontaneously discusses learning to learn computer science:

Samuel_1:	I'd like to say something, perhaps not directly related to all this, but I would like to say that this whole concept of doing courses is an excellent learning opportunity.
Interviewer:	Yeah
Samuel_1:	[...] You get, in some way, challenged to find knowledge in unconventional ways and this is really important this experience. [...]
Samuel_1:	[...] We teach ourselves stuff by participating in the course to discover new things [and to be] creative in locating information. [...]
Samuel_1:	[I] look for information much more often now, 'cause I feel I need to do that in order to complete some tasks, some sub-tasks.

Samuel here stresses it being important to learn to find information, and continues the discussion by comparing the Runestone course to other courses he has taken:

Interviewer:	You do that also in other contexts, not just in this course?
Samuel$_1$:	Yes, but not in the same way, and not to the same extent.
	[...]
Samuel$_1$:	Now it is much more, now it is a new approach to doing things. Earlier we could perhaps pass the course, without going out, sort of, and looking for much information. Now we have to do that.

He particularly emphasises that this feature of the course, to learn how to search for information, is something unique.

The Runestone course serves as a catalyst and a guide to the learning also in this category. In contrast to category 2, where learning of computer science for the purpose of the project has been discussed, we now meet a perspective on the learning in this project, more focused on the process than on the result: Learning here means to learn to find out. A student who strives to learn how to learn computer science has himself taken control over and has the responsibility for his own learning, and is no longer dominated by the formal requirements. The project serves a function, but here as a tool, that enables the learner to learn how to learn and to discern that which is relevant to know.

Category 4. The motive is to learn something new

Doing or learning something new is the topic in the excerpts of the interview with Abraham below.

Interviewer:	[...] Would you say it is good or bad?
	[...]
Abraham$_1$:	That's right. Yeh, it's still good, I still enjoy it because I'm doing something I never did before, and meeting international students. Um, working in, just the IRC chat is very interesting, I think.

The indirect perspective from category 3 is widened in category 4. The important issue is that what is learnt or done is new, not in the particular content of the new experience. To learn something new, the learner has to take his own responsibility for his learning, and has to experience an independence in relation to the formal setting. As in the previous category, the direction in which a student strives is directed towards the learning content.

1.2. Project and team working capacity

In this motive, the attention is turned towards the project and its team. As for the academic results, four categories (summarised in table 11 below) have been identified and are presented, each describing a certain way of experiencing the motive for learning to work in projects and teams.

Table 11. The categories of Project and team working capacity

	Label	
1.	**Pass the project**	To comply to the formal requirements by passing the project is what is striven for
2.	**Gain familiarity with working in projects**	To learn how oneself should act when working in a project is what is striven for
3.	**Learn how a project functions**	To learn how a team should work and a project should function is what is striven for
4.	**Become a better profes-sional**	To learn such things in the Runestone project that could be useful in a future life is what is striven for.

Already at this stage similarities between these categories and those presented in the previous sub-section are apparent. The nature of these similarities, and the reasons for them, are discussed later in this chapter.

Category 1. The motive is to pass the project

The idea of "getting through", "doing the bare minimum" or "just getting rid of" the project is mentioned in some occasions during the interviews. None of these utterances were explicitly expressed as concerning with interviewee himself, his team mates, or other named students. Instead, discussions concerning possible problems in "the other sub-team" or "other teams" have occurred (as has be discussed in chapter 13), sometimes related to the grade of the course, sometimes to the project itself.

Adam makes one of the more explicit statements as a part of a long discussion during the second interview, and makes it clear that he refers to the project:

> $Adam_2$: [...] And whether that was intentional or not, I think it's possible to get through this without learning a new language or learning new concepts or stretching yourself as far as the straight education aspect goes.

He talks about it being possible to pass the project without "learning a new language or learning new concepts". As he points out on another occasion during the interview, this course is a capstone course for the American students, aimed at integrating what the students have learnt in other courses. From this it can be concluded that his statement refers to the learning, or possible absence of learning, in the project. In short, Adam says that a student can pass this project without learning.

Despite the absence of students who state that their motive has been only to pass the project, the argument above shows, that the *idea* that a student can aim for just passing the project is present among the students. This idea is what characterises this category.

Category 2. The motive is to gain familiarity with working in projects

In the following extract from the second interview, Samuel talks about what he has learnt by working in an internationally distributed project:

> Interviewer: Have you learned anything from working in this way?
>
> $Samuel_2$: Yes, [...] it was not very hard to adjust to working in this way. But, I mean, if you want to do it, that is something else, if one wants to work like this in this type of situation. But I understand, I understand, perhaps that things are going in this direction since everyone can't perhaps be located in the same place and work. Then this can be a good thing to know, be something of value, the experience, so to speak, that one has had a chance to get a feel for how it is to work in this type of way and in this type of environment. [...]

Samuel presents what he has learnt from this way of working, namely how to deal with work in project teams. He focuses on the personal experiences from this particular situation, and what he has learnt about working in teams.

The second category of this motive describes the experience of gaining familiarity with working in complex projects as the key aim in the Runestone course.

Category 3. The motive is to learn how a project functions

After a lengthy discussion with Adam concerning what he has learnt, the interviewer rounds up by summarising what he has understood from Adam's words:

> Interviewer: If I understood you correctly, you think the group dynamics and project management area you have learned quite a lot. Is that correct?
>
> $Adam_2$: I think I have, yea, I think my eyes have been opened to what team projects are like and real business situations and how it should be handled, how it probably isn't always handled, how people react to it, things like that. More psychology than anything else.

Adam states that he has learnt "what projects are like". In these words, and in the continuation of the quote, he states that he has learnt about projects as such.

Anthony compares the Runestone project to similar situations in business, as a spontaneous comment to the last question from the interviewer:

> Interviewer: Anything you want to add?
>
> $Anthony_1$: No, other than... the Brio boards, it seems kind of odd that, um, we'd be working on a project of something at this level. I mean trying to make this as real world as possible, and saying that this is what it is going to be like after you leave school, when you get into, into the work environment. Um, we are running into so many problems with

our equipment. And there are just so many things that we have no control over and it seems like that, or at least I would hope, would never be the case in many industries.

Interviewer: Do you think it will not be the case?

Anthony$_1$: I hope not. I am pretty sure it will be, but at least then we can have one of our milestones to be buying a new camera and Brio board.

By contrasting the Runestone project with a business situation, Anthony demonstrates his concern for how the project as a whole functions and suggests changes.

In contrast to the second category, the motive for the learning is here generalized, or abstracted, from a particular situation.

Category 4. The motive is to become a better professional

Before the interview extract presented below, Abraham and the interviewer had discussed if the Runestone project was well-specified or ill-specified:

Interviewer: [...] It is not a well-specified problem, I do agree on this description.

Abraham$_1$: Right, right. Yeh, I think, yeh, I see what you are saying. I think in the real world this will probably help. Because I can imagine going to projects or jobs where, um, the people are not really going to help me very much, or they're not going to have very protected work for me. They are going to have parts missing, they are going, you know, I'm sure I'm going to get a lot of that. Or it is going to be: 'I need you to learn this in 2 days'. I'm sure there will be a lot of that, so, yeh, I suppose that I can see, yeh, now I know how frustrating it will be. (laughter).

Abraham describes possible situations in his future work, where his tasks will not be related to his particular competence. Here, he argues, the experience gained from the Runestone project will be valuable. Abraham's perspective extends from his current situation and also refers to his future career. In the fourth category, the usefulness of working in teams is discussed in the context of a future professional life.

2. What do the students strive for?

The two motives "Academic achievement" and "Project and team working capacity" show many structural similarities, that invite to a further study of the nature of the similarities.

Academic achievement as a motive

The focus on the university world dominates this motive, which describes what, in the form of academic achievement, a student can strive for in the Runestone course. The four categories differ in their foci, the relationship between a dependency on university requirements and own responsibility for the achievements, and which factors dominate in the constitution of the category. These aspects are summarised in the different columns of table 12.

Table 12. The categories of academic achievement as a motive

	Label	What is in focus?	Dependency on requirements vs own responsibility for achievements	Dominating aspect
1.	**The motive is to get a grade**	The grade	Dependence on university requirements	The educational framework
2.	**The motive is to learn computer science for the project**	Computer science concepts	Dependence on university requirements	
3.	**The motive is to learn how to learn computer science**	Learning to learn computer science	Personal learning dominates over formal requirements	The content of the learning
4.	**The motive is to learn something new**	Learning something new	Independent learner	

An important qualitative difference divides the described motive between the first and second category on the one hand, and the third and fourth on the other. The first two describe situations where the learner is dominated by his experience of the formal requirements, while he has autonomy and controls his own achievements in relation to the learning content, in the experienced situations described in the last two categories. Thus, the two broad categories "dependence on formal requirements" versus "own responsibility for achievements" can be identified.

Also with regard to the learning of computer science, differences can be identified between the categories. The subject area is not present in the first category, but appears as a set of isolated concepts, determined by the project and its needs in the second. In the third, learning about how to learn computer science is in focus. Learning, as an effort to learn something new (category 4), does not limit itself to the subject area, but also considers learning of computer science in a larger context.

In the same way, the categories of the project and team working capacity can be analysed. This analysis is not reported here, since it shows such strong resemblance to the analysis leading to table 12.

The differences between the categories, that are content-related, are shown in table 13 below. Identifying the similarities between the first categories is straightforward: Both describe categories of a motive, where the fulfilment of the perceived formal requirements is that which is striven for. The unifying idea for the two second categories is the motive for a specific feature, perceived in the project. The third categories describe motives to learn about something that can be generalised: how to learn computer science, and how to learn how projects function in general. Finally, both the fourth categories point to new achievements outside the current situation.

Table 13. Common characteristics of the categories for Academic achievement and Project and team working capacity.

	Motives		Common properties of the motives	Dominating aspect
	Academic achievement	**Project and team working capacity**		
1.	The motive is to get a grade	The motive is to pass the project	The motive is to adapt to the formal requirements in the situation.	The educational framework
2.	The motive is to learn computer science for the project	The motive is to gain familiarity with working in projects	The motive is to get something specific from the project	
3.	The motive is to learn how to learn computer science	The motive is to learn how a project functions	The motive is to learn something generalisable from the project	The content of the learning
4.	The motive is to learn something new	The motive is to become a better professional	The motive is to learn for objectives outside the current situation	

Silén (2000) has studied how students experience the meaning of their learning and how they relate to responsibility and independence within a PBL-based programme in nursing. Based on a phenomenographically inspired theory of learning, she identifies and describes four categories of a motive[60]. The first focuses on passing exams and managing the requirements in assignments. The second emphasises the need to organise the practical learning situation with respect to time usage and demands external to the learning. The third focuses on understanding and “readiness to act”. The fourth describes a process of discernment that “stands out as a learning process within itself” (ibid., p. 287). Here judgements, and considerations of relevance are in focus.

The similarities between the analysis of two motives from this project and Silén’s results are striking. A comparison is offered in table 14 below, in which data from table 13, and Silén’s categories (in my translation from Swedish) are listed.

The association between the first categories is evident, in that the requirements of the educational system is stressed both by me (the three left columns) and Silén (the right column). While my second categories stress the learning, but state that the learning is directed by the educational situation,

60. Silén uses a different terminology than the one deployed in this thesis. The presentation of her work given here is based on my interpretation of her findings. The translations from Swedish are my own.

Table 14. Reflecting the categories of "Academic achievement" and "Project and team working capacity" with the results by Silén (2000) concerning the motive in a PBL based course in nursing. The results from Silén (2000) are translated by the author of this thesis.

	Motives		**Common properties of the motives**	**Categories of the motive in Silén's (2000) study of project-based learning in nursing**
	Academic achievement	**Project and team working capacity**		
1.	The motive is to get a grade	The motive is to pass the project	The motive is to adapt to the formal requirements in the situation.	Requirements of the educational situation as a motive
2.	The motive is to learn computer science for the project	The motive is to gain familiarity with working in projects	The motive is to get something specific from the project	Bringing order to the learning situation as a motive
3.	The motive is to learn how to learn computer science	The motive is to learn how a project functions	The motive is to learn something generalisable from the project	The discernment process at a meta-level as a motive
4.	The motive is to learn something new	The motive is to become a better professional	The motive is to learn for objectives outside the current situation	Understanding and readiness to act as a motive

Silén takes the opposite stand, and describes an educational situation that has a strong impact on the learning that can take place. Silén describes the third category in the following words: "[T]he students have the intention to discern possible direct objects on a level that can be described as a meta level" (ibid., p. 287). Her characterization can be contrasted to the perspective in my results, that focus on the generalizability. Finally, in the last categories the horizon is widened to encompass the applications of the learning in a professional role or other situations outside the students' current setting.

The work of Silén does not only show similarities with the current study, also the discrepancies are important. Firstly, the subject areas and the settings of the studies differ. Secondly, my study has a different purpose, since the aim to create a holistic picture of the students' experienced learning environment is important.

The students' motives for taking the Runestone courses, revealed in this chapter will serve as a key component, when the picture of the whole is drawn in chapter 14.

Chapter 12. The how

In this chapter my interest relates to one of the key issues: How the students act in order to learn computer science. But before listening to what the students say about how they go about their learning, a few remarks concerning *what* they learn might be in its place.

The previous chapter has shown that there are various aims, of different kinds, that the students strive for. As a consequence, much of the learning that takes place is not directly related to computer science, but is instead concerned with issues as diverse as team work, different cultures, personal development, or how to interpret the formal requirements for passing the course. Also within computer science students learn differently and learn about different concepts. In a project course like this, the students certainly specialize in their teams, and come, in this way to individualize their learning. A quote from Anthony serves to illustrate this specialization:

Interviewer:	What have you learned in computer science?
Anthony_2:	Coding wise I have learned a little bit more in Java, just because I haven't really had that much experience in Java. So I learned some more Java skills. I didn't really learn a whole lot more networking concepts or anything.
Interviewer:	Why not?
Anthony_2:	Just because I wasn't really involved in any of that. I was more involved with just breaking the code and like I said Peter took care of the RMI.

This chapter first explores and presents different ways to approach the act of learning computer science. Then, the logical relationships between the categories are discussed. Finally, these results are related to other findings concerning the act of learning.

1. The act of learning computer science

Seven categories of the act of learning computer science, varying in several dimensions, have been discerned, as is shown in table 15 below.

Table 15. Categories of the act of learning computer science

	Description of the categories
1.	**Learning CS through learning to use application programs**
2.	**Learning CS through learning about isolated concepts**
3.	**Learning CS through consolidating what is already known**
4.	**Learning CS through analysing systems**
5.	**Learning CS through integrating systems**
6.	**Learning CS through giving meaning to concepts**
7.	**Learning CS through developing as a professional**

Category 1. Learning CS through learning to use application programs

The perspective of what it means to learn computer science that is captured in this category is only voiced once by Sven, but his statement is clear:

Interviewer: Have you learned much about computer networking and all that?
Sven_2: I have used a few things.
Interviewer: Such as what, then?
Sven_2: Yes, but I had never used, well some call it IRC, and others mIRC. We had not used before and there is a lot of stuff like that which one had not used before [...] lots like that. I spent so much time with people I don't normally spend much time with. And for instance just a thing like downloading a program. "You can not have that program. You should have that" and so you go and download other programs and records the memory of the little computer. No, but in this way you you get to learn a lot all the time.

The use of the general application programmes that Sven discusses are not regarded as learning aims in their own right for third year students in computer science. Instead these topics are normally discussed in an introductory course, intended to acquaint the students with the use of computers, as a preparation for later, advanced applications or for courses in computer science. IRC is a tool that sometimes is deployed in such introductory courses.

As stated before, it is always dangerous for a researcher to interpret the utterances of an individual student. But Sven's statement is clearly formulated, and contains details that clarify my proposed interpretation. His comments include different brand names of the software. He also points out that the students never have used a particular tool before, a statement that further amplifies this interpretation. The interpretation is also consistent with the interview as a whole, where computer science, or the learning of computer science, are rarely touched upon by the interviewee. Instead, Sven's answers to questions from the interviewer concerning computer science concepts more frequently refer to the educational framework, the tools deployed or the social setting in the team.

What it means to learn computer science is in the first category described as learning to deploy application programs. The category focuses on the tools that are used, and does not refer to the field of computer science at an advanced undergraduate level.

Category 2. Learning CS through learning about isolated concepts

The second category expresses learning of computer science as learning about isolated concepts in computer science. The interview excerpt with Staffan illustrates:

Interviewer:	Have you learnt any computer science through working with the camera then [...]?
$Staffan_2$:	Yes you bet, I have found out a bit more about Linux, how you install things, and download new sources and compile them. That sort of thing, it was quite a lot of fun but at the start we had a lot of problems with the computers as well, both Magnus' and Michael's computers crashed a few times so we had to reinstall everything, yeah, otherwise, well yeah it was a lot of stuff like that as we learned the technical stuff.

After this statement, the theme is continued, and Staffan names more topics, such as compiling and installing the operating system Linux. Sven is more straightforward in his wording:

Interviewer:	Yes, I understand. What do you suppose that you have learned out of all this?
$Sven_2$	One can say that I have learned a damned lot of Java

In the continuation of this discussion, Sven tells the interviewer what Java concepts he has learnt, and offers a catalogue, containing specific topics, such as Swing components[61] and ways to search in the class libraries[62].

Neither the statement of Staffan nor that of Sven considers what learning means, or what is done with that which is learnt. Only the catalogue of concepts that are learnt is offered. What it means to learn is not discussed; learning is learning specific technical concepts.

Category 3. Learning CS through consolidating what is already known

In the third category, the computer science concepts are still seen in isolation. Learning is here understood as getting deeper insights, or consolidating what is already known. Let us listen to Samuel:

---

61. The Swing components are used in Java programmes with graphical user interfaces.

62. The core of the Java language is rather small. Instead, Java has "libraries" that contains classes and procedures that a programmer can deploy as parts of his programme.

Interviewer: What do you expect to learn in computing or computer networking?
Samuel$_1$: I reinforced some of the knowledge I had, so, and perhaps a little, little more, such that, yeah, they became a lot clearer for me. I guess I would have been satisfied with a few labs and so forth, passing a few labs. And so it was not so accurate that I learned a lot after all, since it only was just a lab and so.
Interviewer: Labs in computer networking?
Samuel$_1$: Yeah, TCP, yeah.
Interviewer: Right, TCP, yes

Samuel argues that “some of the knowledge I had [...] became a lot clearer for me”. The underlying meaning, that the insights become clearer by working in the Runestone project is taken for granted.

The third category offers a view of learning of CS which retains the fragmented view of computer science concepts of the second category. But in contrast to the second, that takes for granted what learning “is”, this category enables a different form of handling that which is learnt.

Category 4. Learning CS through analysing systems

This category introduces the idea of a whole, of which different concepts are parts. The interview extract below is taken from a longer discussion, in which Abraham discussed what his team did in order to get started with their project and how he valued the different steps the team took.

Interviewer: Are there advantages of that?
Abraham$_1$: [...] look at the code a lot, mostly ‘cause we were trying to figure out what was wrong and it did make us examine how everything was working. So I guess if the code had worked right away in a way we might not have looked at it so closely, I think.

Abraham had earlier told the interviewer that the team had been frustrated concerning the selection of the code. The fact that the code did not work when the students initially got it forced them to analyse it in detail. Abraham mentions this as an advantage, when explicitly asked by the interviewer. The direct answer, presented without hesitation, indicates the importance he places upon reading code.

Staffan explains why it is useful to study code that others have written.

Interviewer: What do you think you will learn from this?
[...]
Staffan$_1$: [...] Yes, well that is a point, so you learn to program a bit, and there is not so much to program actually. We read code mostly, and try to understand their existing code. Is is also worthwhile to learn how one reads code and documentation and that is not something that we have done much of before. It is really mostly now that one realises how important documentation is. [...]

By studying the code of others, you learn about how you should write your own programmes. In this particular quote from Staffan, documentation is in focus, while other topics in similar ways are highlighted by other students.

In this category, focus is directed towards the whole of a programme that is written by others. To study such a programme makes it possible to explore the role of the parts, and thereby to learn.

Category 5. Learning CS through integrating systems

The need to learn to integrate pieces of code into a whole system is stressed in the fifth category. Adam is one of the students who is explicit about this:

Adam_2:	[...] The interaction between the motor daemon and the video daemon and the game server, you know, just having separate components that were totally independently written. It was very interesting and sort of a new experience to see that kind of thing and how they ended up working together so well. [...]

A software system is built up of different components – each with a different function in the final system – that has to be integrated, Adam argues. Samuel follows a similar line of reasoning, when he discusses how the pieces come to be a whole:

Interviewer:	What do you think you have learned by doing this?
Samuel_2:	Yes, okay, firstly I learned how a complete project could turn out, in fact how the different parts could be integrated into a complete project.
Interviewer:	Yeah, right.
Samuel_2:	What I have learned, that was not something that I had worked out before, now I know, more or less, how you can integrate different bits, how you can communicate various parts [...] but specifically, yes, theoretically I knew earlier how it worked.

In contrast to Adam, Samuel does not specify what kind of components that a system is constituted of. Instead, he stresses that he has learnt to perform such an integration.

Contrasting this category with the previous one reveals an interesting difference. The whole software system is something that “already exists” or is taken for granted in category 4; a way to learn is to split this unit into pieces and analyse it. The process is turned in the other direction in this category. Putting the components together is a way to create a system, an integration that can be made by the students.

Category 6. Learning CS through giving meaning to concepts

Learning computer science means, as it is described in this category, to understand the meaning of the underlying concepts. Let us listen to two of the students, Axel and Alec:

Interviewer: What did you learn there? Did you know Java before?

$Axel_2$: A little bit. When I came here…I actually started here in 1996 so I have been a student for quite awhile. I transferred out and came back. When I started, our programming courses were taught in C++ and when I came back they were taught in Java. So I had the beginning of C++, which I didn't understand nearly as well as I wanted to when I was here. Then I came back and had to take the second level course in Java, which I didn't really understand at all. So I understood a little bit of it, but I learned a lot. I learned a lot of programming, even though we didn't do a lot of programming, we looked at it enough where I know a lot more then I did. We learned a lot about RMI. We read and read and read everything we could find about it. So I learned a lot about how it works. I learned a lot about how clients and servers actually communicate. I had an idea, I think like everybody does about something they don't understand very well, they have a general understanding of how it works. I think I understand a lot more now and I understand networks and how they communicate a little bit better I think.

While Axel tells his story, Alec's statements are more concentrated:

$Alec_2$: [...] But as far as a learning factor, I learned an immense amount about RMI as far as I am concerned because I didn't even know it existed before. It is very interesting to me.

Interviewer: So you learned RMI from scratch to advanced, is that correct?

$Alec_2$: I don't know if I am advanced but….

Interviewer: You learned it anyway.

$Alec_2$: Yes.

If reading these statements superficially, they resemble how Staffan and Sven, both quoted as examples of category 2, presented lists of what they had learnt. However, Axel and Alec both relate their discussions to themselves and put their learning in different contexts, Axel through his story and Alec by stating that he found it "very interesting". The focus of their statements is not on the topics about which they have learnt, but instead of their experience of learning these topics. Factors external to the concepts, such as courses (Axel) or the project (referred to as "a learning factor" by Alec), serve as stepping stones, helping to focus their attention. In short, they are searching for a thorough personal understanding of some concepts.

The sixth category has introduced a way of experiencing learning where personal insights concerning the computer science concepts that are studied and a personal understanding of computer science, form the core.

Category 7. Learning CS through developing as a professional

As for the first category, evidence for the seventh category, which highlights a conscious personal development as a part of the result of learning computer science, can only be found once in the data. Alec answers one of the questions towards the end of the second interview in the following way:

Interviewer:	Did you learn anything from that?
$Alec_2$:	Yes, uh just basic design, how other people think, how you should approach such a large amount of code just thrown at you saying I want to know what it does and I want to know how it works. You can't just start picking it up and reading through it and saying Ok, I got it. So just a strategy I guess as to how I would do it in the future.

