Language, and the learning of data modelling

Dr.Scient - thesis

Christian Holmboe

2005

Department of Teacher Education and School Development
Faculty of Educational sciences
University of Oslo, Norway

§ 2.01 Der Sachverhalt ist eine Verbindung von Gegenständen (Sachen, Dingen)

§ 2.01231 Um einen Gegenstand zu kennen, muβ ich zwar nicht seine externen – aber ich muβ alle seine internen Eigenschaften kennen.

(Wittgenstein, 1961)

Preface

On conclusion of a research project leading up to a PhD thesis, it is interesting to see whether the research questions initially posed have been appropriately addressed and answered. Admittedly, the questions as they appear in chapter 2.5 of this dissertation are not identical to the ones I set in my initial project description. The current versions have been altered a number of times, in accordance with the progress of the research work as well as the growth of my own theoretical familiarity with the research domain in which this work is situated. The thesis serves as documentation after completing a four year subsection of a continuous learning process.

One of the points identified in this thesis is that the students should become aware that it is the semantic meaning and logical soundness of a conceptual data model that counts and not the labels chosen for the different elements included. The same observation should be taken into consideration when reading this thesis. As the work has progressed, my perspectives and choices of wording may have changed accordingly. This implies that there might not always be full consistency in terminology and description of perspectives and conclusions between the different papers. The thesis offers descriptions of the underlying principles for the linguistic or semiotic aspects of data modelling as a socially situated activity, irrespective of the theoretical or methodological label used in such a description.

This work could not have been completed had it not been for the support from supervisors, colleagues, family and friends. My supervisors have been Associate Professor Andreas Quale at the Department of Teacher Education and School Development, and Professor Jens Kaasbøll at the Department of Informatics, both at the University of Oslo, Norway. They have both made an effort to keep up with, but also to restrain me from, my tendency to wander off into new disciplines searching for the ultimate solution or yet another perspective. When I was pressed for time in the closing phase of the writing, they both made themselves available at my convenience, including nights and weekends. I am grateful for their commitment.

The research described in this thesis has been carried out at the Department of Teacher Education and School Development at the University of Oslo. This has provided me with an abundance of colleagues and fellow students as enthusiastic partners for discussion. Even though none of them have their research interests in computer science education, or maybe because of that, this has given me most useful comments and perspectives. I would like to thank everyone who at some point offered their thoughts or advice, or just took the time to listen to my complaints. Special thanks to Dr. Erik Knain and Karl Henrik Flyum who jointly brought me along into the area of semiotics. I also owe a lot to Astri Eggen and Sten Ludvigsen for helping me get on the "right" track for writing the first part of the thesis.

The Nordic Network for Discourse Analysis two times funded a two week stay in Gothenburg, Sweden. This allowed me to improve my understanding of socio-cultural perspectives through working with the research group of Professor Roger Säljö at the University of Gothenburg and to visit Dr. Shirley Booth at Chalmers University. I have also had the privilege of receiving a three month Marie Currie scholarship to visit the Centre for Studies in Science and Mathematics Teaching at Leeds University, UK. This was a very productive visit for me, most of all due to the close and impressingly constructive guidance offered by Dr. Phil Scott, who was my supervisor for the visit. I also extend thanks to Dr. Jenny Lewis for taking care of all my practical concerns during the stay, and to the other faculty members for offering helpful thought and comments on early drafts of my work.

Thanks are due to Professor Nell Dale and her colleagues at the University at Austin, TX, for invaluable shepherding in the writing of paper 1. I am grateful to Professor Mordechai Ben-Ari at the Weizmann Institute, Israel, for pointing me towards the Psychology of Programming Interest Group. Attending their annual conferences has introduced me to a large number of merited researchers and fellow students within the field who all share openly their thoughts and offer their guidance. PPIG represents a unique type of scientific community in its embrace of anyone's opinions as equally valid irrespective of scientific merit or title. Thomas Green, Marian Petre, Sally Fincher, and all the rest, thank you very much.

Extra special thanks go to my wife, Katinka, who have spent countless hours transcribing teenagers' seemingly nonsensical discussions. Being a biologist, she has also forced me to justify and "tighten up" the argumentations of the papers when the sociological smalltalk has gotten out of hand. My father, Jørgen, has also been of great help with adjusting and tightening of the language in most parts of the dissertation. Other proof readers include, Andreas, Jens, Phil, Erik, Siri (my mother), and Karl Henrik. Thank you.

Katinka and our daughter, Stella, deserve special thanks for their tolerance in the final completion period, and not the least for coming with me to Leeds for three months. But most of all you deserve thanks for being the most important people in my life.

Oslo, March 22nd, 2005 Christian Holmboe

Table of Contents

List of papers and appendices

PAPER 1:

Holmboe, Christian (2005). Conceptualisation and Labelling as Linguistic Challenges for Students of Data Modelling. *Computer Science Education, 15*(2), 143-161.

PAPER 2:

Holmboe, Christian (2004). A Wittgenstein Approach to the Learning of OO modelling. *Computer Science Education, 14*(4), 275-294.