Alec could, for example discuss an analysis of a programme (category 4) – a researcher can never "look" into the head of a student. But Alec discusses how "other people think", and how to relate himself to a large software system. He stresses the need to take personal decisions ("I want to know what it does and I want to know how it works") and contrasts this to a superficial way of approaching the system. Finally he refers this learning to his own future. Based on this interpretation, I judge that his way of experiencing his learning differs from what previously has been described, in that a conscious professional development is in focus. These arguments also correspond well with the fact that there are categories of the motives that stress personal development (see chapter 11) and contrasts this to a superficial way of approaching the system.

2. Understanding how the students act to learn computer science

Learners succeed to different extents in their learning. This difference can partly be explained in that they think in different ways about what they do (Marton & Booth, 1997). In the light of these insights the results presented above ought to be further scrutinized. What characterizes the different categories, and what, if anything, can be said about desirable acts of learning.

The first category "Learning CS through learning to use application programs" differs from the others, in that it does not discuss the learning of computer science as an academic discipline, but instead focuses on the learning of computer based tools. Issues that are important in computer science, and to different degrees present in the other categories, such as analysis and construction, cannot be related to the learning of application programs. For this reason, this category will not be considered in the comparisons presented in this section.

Table 16 below highlights the similarities and the differences between the categories. Each of the columns are labelled with a character that indicates a particular dimension of variation. In each column the feature described is valid below the indicated line.

Table 16. Aspects of the different ways of experiencing the act of learning computer science. The letters in the headers of the columns are explained in the text

	Label	A	B	C	D	E	F
1.	**Learning CS through learning to use application programs**						
2.	**Learning CS through learning about isolated concepts**	Learning in computer science as an academic discipline					
3.	**Learning CS through consolidating what is already known**		There are different ways in which CS concepts can be understood				
4.	**Learning CS through analysing systems**			Focus on a whole – parts relationship			
5.	**Learning CS through integrating systems**				Making or transforming something		
6.	**Learning CS through giving meaning to concepts**					Searching for personal meaning	
7.	**Learning CS through developing as a professional**						Professional development

It indicates a certain feature that is present from a certain category, and further in the more advanced ones.

A. The first column, that has already been discussed above, introduces the learning of computer science. It indicates that learning that is not relevant from the point of view of computer science is to be disregarded in this discussion
B. This column introduces the categories in which the meaning of learning is not taken for granted.
C. Here the contrast between learning about unrelated fragments, and focus on the parts – whole relationship is highlighted.
D. The aspect of constructing (or transforming) something (in contrast to analysing something that already exists, in a book or in a program) is singled out in this column.
E. The search for a personal understanding or meaning of concepts, is here contrasted to an impersonal meaning of concepts.
F. The final category focuses on a conscious personal and professional development that is the result of learning computer science. Understanding how others think and being better equipped for a future career are the key features here.

Logical relationships between categories can be formulated in the additional dimensions that opens with each new category. Based on this argument, it is concluded that table 16 also indicates how the categories are related.

3. Exploring the results

Much efforts have been put into revealing how students go about study. In an influential paper (Marton, Beaty & Dall'Alba, 1993), six categories of learning among Open University (OU) students in the UK are described. Table 17 shows these, as they are presented in Marton and Booth (1997).

Table 17. Categories of learning, adapted from Marton et al. (1993)

	Categories of learning
1.	Increasing one's knowledge
2.	Memorizing and reproducing
3.	Applying
4.	Understanding
5.	Seeing something in a different way
6.	Changing as a person

The results from the current study correspond well with the findings of Marton et al. (1993), but there are also differences between the two. The category that discusses learning as related to other topics than those of the academic subject area does not have any correspondence in the OU study. For the remaining categories similarities can be identified by abstracting the results from both studies. While the details of this analysis will be presented elsewhere, some broad themes are presented here. The second category of the current study, in which the issue of what learning is, remains unproblematised and corresponds well to the first categories from the OU. Here focus is on the act of learning itself and that can be described in such words such as "picking up" or "accumulating". The OU study then discusses some of the following categories in terms as "memorizing and reproducing", "application", "integrating [...] into their own worlds" "[making] the world appear in a different way" (Marton & Booth, p. 37). The distinction between the categories 2 to 4 of table 16 above (category 1 is still excluded from the discussion) on the one hand, and the three more advanced on the other, has its similarities in the study of Marton et al., where the differences of surface and deep learning approaches have been identified. The close relationship between deep approaches to learning and a good learning outcome, have previously been discussed in this thesis. Such results clearly point towards the need for educators to design courses that encourages the forms of learning that are described in the advanced categories.

It is interesting to contrast these findings with the results of the studies of Booth (1992) and Bruce et al. (2004) who report findings concerning the act of learning to program. They have revealed fewer, and different, categories compared to the current work. Without a thorough investigation, one can only make informed guesses about the reasons for these differences. The studies are performed in different settings and have different foci on the subject area. The two studies concerning the act of programming are performed with first year students, and focus on the learning of a particular task: learning to program.

It is also worthwhile to consider in what way the issue of subject area influences the results. The current study investigates how advanced students, majoring in computer science, think about learning computer science. As only few individuals have studied other subject areas than computer science, their understanding of what “learning” means is based on their experience of “learning computer science”; “learning computer science” is what “learning” means. Here the study differs from the OU study in that the OU study draws its data from students in the social sciences

A number of studies have been performed that resemble this one, most of them showing results that - more or less - correspond to these. Particularly the work of Marshall, Summers & Wollnough (1999) is interesting, since it focuses on learning in an engineering context. They have identified five categories of the act of learning. They argue that the fine-tuned differences in the results in the various studies are related to the cultural and educational contexts, that makes “different aspects of the learning experience be foregrounded or accentuated in different contexts” (ibid., 305).

Exactly here, in its origin within computer science, lies the key strength in the current work. Each of the categories, as well as the dimensions of variation, that highlight differences between the categories, are sprung from computer science, and are thus described in the terms of the field. It becomes easy to bring the results back into teaching situations, since the results “talk computer science”. The distinction between the categories 2 to 4 (category 1 is still excluded from the discussion) on the one hand, and the three more advanced on the other, has its similarities both in the study of Marton et al. (1993), where a dichotomy between surface and deep learning has been identified, and in the recognition of a reflective act of learning in the advanced categories in the study of Marshall et al. (1999). The close relationship between deep approaches to learning and a good learning outcome, have previously been discussed in this thesis (see chapter 12). Such results clearly point towards the need for educators to design courses that encourage the forms of learning that are described in the advanced categories.

Chapter 13. The where

This chapter focuses on the where of the students' learning, that is the learning environment as it is experienced. The environment is described through some key phenomena, selected to illuminate different aspects.

The aim of the analysis presented here is two-fold:

Firstly, the findings from the phenomenographic work presented in this chapter are key components for the further analysis, that aims towards the creation of a holistic picture of the students experience of learning in their learning environment

Secondly, the insights gained through this analysis have a value on their own for teachers, course producers and students, since the chapter discloses the students' perspective on some key elements in the environment. Such an understanding can be used as a tool to improve teaching and learning.

Figure 9 illustrates (marked as phase 1 in the picture) how the results, in these two ways, become present in the total outcome of the research project.

The phenomena that are selected express the students' experience of being team members, the selection of which of the codes[63] from previous year to develop, the weekly milestone meetings and the experience of being graded.

1. The team

The team has a fundamental role in the Runestone project, since its members collectively develop the software system and jointly report to their teacher. The collaboration within the team, and the organization of its work, thus become factors that influence the individual's experience of being a team member, as well as the quality of the outcome.

Numerous definitions of the term "team" exist, often with only subtle differences. Jaques (1992, p. 13) gives the following definition of a team in a pedagogical context:

- Members are collectively aware of their existence as a team.
- Belonging to a team satisfies a need.

63. The term *code* is discussed in chapter 2.

- The team has a shared aim.
- Members are interdependent.
- A team has a social organisation.
- Team members interact.
- Members want to contribute to the team.
- A sense of membership exists.

This view fits well with the perception of how a team is perceived in the Runestone environment (Arnold Pears, private communication).

What it means to be a member of a particular team is settled in the continuous interaction between an individual and his team. The structure of the teams and the distribution of control are dynamic: the relationship between the actors, as well as the actors themselves, changes continuously. An example can illustrate this interaction: A student who acts (for example by arguing that a particular task should done by the team as their next step) changes not only himself (by taking a stand) but also the team (for example, by making the team focus on something new). The relationship between the student and his team also changes with his action[64]: He might, for example, get a more central position as a result of the discussion. The complex Runestone environment, where the complete team never meets face to face, brings these matters more clearly into focus.

1.1. Analysing the experience of being a team member

Among the first tasks performed by the students was to select a leader in each team. His duties were described in the following way in the course documentation[65] on the web:

> Team leaders coordinate work allocation and check that tasks are being carried out and act as the official interface between faculty and the team. This is a voluntary role, so nominated team leaders are expected to devote just as much effort to the practical aspects of the project as the other team members. Team leaders must register their election [...] Registration MUST be done prior to the first team milestone report.

During the interviews, the students were asked about different aspects of their teams, such as how decisions were taken and which role the team leader had. The issue of being a team member was also discussed during other parts of the interview, sometimes as the result of a follow-up questions, and at other occasions on the initiative of the interviewee.

---

64. The term *action* is here used in a generalized sense, where also talking or taking standpoints are interpreted as actions

65. `http://www.csis.gvsu.edu/class/brio/Management/rules.html`

In the analysis, some key factors, that together play an important role in the students' experience of being team members, have been identified. These are:

a. The experienced structure of control
b. The experience of the parts of which the teams are constituted

The first phenomenon concerns the capacity to take and implement decisions with the team, in this thesis labelled "structure of control". Here a complex picture evolves that expresses itself in issues as the roles of the leader and the team members, as well as their relations. Aspects concerning the organization (or lack of organization) of a team also become visible.

The second phenomenon describes the structure of the team that a student experiences and addresses the issue of which parts, or units, constitute a team. In this way some aspects of what a team "is" are investigated.

1.2. The structure of control

To understand how decisions are taken and how control is exercised is crucial for understanding a team. The findings on this issue are summarised in table 18.

Table 18. The categories of the control structure in the teams

	Label	Description
1.	**Control is missing**	Decisions are taken ad hoc in a social game
2.	**Control is taken by a few**	The leader or a sub-group take decisions and act on his/their own
3.	**Control is allocated**	The team selects a leader, that coordinates and take decisions
4.	**Control is distributed**	The decisions are shared in the team

Category 1. Control is missing

This category has only been voiced a few times, and then always in discussions about teams that have not been experienced as functioning well. Sven presents the most downright expressions of this situation in data:

Sven$_2$: And so, then, it actually became two in the US and we three here, since we had divided up the project from the beginning. We had two people who should work on the applet, two on the navigation algorithm, two on the video daemon, and then, because the applet was working as it was, we thought that it was a bit much for two people to continue to work on it, so we split ourselves up so that we could help with work in other places. It was me and Mats who did this, but it seemed like no-one was interested in any help in the other areas of the project, so we just kept going on the applet, to fresh it up. It was interesting in the presentation later, when we had it, because then the person who was working on the navigation algorithm, a guy in the US, who said that he had done a lot of work suddenly said that [...] his work partner quit and that he was alone and didn't get any help from anyone, and so he didn't get far with his part.

Interviewer: Where was that?

Sven$_2$: It was in the presentation that he had on the overhead projector.

Interviewer: Yes.

Sven$_2$: So this was rather a shock for all of us since both Mats and I had said that one of us could jump in and help out on the navigation algorithm. He had said that he made changes and everything and after all that, nothing. And Martin is working on the video daemon and what I did with the applet I did special stuff so that you click on the applet and then one can change the camera settings so that it wasn't necessary to change the video daemon in the USA.

The team as a whole had not taken any conscious decisions concerning the distribution of work. Instead different team members worked – or did not work – with tasks they had selected themselves. Sven's statements are the strongest, but they are not alone. Sebastian reports about a team where the organization of the work partly failed:

Interviewer: What do you think about your result? Are you happy?

Sebastian$_2$: Oh yeah, somewhat I think that everything is working out really well except for the navigation algorithms and that was mostly done on the American side, but since our American side didn't work at all until about seven weeks had passed we got to do this stuff, under a bit of stress towards the end, and hope to adjust everything to the new situation a bit, more or less, but I am pretty pleased anyway.

No one in the team had, Sebastian tells, insights into which tasks that were finished, and which were not, until very late in the project. An ad hoc group, with Sebastian as one of its members, then took control and finished the tasks.

In the absence of an organised structure of control, different ad hoc created sub-groups take their own, possibly even contradictory, decisions. The control structure, as well as the positions of the individuals, are here settled in an on-going social game. As the control continuously moves, it is impossible for a team member to know with whom to communicate in a certain situation. In all such cases found in the data this has led to situations that were experienced as more or less troublesome.

Category 2. Control is taken by a few

The experience of a situation, where a leader or a sub-group take decisions and act on his/their own, without the consent of the team, is identified as the second category. Let us listen to Stig:

Stig$_1$: They, they think that, to decide stuff, well he thinks that he has the right [to decide]. I think, and that goes for the whole Swedish group, that his job is only to get a grip on things, see that stuff happens, see that communication works out. Mostly so that there is someone, like for instance our teachers, who should have, well know, that it is him that, that they should contact. And he should manage that communication, like, so that all six don't send the same mail to the teacher, instead he should do it. And, to some extent even allocate work, and say "If you can do this it would be great?", and make sure that everyone always has something to do [.....]

Interviewer: Yeah?

Stig$_1$: Decide, I think.

Interviewer: Sometimes you say "the Americans" and sometimes you refer to your group leader. What's your experience there?

[...]

Stig$_1$: Yes, but it is you know, well like it is all three of them, well at least two of them who are good friends I think. And it is not just the group leader, in fact, them two of them, who think ... decide things together.

When pushed to elucidate if the team leader alone or some team members together take the decisions, he hesitates. How the decisions are taken is not transparent to him, and he does not feel that the situation is satisfactory to him. Anthony, on the other hand describes the situation more exactly:

Interviewer: O.K., ja. How do, how did you manage to do this division? How did you do it?

Anthony$_1$: In the IRC meeting we said, O.K., these are possible milestones we can do, is there anything you guys feel very strongly towards? We kind of pushed the camera upon the Swedes, saying - we can't do it on our end because obviously we don't have the camera...

Anthony$_1$: Ja, so some of you guys are going to have to work on the camera. [...] And the people who said, who really kind of stayed in the shadows and didn't really say they want to do anything, we said well, you are over here on the Swedish side, why don't you help these guys with the camera and like you know,?[...] and help with the navigation of it.

Anthony here makes a clear division between "we" (the Americans) and the Swedes. The Americans propose tasks to the Swedes, as well as to themselves. Later in the interview he explains that he finds this to be an efficient organisation of the work. Decisions can be taken without time-consuming discussions.

The second category interprets a situation, where a team leader or a sub-group take decisions on their own without prior agreement of the whole team. The control is in the hands of few, or even a single person. A team member who is not in possession of control, does not have a working two-way communication with the owners in control. Instead, he is talked to and told what to do, but cannot influence the situation himself.

Category 3. Control is allocated

The third category expresses the situation where an elected team member can take decisions and make arrangements. Abraham's statements illustrate this idea:

> Abraham$_1$: Yeh, yeh. We tend to just, once we're given an assignment we'll do it until the team leader says otherwise. You know, or if the team leader.., we look for him, to him for to tell us if he needs, if they need anything more from us, or if he needs some like special work from us, then we look for the team leader to give us that, sort of command.

The team leader should not only take initiatives, but also organise and distribute the work. Being selected by the team members, he is authorized to do this. Adam discusses the responsibilities that come with being in control:

> Interviewer: So what do you say is the function of a group leader?
>
> Adam$_1$: Um, to um, help people, divide up tasks, um, keep track of requirements and make sure that there is somebody to do each of the requirements for each milestone. Um, if there's problems that people don't understand what is going on or something, you know sort of be in charge of getting answers to questions, if I don't know the answer myself, I think to figure out where to go for it, and just try to keep everyone on the same page.. pretty much.

The duties of the team leader are extensive in category 3. He both has to take decisions for the team and coordinate its work. By offering support to the team members, for example concerning technical problems, he aims to make sure that everyone contributes to the result of the team. Being selected by his peers, he is a representative for them, and owns his control as the result of a joint decision of the team. That is, the team members have allocated part of the right to take decisions on their behalf to the team leader, in exchange for management of the project and of individual support. A team member here has a working two-way communication with the owner of the control.

Category 4. Control is distributed

The control of the work of the team is experienced as distributed and shared in category 4. In the extract, Alec presents his opinion:

> Interviewer: [...] What would you say is the function of the group leader?
>
> Alec$_1$: To organize the entire group as a whole, to um, manage what is done by who, by what time, um, fill in the gaps so to speak. Um, there is a big distance between the two of us, I feel that he should be the main communicator, or assign some sort of organization to the communication. Other than that, basically just making sure everything gets done.
>
> Interviewer: You said 'manage', what do you mean by manage?

Alec$_1$:	Um, making sure that communication is happening. Controlling misunderstandings between the two groups or group members. Attempting to clear that up, and when all else fails setting precedence.
Interviewer:	But you don't think, if I've understood you correctly, that his role is to distribute the work.
Alec$_1$:	Not exactly, no. I don't believe that.
Interviewer:	It's more to coordinate the work, is that correct?
Alec$_1$:	Right, yes.

The focus in the utterance of Alec is on communication. The team leader should coordinate by communicating. It is his duty to keep an environment, where the conflicts can be solved and work be done. This point becomes still more visible as Alec contrasts it to the hypothetical situation "when all else fails setting precedence". An underlying assumption in what he says is that team members are active and independent. If they were not, their communication should not need to be organized, or possible misunderstandings solved.

Albert discusses the implications of the open communication on the ways in which decisions are taken.

Interviewer:	O.K. What is the role of the group leader as you see it?
Albert$_1$:	Um, I see it in the way that it seems to be currently being held is that, kind of like, the team leader is keeping the direction of the team. [...] Um, agree with the group members who is going to be doing what, you know. And then say, you know, we need this by this date for our meeting and .. so far it seems more in our group, at least I think, that it seems more of a, there is a leader but it seems almost like a mutual leadership where he might suggest. Peter might suggest um, who wants to do this particular section for this time and if someone responds then great, they've got it and if not then he says well how about such and such doing it. That seems to be the, the, what is currently happening.

In the next statement the interviewer summarized his understanding of Albert's statements, and got an answer that confirmed his conclusion.

Interviewer:	It is a very democratic group, in a sense. Is that what you're trying to say?
Albert$_1$:	Yeh, yeh, I would say that [...]

The control is shared within the community, in a way that could be called "democratic" in the fourth category. The position of individual team members, being integrated parts of the structure of control, are strong, and they play active roles of the life of the team. Communication and coordination are the main tasks for the team leader.

1.3. How is a team experienced by its members?

The categories of the structure of control have been analysed to show their hierarchical nature with respect to key aspects (see table 19).

Table 19. Summary of the categories of a control structure within the teams

	Label	By whom is control exercised?	What does it mean to exercise control?	Communication between an individual and the centre of control	Role of the individual
1.	**Control is missing**	Defined ad hoc in a social game	Varying	No stable communication is possible	Varying
2.	**Control is taken by a few**	Leader or a sub-group	Decisions are enforced	One-way communication	Uneven distribution of control
3.	**Control is allocated**	Originally by everyone, but is channelled through someone	Distribution of duties, obligation to offer support	Two-way communication	Offer control and get support
4.	**Control is distributed**	All individuals	Communicate and coordinate	An individual is integrated in the control structure	Active

Although the categories, as they are defined at a collective level are stable, the situation that gives birth to them changes dynamically throughout the life of the team. The Runestone course demands a change over time: At first the members of the team at the two sides do not know each other, but eight weeks later, they present a complex software system that has been jointly developed. The teams need to evolve along this journey, during which the members jointly work towards an accomplishment of the task. In the data this is expressed in the shifts between the categories. The categories do thus not express how particular teams “are”, but reveals the patterns that can be discerned from the students’ descriptions of their experiences of being team members.

1.4. The team structure

The analysis reveals three qualitatively different ways in which the structure of the team is experienced, as being fragmented, as being constituted of halves and as being a whole.

The categories will only be summarised here and will be presented without interview extracts, as an account of the analysis does not contribute to the full picture. The three categories are summarized in table 20 below.

Table 20. Summary of categories of the team structure

	Label	Description	Purpose of the entity	Structural aspect In focus
1.	**Fragmented**	A team consists of smaller entities, individuals or sub-groups	Smaller tasks to solve	The individual or small sub-group
2.	**In halves**	A team consists of one half at each university	Coordinating, collaborating locally	The half-team
3.	**A whole**	A team is a single unit	Acting as a team, being visible outside the own team	The team

Category 1. Fragmented

In the interviews, evidence is found that the team is experienced as a set of smaller entities: individuals or pairs, cross-atlantic or local. The purpose of this organisation is to work on specific sub-tasks, such as coding.

Category 2. In halves

The most frequent category describes the team as consisting of two geographically separated halves. While this situation is regretted by some students, others find it valuable, because it promotes a good result.

Category 3. A whole

The third category sees the team as a unit and can be identified in situations in which the team needs to act, for example when dividing the work, choosing a code, having a meeting with the teacher, or presenting the results of their work. This way of experiencing the team is rarely voiced, although frequently present implicitly, since it is a way of working that is taken for granted.

1.5. The results come into play - seeking contradictions

An intellectual tool, offered by activity theory, is to seek for contradictions. This step in the analysis corresponds to phase 2 in figure 9. The results are only presented here for the structure of control, since the analysis of the team structure that has been performed does not reveal any interesting results.

Category 1. Control is missing

The possibilities for successful communication and distribution of work are low within a team that is experienced to be without a functioning control structure. Figure 10 below illustrates such a situation.

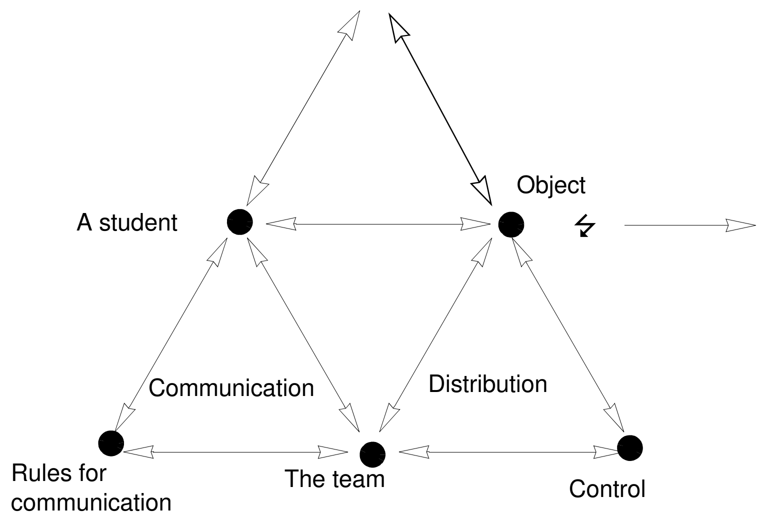

Figure 10. Relating the category *Control is missing* of the control structure to an activity system. The flash arrow indicates a contradiction in the object node, related to problems with defining a clear motive for the activity. Without a clear object, the activity loses its meaning.

Activity theory suggests that problems of the kind that are reported here related to contradictions in the object node. Without a clearly defined, jointly agreed object, the activity loses its purpose and meaning. In the situation described by Sven (page 143), the conflict about what the team should do indicates an object with inner contradictions. This is indicated by the flash arrow in figure 10. Thus, a situation experienced in this way can only partially fulfil its intended purpose: to facilitate the subject's aim to transform the object. The weak relationship between the nodes is illustrated in figure 10 by the absence of highlighted arrows.

Category 2. Control is taken by a few

In the second category two contradictions come to the surface. The first stems from the situation being judged differently by those who experience themselves to be in control, and those who do not. A student who is not in control here experiences a one-way communication, where he is talked to (see table 19), as is indicated by the dashed arrow in figure 11 below. In a similar way, students in possession of control have reported instructing others what to do. Clearly this is a source of frustration for those who lack control. This contradiction between the subjects' expectations and the experienced structure of control is illustrated by the flash arrow in figure 11.