PAPER 3:

Holmboe, Christian, & Scott, Phil H. (2005). Characterising individual and social concept development in collaborative computer science classrooms. *Journal of Computers in Mathematics and Science Teaching, 24*(1), 89-115.

PAPER 4:

Holmboe, Christian, & Knain, Erik (2005). A semiotic framework for learning UML class diagrams as technical discourse. *Systems, Signs & Actions, submitted for review*.

APPENDIX A
The original transcripts in Norwegian for the excerpts presented in paper 1

APPENDIX B
The original transcripts in Norwegian for the excerpts presented in paper 4

1. Introduction

1.1. Setting the scene

Data modelling as activity operates in the intersection between software design and programming. It takes input from the problem domain to be addressed by the information system, and creates a description of this domain in terms that lend themselves to the rigorous procedures of programming (i.e. coding). Some sort of data modelling is often required to provide a manageable overview of a problem domain prior to embarking on the development of the implemented solution. In this respect, data modelling stands out as a particularly important topic for novice students to master in order to handle the complex tasks involved in system design and development. Accordingly, data modelling is increasingly taught as an essential part of system design and development in introductory computer science courses. A significant amount of research has been carried out, providing insight into various aspects related to the teaching and learning of computer science – in particular, psychological and organisational issues concerning introductory courses in programming, in addition to studies of expert behaviour. Some of the contributions made, and topics covered, are presented and discussed in chapter 2. The learning of system design and data modelling has, however, been far less focused on in computer science education research than is the case for the more traditional issues related to the learning or understanding of programming (McCracken, 2004). Contributing to the body of knowledge in computer science education research, this thesis addresses the learning of data modelling in school and undergraduate university computer science classrooms. Special attention is given to some aspects of this learning process where language plays an important role.

The first aspect studied, which was also the initial focus for this project, concerns the scientific concept building of students learning data modelling. Data modelling as an activity relies on scientific concepts like *connectivity*, *attributes* and different types of *keys*. The results presented concern students' understanding of *candidate key*, *primary key*, and *foreign key*. Emphasising that scientific concepts are not absorbed ready-made, but formed under influence from teaching and learning in social settings, Vygotsky states that "to uncover the complex relation between instruction and the development of scientific concepts is an important task."

(Vygotsky, 1986: p162). The study of conceptual knowledge in novices is accordingly seen as an important source of information for future design of teaching and facilitation of learning.

Furthermore, a conceptual data model is supposed to represent a subset of some problem domain (Peckham & Maryanski, 1988). In order to maintain a comprehensible link between the different parts of the data model and the “real world” features that they represent, it is common to label the components of the data model using terms from the language of the problem domain. It has been shown in studies of programming (e.g. Bonar & Soloway, 1985) that this mapping is not necessarily trivial. This thesis addresses the issue of labelling as the second major aspect in which language relates to the learning of data modelling.

Across both of these aspects, it is possible to discuss cognition and learning both on an individual level and as a socially distributed construction of knowledge. I will take a *distributed cognition* perspective adopted from Salomon (1993) in order to allow for discussions of both these levels of cognition as well as the interaction between them. This perspective will be discussed in section 3.2.

The inclusion of socially constituted cognitions introduces a third aspect of the relationship between language and the learning of data modelling. This last aspect concerns the collaborative problem solving activities in the classrooms as discursive practices constituting and shaping the collective construction of knowledge within both of the two first aspects. This third aspect has methodological implications, as it forms the rationale behind the link between choice of data collection method and research questions.

1.2. Outline of the thesis

The thesis consists of two parts. The second part comprises the four research papers, while the first part (chapters 1 through 5) includes the rationale and motivation for the studies as well as discussions of theoretical background, methodological considerations and a summary of the main findings.

Chapter 1 lays out the scope of the study in broad terms. The research field of computer science education is then briefly described in chapter 2 and the present work positioned in that context. Through the presentation of existing work, some issues that merit further research efforts are identified, leading to a set of research questions that will be addressed in this thesis.

Chapter 3 addresses the concepts included in the title of the thesis (i.e. language, learning and data modelling) and thus provides the theoretical framework and rationale both for the analysis and the discussion of the results. In the section dealing with language, emphasis is put on language and discourse as tools for mediation of meaning and the relationship between language and thought is discussed. This leads to a discussion of the nature and status of knowledge and learning as individual properties or as distributed social constructions. And finally, the activity of conceptual data modelling is described.

Chapter 4 presents some general methodological considerations for the data collection, and describes choices made in design of the study as well as in the analysis of the data. More detailed accounts of the specific methods for data collection and analysis are given in each of the research papers.

Chapter 5 is organised around the research questions with an aim to demonstrate the contributions of the individual papers to the addressing of each of these issues. This chapter also comprises a summary of some implications for teaching, as emergent from the main findings of the research papers, and a discussion of some limitations and shortcomings of the present study, with suggestions for further work.

For simplicity, the research papers are referred to as *paper 1* through *paper 4* throughout the thesis.