Another contradiction can be identified in the interpretations of the rules and the previous decisions of the team as is indicated by the flash arrow at the rules node at figure 11. A student who is not a part of the leading sub-group experiences that his interpretation of the rules is by-passed by those who have taken the leadership, and that he is not offered the possibility to make his voice heard and to influence the work organisation. Those in power, on the other hand, argue that the rules they implement are intended for the common good, since they create an efficient organisation, as stated by Anthony on page 145. The situation here, with these contradictions, is not stable.

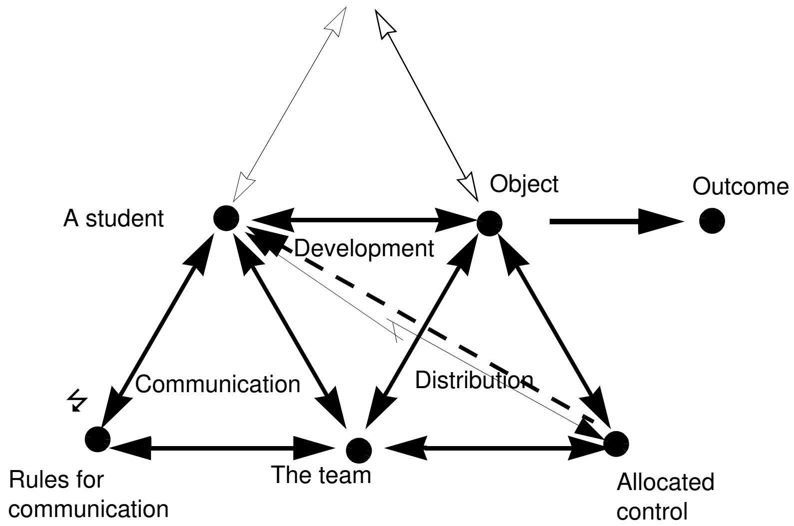

Figure 11. Relating the category *Control is taken by a few* of the control structure to an activity system. The flash arrow at the rule node illustrates the different interpretations of the rules of the team. The dashed arrow here indicates one-way communication from the leadership to a team member.

How contradictions can be a source of change is seen in the first interview with Stig. He tells about an attempt to change the situation in the team in a discussion with the interviewer concerning a particular milestone meeting:

Interviewer: What have you all done to deal with all this, ‘cause it seems obvious that you don’t think that things are going well?

$Stig_1$: Hmm, we had that sort of things [the team building exercises] at the start, so I think absolutely that you speak out directly [if something is going wrong]. [...] We have accepted a great deal, but then, last week, we all three of us wrote [an email to the Americans]. Because at that point we all got angry again, when they had changed something again at the last minute.

[...]

Stig$_1$: Yeah, the content [of the web site], it should have been updated with information, the day before or the night before, we had the meeting with the teacher for our progress report. But there was a reason for that we heard. So then we wrote an angry mail to them and explained exactly how we felt and explained what we thought the team leaders' role was and so on. But it felt afterwards as though they had understood and so that was good.

Whether the efforts led to a new organisation of the group cannot be told from this interview. Unfortunately Stig was not available for a second interview. Still, the example indicates that attempts are made to solve the contradictions by reorganising the control.

Category 3. Control is allocated

A similar picture can be discerned for the third category, but here there is a two-way communication, as is indicated by the dashed line in figure 12, below.

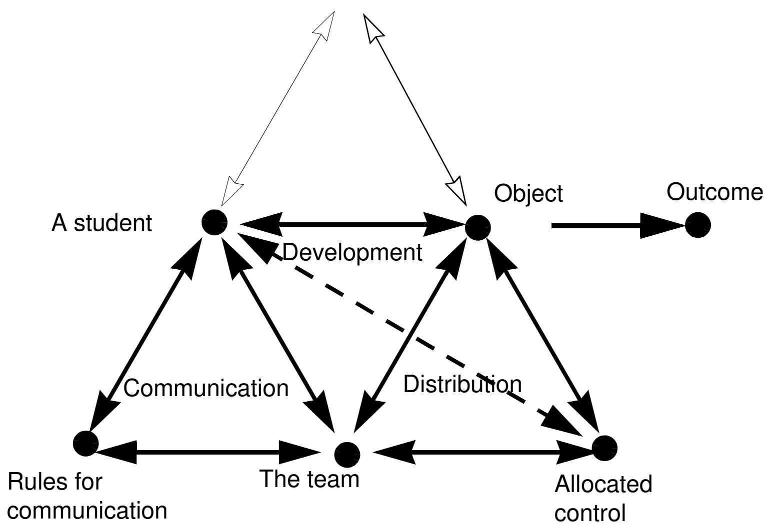

Figure 12. Relating the category *Control is allocated* of the control structure to an activity system. In this situation, the dashed arrow indicates two-way communication; that is a team member can communicate with the leadership.

This category explicitly describes a change. Initially, as a new team is constituted, it selects a leader, to whom the right to take and implement decisions is allocated. The rules for the collaboration as well as for the division of control and labour are negotiated. At later stages, when the rules are set, the situation develops so that the leader comes to act on behalf on his team taking decisions and distributing tasks according to the rules initially set. The roles of members in non-leading positions also change to become dominated by the rules. The sketches in figure 13 visualize this change. After the initial phase (upper triangle), the leader experiences a situation where distribution and development form the core, as illustrated in the lower left sketch. For a team member in a non-leading function instead communication and development dominate their experience of the situation (lower right sketch)

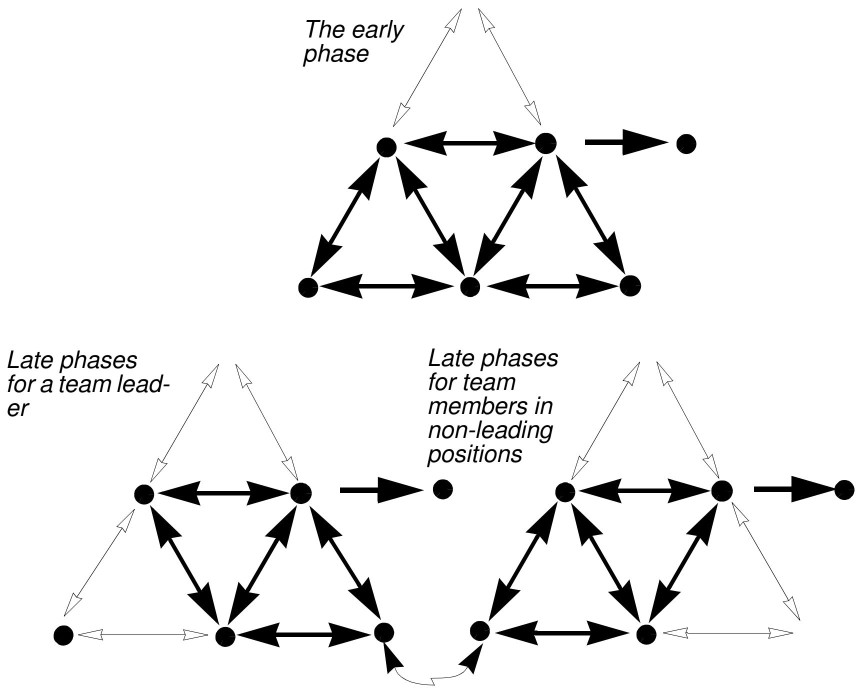

Figure 13. The change over time for the experienced control structure of the category *Control is allocated*. The top sketch indicates the situation as experienced in the early phase of the team when the whole team collaborate in selecting a leader, the lower right sketch depicts how the situation is experienced later, after the selection, by a team member in a non-leading position, while the lower left shows the situation later as experienced by the leader. The flash arrow indicates a contradiction between the experience of the rules that govern the team and the experienced division of labour. The labelling of the nodes is shown in Appendix B.

In general, this category represents a functional structure of control. Problems have rarely been reported by team members who do not have a leadership position. One team leader, Adam, has, on the other hand, vented frustration. He dresses his concerns in the following words:

Interviewer: Isn't that [...] a rather normal situation, that the code doesn't work and the hardware doesn't work?
$Adam_1$: Right, but I think because of being a school situation attitudes are different about it.
Interviewer: Yeh, O.K. I get your point.
$Adam_1$: That is something that I am not.. I personally, very much dislike, being a leader in a team where I have no power to enforce my leadership.
Interviewer: O.K.
$Adam_1$: Because if I see a problem and I say to somebody, 'this is the problem, this needs to be fixed', they blow me off, they don't care that I have said this. I have no impact on it and it is very difficult then to, to accomplish things and to function as a team when people aren't really seen as, as a serious team effort and project that needs to be treated as if it were a work situation.

Adam is expressing a conflict between his duties as a team leader, and the absence of meaningful ways to enforce his leadership. Although the exact reasons for his frustration cannot be determined, some aspects of the activity theory framework can shed further light on the problem he discusses by relating the problem to a larger situation. When the team is working with the code, after the initial decisions, the rules about how the work should be performed are applied. If a problem of some kind arises in the work of a team, it cannot easily be solved through a redefinition of rules in the on-going process, since the rule construction has now ceased, and the leader no longer has the capacity to influence the rules (as is illustrated in the lower left triangle of figure 13). The process of redefinition is not directly available at later stages of the second. A possible contradiction experienced by the leader here becomes visible. He has to obey the rules decided earlier in his work with distributing tasks, but without being capable of influencing the rules. This contradiction between the rules and the division of control is visualized by the flash arrow in figure 13. To overcome this contradiction, a developmental change of the system itself is needed (Engeström, 1987), that is, new decisions about the structure of control need to be taken within the team.

Category 4. Control is distributed

The individuals jointly take decisions that are intended to move the team and its members closer to their aims. The category that is described here, where power is experienced as shared among active independent participants, offers descriptions of three sub-triangles (as is illustrated in figure 14). Each of these can then be further studied, deploying elements of activity theory to shed light on their particular characteristics.

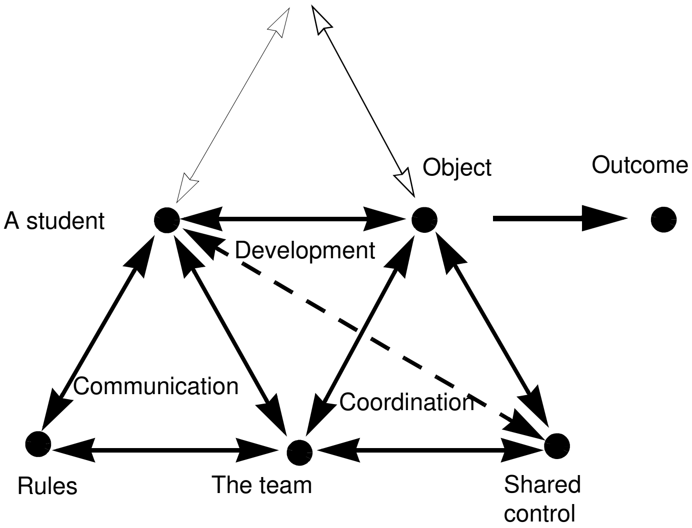

Figure 14. Relating the category *Control is distributed* of the experienced control structure to an activity. The dashed arrow here indicates participation. Every team member is a part of the leadership.

The forms for the communication in the team are negotiated in the *communication* triangle (lower left corner in figure 14), which is constituted by a subject (a team member), the community (the team) and the rules, norms and traditions that evolve in the discussion or that are brought into the situation. Here, the relationship between an individual and the team crystalizes. At the same time, and as a part of the same process, a structure for how decisions are taken is created. The sub-triangle in the lower right corner, depicts how the object (the code that is developed) is distributed over and coordinated between the team members in an on-going process. In the work of Engeström (1987) it is labelled *distribution*, but is here renamed to *coordination* to emphasize the coordinating structure of control. Finally, the *development* triangle characterizes the subject's efforts to transform the object, an effort that takes place within and with the assistance of the community. Thus, the sub-triangle in the centre, delimited by the subject, the object and the community is visible as well.

A particular feature of this category is the strong position that is held by the team, both in the creation of rules and in the distribution of control and labour. Alec reports in the quote on page 146 about the open communication that offers possibilities renegotiating the rules, while the main point in

Albert's statements (page 147) is how control is shared. The decision taken concerning the forms for the collaboration, as well as the implementation of the control over the object, have thus been demonstrated to be performed jointly by the members, orchestrated by a team leader with a coordinating function. Another characteristic attribute of this category is the subject's experience of himself as *being a part of* the structure of control. In figure 14 this is illustrated by the dashed arrow. The relationship between the student and the object is thus not only influenced by the team, but also by the distribution of control. This picture of the structure of control suggests, both in the absence of visible contradictions and with its capacity to cope with changes, that the category under investigation is harmonious and functional.

1.6. Case studies

Now, with the two phenomena, team structure and distribution of control, analysed and documented, the experience of being a team member can be described in terms of the categories obtained. An individual student's experience of these entities transforms continuously, as he, the team and their relationship evolve over time in the changing environment. Furthermore, evidence indicates that students can simultaneously experience the entities in two (or more) different ways.

Case 1

Staffan, Samuel and Allan, members of the same team, have described their team as being constituted by two halves (team structure, category 2). When the distribution of the work is discussed, here by Allan, who shares the leadership with Staffan, the picture gets confirmed:

Interviewer:	That is how you have divided it between you, and if you look at the whole group how, how is the division? [...]
Allan$_1$:	Working on every little thing we're doing as well. He's, doesn't really have a leadership position in any one part 'cause we only have three milestones
Interviewer:	So, but you don't have a very clear division of the tasks between you?
Allan$_1$:	Um...
Interviewer:	Except for the cameras in Sweden.
Allan$_1$:	Yeh, and the documentation in..
Interviewer:	Yeah, and the documentation here.
Allan$_1$:	And in most parts we're going to be working on the applet in Sweden and we'll be working on our game server here. Because while they're working on the camera we're working on our algorithm with getting the ball around and it seemed like a natural division. If we worked on the navigation of the ball we should also work on the game server, overall as well. And then let them work on the applet because that seems to have less work that actually needs

to be done from what we actually want to accomplish in improving what we were given.

The sub-teams, not the individuals, are in the centre even when specific tasks are discussed. Although he is the leader of the American half, he is hesitant about some of the tasks for US students, and tells the interviewer that the work to a large degree floats between individuals.

Later during the interview, Allan describes the distribution of control, as he experiences it:

Interviewer: What are the responsibilities of the team leader, would you say?
Allan_1: Um, telling Greg who's working at what and directing questions from what Greg has, to the correct person. So, um, you've got to have an idea of who is doing what at what times. So, it's not that hard to do.
Interviewer: Did everyone in the group agree on this?
Allan_1: Um, yeh.
Interviewer: Or is, is there someone want.., having a different view about what the group leader really should do?
Allan_1: Um, not from what I can tell, for the most part. Um, if someone wants to say something I just let them go on and, like, we go from there. So it doesn't really make too much of a difference to anyone, I think, who's the lead person. Um, we're not really too control hungry or anything.

The role of the leader is to coordinate. Initiatives from other team members are encouraged. The communication within the team is normally unproblematic, as is here pointed out by Staffan:

Interviewer: But in general you would say that communication within the whole group has been OK?
Staffan_2: Yeah, it has gone pretty well, yeah, obviously they had a little, well they complained about a guy, Philip, in the US because he never turned up to lectures and so forth. And then they said that he didn't answer mail sometimes, he worked quite a lot as I said.

Staffan's statements are confirmed by Samuel. The team is characterized by category 4 ("Control is distributed"), together with category 2 of the team structure ("A team consists of one half at each university")

Certainly, this team shows some uncommon characteristics, both with regard to the research process and to the characterisations of the teams that are presented in this section. Firstly, it is the only team from which I have interviewed three students. As was noted on page 82, the students who participated in this study were selected on criteria other than to which team they belonged. Secondly, the statements of the three informants are not only congruent between each other, but the different statements of each informer also show strong coherence during the interviews. In only a few cases during the interviews they turned their attention to individuals, and then only when coding or the results of coding efforts are mentioned. The full team was only discussed when called for by the situation, as when deciding a team leader. For this particular team, the results can thus be taken as evidence of a shared experience of what it means to be a team member, and possibly even as an indication of how the team "is".

Case 2

A different situation is described by Sven, who is not a team-leader. The team, however reduced in size to be a subset of its original size, is the fundamental entity. This experience is characterized in category 3 ("A team is a single unit"). In the full interviews, a complementary picture evolves, where Sven also presents the actions of the different individuals, and describes their contributions to the work. Here he describes an experience that is categorised in category 1 ("A team consists of smaller entities, individuals or sub-groups").

The quote on page 143 illustrates a situation in a team where the control is undefined (category 1) and ad hoc constellations take control. His idea about how the team ought to function is different:

Interviewer: So what do you think the group leader's role is?
$Sven_1$: Yes, if we say... think that the group leader should be the person who has overall responsibility as we have now, then, that the web site is up to date and see to it that everything is going as it should. For example, the group leader should make the list of who was going to be in charge of each milestone report [...] Then I think that for instance when one has an IRC meeting it's clearly the case that people are not always able to be there, and then I reckon that the group leader should see to it that even if they don't do it themselves that someone sends out a mail [...]

A team leader with a coordinating function (category 4) is here present in his discussion. In summary, Sven describes a situation where he experiences his influence on the team as small, the structure of controls as floating, and the team structure as consisting of individuals in the team. The situation frequently changes. He is not satisfied with the situation.

Learning from the case studies

Comparing the team that is presented by Sven, with the team discussed in case 1, two different pictures of what it means to be a team member evolve. While the first team has a stable situation with a well-defined, distributed structure of control, ad hoc decisions taken as a the result of a social game characterize the team in case 2. Being a team member, with an aim of participating in the joint construction of a software package, cannot be the same in these two teams.

The team described in case 1 has succeeded in defining, and redefining, which kind of result they aim for - their object. With the participants being active members in the activity, their negotiations are successful. In the case

2, the negotiations concerning a joint object fail. Instead, social issues and issues concerning group dynamics and individual members come to the surface.

1.7. Working in teams

The analysis performed at a collective level of particular constituents of a single category is not an analysis of a whole activity. But here, through the isolation of a particular aspect of a category, the characteristics of the particular category gets exposed. In this way, the researcher gains tools that supports him in his efforts to discern aspects that otherwise are hidden by the complexity of a situation. Particularly, he is offered possibilities to examine hidden contradictions and other features of a particular situation that participate in making this situation more (or less) functional.

General conclusions can be drawn already at this stage as the inner contradictions can be studied. The first category, with an absent structure of control, is the most vulnerable in this sense, while the fourth is the most stable.

In the context of the analysis performed above, it must be remembered that the categories are idealized descriptions, created by the researcher, and that should be interpreted at a collective level. This implies that the categories do not correspond to specific "real" teams. Instead, they serve as patterns, that help the researcher in his task to analyse the learning. Thus one can not draw the conclusion that a team, which by a participant is described as having a distributed structure of control, automatically is a functional team. Turning the statement the other way around is however possible: A distributed structure of control is one (of the) desirable structure(s) of control, and thus ought to be encouraged in the course design.

A literature review reveals some studies in the phenomenographic research tradition which explicitly address students' experience of studying or working in teams. The work of Booth and Petersson (1998), concerning team work in an introductory course for students in an educational programme that leads to a master's degree in computer engineering, is relevant in this context. In the course, learning of different ways to study, mainly collaboratively, was stated as one of the main goals. Booth and Petersson have identified three qualitatively different ways, in which learning in teams is experienced by individual students. They discerned learning (1) as experienced in isolation within a team, (2) as part of a distributed effort, (3) as part of a collaborative effort. Contrasting their results with the findings in this section, we find that more complex and differentiated ways of experiencing the function of the teams are described in this study. This indicates a more developed perspective of the more advanced students in the current research.

The analysis performed in this section resembles in many senses that which is proposed by Engeström (1987) and others in the same tradition. The differences are however important. The analysis in this chapter is built on experiential data, and does thus not describe how the situation "is", but focuses instead on the different ways in which the situation is experienced. The physical environment, being so important in activity theory, is only implicitly present here. In the same way, the historical background only manifests itself indirectly in reported experiences.

The results presented in this section are well in line with the findings of Hause (2003), earlier discussed in chapter 3. She has shown that high-performing Runestone teams more often have decentralized communication patterns, where communication links are open and information is disseminated. The decentralized structure is thus more desirable than a centralized communication structure. Furthermore, low performing teams had more conflicts than the high performing teams. Building on Belbin (1996), she distinguishes between *team leaders* (who delegate, seek talent and build on diversity) and *solo leaders* (who interfere, collect acolytes and strive for conformity). High-performing teams more often have "team leader" types, while low-performing teams show both "team leaders" and "solo leaders". While her findings are built from data of the performance of the teams as assessed by their teachers, the current study goes further in that it gives insights into *why* certain ways of working are better. From such insights, conclusions about teaching can be drawn.

2. Selecting a program code to develop

The three code packages, that the teams were to choose from were prepared by the teachers using the systems that were produced by the students during the previous year as starting points. They were selected to differ with respect to a number of factors, such as the code structure, documentation style, programming languages, and computer communication solutions. The codes should also have different strength and weaknesses and offer platforms at a similar level of difficulty (Arnold Pears, private communication). I will refer to the old software packages as *codes* in this section, in order to stay close to the terminology used by the participants. Also the numbers assigned to them (1, 2 and 3) remain unchanged from the original setting. One of the codes, number 3, was demonstrated by a staff member at GVSU. Apart from this, the teachers did not offer any advice about which code to choose, or what advantages or disadvantages a certain code had over another.

What determines a team's selection of a code?

Two phenomena, related to the selection of a code were selected for analyses:

1. The students' motives for selecting a certain code.
2. The technical properties of the code.

The first phenomena relates the code to the students' experienced situation in which the project was conducted, and encompasses issues contextual to the code itself, such as time constraints or motivation to take this particular course. The second focuses on the technical qualities of the code.

2.1. What is a "good selection of a code"?

Four categories of the motives for selecting a certain code have been discerned and are presented in table 21.

Table 21. The categories of the experienced motives for selecting a certain code

	Label	Description
1.	**Pragmatic or opportunistic**	Issues not related to the learning aims are in focus in the selection
2.	**The technical features**	The technical features of the code are evaluated in order to form the basis for the selection
3.	**Strategic**	External factors, such as schedules for work and parallel courses constitute the basis for the selection
4.	**Personal**	Personal aims constitute the basis for the selection

Category 1. Pragmatic or opportunistic aspects constitute the basis for selecting a code

The idea in this category is clearly described in the following excerpt from an interview with Anthony:

Interviewer: [..] Um, what code are you working on?
Anthony_1: Ah, code 3.
Interviewer: Ja, How did you select this code?
Anthony_1: That was the one demod to us by, um, Greg, and what it seems like it was, had better documentation. It was kind of funny because when Greg first started demoing
Interviewer: Greg Staff?
Anthony_1: Greg Staff, he started demoing the code and we thought it was code 1, so we went into our first IRC meeting, being the kind of pig-headed Americans that we are, and we're kind of saying we're going to use Code 1, we're going to use Code 1, does anyone have any reasons why not? And no-one really presented anything or kind

of wanted to argue about it, but there was no points to argue. So we originally started going with Code 1. Then we went in with a meeting with Urban, that very first week, a kind of getting to know each other, and all of a sudden we learn that Code 1 is not what Greg demo'd to us. And so we pretty much picked Code 1 without even looking at the code, we just assumed that was what was demo'd and that was the one that works the best. And then we learnt that code 3 was the one that was actually demo'd and that was the one that works the best, better documented after looking through all of them. And so we ended up going with Code 3 and no-one had any complaints for it.

Interviewer: Do you think it worked the best?

Anthony_1 I think it did.

The interpretations of the staff member's demonstration, the numbering of the code, or the selection of other teams are taken for indications about the quality of the codes and comes to serve as a guideline for the selection. A factor, that is neither related to the course itself nor to computer science, is here in the centre of the attention, while computer science and course objectives are relegated to the background. The aims of the course are "by-passed". In this way, the category represents an opportunistic choice. Clearly, it is not a desirable category from a computer science perspective.

Category 2. The technical features of the code constitute the basis for selecting a code

The technical assessments become the foundation for taking a decision about which code to work on. Allan, for example, points out a particular feature as the foundation of the decision:

Interviewer: How did you take the decision then?

Allan_1: [...] I really, I really didn't care one way or the other which code it was, there is pretty much the same idea going through so it really didn't make much difference to me.

Interviewer: Ja, ja

Allan_1: .. it didn't make much difference to me. And I'd feel that most of the team members had the same feeling as me.

Interviewer: OK. You wouldn't have felt bad if the choice would had been another of these.. O.K.

Allan_1: Right, and I think the major decision was the RMI factor.

Which technical aspects of the code that the students focus on is the theme for the next section of this chapter. What this category captures is that a technical feature can serve as a guide for a decision.

Category 3. Strategic factors constitute the basis for selecting a code

Abraham stands as a representative for the third category:

Interviewer:	You hadn't looked into, into the strengths of the other two codes?
Abraham$_1$:	No, we didn't feel we had much time to, to look into their strengths very closely. So we really needed to pick them quickly and get, that's the feeling that we kind of all had, that we had to really get at it quickly.

This category sees a wider focus that also includes the experience of the life situation of the student, where time, in the light of competing courses, jobs, and social life, becomes crucial.

Category 4. Personal factors constitute the basis for selecting a code

This category is characterized by a wish to let personal preferences, such as aims for taking the course, form a decision criteria. Alec formulates this in the following way:

Alec$_1$:	Yeh. And we had chose code 1 because we liked the organization of how um, they created a package, and um, I myself would rather work in, um.. I decided we.., suggested we work in Java because my two other group members, not knowing Java very well, as an opportunity for them, and Java lends itself as easy, easily readable.
Interviewer:	Yes.
Alec$_1$:	So there was that opportunity there, and the fact that it used RMI interested me.
Interviewer:	So these were actually your arguments for choosing number 1?
Alec$_1$:	Yes.

To learn about or practice his knowledge of RMI, as well as the possibility for his team mates to learn more Java, is presented as a feature of the selection of code 1.

This category focuses on the reasons for the individual to take the course. The outcome of the project work (learning computer science, or the pleasure of working in an interesting environment) is present and points towards a future situation, in which the result of the project work is available.

Discussion

An hierarchical structure of categories 2 – 4 can be discerned in the widening focus and inclusive structure. The first category can not be related logically to the other categories, as it is not based in computer science and the course situation. The situation is summarised in table 22 below.

2.2. Technical features of the codes

The code serves as a starting point, or raw material, for the students' project. Thus, its technical properties must be considered by the students.

Four different categories describing technical properties have been discerned, each focusing on a particular feature or a "blend" of features. Categories of this kind, where each category highlights a specific issue, do

Table 22. Summary of categories of the relationship between a student/team and a code in a learning situation

	Label	Nature of relationship	Focus	Background	Relevant for course aims
1.	**Pragmatic/ opportunistic**	Pragmatic aspects constitute the relationship	Interpretations of statements about the code	Interpretations of social rules	No
2.	**Technical**	Technical features of the code constitute the basis for the relationship	Evaluation of code features	The project	Yes
3.	**Strategic**	Strategic decisions based on the experience of factors in the environment constitute the relationship	Time constraints, parallel courses and other environmental factors	Study situation	Yes
4.	**Personal**	Personal factors constitute the relationship	Aims for taking the course	Personal interests and motivation for studying computer science	Yes

not represent a phenomenographic outcome space, but are important for a reader who wants to explore how the students understand a specific computer science topic. The results concerning technical properties are summarised in table 23.

Table 23. Categories of relevant technical properties of the code

	Description of category
1.	Behaviour of the program that is represented by the code
2a	Documentation as a feature of the code
2b	The programming language(s) as a feature of the code
2c	Readability, or layout, of the code
3.	The structure that underlies the code
4.	Technical quality is based on a compound judgement of several features

At the beginning of the course, the codes were made available for the students to down-load. They could not only read the codes, but also compile and test them. With the different qualities of the codes and their documentation, as well as the modifications made by the teachers, the results of this endeavour (whether the compilation and the test, only the compilation, or neither of them, succeed) varied between the teams and with the different codes.

Category 1. Behaviour of the program that is represented by the code
The technical feature that characterise this category is the focus on the results of tests of the codes on the hardware, the Brio-boards. The following statements from the first interview with Adam illustrates the category:

Interviewer:	You said you selected code number 3.
Adam_1:	Right.
Interviewer:	Can you tell me about this code?
Adam_1:	Well, um, it was, the one that was working the most, of the codes we had available. [...]

Focus in the first category is on the results of testing a code, that is, on the behaviour of the programme that is represented by the code. To evaluate if the movements of the board are good or not, a comparison is needed. The behaviour of the board, when the code is running, is then compared to the behaviour of an ideal programme. This idealized programme, although not physically present[66] can be experienced as a frame of reference.

Category 2. Textual features of the code
The second category studies and analyses the technical characteristics of the codes, both the programs and the documentation, as texts. The words and symbols that it is composed by (rather than the behaviour of the program that the symbols represent) are studied. Documentation, the programming language(s) used in the codes, and the readability, or layout, of the code are discerned as important issues. In many of the interviews, the students referred to two or the three of these issues.

Subcategory 2a. Documentation
The value of the documentation is visible in the following discussion between the interviewer and Abraham:

Interviewer:	Which code are you working on in your group?
Abraham_1:	Three.
Interviewer:	Um, why did you select this code?
Abraham_1:	Um, it was the best documented. That was really the main reason, and we could get it to run, that was pretty nice. But it was just really, really well documented and that was our main reason for doing it.

66. During the years of the Runestone course, on one occasion one team has been close to solving their task according to the specifications. The remaining teams have, without exception, achieved working prototypes, with different levels of perfection (A. Pears, private communication)

Interviewer: What are the advantages and disadvantages? Except for, except for the documentation.
Abraham$_1$: Um, you mean the advantages of Code 3 over the other ones?
Interviewer: Ja, and the disadvantages compared to others.
Abraham$_1$: Well I don't know, 'cause we really just based it on documentation. We got it to run and then we saw..
Interviewer: Ja, you got it to run before you selected it?
Abraham$_1$: Correct. Yeh, we tried the other two. We couldn't really get those to run, 'cause we couldn't, there was no documentation really to tell us how to run it. So once we got three to run we realized, you know, we don't have much time on this project. We really have to get familiar with it really quickly. So we all decided hey, lets go with three, we'll dump the other two and that's what we did. Just based on the documentation.

Although the interviewer encourages Abraham to discuss other issues than documentation, his story about how the code was selected remains a story about documentation. He particularly points out that the absence of documentation in the other codes, hindered the team from testing them.

Subcategory 2b. The programming language(s) used in the codes

Samuel puts the emphasis on the choice of a programming language:

Interviewer: Why? What's the reason for number two?
Samuel$_1$: Ah, that's mostly since some of the code is in C and there was one of the Americans who thought that he was good at C.

A focus on the programming language is by far the most frequent issue in this category and is mentioned by a majority of the interviewees.

Subcategory 2c. Readability, or layout, of the code

How the code "looks" in the sense of readability and simplicity is mentioned in this sub-category. When the issue is explicitly raised by the interviewer, Samuel presents why he finds code 2 disorganised:

Interviewer: Was the messy? That can be the case sometimes.
Samuel$_1$: Yeah, especially, these, sort of motion things. We had a file with some six, eight hundred lines of code and there were parts where one thought, sort of, what on earth is this, we didn't understand a thing huh, and then there were some parts of the code which were not needed either [...] there were some areas that were confused and we had no idea how they worked. Honestly one couldn't say that we really understood all the code, no, that one could not say at all

By stating that some parts of the code were not needed, while others were disorganized, Samuel makes clear, that his judgement here is about the layout of the code. Staffan argues for the importance of the layout in another way:

Staffan$_1$: Number two was pretty simple, probably it was the one that worked worst, of the codes. But it was the easiest to read and understand, so we took it.

Since code 2 was easy to read and understand, it was the choice of the team, although the other codes worked better when tested. The readability of the code is important.

Category 3. The structure that underlies the code

This category goes further than just analysing the code (or the documentation) as a text. The design ideas underlying the code, expressed in the structure of the resulting program, with such properties as modularity and relationships between classes and objects are in focus. This standpoint is clearly vocalized by Albert in the following way:

Interviewer: O.K., O.K. Which code are you working on?
[...]
Albert$_1$: Um, we're working on Code 1. Uhum.
Interviewer: O.K.
Albert$_1$: We felt that it was, um, better.. We looked, with a program called BlueJ, it looks at all of the different classes and shows the relationship between the classes, which class uses which class and ?? And we compared them and we felt that Code 1 had a better design structure than Code 2 did. And so, um, without too much, um, work on the codes, like, it took us a long time to actually get the codes to compile and to run on the Brio machines. We chose Code 1 before we did that and it seemed to work out pretty good.

Albert refers to their use of BlueJ. It is a programming environment for Java, intended for educational use. BlueJ offers several features to its user, among which are the possibility to display a graphical representation of the class structure of a Java program. The initiative to use a tool as an aid to analyse the class structure stems from the students, and has not been proposed by their teachers. In the interview with Sebastian the structure of the code is also discussed:

Interviewer: Yeah, you said that you use code number three.
Sebastian$_1$: Mm...
Interviewer: Why?
Sebastian$_1$: Yeah, I was sold on it because it had the possibility...one could swap navigation algorithm directly, and I though that was cool, and then it felt like it was pretty clear, and there was, yeah, there was ah well one can't really say, it was also pretty confused, with files everywhere, here and there, but it felt like they had thought about things a bit anyhow [...]

Although he mentions the code as “it was also pretty confused” concerning the files, he still argues that “they had thought about things a bit anyhow” and says that the navigation algorithm was an independent module. In this way he makes clear that there is an underlying modular structure of the code that pleases him, and that not should be confused with the layout (that is messy).

The underlying ideas, that are found by interpreting the code, manually or through the use of a tool, is in focus in the third category. The structure, that represents these ideas, are compared to an idealized structure, that comes to represent the background, against which the properties of the structure of a certain code stands out and can be judged.

Category 4. Technical quality is based on a compound judgement of several features

While the first three categories focus on one single technical feature of the code, category 4 contains multiple foci. The different foci are brought together to form a coherent argument in a technical judgement of a code. Axel expresses his view on code 1 in the following way:

Interviewer:	Um, how come you chose code number one?
Axel_1:	Um, we looked at all three codes, everybody in the group did, and we pretty much all agreed that Code 1 seemed to be, to us anyway. It was all in Java except for motor server and the video server, those were in C. And it seemed to us to have the most organized layout, and make the most sense to us. It seemed like the objects were distributed right and um, it was pretty clean. Some of the other codes we looked at, like Code 3, we tried to look at it and there was just.., the documentation was really bad. We couldn’t figure what the intention was of the different parts? and there was so much stuff with it that just seemed to be completely extra, that we didn’t.., we just couldn’t wade through it quickly enough, really. And Code 2 was kind of the same. It seemed pretty well organized too but it was, um, really confusing and the documentation was awful. It was really, really bad. And so we decided to focus on the one that seemed to make the most sense and it was pretty well documented, and the objects seemed to do the right things, to communicate correctly.

After mentioning the programming languages (corresponding to category 2b), he discusses the structure (3) and then turns to issues of documentation (2a). By stating that “there was so much stuff with it that just seemed to be completely extra” he touches upon the layout of the code (2c). Later in the dialogue, he tells the interviewer, as an answer to his question, that they have tested the three codes (1).

His argument creates a complete picture, clearly marked in his summary, where he again stresses the importance of a good documentation and a good structure for the communicating objects. In this way, his use of a multiple focus becomes more than the sum of the focuses that it is constituted of. The possibility of judging their relative merits, as he does in the last phrase, and thus draw conclusions based on his compound perspective, is new.

Of course, Axel is not alone in discussing several criteria for evaluating the codes. Most, if not all, students express understandings expressed in two or three of the sub-categories of category 2. Still, not all interviewees, but rather a minority, bring their analyses further to encompass several features of the code with an aim of drawing an integral conclusion.

A discussion concerning "a technically good code"

If these results are analysed in a framework of computer science further conclusions can be drawn. The level of abstraction differs between the categories. The first sub-category sees something concrete: A board that tilts – or that does not tilt – according to a certain pattern. Category 2 looks at a representation of the program that controls the board, and has thus a more abstract perspective on the code. The third category describes an interpretation of this representation, that is, it focuses upon the structure that underlies the program. Related to this increasing level of abstraction is an increasing need of expertise to understand and evaluate the computer science content seen in the different categories. It demands more of computer science competence to read and assess the choice of a programming language (category 2) than to study the movements of the board (category 1). To analyse the program structure (category 3), whether manually or by a tool, demands still more expertise to be drawn upon. Turning to the last category, in which two or three of these considerations are taken into account to form a coherent judgement, a different picture evolves. This category is clearly more complex, since it reflects a simultaneous awareness of different aspects of the topic, and thus a more advanced way of experiencing it. Three variables can be identified within the category: (1) the number of features of the code that are taken into consideration in the coherent argument, (2) how the quality of a particular feature is evaluated, and (3) how the importance of the features are evaluated in relation to each others. A summary of these findings can be found in table 24 below.

2.3. The results come into play - seeking contradictions

The analysis of the students' relationship to the codes shows an important differentiation between the four categories.

Category 1. Pragmatic or opportunistic aspects constitute the basis for selecting a code

This category describes a situation in which the aim that a student strives for differs from the learning goals defined by the university. When discussing

Table 24. The categories for the technical properties of the codes

	Description	**Referential aspect**	**Structural aspect**			
	Relevant technical properties	**Meaning**	**Which technical feature is in focus?**	**Background**	**Variation lies in**	
1	**Program behaviour**		The results of testing the code	Ideal results of a test	The different results	
2	**Choice of programming language(s)**	A single feature is highlighted to judge the code	Programming language(s) used in the code	Programming languages known	The class of programming languages that could be used	A single feature in focus
3	**Underlying program structure**		The identified program structure in the code	Ideal program structures	The different program structures	
4	**Based on a compound judgement**	Quality of a code is determined on a coherent argument based on several features	The selected features, their qualities and their relative values	The codes seen as a whole and related to ideal codes	1) The features that are taken into account 2) The quality of a particular feature 3) The relative value of the features identified	Multiple features in focus

the tools, two perspectives can be identified: While the students see the social environment (or at least the teachers' acts) as tools for their choice, the universities certainly would prefer the students to identify and deploy other tools, related to their competencies as computer scientists. The community node in an activity system comes in a similar way to possess a dual function: At some moments, the teachers must be considered to be a part of the community, offering words that could be interpreted. At other occasions, they must be excluded from the discussion to make the different interpretations of their words possible. Not only does this situation create contradictions within the nodes, there are also contradictions between nodes. The role of the universities, to serve as a tool and to be a part of the community at the same time is untenable. In similar ways it can be argued that there are contradictions in the object node and in the tool node. The situation is illustrated in figure 15 below, where the flash arrows depicts contradictions.

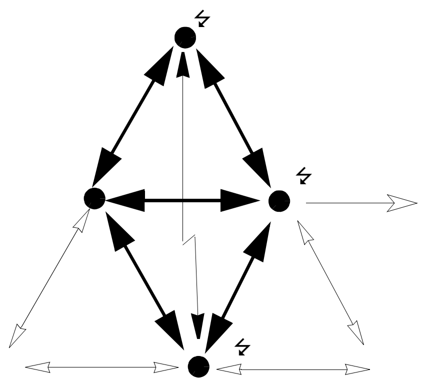

Figure 15. Relating the category *Pragmatic or opportunistic aspects constitute the basis for selecting a code* to an activity system. The meaning of the flash arrows are described in the text.

Category 2. The technical features of the code constitute the basis for selecting a code

In contrast to the first category, in which many contradictions were present, we here find a category that describes a harmonious situation. A student here strives to find a good code (as is indicated in the object node), deploying his competence within computer science and the project as a tool for his judgement The upper sub-triangle of the activity system (see figure 16), labelled *production* is here complete. This indicates that the student

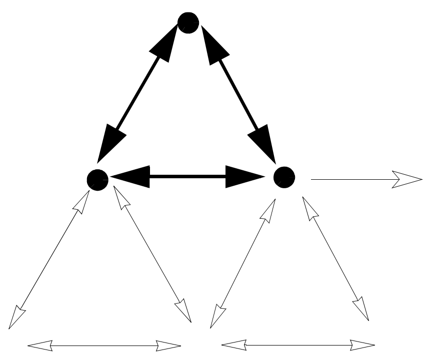

Figure 16. Relating the category *The technical features of the code constitute the basis for selecting a code* to an activity system. The student here selects a code, using his competence in computer science as a tool for his efforts.

experiences a working relationship in his work to make, or produce, a selection. The production of the choice is not only taking the code into account, but is mediated by his intellectual tools offered by computer science.

Category 3. Factors in the environment constitute the basis for selecting a code

As in category 2, a student is here trying to find a good code, and is working in a harmonious situation. The key difference is that the view of the code is here related to the student's full study situation, rather than to the technical features of the code. This is illustrated in figure 17, where the study situation is depicted in the rules node (indicating both rules for what needs to be done in the project, and rules that are implied by a daily life, for example working hours for a student who finances his studies through a job), community rule (both the project team and broader circles, such as family, that the student considers) and the division of labour node (both division of labour within the project and within the broader circles in which the student acts). The three bottom triangles, labelled *communication*, *development* and *distribution* are present. These are necessary for the success of this particular form of the process of selection, since interaction with others, both in the form of a working communication and a meaningful distribution of work and responsibility are needed for the development of the situation.

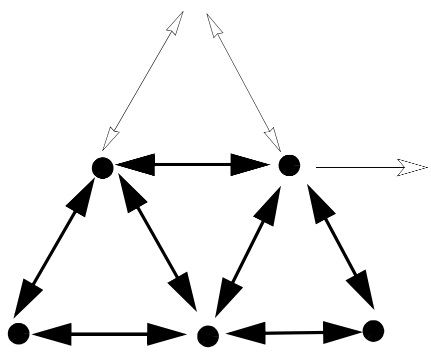

Figure 17. Relating the category *Factors in the environment constitute the basis for selecting a code* to an activity system. The rules are extended beyond the current situation and also include social obligations. The team, as well as a working division of labour, is needed for the situation.

Category 4. Personal factors constitute the basis for selecting a code
In this category (see figure 18), we find a situation where a student's objectives with his studies forms the ground for choosing a specific code. When depicting a situation of this kind on the triangle, clearly the student himself, as a subject who wants to achieve something and the object, the motives for his efforts, are present. But the situation is more complex than can be described in these two nodes. To create the full picture of the situation, it is not sufficient to consider the aim (in the object node), but also the results of the project work, and the benefits that can be gained by doing the project, must be considered. For this reason, the outcome or the activity is also highlighted figure 18. Other factors are relegated to the background. The absence of highlighted sub-triangles emphasises the personal character of this category.

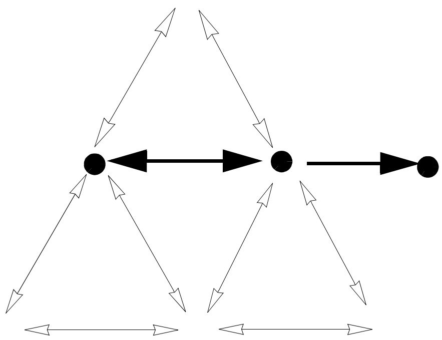

Figure 18. Relating the category *Personal factors constitute the basis for selecting a code* to an activity system. Here the object as well as the outcome of the efforts in the activity, in the form of personal benefits are highlighted

2.4. Selecting a code

The first category of the experience of selecting a code, *Control is missing*, shows, when studied in the light of activity theory, itself to be problematic with its many contradictions. The remaining three describe ways of experiencing the selection of the code, that are harmonious. Figure 16 and 17 illustrate well the difference between the second and third category, where focus is on the code itself on the former, while in the latter it is extended to the students' full study environment.

The fourth category stands out as different from the others, in that the category is not centred on the code itself, but instead considers a desirable outcome of the course. Phenomenographic research has shown that students who search for meaning and/or strive for an aim (see summaries in Ramsden, 1992 and Marton & Booth, 1997) tend to learn better than those who have a more fragmented or restricted view of their learning. Still, with the broad and varying contexts in which the previously mentioned studies have been carried out, it can be assumed that their results also are relevant for this project and that the fourth category is the most desirable.

3. The weekly milestone meetings

The milestone meetings were the only scheduled teaching occasions during the course, and served as the main communication medium between teachers and students. The seven meetings were conducted over the Internet, using IRC, and lasted for 30 minutes. The aim was, according the course web-page, to "present the work that they have done in the last week"[67].

The rather weak formal description of the meetings gave openings for different interpretations. The two teachers came to carry out their meetings in rather different ways. Since each team was assigned an individual teacher, either in Sweden or in USA, this resulted in the situation that different teams encountered meetings in different forms. Stig explains the difference in the following words:

Stig$_1$:	[...] We saw that Urban here and Greg in the US, they have different... they have half of the groups.
Interviewer:	Yeah.
Stig$_1$:	But they have different approaches, milestones mean different things, 'cause they different, for different groups. And this is a bit confusing now
Interviewer:	In what way do they mean different things?
Stig$_1$:	So, for Greg, he thinks we should have three milestones and that there are, sort of, things that we should achieve, changes to the code that we should do, there are three things so that it shall be fair that everyone does the same in all this. So that is like, when we have gotten there, we also have milestone meetings and report on changes and so forth, but Urban he has a milestone every week more or less, with smaller sub-goals, sort of.
Interviewer:	Yeah.
Stig$_1$:	So if one has him, then you get, like, 10 milestone reports [...]
Interviewer:	But Greg has three?
Stig$_1$:	Yes, Greg has three.
Interviewer:	But do you have weekly meetings with Greg as well, or is it just Urban who has weekly meetings?
Stig$_1$:	No, we have a time with Greg every week, but, yeah, but last week we didn't have any type of reporting with him or anything, but he came to the meeting anyhow. [...]

67. http://www.csis.gvsu.edu/class/brio/Management/staff.html

3.1. The role of the milestone meetings

The analysis has revealed four ways, in which the role of the milestone meetings are experienced. These are summarized in the following text and in table 25.

Table 25. Summary of the function of the milestone meetings

	Label	The purpose or meaning of the meeting	What is in focus?	Communication
1.	**Meetings are taken for granted**	The meetings are a part of the what it means to work in a project	Not an issue	Not an issue
2.	**Meetings have a value of their own**	Meetings offer a structure by their pure existence	The existence and rhythm of the meetings	Unidirectional, presentation by team or demands from teacher
3.	**Meetings concern the project**	The meetings are for communication concerning the project	The discussion about the project	Bidirectional, discussion concerning the project
4.	**Meetings support a sense of responsibility**	The meetings serve to develop a sense of responsibility	How the meeting affects the participants	Bidirectional, discussion aiming beyond the project

Category 1. The meetings are taken for granted

Staffan$_1$: Isn't that the way the project is constructed, I thought so?
Interviewer: I was asking you?

This interview excerpt from the first interview with Staffan, summarizes this category in a clear way. The meetings are taken for granted. What purpose they are intended to serve, and if they come to fulfil any purposes, are not questioned. The meetings just "are".

Category 2. The meetings have a value of their own

In contrast to the situation described in first category, the milestone meetings are here understood to influence the work of the teams. They offer a structure, or a rhythm, for the project. Understood in this way, the communication is directed either from the students with the aim of the students being to answer the perceived requirements of the teacher, or in the opposite direction, with a teacher giving instructions. In either case, the communication is uni-directed with a content that is less important than its mere existence.

Category 3. The meetings concern the project

In this category communication between the team and their teacher concerning the project is in the fore.The dialogue is what characterizes the third category. Here technical or organisational issues of the projects are discussed. The teacher listens to and guides the students in their work with the project.

Category 4. The meetings support a sense of responsibility

The effects of the communication in the form of a sense of responsibility become a key feature in this category. A sense of responsibility for someone or a sense of responsibility for the project is here the result of the discussion. The discussions thus aim further than simply focusing on the project.

3.2. The results come into play - seeking contradictions

Elements of activity theory have been used to advance the analysis for the previous topics discussed in this chapter. A similar pattern will be followed in this section, when discussing the milestone meetings.

Category 1. The meetings are taken for granted

Relating the first category to the proposed framework is difficult, since that which is taken for granted (here the meetings) cannot consciously be experienced. Still, the category is relevant to discuss in terms of activity theory. To have regular meetings is an implicit rule that exists in projects. This is illustrated in figure 19. The absence of complete sub-triangles

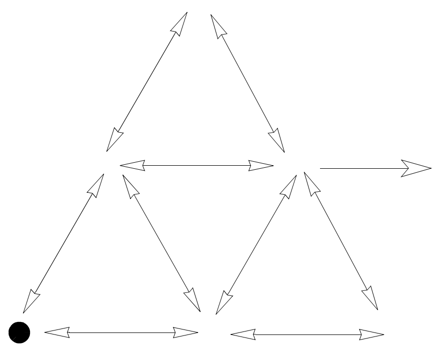

Figure 19. Relating the category *The meetings are taken for granted* to an activity system. When the meetings taken for granted, the meetings are experienced as a rule, and are not related to the remaining parts of the activity.

clearly indicate that the meetings, experienced in this way, do not add any contributions to the work of the team.

Category 2. The meetings have a value of their own

The second category is distinguished by the presence of the team and the teacher, but by the absence of a meaningful two-way communication. Relating this to an activity system, a situation where the student himself is present arises. His team, and the rules that are followed are important for understanding the situation, and are therefore highlighted in the model. Figure 20 below summarizes how this category is related to an activity system.

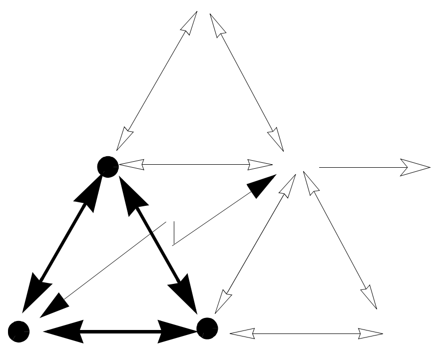

Figure 20. Relating the category *The meetings have a value of their own* to an activity system. The flash arrow indicates a contradiction between the rule node and the object node. It illustrates that the meetings exist, without being related to the object of the activity.

The triangle in the left bottom corner, *communication*, is complete in this instance of the model, which here indicates that a forum for communication (the milestone meetings) has been established. It is also worth noting that this triangle exists in a "vacuum", without being related to the other nodes, particularly not to the object node. This illustrates that milestone meetings exist "for their own sake", and that they, in themselves, do not serve to develop the object through that which is said.

In the data, utterances expressing frustration concerning the milestone meetings can be heard at a few occasions. The complaints are related to the rules for the meetings: some voices express that the rules are inappropriate. The model inspired by activity theory can shed light on these problems. This category does not describe an object that is jointly agreed on by all participants, teachers and students. Since the object is that which defines the activity, and that which unifies the participants, the model indicates that the participants might (or might not) experience different activities and thus "need" different sets of rules. This contradiction is marked with the flash arrows in figure 20.

Category 3. The meetings concern the project

An open discussion concerning different aspects of the code development is in focus in the third category. Two important differences compared to the second category are visible in figure 21. Firstly, the meetings now also serve the function of being a tool, that facilitates the team's effort to develop the code, since the discussions with the teacher are experienced as useful or productive. Secondly, the meetings serve an object, namely to transform and develop the code. The joint perception of an object gives meaning to the meeting and makes the participants strive for a common goal. In the data underlying this category, there are no evidence of any conflicts, an observation that fits well with the object being present and defined.

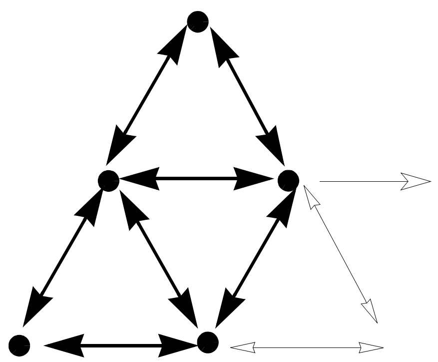

Figure 21. Relating the category *The meetings concern the project* to an activity system. The meetings serve as rules for how to work, tools for reaching the object and objects as they transform the situation and the code.

Category 4. The meetings support a sense of responsibility

The fourth category shares most characteristics with the third. There is only one difference, but this difference is vital. The discussions, as they are perceived in the fourth categories lead further than to the code that is to be developed, and influence the developers as well.

Figure 22 below illustrates this new situation, where the outcome is highlighted. The definition of the outcome is, as of the object, problematic in activity theory (see chapters 5). Factors such as international experience, learning for professional life or personal development are in different ways included in the students' aims. The outcome of the work becomes not only the transformed code, but also the personal development. This can be illustrated by highlightening the outcome component in the activity system.

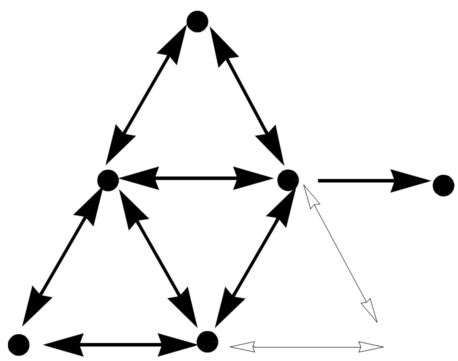

Figure 22. Relating the category *The meetings support a sense of responsibility* to an activity system. The highlighted outcome node symbolises the personal development that is a result of the activity.

4. Being graded

Grading and being graded is something that most, if not all, students have strong feelings about. In project-based courses the issue of grading is more complex, since the grade of the individual cases is related to the performance of the whole team. In the Runestone course, these matters are brought to a head, as the full team never meets face to face. Furthermore half the team members never physically meet their teacher (see Chapter 2), and are thus graded by a teacher whom they do not "know" and who does not recognize their faces.

The two university systems, with different rules, grading systems, cultures and attitudes towards grading constituted another intricacy in the Runestone environment. While the American students are awarded letter grades (A – D as passing grades, and F for failure), the Swedish students receive "pass" or "fail". As has been previously discussed in detail in Chapter 2, the course had its own "neutral" grading system, according to which the teachers could calculate an individual number between 0 and 100 for each student. This number is then transformed into a local grade at each department.

4.1. Researching the experience of being graded in the Runestone course

During the two interviews the students were asked about their experience of being graded in this particular course. Emphasis was put on the two different grading systems and their possible influences on issues such as motivation, collaboration and experienced engagement in the work of the team. The opening question for this theme was “How do you experience the different rules governing the course?”. If an interviewed student did not mention grading and the two grading systems, this issue was brought up by the question “You have different grading systems in the two countries. Does this influence your collaboration?” and was followed by questions concerning different aspects of the grading.

The answers obtained from the students often referred to how the interviewee perceived the feelings of his team mates at the other university. Such answers could, for example, discuss how he found that the students in the other sub-team judged the work at the site of the interviewee, and the relative importance of grading for the two sub-teams. The answer could also go one step further, in that it stated how members of a sub-team perceived themselves being perceived by the other sub-team. Two quotes from Samuel both from the first and the second interview clearly voice these considerations.

Samuel_1:	[...] But in a project like this, where several people are involved I really don’t think that it should not be so competitive as this. [...]
Interviewer:	Have you experienced this type of problem, or?
Samuel_1:	No, not really, not experienced it, in fact it is the attitude, especially on the Swedish side. We have not experienced anything really about how they feel on the American side. [...]

During the second interview, the argument is carried further. This extract should be read against the background of the concerns that Samuel expressed earlier during the same interview concerning his disappointment with the first milestone meeting that his team had with their teacher.

Interviewer:	Have you had problems or positive experiences related to things like the course regulations, uhm, different grading systems, different teachers in the different countries?
Samuel_2:	Yes, we are a little worried about having two different grading systems and so forth, because, as we noticed, our Americans are very competitive as a result of having different grading levels. [...] [...]
Interviewer:	How does that affect you all?

Samuel$_2$: Yes, well of course one doesn't want, you know, have disastrous meetings, one after the other. That's for sure. We didn't feel too good about it either, eh, to make a mistake. Of course we want to have successful meetings. But occasionally that was not really connected, connected to their concerns. After all it was how it was. And that was our attitude. That didn't really get accross to them. Really they, Americans, didn't really understand that.

The indirect reference is brought one step further by Samuel on this occasion. He expresses that he experiences that the Americans worry about the performance of the Swedes. These concerns are ungrounded, he argues, and might be based on misunderstandings.

The issue of how well "the other" sub-team works has also been a concern, in particular related to the two grading systems. Some students, most frequently on the American side, have said during the interviews, that everyone in their own team, independent of their location, worked well, but that they had feared or heard of problems in other teams. Such a view is pronounced by Adrian in this way:

Interviewer: There are, there are different grading systems in the two countries.

Adrian$_1$: Yeh, we found that out on Friday when Urban was here. He told us that most of the classes are credit/no credit. And over here we actually have an A, B, C or D or..

Interviewer: Do you think this affects your collaboration?

Adrian$_1$: I haven't seen it in our group, but I've heard a couple of other groups complain that they seem to be, the Sweden students seem to be trying to do the bare minimum so they can get the credit. Whereas um, American students have to push a little bit harder in order to get their A. But I haven't seen it too much in our group. [...]

To address certain issues of grading, it needs to be considered to what extent the statements about "other teams" should be interpreted as hidden messages concerning their "own" team, or if such statements describe hypothetical situations rather than real problems. It is here worth noting that only one of the interviewees has complained that his team-mates at the other university work too little, or too much, as a result of the different grading system. The student in question was a member of an ill-functioning team, and was angry during this part of the interview.

Not only are there many indirect references and discussions about hear-say in the material, but there are also unspoken, taken-for-granted assumptions about what grading "is" and "means". For example, the material indicates that terms like "career" and "competition" might have different values in the two countries and form different collective contexts (see page 59).

In order to analyse students' experience of being graded in such situations, it is thus important to formulate questions where aspects of grading come to the fore with the cultural and societal aspects coming to serve as a background. One such relevant research question, or phenomenon, which has been discerned in the data, can be expressed as *What is the purpose of being graded in this course?*

4.2. What is the purpose of being graded in this course?

Three qualitatively different ways of experiencing the purpose of being graded have been discerned. These are summarised in table 26.

Table 26. Categories of the experience of being graded.

	Label	Description
1.	**Grading is an aim in itself**	Getting a good grade is important, since the grade has a value of its own
2.	**Grading is an instrument for future life**	A good grade helps to advance a future career, and is for this reason important
3.	**Grading is subordinated to social obligations**	Other obligations in this course, concerning my relation to the team or the team in relation to other teams are more important

Category 1. Grading is an aim in itself

In a discussion concerning problems that could arise from the course regulations, Adam expresses the following view:

Interviewer: Course guidelines? With grading, for example?
Adam_1: I guess um, everybody has concerns about the grades. That's something that has been brought up quite a bit in the class, you know, not just in my team, but in, among everybody that, um, if things, if the code doesn't work but we've worked really hard on it and had to get through all these team issues and stuff, are we still going to fail the class because our code didn't work. Um, that's something that there is a lot of concern about and I, you know, personally I'm not concerned about it myself, but I think that it is something that ought to be addressed in the future, you know

Grading is here presented as something important, that everyone is concerned about. For this reason, clear guidelines concerning how the grading is performed would be useful. This line of reasoning does not question the grading as such, neither does it put forward any reasons for its existence or reasons why it should be abandoned. It is taken for granted and is considered to be an aim in its own right.

Also among Swedish students such statements can be found, but only as a perception or description of which values are important within the American institution. Let us here listen to Samuel:

Interviewer: The course rules and marking scheme and so forth, in general?
Samuel_1: Yeah, I don’t know now, now we have two separate, so the Americans have one different way to grade, than... than we have here in Sweden, but we knew that. [...] Greg uses, I don’t know, uses

	the American system, I believe, he grades based on a certain scale from one, I don't know, but 1 and 5 I think he said, I don't know. After all the whole American grade system is built on competition and such and everyone should be competitive, that's how it is.
Interviewer:	Yeah
Samuel$_1$:	The best person gets the best grade

The American system is according to Samuel, based on competition. The best students are awarded the highest grade. This is not questioned or discussed, but taken as a fact by Samuel in this statement.

Category 2. Grading is an instrument for future life

Alec talks about the importance of getting a good grade during his first interview, in a discussion that is earlier presented on page 60. During the continuation of the discussion, the following dialogue takes place:

Alec$_1$:	If my class were pass/fail here, and the rest of my classes weren't graded as they are, I would not put as much effort into it. I've already noticed a lot of my classes are lacking in grades because I have to put so much effort into this project.
Interviewer:	Ja, ja.
Alec$_1$:	And that's a concern over here. How much time can you put in without it affecting the other courses?
Interviewer:	Ja, I understand that problem.
Alec$_1$:	And we all work to, it's, it's a.., time management.

In the quote on page 60 Alec argues that a good grade is a requirement for a good job. To get a good grade requires tactical considerations, he continues.

Where grading in the first category is an aim in its own right, it points in the second to a use outside the individual's current world.

Category 3. Grading is subordinated to social obligations

One of the many students who voice the importance of social obligations is Albert:

Interviewer:	But this different grading, do you think it would be um, is a factor that might be a problem for the project?
Albert$_1$:	Um, I don't think so, not in the project aspect, I think that might be disappointing maybe to hear that the Swedes or to the US that one is being graded differently to the other. But, you know, I feel that, you know, if you are a group you should work hard to the best of your ability whether you are being graded on it or not, or which style of grading.

A similar statement can be found in the interview with Stig:

Interviewer:	That does not affect how you work, does it, do you think? You don't think that you do less as a result?
Stig$_1$:	No, we, well we need to do what we have agreed on together anyhow.

Both students here refer to their obligations to their teams, even though Albert acknowledges the potential for unfairness. Being graded is overshadowed by the obligations that stem from the situation of being a team member, and of having obligations to the team.

Andy presents another factor, the final presentation, as more important:

> $Andy_1$: No, OK, that's certainly how they feel, they're worried they won't bother because they're going to pass anyway. But it's generally not a problem at home, I think, because everyone wants to do a good job and not be ashamed when they present it

Since Andy frequently during the interview refers to his the team as a unit, his statement about not being ashamed does not refer to him being ashamed in front of the team-mates, but rather for him not to feel ashamed of the achievements of his team in front of other teams. This pride is more important than the different grading systems.

In this category, a description of grades has been offered where grading is not an important factor for promoting the work of the individual or the team. Other mechanisms have been put forward as stronger, the individual's relationship to the team, and the pride of the work of the own team, when contrasted to other teams. In the interviews, the role of the grading as an instrument for feed-back has also been stressed. Grading here takes another role than being a goal for the individual, namely to enhance the quality of the work.

The purpose of being graded

Three distinctly different ways of experiencing the purpose of being graded have been discerned. Table 27 below describes these categories and highlights their differences. A relationship between the categories, in the form of a hierarchical structure, can be identified. The focus is broadening from the first category over the second to the third.

4.3. The results come into play - seeking contradictions

The analysis of the categories has been elaborated by using tools of activity theory.

Category 1. Grading is an aim in itself

The first category illustrates a situation, where the achievement of the grade itself is an aim, and the relationship between the student and the grade comes to the fore. A student has to work for the grade; the grade, in its turn, shines back on the student. However, a student does not act directly on the grade; the grade is only accessible for him through the project he is doing.

Table 27. Summary of the experience of purpose or function of being graded

	Referential aspect	**Structural aspect**		
	Getting a good grade in this course	**Focus**	**Variation is discerned in**	**Background**
1	is an aim in itself and has a value on its own	The grade *per se*	The different grades	Expectations and ideas about own performance
2	is an aim, since it is a tool to reach other aims as making a career	The value and benefits of a good grade	The different grades and the following openings that good grades offer	Own future life and personal career
3a	is subordinated to other aims	The value of a good grade judged against the concerns of my relation to my team	The different issues that are important in the course	The project or the course as a whole
3b		The value of a good grade seen judged against concerns of my team in front of other teams		

Thus, with this perception the experience of the grade is mediated through the project, making the project here serve as a tool for “grade-production”. This situation is illustrated in figure 23.

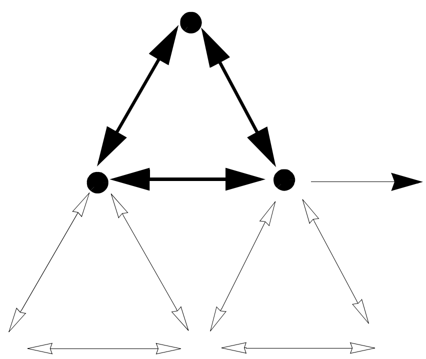

Figure 23. Relating the category *Grading is an aim in itself* to an activity system. The project serves as a tool for obtaining a good grade

The category does not in itself contain the seeds of any contradictions. Studied in isolation, it is stable and harmonious. As is indicated by the full upper sub-triangle, the system serves as an isolated unit for the production of a grade.

Category 2. Grading is an instrument for future life

Many similarities can be found between the first and the second categories, based on the project being experienced as serving the function of a tool. The new contribution in this category consists of the focus on the outcome of the grade, on that to which a good grade can lead. For this reason, the outcome is highlighted in figure 24, which depicts this category in the framework of an activity system.

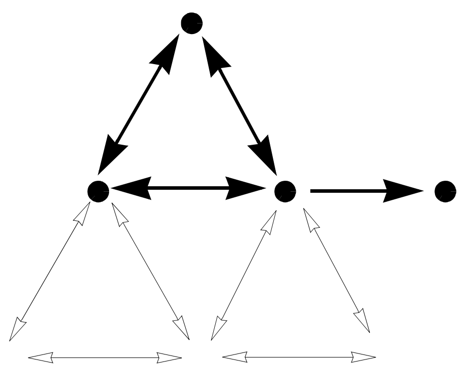

Figure 24. Relating the category *Grading is an instrument for future life* to an activity system. The outcome of the activity, in the form of a future life, is highlighted. The project serves as a tool to reach this outcome.

As the previous category, the second is harmonious, when studied on its own. It is only when this category interacts with others in a team that conflicts may, or may not, arise.

Category 3. Grading is subordinated to social obligations

Social obligations overshadow the effort to obtain a good grade in the third category. This is illustrated in figure 25 below, where the community node is highlighted to show the important role of the team. It would be misleading to compare the role of the team in this category with that of the project in the previous. While the project in the first and second category served as a tool, that, in the extreme, could be used and possibly even misused, this category discusses obligations towards team members, and is based on a sense of responsibility towards other team-members. The norms (or rules) that a subject experiences come in this way to regulate, or mediate, the relationship between an individual and his team. This category has been voiced by the students using expressions such as “work hard to the best of your ability” (Albert, cited on page 183), “everyone wants to do a good job” (Andy, cited on page 184) or other similar expressions. These statements all refer to the need to do a fair share of the total work needed in the project, that is to the importance of a reasonable distribution of labour. By studying the three complete sub-triangles: communication, distribution and development, the key components in the working process implied by the category become visible: Communication in order to relate oneself to the community, and a reasonable distribution of the work are needed for development in this situation. In the figure the output is also highlighted, to indicate that a result in the form of a development of the social situation is expected.

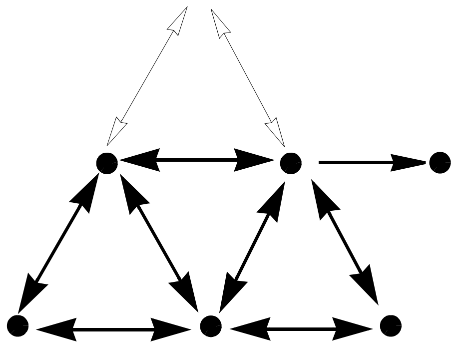

Figure 25. Relating the category *Grading is subordinated to social obligations* to an activity system. The norms (in the rule node) mediate the relation between the individual and the team. The distribution of labour is important since the good of the team is in focus. The outcome node, that is highlighted here illustrates the development of a social situation.

4.4. What does it mean to be graded in the Runestone course?

The three categories together discuss and present the students’ experienced purpose of being graded, but does not focus on the issue on whether grading is desirable or not in the Runestone course.

There have been vivid e-mail discussions among the staff in Runestone course concerning the grading. Fears have been expressed that Swedish students would have less incentive to work to the maximum of their ability (Daniels, Berglund, Pears & Fincher, 2004), because of the Swedish pass/fail grading scheme. In the debate, issues of trust have been ventilated. It has also been argued that the perception of how the team "at the other side" works is so important for the learning environment, that such perceptions might overshadow the actual contributions by the distant team mates.

According to a literature overview performed by Last (2003), research concerning virtual teams shows differences on collaboration in virtual and co-located teams. It is argued that a distributed learning environment invites misunderstandings, low trust and assumptions about the (low) achievement of the team members at a distance. In a recent study on virtual teams in engineering education, Herder and Sjoer (2003) have investigated differences in the perceived quality of the contributions, feed-back and commitment between local and non-local team members in an engineering course. They conclude, based on a statistical analysis of a survey offered to the participating students, that the students in the local sub-team are perceived as making a better contribution than their distant colleagues. Moreover, over half of the number of students reported that they did more work than their colleagues, both local and non-local.

In order to advance the debate on these issues, I have complemented the results concerning the students' experience of being graded, by investigating the students' opinions about grading in the Runestone course, the students' evaluations of their team-mates' work and the teachers' distribution of grades. The results of these additional studies bring light to aspects of the grading other than those that are visible in the phenomenographic outcome. The phenomenographic results represent a structural outcome, while the opinions about grading are value statements and the peer evaluation and the distribution of grades highlight particular aspects of the environment.

Attitudes towards grading

The students' opinions concerning the eventual desirability of being graded in the Runestone course is closely intertwined with the experienced purpose of being graded. For example, aiming for a good grade as a tool to promote a personal career, as described by Alec in category 2, certainly invites a different judgement of the value of being graded than the standpoint expressed in category 3, where obligations to the team or a personal pride overshadow grading. It can also be assumed that the two different grading schemes that are deployed in the course would lead students at the different universities to have different standpoints in this issue.

These two aspects of the grading, its purpose, and whether it is desirable or not are thus intertwined and even inseparable, in the sense that one cannot be imagined without the other, in university environments such as the Runestone course. If there is a grading system, there will be opinions about it.

Still, the two can be analytically separated, they can be "thought apart". Such a separation has been performed and has revealed the students' judgements concerning how well the grading serves the purposes of the course. The three categories listed in table 28 covers judgements concerning grading expressed of students at both universities.

Table 28. Categories of the judgements concerning grading in the Runestone course

	Description of category
1.	Grading is useful for the project
2.	Grading is irrelevant
3.	Grading is an obstacle

In the quotes presented here, the terms "grade" and "being graded" is used, both by Swedish and American students, to denote grades and grading in a fine-grained system, as the American.

Category 1. Grading is useful for the project
Some excerpts can be found where students imply that grading is useful in this particular course. Clear examples of statements stating that grading serves as feedback for the students can be found in the data. The feedback helps the team to decide what to do, since it gives a judgement from the teacher. Anthony's statement on page 60 can be interpreted in a similar way. He declares that grading has a value in the course in that it reflects the performance and thereby stimulates the team to do well.

Examples of some statements in this direction can also be found in the data from Swedish students. They argue, in different ways, that a more fine-grained grading system than the Swedish would serve the purpose of the project. The arguments are presented in indirect ways, and point to a disadvantage of the Swedish grading system.

Category 2. Grading is irrelevant
In this category, it is stated that the grading is irrelevant for the project. Certainly, many of the statements, categorised in the third category *Grading is subordinated to other aims* above, expresses the stand that grading is irrelevant, since other issues are more important. Albert and Stig, quoted above on page 183, serve as representatives for this opinion. The discussion with Albert later returns to the issue, and he then clarifies his view:

Interviewer: Ja, ja. There are different grading systems between the two different countries?

Albert$_1$: That's a little, um, I don't know if unfair is the word, it just doesn't seem, um, equal. Um, from what we understand, we understand that the Swedish are pretty much a grade, no grade on the class, and we are graded on the class. And I know that, I don't know if it would

make it any easier, but I guess it would be more comforting if there was the same or similar. I mean if they were graded, or they were grade/no grade and we were grade/no grade.
[...]

Interviewer: But this different grading, do you think it would be um, is a factor that might be a problem for the project?

Albert$_1$: Um, I don't think so, not in the project aspect, I think that might be disappointing maybe to hear that the Swedes or to the US that one is being graded differently to the other. But, you know, I feel that, you know, if you are a group you should work hard to the best of your ability whether you are being graded on it or not, or which style of

Albert argues, that it is unfair with different grading systems, without pointing out a particular team as those who get an advantage or disadvantage under the current system. As long as everyone is treated the same, it is irrelevant if there is grading. Instead, the issue of fairness is more important.

Category 3. Grading is an obstacle

Grading is expressed as an obstacle, that in different ways, and to different extents, prevents the students from doing well in their projects. This is an opinion that is commonly expressed, particularly by the Swedish students, but also by some of the American. Axel verbalizes this point clearly as an answer to an open question:

Interviewer: Anything else you want to say about the group or the project?
[...]

Axel$_1$: [...] And I don't know if it's because of us having less time [...], but I feel kind of handicapped on this project by the fact that we're getting a grade because it um, the requirements for each report have been a little bit unclear. We're not sure what we have to give and we know that behind the scenes we are doing the work, but we don't know what to show Urban, and I think it was Greg that was grading it, that we've done it. And so we're spending a lot more time than is really useful getting together reports and making sure that we've got all the stuff that we need to get a good grade, instead of working to, to make sure that we do the project and get it up and running.

Axel's comment implies that the students perform a play for the galleries. The team spends time on writing reports, that are aimed at enhancing their grades, instead of doing work that would advance the project. Samuel reaches a similar conclusion, but by arguing from a different perspective, in quotes that are a direct continuation on the dialogue reported about earlier on page 183.

Samuel$_1$: And such, I don't know that it is suitable for this type of course, in any case. I don't think that is suits this course, absolutely not. I can understand that one can have such grading in a written test and that is OK. I understand that, that is something that one can evaluate knowledge, but in a project like this one when many people are involved I absolutely don't think that it is appropriate to be competitive in this manner.

Interviewer: Can that create problems for you all?

Samuel$_1$: Absolutely, absolutely, perhaps, sort of, take those who know the most in the group. Maybe they are not happy with the grade that the whole group gets, perhaps they believe that they could do a better job themselves and get a better grade than they will get in the group, sure this will create problems, I don't know, it has to.

Samuel takes the perspective of the best performing students in the team, whom he names later during the interview. They could be disadvantaged by getting a grade that is partly based on the performance of the team, he argues, since they could have done better without their team-mates. He clarifies later, when he mentions the names of those students, that they do not seem to be concerned about this issue.

The personal advantages of not being graded are put forward by Stig. Part of this discussion has been discussed earlier on page 183.

Interviewer: What expectations do you have concerning your solution? How do you think it will go?

Stig$_1$: I think that it is going to go well, eh, I believe so [...] then, yeah, this motor bit, that's something two guys in the USA are going to do, so I know that I don't have to worry about that. And then, I think that it feels like it isn't too hard to get a pass on the course, not for us in Sweden in any case, we don't get a grade, so I think, yeah, it should not really require so much.

Interviewer: Does that affect you?

Stig$_1$: It affects us

Interviewer: Do you work in different...

Stig$_1$: No, not if one works perhaps, but how it feels to work, one is not as stressed perhaps, but

Interviewer: That does not affect how you work, does it, do you think? You don't think that you do less as a result?

Stig1: No, we, well we need to do what we have agreed on together anyhow.

Interviewer: Yeah

Stig$_1$: Which we would have done anyhow.

Stig feels less stress in the project, since he is not graded. This, however, does not affect how the team works. The obligations to the team, and promises made to others, serve as driving forces. By reading the full dialogue, it is clear that the statement "it isn't too hard to get a pass on the course, not for us in Sweden in any case, we don't get a grade, so I think, yeah, it should not really require so much" does not imply that he lowers his requirements on his own work. Instead, the comment is an answer to the question "How do you think it will go?", in which Stig states that he is convinced he will pass the course. But passing is not enough, the team is more important. In this category, the stand that a fine-grained grading is a disadvantage has been expressed. The students expressing this opinion have been arguing both from a personal perspective and from the perspective of the good of the project.

A discussion of the attitude towards being graded

The categories of the attitudes towards the grading in the Runestone course are summarised in table 29. The students shift between categories.

Table 29. The attitude towards being graded in the Runestone course

	Label	Category	Motives
1.	**Grading is use-ful for the project**	Grading serves the purpose of the course in that it encourages a good job	• Grading offers an incitement to work hard. • Grading offers feed-back • Grading shows progress
2.	**Grading is irrel-evant**	Grading is irrelevant for the purpose of the course	• Using *different* grading systems within the same team is unfair • Other aims are more important
3.	**Grading is an obstacle**	Grading is an obstacle that prevents the participants from reaching the purpose of the course	• Grading results in a play to the gallery • Grading in a group project is unfair • Grading results in a stress that does not contribute to a better result

In the column describing the motive a difference between the two standpoints "Using different grading systems within the same team is unfair" and "Grading in a group project is unfair" can be found. Although seemingly similar, there is an important difference between the two. The former states that *if* the course is graded, all team members should be graded in the same way. This statement does not judge whether grading is desirable or not. The latter goes further, in that it argues against grading in group projects, on the ground that the "best" students are not judged for their own abilities, and can thus not achieve the grades they deserve, when the grade is based on a collective project.

Attitudes towards being graded in this course, in the way they are expressed here, are value judgements. The categories created based on the students' judgements come to describe varying *opinions about* a phenomenon (or some phenomena) and put different arguments for, or against, this opinion in focus. These categories differ from the phenomenographic categories of description, in that they summarise *different opinions* on the phenomenon (as opposed to perspectives on, or different ways of understanding a particular phenomenon).

The students' recognition of the work of their team mates

The results of a peer evaluation that formed a part of the assessment, have been analysed. Each student was asked how they would distribute an amount corresponding to USD 20 per member over their team mates in relation to their performance. Since the majority of the teams consisted of six members, most individuals distributed USD 120. The outcome, presented in table 30, of this distribution has been analysed with regard to how the students at each university distributed the amount, and which amount the students received on average from colleagues at the two different locations.

Table 30. The amounts given to students in the peer evaluation of 2001, divided on the location of the students giving and receiving.

Location of donor and receiver	Average amount given
From Swede to Swede	22,25
From Swede to American	18,79
From American to American	20,07
From American to Swede	20,07

It is sometimes tempting for a researcher to draw strong conclusions from data of this kind. However, when studying limited statistical data, as this, care must be taken. In this setting, reasons to be cautious include that the distribution pattern varied considerably between and within the teams, as well as that the two student populations had different backgrounds and expectations. By judging solely from these statistically based findings, without relating it to other research results, the conclusions would not offer a nuanced picture of the interesting issue of grading. For example, it would be impossible to state the motives that a student might have to give a certain amount, and to determine to what extent other concerns than the evaluation of their peers for the purpose of the grading come into the picture. A system of this kind, might be (mis-)used to favour personal friends, or might invite offering larger amounts to those, who need to increase their grade. Used with care, such findings might however provide a building stone to the construction of a larger understanding of the role of the grading in the Runestone course.

The trustworthiness of such data must thus be seen in relation to other findings[68]. In the Runestone environment, some comparisons can be made. A similar evaluation was made in the year 2000 (the year before the data used

68. The discussion here does not concern how statistical data in general should be interpreted and judged, but refers only to the use of this particular data in this setting. A further discussion concerning quantitative research methods is outside the scope of this thesis.

in this thesis was collected). The overall image offered by that evaluation is similar (Pears et al., 2001). Still, a comparison with data for 2001 is relevant and indicates that tendencies at large remain similar. Another comparison can be made with the results of Last (2003). In her research she compares the peer evaluations from year 2000 with the IRC-logs and e-mails among the students. From this comparison (or triangulation with her terminology), she draws the conclusion that the students were honest in their peer evaluations.

The distribution of grades

The peer evaluation of 2001 can also been studied in relation to the distribution of the grades that the students were awarded by the teachers presented in table 31. Here a similar pattern can be found, where the students in Sweden are awarded higher grades by the teachers than their American colleagues.

Table 31. Average grades awarded by the teachers to the students at the two location in the Runestone grading scheme (see page 141)

Location of students	**Average grade awarded**
Average grade awarded all students	83,61
Average grade awarded all students in America	81,55
Average grade awarded all students in Sweden	85,05

Since the findings described in this sub-section are consistent with other research results concerning the peer evaluation and other data concerning the distribution of the grades in the Runestone course, they can serve to complement the qualitative results that dominate this thesis.

Discussing grading

The worries expressed among the staff that the different grading systems would result in different incentives to work hard in the course must be seen as ungrounded in the light of these results. Certainly, this could be the case for certain individuals, but if it were common among the Swedish students to be satisfied with a low degree, the statistical data would have shown a different picture, and the qualitative analysis would have given different results. The mechanisms that motivate the students to do a good job are complex and diversified. All claims, that might previously have been recognised by teachers and students concerning “how the students think” about grading and results are thus established to be oversimplifications of such a degree that they are incorrect. Reality has proved richer and much more complex.

The results also show that the factors that can overshadow the grading have a social dimension. These results point towards social aspects *being a* *part of* the motivation to learn, a question which is touched upon in chapter 11, and has also been discussed by Coupland (2004) who argues that "learning together" is a strong motivational factor for the students. It would be rewarding to explore further the relationship between team-learning situations and motivation.

Certainly, the particular environment of the Runestone project can be assumed to be an important factor that influences the priorities of the students.

IV. THE WHOLE AND BEYOND

Chapter 14. A full picture

Not until now, I can analyse the whole. In this chapter, I use the methodological framework to synthesise the results from the previous analyses into a whole. The holistic picture that evolves is constituted both of the partial results that I have previously discussed, and of the synthesis described in this chapter.

1. Drawing the full picture

The purpose of this chapter is to tie the threads together and to reveal the different ways in which the students' experience of different phenomena interact in an experienced learning environment.

To draw the holistic picture of the students' experience of learning in the Runestone environment, I have taken two different perspectives on the previously presented results.

Firstly, the different motives that the students strive for in the Runestone course (see chapter 11) are used as a structure to organise the earlier revealed categories of different phenomena in the learning environment. This analysis offers means to relate different categories of phenomena, that at a first glance seem unrelated, into a structure based on the experienced motive.

Secondly, the contradictions, earlier discussed in relation to particular phenomena in chapter 13, are now further analysed with the aim of distinguishing between different forms of contradictions. The categorisation, that results from this endeavour can be deployed to explore in which situation and "where" within the activity system contradictions can arise, and what their roles are in the students' experienced learning environment.

The results of the previous two analyses are related to activity systems, with the ultimate aim of describing the different ways, in which the learning environment is experienced, in a unifying picture.

2. Relating categories of the environment to the students' motives

Categorising the different categories of phenomena in the learning environment (described in chapter 13) in relation to the motive (see chapter 11) serves two purposes.

Firstly, it highlights similarities and differences between categories of different phenomena, and in this way contributes to the creation of a holistic picture of the students' experience of learning in this environment.

Secondly, it serves as an instrument to reveal underlying patterns in the experienced environment. Understanding such patterns, particularly in relation to contradictions - where they arise and for what reason, would be useful for a teacher or a course designer.

Phenomena in the learning environment can be organised into four different classes[69], differing in their focus of the motive as is shown in table 32 below. The purpose of the classes is to organize the previously described categories according to certain criteria. Thus, they are not entities on their own right. The classification into four classes, for which the categories of the motives form the point of departure, can be contrasted to the results concerning the motive (see for example table 13) that indicate two main foci: (a) the categories that are related to the educational framework and (b) those that are geared towards the learning object.

Two "new" classes can be described in terms of the previous results, in that one describes a situation, in which the motive is left unproblematised (category 1), and the other in itself contains a "blend" of two motives (category 3). Thus the difference between the number of classes and categories, is a result of a new organisation of data and does not in itself highlight any new insights. In table 32, the first column indicates the different foci of the motive, the remaining columns mirror how the categories of the different phenomena from the learning environment are categorised. Internal contradictions within a category are highlighted.

In the following, the analysis will be presented "through the eyes of" a "hypothetical student"[70] and will be heavily slanted towards the students' experience of learning computer science. This presentation offers a

---

69. The term *class* is introduced here to represent the new categories. The purpose of introducing a new term is to avoid confusion, rather than to highlight a qualitative difference compared to the term *category*. It should be pointed out that the term here is used without any sense of rating.

70. Again, it must be stressed that the categories describe experiences *at a collective level*, that students shift between categories, and that they might experience different phenomena simultaneously in qualitatively different ways. For these reasons, the *hypothetical student* that is referred to in this section does not have any physical correspondence, and does not even illustrate how an individual student could experience a situation at a certain moment. It is purely a theoretical construct serving to illustrate the constitution of experience.

Table 32. Classes that describe how phenomena in the learning environment are organized according to their different foci in terms of the motives. The dashed line differentiates between those classes in which other aspects than learning of the content comes to the fore, and those in which the content of the learning is a key aspect.

		Categories of...			
	The focus of the motive	**... the structure of control**	**... the code selection**	**... the milestone meetings**	**... the experience of being graded**
1.	**Unproblematised**	ϟ Control is missing	ϟ Pragmatic/ opportunistic	Meetings are taken for granted	–
		–	–	ϟ Meetings have a value of their own	–
2.	**Focus on the requirements of the environment**	–	Technical	Meetings concern the project	Grading is an aim in itself
		–	Strategic	–	Grading is an instrument for future life
3.	**Focus on both the requirements of the environment and the content of learning**	ϟ Control is taken by a few	–	–	–
		ϟ Control is allocated	–	–	–
4.	**Focus on the content of learning**	Control is distributed	Personal	Meetings support a sense of responsibility	Grading is subordinated to social obligations

recontextualisation of the categories discussed earlier in this thesis, now into a context of the students' different motives for taking a course. The purpose of the abstraction that is obtained in this way, is to help the researcher to see the situation with a minimum of "noise", so that he can concentrate on some key aspects, here on the various holistic pictures of the experienced learning environment.

Class 1. Unproblematised

In the first class the issue of a motive remains unproblematised; that is, it describes situations where students do not experience a clear meaning of their work. Experiencing a team to be without a control structure, as well as the pragmatic or opportunistic ways of selecting a code, are examples of such situations.

Three of the four categories of phenomena in the learning environment that form the basis for the first category show internal contradictions. This is well in line with activity theory that state that a mutually agreed motive is necessary for constituting an activity (see for example Holland & Reeves, 1996). These four categories do not appear to be related to a motive, and come, as a consequence, to represent ways of experiencing phenomena as if they existed in a vacuum; that is, they describe ways of understanding something, where this "something" is not seen as a part of a whole. Consequently, they are depicted on isolated, or a few, constituents of the activity systems (see figures 10, 15, 19 and 20).

Our hypothetical student, who experiences his learning environment in this way, has a fragmented perspective of the environment, and cannot see a strong correlation between his work in the project and the aims he has for his university studies. Furthermore, the phenomena within the learning environment are experienced as isolated. This leads to difficulties for the student in finding a solid position to use as a lever to get out of this situation. Support from others is needed.

A student who experiences the environment in this way acts pragmatically and takes decisions on an ad hoc basis. Needless to say, learning computer science does not play an important role in these situations. It is hard to imagine how efforts to learn could advance beyond the second category of learning computer science, *Learning CS through learning about isolated concepts* (see table 16), since the unproblematised way of understanding phenomena in the learning environment suggests a similar attitude to the object of learning.

In short, this category is not desirable.

Class 2. Focus on the requirements of the environment

The second class organises those categories whose motives are focused on experienced requirements from the educational system. Two different ways of experiencing the selection of a code, one way of experiencing the function of the milestone meetings and two ways of experiencing grading can be found here.

The underlying categories may, or may not, show internal contradictions. Even here, the results align well with theory. A motive that gives meaning to the activity exists and renders a certain stability to the situation. The categories are not advanced or inclusive (with the terminology of phenomenography), or not culturally advanced (with the terminology of activity theory) compared with many other categories.

The better functionality in this class, compared to the previous, is further illuminated by the more complete activity systems that depict these categories (see figures 16, 17, 21, 23 and 24 respectively), filling at least one of the sub-triangles, and including the subject and the object node in the activity triangle. Here, a complete, fully functional activity system, which serves to give meaning to the categories is obtained. This is illustrated in figure 26.

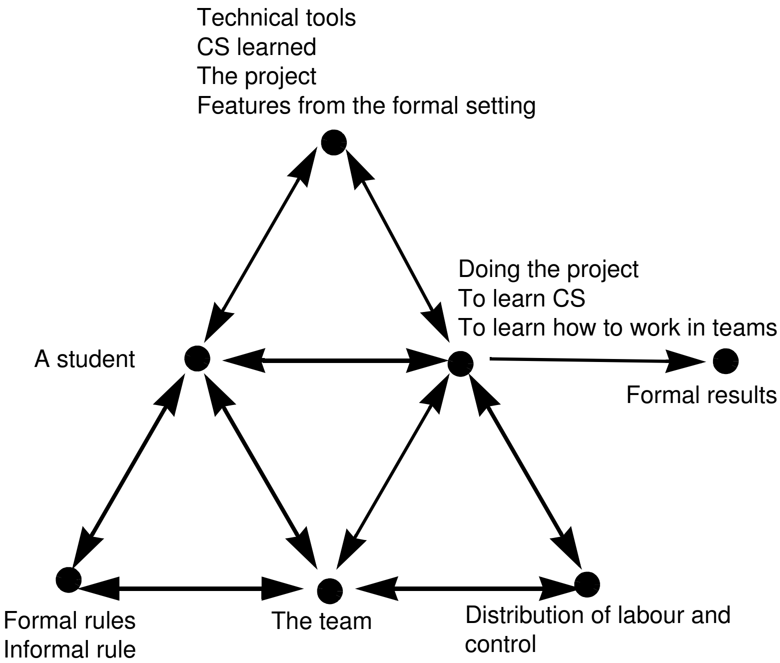

Figure 26. The learning environment, as experienced in class 2, *Focus on the requirements of the environment*. The nodes are described in the text.

How the different constituents of the activity are experienced by our hypothetical student, who represents this class, varies between the categories; claims about specific issues are impossible to make. However, some conclusions can be drawn.

The tool node represents something concrete that is experienced to help the work. Certainly a student's competence and previous experience of computer science constitute important instruments in his work towards his aims. The empirical data shows several examples of such instruments, such as the role of technical features of the code, or the use of milestone meetings, that serve as vehicles to reach the aim of finishing the project. The project itself can also serve as a tool to reach the aim of obtaining a grade. The nodes representing the rules have concrete denotations. For example, data indicate that the rule node could correspond to restrictions that are set by the environment (frequently time constraints), or formalised rules, such as grading schemes. The division of labour is concerned with control or concrete distribution of tasks.

Our hypothetical student here experiences a situation where the environment both expands the possibilities of doing the project and, at the same time, delimits these possibilities. The restrictions that are constituted by the environment also influence the learning of computer science. The most advanced categories of the act of learning are unreachable in this class, since they are related to the motives of personal development, that are not represented here. Hence, only categories number 5 of the act of learning computer science, *Learning CS through integrating systems*, and below are available. For similar reasons, it is not possible to use tools based on a personal meaning of computer science concepts for a student who does not search for such meaning. Since a personal meaning is useful for judging the relevance of a particular concept in a certain context, this constraint sets an upper limit for the strength and generalizability of the intellectual tools that are related to competence in computer science.

Class 3. Focus both on environmental requirements and the content of learning

The third class organizes those categories, in which two different motives meet, a situation that has only been identified in two categories both describing the experienced control structure (see figure 11 and 12, respectively). As expected, these are subject to contradictions, related to the dual motive. In the categories *Control is taken by a few* and *Control is allocated,* contradictions can arise when some want to finish the work for reasons of expediency, while others see that as an obstacle to focusing on the actual learning content.

These two categories are characterized by the situation in which the students experience a conflict between different objectives, as is clear from the underlying interview excerpts (see chapter 13). This contradiction can now be explained in terms of a conflict between a focus on formal requirements and on learning objectives.

Class 4. Focus on the content of learning

Finally, the fourth class groups the most advanced categories for each phenomenon. The key characteristics of this class are the outcomes in terms of personal development or ethical standpoints, highlighted in all the underlying categories (figure 14, 18, 22 and 25, respectively). The categories are in themselves stable and functional, and describe harmonious situations. An activity system, where the categories are superposed on each others, is illustrated in figure 27 below.

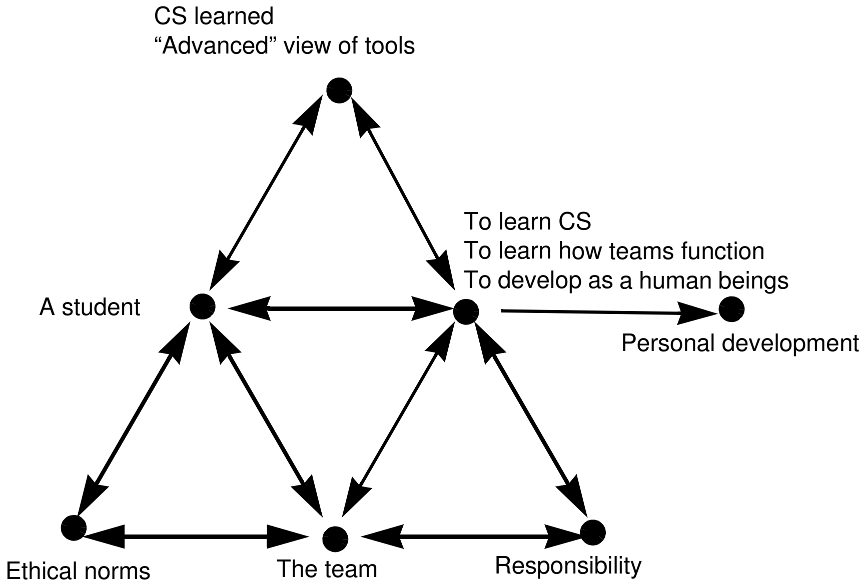

Figure 27. The learning environment, as experienced in class 4, *Focus on the content of learning*. The nodes are described in the text.

No contradictions can be identified in these categories. a theoretical explanation to this is offered by the phenomenographic theory of learning. In general, a more advanced category includes (aspects of) underlying categories and adds new aspects or relationships, that makes the more inclusive category qualitatively different. This means that a more inclusive category comes to include a less inclusive, and, at the same time offers a qualitatively different perspective. In this way, it offers tools to resolve contradictions, that seem unresolvable in the perspective offered with the less advanced perspective. A similar line of thought can be found in Engeström’s work, where he argues that contradictions might lead to a change, where the new situation is “culturally more advanced” (Engeström, 1987).

An illustrative example can be found in the different categories of the milestone meetings. The most advanced category includes all others and can, through its broader perspective, offer solutions to problems in less advanced categories. Particularly, the contradictions concerning the rules for the meeting, caused by unclear ideas of what the motive for the Runestone meeting is (category 2, *The meetings have a value of their own*), can be resolved since focus is shifted from the meeting itself (category 2) to the motive for the whole Runestone project, with the meetings now serving as a tool, to reach other motives.

Exact interpretations of the different constituents of an activity system, as experienced by our hypothetical student cannot be given, but common patterns, described in terms as personal engagement, participation, and respect for others can be discerned.

The tools in this class are experienced differently from the second, in that they here serve as instruments to reach different motives. For example, a computer science competence of a different kind is needed in order to finalize the code (class 2), than to develop as a person (class 4). To finalise the code, programming and a basic knowledge of design is needed, but when striving to develop as a person, more advanced ways of experiencing a computer science concept, that puts it in a context, is required. Data suggests that the two nodes rules and division of labour differ compared to the second class. Here ethical norms - the issue of what is right or wrong - as well as a sense of responsibility that encourages a student to work for the common good comes to the fore.

For our hypothetical student, the environment experienced in this way can serve as a platform for the most advanced ways of learning computer science; that is, the environment encourages learning that renders a personal meaning to computer science.

Certainly, this is a desirable way of experiencing the learning situation.

3. The experienced learning environment

The "life" of a team is more complex than what is revealed by the outcome of the phenomenographic study of the phenomena in isolation. Not only do different students experience phenomena in different ways, but an individual student might simultaneously experience something in different ways and/or might shift between different ways of experiencing something both spontaneously or when triggered. Still, the underlying structures that are revealed serve as lenses for the researcher when studying the learning that takes place, and can serve as a tool for a teacher.

A team then "lives" in a situation that "stretches over" several of these categories. Superposing the fourth onto the second, as is done in figure 28 below, summarises the different ways in which it is possible for a team member to experience his learning situation. The first category is excluded since, as was pointed out above, it does not offer good possibilities for learning; the third since it is constituted by (an aspect of) the contradictions between the second and the fourth.

It is interesting to compare the picture that summarises the students' experience of their learning environments with results from similar studies within "pure" activity theory (see for example Engeström, 1987).

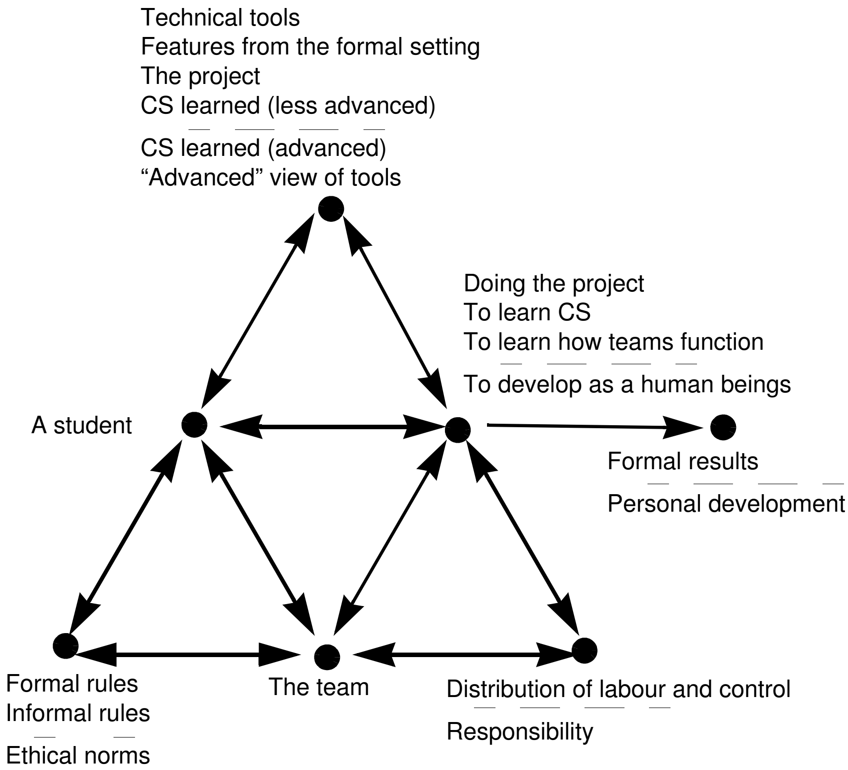

Figure 28. The variation in the learning environment, as it could be experienced by a "hypothetical student". A student's experience of the learning environments as described in class 2, figure 26 (above the dashed lines) is superposed on that in class 4, figure 27 (below the dashed line). The nodes are further described in the text.

In order to search for underlying differences, the researcher has to go beyond the individual nodes, and extend his investigation to the general meaning of the nodes. In doing so, an important difference becomes visible in the nature of the relationships between different ways of experiencing something, that can be found within a node. In contrast to the activity theory model which *takes for granted* that there are contradictions of the same kind within a node, the empirical findings in this study indicate that there are relationships of different kinds between the constituents of the node: contradictions, harmonious coexistence, or simply situations where different ways of experiencing the node collapse into one main category. The team node can serve as an example. The variation that has been found in the phenomenographic project is not related to the different motives and is thus not "visible" from this perspective of an activity system. Since the nature of this variation is different, it cannot be illustrated, and the community node is simply denoted “team” in this model. The picture outlined by Engeström, with its different theoretical foundation and externalist perspective, allows the researcher to describe a contradiction in the community node.

The empirical and experiential basis in the current work makes it possible to go further to examine the structure of the situation *from the students' perspective*, and thereby to reveal how the students experience and can constitute a situation. An activity theory approach, on the other hand, takes the researcher's perspective and becomes, as a consequence, less focused on the students. Thus, the feature of the results obtained by the framework deployed in this project, is that it aids the researcher to draw conclusions concerning learning in the subject area.

Furthermore, the empirical data allows to perform an analysis at a detailed level, and in this way make claims about specific phenomena and particular questions. This aids the researcher in drawing the kind of conclusions concerning learning in the subject area that can be valuable in teaching situations.

4. Categorising contradictions

Now, having established a model for how the learning environment is experienced, it is fruitful to continue the analysis of the contradictions according to the framework discussed in chapter 7. This time the motive is to settle “where” in an activity system contradictions can arise and what their functions are for learning.

By now taking the analysis one step further, corresponding to phase 3 in figure 9, the focus shifts from mainly being on the data and seeing the categories as something that represent (aspects of) data (as in chapter 13) to the categories themselves and their different properties. The categories can then in their turn be studied, organized and categorised. The results of such an analysis of “where” in an activity system the contradictions appear, are illustrated in table 33 below. The purpose of such categorisation of the contradictions is to offer means for the teacher to recognize when and where contradictions can appear.

A situation where a team meets, or a student reflects, might, or might not, contain contradictions that might, or might not, come to the surface. The distinctions and unifications concerning the contradictions that will be introduced is a valuable tool in the analysis of the students' experience of learning in the Runestone project.

Category 1. Contradictions within a category
Contradictions appear in many categories, particular in those that are the least advanced. The three sub-categories presented in table 33 have been

Table 33. The categories contradictions in and between the students' experience of phenomena in the Runestone course.

	Label	Description
1a	Contradictions within a category	Contradictions in a particular constituent of a category
1b		Contradictions between two particular constituents of a category
1c		Contradictions between a particular constituent of the phenomena and its background
2.	Contradictions between categories of a phenomenon	Contradictions between the two different categories of a phenomena
3.	Contradictions between two phenomena	Contradictions between categories of two different phenomena

further presented and illustrated with empirical data from the learning environment in chapter 13.

Category 2. Contradictions between categories of a phenomenon

As has been noted above, contradictions can arise between different ways of experiencing a particular phenomenon that might coexist and meet in a team or in the reflections of a student.

Although the process of relating the three categories of the experience of being graded in the Runestone project to an activity system has been rather straightforward (see page 179), the discussion has exposed new issues that need to be addressed in order to understand the implications of these results.

Firstly, in order to understand the nature of the contradictions that have been discerned in the analysis of the students' experience of being graded, the relationship between the object and the tool in an activity deserves more attention. Secondly, possible contradictions between the categories that coexist in a team must be considered.

What an object "really" is and how different concepts tend to move into and out of the object node, have previously been discussed in chapter 6. The Runestone project with its dual role as a tool for getting a good grade (category 1 and 2) or as object (category 3), can serve as an example that could lead to conflicts within a team. This is illustrated in figure 29 below, which shows the three categories superposed. The flash arrow indicates the possible conflict.

The second question, outlined above, concerns the possible contradictions that might arise between the categories of being graded that meet in a team. The object node comes to encompass different objectives for taking the course (see figure 30), both the wish to get a good grade (categories 1 and 2), and the desire to realise a successful project (category 3). This contradiction,

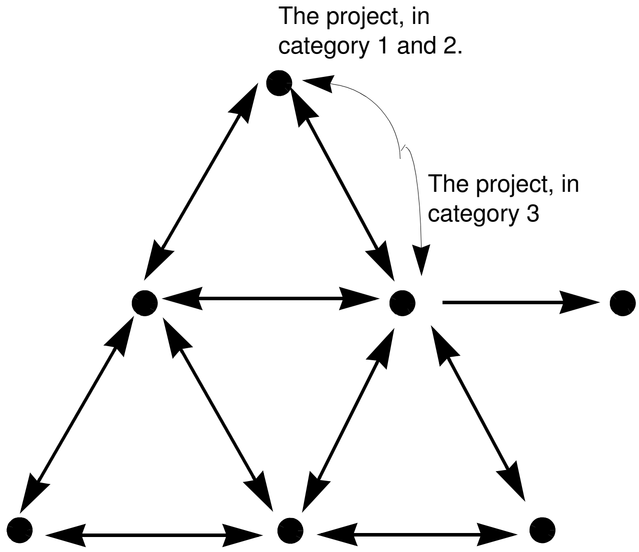

Figure 29. Superposing the three categories of the experience of being graded in the same activity system. The flash arrows indicate a contradiction between the experience of the project as a tool and as a motive on its own right.

that might or might not come to the surface, is illustrated by a flash arrow by the object node in the figure.

Category 3. Contradictions between two phenomena

Anthony´s statements serve as an illustration of a contradiction with one of its poles in the experience of being graded. In the discussion cited on page 60, he argues that a good grade is important in itself. Those statements can be read against the following fragment of a discussion (previously discussed on page 145):

> Interviewer: Do you think [...] your way of working is it a better way to get the job done?
>
> Anthony$_1$: I don't think it necessarily is better. I know it is a way to get it done. If we had more time, if the project was any different, I mean I would love to take time and to sit down and all of us discuss, and say this is what is going to be going on, what do you guys think? What should we do about this? But with the time constraints, it's like we're doing this. Any complaints, let's do it.

The long discussions that would characterize team work, if all team members were to participate in all decisions, might, according to Anthony,

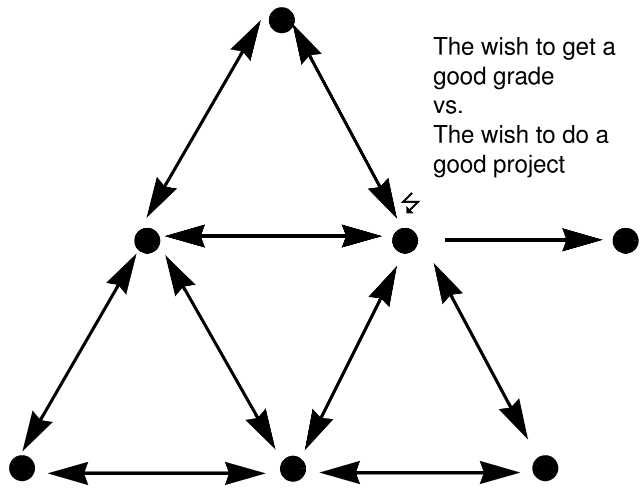

Figure 30. Superposing the three categories of the experience of being graded in the same activity system. The flash arrow indicates contradictions and are further described in the text.

lead to a poor outcome. The structure of control, that he criticizes has earlier been discerned and labelled *Control is distributed.* Thus, a contradiction has been identified between categories of different phenomena, the experience of being graded and structure of control in a team.

5. Contradictions as a way to encourage learning

The learning of computer science is presented as the primary objective of the Runestone project, whereas the international experience and the learning to work in distributed teams take second place. For the course to comply with its motives, the students' learning of phenomena within computer science should be encouraged, while a development of the students' experience of phenomena that are contextual to learning are less interesting. For this reason, it is important that the students face contradictions that are related to concepts within the subject area, and that the situation encourages the students to tackle such problems and that they are offered the tools to solve them.

On the other hand, such a development is not even wanted, for those phenomena that basically express a social dimension. The short time frame for this course (approximately eight weeks) is here an obstacle. A conclusion is that the teachers ought to create an environment that minimizes the possibility for such contradictions to dominate the study situation, and to offer support in the process of overcoming such contradictions.

6. Summary

In this chapter I have proposed a picture of the learning environment, as it is experienced by the students. The basis for this picture is a reanalysis of earlier results, now organised to describe the complex relationship between the motives that the students strive for and the different ways in which they experience their environment. I have shown a relationship between a focus on the content of the learning and the advanced (or inclusive) ways of experiencing the learning environment. The analysis also has indicated that the less advanced ways of experiencing the learning environment lead to a more problematic view of the environment, with a considerably larger degree of contradictions. The contradictions can arise, not only within particular categories, but also in the meeting between categories from the same or different phenomena.

Since contradictions in and between phenomena in the learning environment tend to draw a student's attention away from the object of his learning, ways of experiencing the learning environment that contain contradictions are less desirable. A relationship between the different ways of experiencing a motive and the categories of phenomena in the learning environment are established.

With the intention of offering a tool to analyse the contradictions, and predicting what forms of contradictions can be expected and in which situations, I have also analysed and organised the contradictions into categories.

These results were then taken together and depicted as an activity system. In this way I have provided a structured result of this analysis, and revealed how the different constituents of an activity system are experienced by its participants and how the whole is shaped from these pieces.

Chapter 15. Significance

With this chapter, I conclude my thesis. It offers a holistic picture of *what*, *why*, *how* and *where* students learn computer systems in an internationally distributed project-based course.

- The *what* aspect describes the students understanding of that which is aimed to be learnt about.
- The *why* aspect illuminates the students' motives for their learning.
- The *how* aspect explains how the students go about to learn.
- The *where* aspect presents the students' experience of their learning environment.

These different aspects are synthesised to form a whole, that focuses on the students' experience of learning computer systems. A theoretically anchored research approach is deployed to reach these aims. Both in the results and in the ways in which the results are obtained, the project offers novel contributions.

The project makes the complexity of a learning situation for students in a distributed project course in computer systems visible. It also offers insights in how this complexity can be unpacked and draws conclusions on teaching and learning computer science. In this way, the project positions itself in the cross-disciplinary field of computer science education research.

Being a study in a cross-disciplinary field, it "speaks" to different audiences, who will see different results, different contributions to their interests and different sorts of significance in the results and contributions. To tackle this diversity, this chapter is divided into four main sections, each of which focuses on a particular field.

1. Significance to computer science education research

The literature overview has shown that only a few projects has been performed in computer science education research, where the students and their experience of learning computer science are in focus. Thus, by its pure existence, the project contributes to the field of research. It serves to ask questions about how such research ought to be performed, what kind of results that are desired, and how the results could be used in teaching situations. The role of the experienced learning environment is debated, and the students aims with their studies are examined. In this way, it contributes to the debate and furthers the limits of computer science education research.

The results also illuminate the value of studying the full picture. Since a student experiences a whole in his studies of computer science, a project that is aiming for a full picture contributes to a valuable understanding of our students' learning.

A non-deterministic relationship between the *what*, the *how* and the *why* of the learning of computer systems has been identified. The *where* aspect plays an important role as factor that influence this relationship; that is, by a conscious change in the prepared environment, a teacher can influence the relation between the three aspects of the object of learning and thereby improve the outcome of her teaching. Although these findings are firm, more research is required on the nature of this relationship and how these results can be used in teaching situations.

Concrete results also include:

- The methodological development has resulted in a framework for studying the students' experience of their learning in a context. It can now serve as a platform for research into students' understanding of computer science also in other situations
- The empirical investigation in how students experience some key concepts in computer science are established. This can be used by a future researcher as a stepping-stone for his research.

2. Significance to computer science education

The process of the production of this thesis is a key contribution in itself, from the very beginning, when I started to discuss if and how it could be feasible to do theoretically solid research within computer science education.

At the departmental level, the study has shown that research in computer science education research can be placed on a solid foundation, on equal footing to the foundations that are expected of main-stream computer science research. This insight has brought with it new ways of putting questions about teaching and learning computer science. The research approach and the research results have also shown that statements about teaching and learning, previously taken for granted, can be challenged. Within the department, ways of going about research in computer science education, the similarities and discrepancies between research in "pure" computer science and computer science education, and what kind of results that can be obtained have also become subjects for discussion. A language for these questions has evolved. In short, the department has obtained tools in the development of the education in computer science that are more solidly grounded than before. Recent research has pointed out the importance of an open debate concerning learning and teaching and an institutional commitment to the quality of teaching at the department (Hilborn & Howes, 2003).

The influences on the computer science education community can be outlined in a similar way, but of course, the direct effects are less apparent. On the other hand, influence can be pointed to at a large number of sites. My work has been discussed at several universities on at least four continents, through its publications, during conferences and seminars and, maybe most importantly, during informal discussions with Swedish and international colleagues. In this way, the production of this thesis has contributed to putting the kind of research in computer science education that I have carried out on the map, and has contributed to the dissemination and development of research within teaching and learning computer science. There are some direct contributions to education within computer science to be read in the results of this thesis, but because of discrepancies between groups of students and learning environments I encourage teachers to study the research approach and its outcome to seek insights relevant to her and to her situation.

The contributions of this project can thus be described in the visibility it puts on the need to understand how our students learn computer science, on the ways to gain insights in such questions, and on the results of such investigations. These outcomes, combined with the development of a language for and an environment to discuss computer science education, enables the field to advance.

3. Significance to computer science

As computer science education research shares aspects of its research object with core computer science, computer science education research fertilizes core computer science, in the sense that in both fields of research better understanding of core concepts are being worked on.

My study opens the way for researchers in main-stream computer science research to tackle certain categories of computer science problems in new ways, in particular when it is the ways in which other people experience computer science concepts and devices that is in question. Human-computer interfacing problems is one obvious example of this.

The development of core concepts within computer science can be traced in a historical evolution. However, the students' learning of the concepts follows another pattern, in that it starts in the current situation as it is presented to them, and in that it takes place in a far shorter period of time. Thus, their perspective becomes different from the historical evolution that has formed the concepts to what they are today. In the tension between these two perspectives, insights can be gained about other ways, in which the concepts are developed, constituted and related to each others.

Also when it comes to the relationship between core computer science and the world around it, this study, that goes beyond "pure" computer science, offers contributions. It offers tools for evaluating the research in a context and offers methods and methodologies aimed for exploring, evaluating and judging aspects of the interplay between computer science and the world and people around it.

Research in computer science education can also, thanks to its cross disciplinary character, serve as a bridge between different areas of research and in this way transfer insights gained within one field of research to another. For example, theories about learning communities and group development processes are relevant for ad hoc networking.

In short, the direct relevance of computer science education for core computer science can be traced to the fact that the former offers new perspectives on and approaches to certain issues of the latter.

The indirect effects lie in the opportunities for improved teaching and learning of computer science. The insights about how students learn about concepts within computer science, gained through computer science education research, serve as tools for teachers to improve education, and thereby educate better computer scientists.

4. Significance to educational research

The methodological framework, based on phenomenography and extended with elements of activity theory, forms a novel contribution to research on learning. Not only does it have direct applications in different settings, but it also demonstrates in what ways the outcome of research, here phenomenographic research, can be enhanced by bringing another approach with different a theoretical basis into the project.

Much educational research of today does not consider that which is learnt. Such a delimitation of a research project can, of course, be fruitful. But if the current project can open for a discussion of the role of the content of the learning, this project has contributed in a valuable way to educational research.

The focus on advanced university students, that characterise this study, is uncommon in educational research. Since the students already have a personal relationship to their subject area, and this relationship is crucial in phenomenographic research, the outcome also mirrors computer science

Much of what is mentioned in the section concerning computer science education research is also relevant to educational research. Particularly, the value for a perspective, where the students' experience of their learning of a certain subject matter is studied in, and as a part of, an experienced learning environment should be stressed. With this research perspective, the students' learning in a context is explored while focus still is kept on the subject matter.

5. Open questions

By taking a step into the unknown, as this study does, some questions are answered, but a still wider set of unanswered questions become visible and open for future research. Many of these are touched upon in relevant passages in the text. Here I will summarise some that are of particular interest.

During my work, I have been asked about possible cultural differences between the Swedish and the US students, and in the case such differences existed, what their nature were. I have become convinced that there are differences, possibly cultural, but that these differences, as well as the students' experience of the differences, are of secondary importance for the outcome of the course.

Another different, but related, issue concerns the relationship between what the students say and what they mean by their words. Here I believe that some students, predominantly Americans, used a language that was more dominant and persuasive compared with others, often Swedes – who speak English as a second language – and who talked with a lower profile. Since written language is so important for the communication in the Runestone environment, this issue can be relevant for the collaboration within the teams. Whether this impression is correct, and in that case, what influence it has on learning and collaboration are, however, questions left for future research.

Differences between individuals have also caught my attention. They became visible in the Runestone course, in that the students "took different courses" or "had different stories to tell" with regard to what the course was about. For example, Sebastian's story was that of a technical development and technical competence. He associated to these issues in his answers to most interview questions, even to those where I had expected different kinds of answers. In the same way, Sven's answers were, with few exceptions, turned towards the social interaction in his team. There are many examples of this kind – in fact, the answers of most, if not all, students could be described in terms of one or a few specific themes, that shone through. Now, the question that can be asked, is how these differences influence what the students' learn, what they are aiming to learn, and how they go about learning.

With the current results as a background, and with the rich interview data, other questions than those mentioned above can certainly be explored. Topics include how students develop a relationship to different ICT tools and how they come to use them as tools for learning, as well as issues such as the social roles of the students in their teams.

It would also be profitable to further develop the methodological framework with the aim of deploying it in new situations. The core question that it highlights, that of the experience of learning in an experienced learning environment, is certainly important to understand in order to improve education both in computer science and in other fields.

6. A final word

With this chapter, I leave my project and its results to others to use in research and teaching. My hope is that the project should put theoretically anchored research in computer science education on the map, and that it should inspire others to take up such research. Possibly, this would be the most important contribution of my work. It would advance the debate concerning good teaching and learning computer science and would serve to extend our knowledge to students in other sub-fields of computer science. The results of such a development would serve my overall aim of this project: to improve teaching and learning in computer science.

Sammanfattning

Hur kan man på en pedagogisk vetenskaplig grund förstå studenternas lärande av datorsystem i en komplex läromiljö? Frågeställningen står i fokus i denna avhandling i ett sammanhang av en internationellt distribuerad projekt-kurs.

Studenterna, som följde denna kurs som en del i en utbildning i informationsteknologi eller datavetenskap, arbetade i grupper om sex, där tre studenter i varje grupp fanns i Sverige, medan de tre återstående befann sig i USA. Tillsammans utvecklade gruppernas medlemmar styrprogramvara för en datoriserad träleksak. En väsentlig datavetenskaplig frågeställning som studenterna stod inför var att lösa datakommunikationsproblem i det system de skulle konstruera. Valet av nätverksprotokoll, eller regler för hur datorer kommunicerar, blev här väsentligt.

För att förstå hur studenterna erfor sitt lärande och sin läromiljö har en analytisk separering genomförts av hur studenterna uppfattade vad, varför, hur och var de lär. Vad-aspekten belyser det inom datorsystem som studenterna lär: nätverksprotokoll. Varför-aspekten belyser hur studenterna uppfattar sina motiv för att läsa denna kurs. Hur-aspekten belyser studenternas sätt att gå tillväga för att lära datorsystem. Var-aspekten belyser läromiljön, som den uppfattas av studenterna.

Data är insamlad genom intervjuer med studenter kring olika sidor av deras lärande och läromiljön. Analysen har skett för att blottlägga kvalitativt olika sätt att uppfatta de fyra aspekterna utan att för den skull kategorisera individer.

Vad: Fyra olika sätt att uppfatta nätverksprotokoll identifierades: som kommunikation mellan två datorer; som en förbindelse över ett nätverk; som en uppsättning regler; som en standard. Samtliga sätt att uppfatta protokollen kan vara relevanta i förhållande till förekommande situationer i datavetenskapliga utvecklingsprojekt.

Varför: Tre olika inriktningar i studenternas intentioner med att följa kursen identifierades: mot akademiska resultat; mot att lära sig arbeta i projekt; mot en social dimension av lärande. Dessa olika inriktningar kan i sin tur uppfattas på olika sätt. Exempelvis har fyra olika sätt att erfara akademiska resultat urskiljts: att få betyg; att lära sig datavetenskap för projektet; att lära hur man lär datavetenskap; att lära sig någonting nytt.

Hur: Sju sätt att gå tillväga för att lära sig datavetenskap identifierades. De olika sätten skiljer sig åt i aspekter såsom om lärandet handlar om att lära sig enstaka orelaterade begrepp, huruvida delarna i ett datorsystem eller hela systemet är i fokus, eller om lärandet inbegriper en personlig utveckling.

Var: Den miljö i vilken studenterna lär är komplex och inrymmer många faktorer. Olika sätt att uppfatta kontrollstrukturerna i en grupp, ett beslut taget i gruppen, värdet av möten med lärarna och uppfattningar om betyg analyserades och beskrevs i kategorier. Mellan vissa av dessa faktorer råder spänningar eller motsägelser. Exempelvis medför en ad hoc-baserad gruppstruktur, där ett socialt spel är en huvudfaktor i gruppens beslut, att motsättningar kan dominera gruppens arbete. Å andra sidan kan en struktur, där alla viktiga beslut tas gemensamt, leda till en stabil arbetssituation.

Ett önskat fokus på studenternas erfarenheter har motiverat en empirisk, kvalitativ fenomenografisk forskningsansats som bas. De olika komponenterna av studenternas erfarenheter av sitt lärande integreras med hjälp av element ur verksamhetsteorin. För att möjliggöra den syntes av fenomenografiska data, som erbjuds av verksamhetsteorin, har frågan om sammanhang, eller kontext, till lärandet belysts ur olika perspektiv. En åtskillnad gjordes på basis av vem som erfar en viss kontext i en viss situation. En student erfar en kontext som en bakgrund till det fenomen som diskuteras i en intervjusituation. Den kollektiva kontexten syns i analyssituationen och innesluter de individuella kontexterna, men överskrider samtidigt dessa i och med att helheten ger nya insikter om sammanhanget. Forskaren erfar en kontext, som innesluter hans eller hennes förståelse av studenternas kontexter, både de individuella och den kollektiva, och dessutom innefattar hans eller hennes egna erfarenheter av situationen, ämnet och forskningen. Ett metodologiskt ramverk är utvecklat för att integrera de insikter som vunnits till en helhet där studenternas lärande beskrivs i sitt sammanhang.

Syntesen visar på samband mellan de fyra aspekterna beskrivna ovan. Två kvalitativt olika, övergripande sätt att uppfatta lärandesituationen har identifierats. Det kan vara inriktad mot att uppfylla de formella kraven i en situation, eller mot att lära datavetenskap. Medan det första kan uppfattas som fylld av motsägelser, ger det senare bättre möjligheter för ett tydligt fokus i lärandet.

Som ett resultat har en modell utvecklats där det som studenterna lär, hur de går tillväga med sitt lärande, deras motiv till lärandet och hur de uppfattar sin lärandemiljö, studeras i relation till varandra. Ett exempel får illustrera: Vi tänker oss en student som uppfattar studiesituationen såsom huvudsakligen inriktad mot att lära sig om datorsystem och som inriktar sig emot att lära någonting nytt genom att tillägna sig personliga erfarenheter.

Han eller hon är i en bättre situation för att utveckla en nyanserad förståelse av nätverksprotokoll än en annan student som domineras av kraven från omgivningen och försöker att ta sig igenom kursen genom att lära sig vissa begrepp utantill.

I detta komplexa samspel ligger den möjlighet och utmaning som en lärare står inför, när han eller hon försöker förbättra sin undervisning. Att identifiera och utveckla de faktorer i omgivningen, som verkar för ett meningsfyllt lärande, är ett kraftfullt, men svårbemästrat, verktyg.

Ett viktigt bidrag av denna avhandling är att indikera vilka frågor som bör stå i fokus för den som vill arbeta för att förbättra studenternas lärande inom datavetenskap. Avhandlingen pekar på komplexa samband och visar resultat som går bortom det som ofta tas för givet hos lärare och studenter. I stället formulerar arbetet grundläggande frågor kring hur studenternas lärande av datavetenskap hänger samman med hur de uppfattar sin totala studiesituation.

Denna avhandling vänder sig, såsom ett tvärvetenskapligt arbete, till olika läsare med olika behov. Arbetet placeras mot en bakgrund av datavetenskap, datavetenskaplig utbildning, forskning inom datavetenenskapens didaktik (computer science education research) och pedagogisk forskning. Avhandlingen sträcker sig över ett vitt spektrum av frågor, som tillsammans beskriver hur studenter lär datavetenskap i en internationellt distribuerad projektkurs.

Acknowledgement

First I wish to thank my supervisors: Shirley Booth, who has supported me throughout the process that has led to this thesis, and who has introduced me to educational research and phenomenography. Arnold Pears has been my support during the tough of this thesis, while Lars Asplund helped me to set off on my journey that led to this thesis in computer science education.

My research project would not have been possible without the strong support from my friends and coaches. Åke Ingerman has been my intellectual sparring partner and has happily read all my drafts, while Mats Daniels has offered both his moral and practical support. Tony Clear has helped me to transform my writing into proper English.

The warmest thanks to the six of you.

I am in deep debt to many more. Without all support from the encouraging members of the community of computer science education research, I would not have succeed this project. Thank you.

I also want to thank the students, who shared their experiences of participating in the Runestone course with me. Your contributions have been of a tremendous value for me.

My work on this thesis has mainly been financed by a doctoral studentship from The Knowledge Foundation of Sweden's research programme "Learning and IT", for which I am grateful. The Swedish Research Council have kindly founded the later parts of my doctoral studies. The Runestone project has been financed by the Council for the Renewal of Higher Education, NyIng (The project of renewal of engineering programs) and from Uppsala University. The Swedish Foundation for International Cooperation in Research and Higher Education and Uppsala University has also contributed to finance my doctoral studies.

Finally, but most importantly, I would like to thank my family and friends for their caring support.

References

Åberg-Bengtsson, L. (1998). *Entering a Graphicate Society. Young children Learning Graphs and Charts*. Gothenburg, Sweden: Acta Universitatis Gothoburgensis.

Adawi, T. (2002) What is hot and what is not: A phenomenographic study of lay adults' conceptions of heat and temperature. Revised version of paper presented at the Second Hong Kong symposium on phenomenography. In Adawi, T. *From Branes to Brains: On M-theory and Understanding Termodynamics.* Uppsala, Sweden: Acta Universitatis Uppsaliensis.

Adawi, T., Berglund, A., Booth, S., & Ingerman, Å. (2001). On context in phenomenographic research on understanding heat and temperature. Revised version of paper presented at EARLI 2001, Fribourg, Switzerland, 2001. In A. Berglund (2002). *On the understanding of Computer Network Protocols.* Licentiate thesis, Uppsala University, Uppsala, Sweden. Also available at http://www.docs.uu.se/~andersb/lic/

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.

Baffes, P. (1994). *Automated student modelling and bug library construction using theory refinement*. PhD thesis, University of Texas at Austin, Austin, TX, USA.

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

Bannon, L. (1997) Activity Theory. Available at http://www-sv.cict.fr/cotcos/pjs/TheoreticalApproaches/Actvity/ActivitypaperBannon.htm

Barab, S., A., Barnett, M., Yamagata-Lynch, L., Squire, K., & Keating, T. (2002). Using activity theory to understand the contradictions characterizing a technology-rich introductory astronomy course. *Mind, Culture and Activity* 9(2) 76 - 107.

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

Belbin, R.M. (1996). *Team Roles at Work*. Oxford, UK: Butterworth Heinemann

Bellamy, R. K. E. (1996). Designing Educational Technology: Computer-Mediated Change. In: B. Nardi, (ed.) *Context and Consiousness, Activity Theory and Human-Computer Interaction*, Cambridge, MA, USA: MIT Press, 123 - 146

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. (2002). *On the understanding of Computer Network Protocols.* Licentiate thesis, Uppsala University, Uppsala, Sweden. Also available at http://www.docs.uu.se/~andersb/lic/

Björkman, C., & Trojer L. (2002) Compuer 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. (1997). On phenomenography, learning and teaching. *Higher Education Research and Development*, 16(2), 135 - 157.

Booth, S. (2001). Learning Computer Science and Engineering in Context. *Computer Science Education,* 11(3). 169 - 188.

Booth, S. A., & Petersson, J. (1998). Getting a grasp of computer science and engeineering: On the experience of learning in groups. In C. Rust (ed.). I*mproving student learning. Improving students as learners*. Oxford, UK: Oxford centre for staff learning and development. 392 - 403

Bowden, J., & Marton, F. (1998) *The University of Learning. Beyond Quality and Competence in Hinger Education.* London, UK: Kogan Page.

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.

Bruner, J.S. (1966) *Towards a Theory of Instruction.* Cambridge, MA, USA: Harward University Press

Budd, T. (1999). *Understanding Object-Oriented Programming Using Java*. Reading, MA, USA: Addison-Wesley.

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

Chang, R. K. C. *Teaching Computer Networks with the Help of Personal Computer Networks*. In Proceedings of the 9th annual SIGCSE conference on Innovation and Technology in Computer Science Education, Leeds, UK, 208 - 212.

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.

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.

Cope, C.J. (2004). Ensuring validity and reliability in phenomenographic research using the analytical framework of a structure of awareness. *Qualitative Research Journal*, (2)4, 5-18.

Coupland, M. (2004). *Learning with new tools*. Unpublished doctoral thesis, The University of Wollongong, Australia. Available online: Australian Digital Theses. http://adt.caul.edu.au/.

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. , Faulkner, X., Newman, I. (2002). Open Ended Group Projects, Motivating Students and Preparing them for the 'Real World'. *In Proc. Conference on Software Engineering Education and Training*, Covington, KT, USA

Daniels, M., Berglund, A., & Pears, A. (2003) *Runestone - the story.* Paper presented at Utvecklingskonferensen för högre utbildning, Gävle, Sweden.

Daniels, M., Berglund, A., Pears, A., & Fincher, S. (2004) *Five myths of assessment*. Proceedings of the sixth conference on Australian computing education, Dunedin, New Zealand. 57 - 61.

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

Ekeblad, E., & Bond, C. (1994). The nature of a conception: questions of context. In: R. Ballantyne, & C. Bruce. (Eds.) *Phenomenography: Philosophy and practice.* Centre for Applied Environmental and Social Education Research, Faculty of Education, Queensland University of Technology, 147 - 162.

Engeström, Y. (1986). The zone of proximal development as the basic category of education psychology. *Quarterly Newsletter of the Laboratory of Comparative Human Cognition,* 8(1), 23 - 42.

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

Engeström, Y. (1990). When is a tool? Multiple meanings of artifacts in human activity. In: Engeström. *Learning, Working and imagining. Twelve studies in activity theory.* Helsinki, Finland: Orienta-konsultit.

Engeström, Y. (1991). Activity theory and individual and social transformation. *Multidisciplinary Newsletter for Activity Theory*, No. 7/8, 14 - 15.

Engeström, Y. (1993) Developmental studies of work as a test of activity theory: the case of primary care medical practice. In S. Chaiklin, J. Lave (eds). *Understanding practice - perspectives on activity and context*. Cambridge: Cambridge University Press, 64 - 103.

Engeström, Y., & Escalante, V. (1996) Mundane Tool or Object of Affection? The Rise and Fall of the Postal Buddy In: B. Nardi. (ed) *Context and Consiousness, Activity Theory and Human-Computer Interaction*, Cambridge, MA, USA: MIT Press. 325 - 374.

Ernest, P. (1995) The one and many. In Steffe, L.P., & Gale, J. (eds.) *Constructivism in education.* London, UK: The Falmer Press. Hillsdale, NJ, USA: Laurence Erlbaum. 459-486.

Feit, S. (1998) *TCP/IP*, New York, NY, USA: McGraw-Hill.

Fjuk, A. (1998): *Computer Support for Distributed Collaborative Learning. Exploring a Complex Problem Area.* Dr. Scient. Thesis. Dep. of Informatics. University of Oslo, Oslo, Norway.

Fjuk, A., & Ludvigsen, S. (2001) *The Complexity of Distributed Collaborative Learning: Unit of Analysis*. Presented at Euro-CSCL, Maastricht, Netherlands.

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.

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.

Gall, M. D., Borg, W. R., & Gall, J. P. (1996) *Educational Research. An introduction. 6th edition.* White Planes, NY, USA: Longman.

Glaser, B., & Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, IL, USA: Aldine.

Glasersfeld, E. von (1995). A Constructivist Approach to Teaching. In L.P. Steffe & J. Gale (Eds.) *Constructivism in Education*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates. 3-15

Gordon, S. (1998). *Understanding students learning statistics: an activity theory approach*. Unpublished doctoral dissertation, The University of Sydney, Sydney, Australia. Available at http://adt.caul.edu.au

Gordon, S. (2004). Understanding students' experiences of statistics in a service course *Statistics Education Research Journal,* 3(1). 40 - 59.

Gordon, S., & Nicholas, J. (2002) *Phenomenography and Activity Theory: Do They Fit?* Paper presented at the tenth annual symposium: Contemporary Approaches to Research in Mathematics, Science, Health and Environmental Education, Melbourne, Australia.

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.

Hause, M. L. (2003) *Software Development Performance in Remote Student Teams in International Computer Science Collaboration*. Unpublished PhD thesis. The Open University, Milton Keynes, UK.

Hause, M., & Woodroffe, M. (2001). *Team Performance Factors in Distributed Collaborative Software Development.* Proc. Psychology of Programmers Interest Group (PPIG), Bournemouth, UK.

Herder, P. & Sjoer, M. (2003). *Group-basde Learning in Internationally Distributed Teams: An evaluation of a Cross-Atlantic Experiment.* In the proceedings of IEEE Frontiers in Education, Denver, CO, USA.

Hilborn, R.C. & Howes, R. (2003). Why many undergraduate physics programs are good but few are great. *Physics Today*, 56(9), 38

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.

Hultén & Booth (2002) *Considering context for networked learning in a phenomenographic perspective.* Paper presented at Networked Leaning 2002, Sheffield, UK.

Ilenkov, E. V. (1977). *Dialectical logic: Essays in its history and theory.* Moscow, Russia: Progress.

Ingerman, Å. (2002). *Exploring two facets of physics*. PhD Thesis. Chalmers Univerity of Technology, Göteborg, Sweden.

Jaques, D. (1992) *Learning in groups.* London, UK: Croom Helm

Jard, C., & Jéron, T. (2000). An educational Case Study in Protocol Verification and Distributed Observation. *Computer Science Education*, 10(3), 203-224.

Johnson-Laird, P. (1993) *The Computer and the Mind*, London, UK: Fontana

Kaptelinin, V. (1996). Activity Theory: Implications for Human-Computer Interaction. In: Nardi, B. *Context and Consiousness, Activity Theory and Human-Computer Interaction,* Cambridge, MA, USA: MIT Press. 103 - 116.

Kaptelinin, V., Kuutti, K., Bannon, L. (1995) Activity Theory: Basic Concepts and Applications. In B. Blummenthal, J. Gornostaev, C. Unger (Eds): *Lecture Notes in Computer Science.* 1015 (188 - 201)

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.

Kuhn, T. S. (1962) *The Structure of Scientific Revolutions*. Chicago, IL, USA: University of Chicago Press.

Kuutti, K. (1996) Activity Theory as a Potential Framework for Human Computer Interaction Research. In: B. Nardi, B. *Context and Consiousness, Activity Theory and Human-Computer Interaction*, Cambridge, MA, USA: MIT Press. 17 - 46.

Kärkkäinen, M. (1999). *Teams as Breakers of Traditional Work Practices. A Longitudinal Study of Planning and Implementing Curriculum Units in Elementary School Teacher Teams*. PhD thesis, Helsinki University, Helsinki, Finland.

Last, M. (2003) *Investigating the Group Development Processin Virtual Student Software Project Teams.* Unpublished PhD thesis, Kingston University, UK.

Last, M., Almstrum, V., Daniels, M., Erickson, C., Klein, B. (2000). An International Student/Faculty Collaboration: The Runestone Project. In *Proc. ACM SIGCSE/SIGCUE Conference of Innovations and Technology in Computer Science Education (ITiCSE'00).* Helsinki, Finland

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

Leontev, A. N. (1978). *Activity, consciousness, and personality.* Englewood Cliffs: Prentice-Hall.

Leontev, A. N. (1981). *Problems of the development of the mind*. Moscow: Progress.

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.

Lipnack, J. & Stamps, J. (1997). *Virtual Teams: Reaching Across Space, Time, and Organizations with Technology.* New York, NY, USA: Wiley

Marshall, D, Summers, M & Wollnough, B. (1999). Students conceptions of learning in an engineering context. *Higher Education,* 38, 291 - 309.

Marton, F. (1994) Phenomenograhy. In T. Husén and T.N. Postlethwaite *The International Encyclopedia of Education*, 2nd Edition. Oxford: Pergamon Press. 4424 - 4429

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

Marton, F., & Säljö, R. (1976). On qualitative differences in learning II - Outcome and process. *British Hournal of Educational Psychology,* 46, 115 - 127.

Marton, F., & Tsui, A.B.M. (2004) *Classroom Discourse and the Space of Learning*. Hillsdale, NJ, USA: Lawrence Erlbaum

Marton, F., Beaty, E., & Dall´Alba, G. (1993) Conceptions of learning. *International Journal of Educational Research*, 19, 277 - 300.

Marton, F., Watkins, D., & Tang, C. (1997) Discontinuities and continuities in the experience of learning: an interview study of high-school students in Hong Kong. *Learning and Instruction,* 7(1), 21 - 48.

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. *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, 2003

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

Nardi, B. (1996a). Activity Theory and Human-Computer Interaction. In: B. Nardi (ed.) *Context and Consiousness, Activity Theory and Human-Computer Interaction*, Cambridge, MA, USA: MIT Press.

Nardi, B. (1996b). Studying Context: A comparison of Activity Theory, Situated Action Models, and Distributed Cognition. In: B. Nardi (ed.). *Context and Consiousness, Activity Theory and Human-Computer Interaction*, Cambridge, MA, USA: MIT Press.

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.

Pears, A., Daniels, M., Berglund, A., Erickson, C. (2001). *Student Evaluation in an International Collaborative Project Course*. Presented at the Wise workshop of the SAINT conference, San Diego, CA, 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.

Pong, W. Y. (1999). *The Dynamics of Awareness*. Paper presented at Bi-annual conference of the European Association of Research on Learning and Instruction, Gothenburg, Sweden.

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. (1988). Learning in educational settings: Methods of inquiry. In P. Ramsden . *Improving Learning: New Perspectives*. New York, NY, USA: Nichols Publishing. 32-48

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

Sandberg, J. (1995) Are phenomenographic results reliable?, *Nordisk Pedagogik,* (15)3, 156 - 164.

Silén, C. (2000). *Between chaos and cosmos - about responsability and independence in learning.* (in Swedish), Linköping, Sweden: Linköping Studies in Education and Psychology.

Stallings, W. (2004). *Computer Networking with Internet Protocols and Technology.* Upper Saddle River, NJ, USA: Pearson Prentice.

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

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..

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA, USA.: Harvard University Press.

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.

Appendix A. Published work related to this thesis

Journal publications

Berglund, A. (2004) A framework to study learning in a complex learning environment. *ALT-J, Research in Learning Technology*, 12(1), pp 65 - 79.

Daniels, M., Berglund, A. and Petre, M. (1999) Reflections on international projects in undergraduate CS education. *Computer Science Education*, 9(3), pp 256 - 267.

Reviewed conference publications

Berglund, A. (2003). What is good teaching of computer networks? *Proceedings of 33rd Annual Frontiers in Education.* Champaign, IL, USA: Stipes pp. 2SD13 - 2SD19.

Berglund, A. (2003). Learning Computer Networks in an Internationally Distributed Project-based Course – Work-in-Progress. *Proceedings of 33rd Annual Frontiers in Education.* Champaign, IL, USA: Stipes, pp. 1S19.

Berglund, A. (2003). A Framework to study learning in an internationally distributed course. In J. Cook and D. McConnell (Eds.) *Communities of Practise. Research Proceedings of the 10th Association for Learning Technology Conference (ALT-C 2003)*, pp 27 - 39.

Berglund, A. and Pears, A. (2003). Students' Understanding of Networks in an Internationally Distributed Course. In V. Devedzic, J. M. Spector, D. G. Sampson, Kinshuk (Eds.) *Advanced Learning Technologies.* IEEE, Los Alamitos, CA, USA. pp 380 - 381.

Daniels, M., Berglund, A., Pears, A., and Fincher, S. (2004) *Five myths of assessment.* Proceedings of the sixth conference on Australian computing education, Dunedin, New Zealand. 57 - 61.

Pears, A., Daniels, M., Berglund, A. and Erickson, C. (2001). *Student Evaluation in an International Collaborative Project Course*. Presented at the Wise workshop of the SAINT conference, San Diego, CA, USA.

Other publications

Adawi, T., Berglund, A., Booth, S., & Ingerman, Å. (2001). On context in phenomenographic research on understanding heat and temperature. Revised version of paper presented at EARLI 2001, Fribourg, Switzerland, 2001. In A. Berglund (2002). *On the understanding of Computer Network Protocols*. Licentiate thesis, Uppsala University, Uppsala, Sweden. Also available at http://www.docs.uu.se/~andersb/lic/

Berglund, A. (2002). *On the understanding of computer networks*. Licentiate thesis, Uppsala university, Uppsala, Sweden.

Berglund, A. and Booth, S. (2002). *"Are you guys really concerned about the grades? On the experience of grading systems as contextual to learning in an internationally distributed computer science course*". Presented at ISCRAT2002 Dealing with Diversity. Tools and resources for human development in social practices, Amsterdam, Netherlands.

Berglund, A. (2002). *Learning computer systems in a distributed course: Problematizing content and context*. In the electronic proceedings of at European Association for Research on Learning and Instruction, SIG 10, Current Issues in Phenomenography, Canberra, Australia. Available at `http://www.anu.edu.au/cedam/ilearn/symposium/abstracts.htm`

Appendix B. An activity system

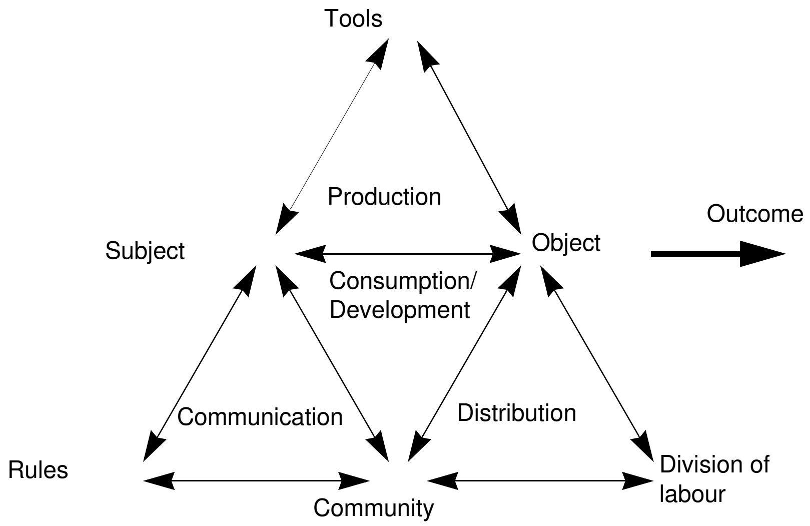

Figure 31. The components in an activity system and their relationships (developed from Engeström, 1987, p. 78)