2. CSE research and scope

A large body of work has been published on topics related to CSE in different forms and places over the past four decades. Most of this work has emerged from one of three scientific research domains or academic fields: (1) Cognitive psychology, (2) Computer science teaching, and (3) Human Computer Interaction (HCI) or Computer Supported (Collaborative) Learning (CSCL). In this thesis there will only be room for a brief introduction to each of these, mentioning examples of work that are directly relevant for the issues addressed here. More comprehensive recent discussions of the history and scope of CSE as a research discipline can be found for instance in Détienne (2002), Fincher and Petre (2004), and Robins et al. (2003).

2.1. Cognitive psychology and programming expertise

The first category of research addressing issues relevant to CSE emerged within the area of cognitive psychology. In the 1960s and 70s there was a scientific focus on expertise and proficiency, and psychological experiments and measurements were made aiming to describe the characteristics of experts in domains like chess, mathematical problem solving, and, to an increasing extent, computer programming. The book, "Psychology of Computer Programming" by Weinberg (1971) is considered by many to be the first major contribution recognizing this field. The book was written with the purpose "to trigger the beginning of a new field of study: computer programming as a human activity" (Weinberg, 1971: p1 of preface), and deals with behavioural science aspects of programming as performance, including the use of tools, both on an individual and a social level of activity. The thoughts introduced by Weinberg were followed up by some further research in the 1970s, comprising few, but significant contributions (e.g. Brooks, 1977; Sime, Green, & Guest, 1973). Brooks outlines a psychologically based theory of programming behaviour. Keeping a cognitive psychology perspective, he uses theories of long and short-term memory as basis for an analysis of expert programmer behaviour as consisting of understanding, method finding and coding. Brooks work represents one of several noteworthy exceptions to the unfortunate pattern of lack of, or misapplication of, theoretical frameworks within CSE research as noted by Détienne (2002).

Over the following decade, substantial contributions were made to the knowledge of programming expertise. Some of the main findings from these studies are presented in various chapters of Hoc et al. (1990). An overall pattern for expert behaviour, as documented in these studies, is the ability to handle information at different levels in parallel (Petre, 1990; Soloway & Ehrlich, 1984). Détienne similarly emphasises that designers (and thus data modellers) "use knowledge from at least two different domains, the application (or problem) domain and the computing domain, between which they establish a mapping" (Détienne, 2002: p22).

This thesis will address students' knowledge and concept building related to the two domains introduced by Détienne. Successful data modelling depends on conceptual knowledge from the scientific domain of computing. The learning or acquisition of such knowledge will be referred to as 'scientific concept building'. In order to make a data model that maps sufficiently well to the problem domain, the data modeller also need a certain level of domain familiarity – including knowledge of the domain-specific terminology that will be used in labelling the elements of the data model. This second type of knowledge will be discussed under the heading of 'labelling'.

In accordance with previous findings (e.g. Visser & Hoc, 1990), Détienne (2002) furthermore describes the seemingly unstructured behaviour of experts as *opportunistic design*, with emphasis on the multi-dimensional nature of program design. A further characteristic of expert behaviour is the application of programming plans (Soloway, 1985) or schemas (Rist, 2004) in the problem solving process of program design. Plans have been defined as "generic program fragments that represent stereotypic action sequences in programming" (Soloway & Ehrlich, 1984: p 595). A brief overview of studies describing behaviour of expert programmers and designers can be found in Robins, Rountree and Rountree (2003). It has furthermore been shown that data modellers similarly rely to a large extent on heuristics and pre-memorized rules in their modelling (Batra & Antony, 1994b; Batra & Sein, 1994; Srinivasan & Teeni, 1995). These observations resonate well with the theories of pattern use in design (Gamma, Helm, Johnson, & Vlissides, 1995). Efficient use of programming plans or design patterns requires some experience, and is therefore less relevant for my study of novice data modellers. More relevant is the general characteristic implicit in these accounts of expertise as not needing to be consciously aware of what techniques to employ when solving a problem (Dreyfus & Dreyfus, 1986).

2.2. CSE practitioner reports

The second main strand of contributions has come from the community of computer science educators. Professors and lecturers worldwide have struggled with similar challenges trying to help their students to come to grips with the apparently difficult and complex activity of programming. This has generated a large body of practitioners' reports and a market for sharing of experiences and helpful suggestions. A main forum for these publications has been the annual north American based conferences of the Association of Computing Machinery (ACM) special interest group for computer science education (SIGCSE), and the equivalent European conference on Innovation and Technology in CSE (ITiCSE), also hosted by the ACM. These conferences are gradually shifting towards a firmer emphasis on theoretically and empirically founded scientific research work – something that is applauded by many, but also raise some scepticism among practitioners who fear to lose their valuable forum for the informal exchange of thoughts.

2.3. HCI and CSCL

Thirdly, there is an immense amount of research within the disciplines of Human Computer Interaction (HCI) and Computer Supported Collaborative Learning (CSCL) of which several studies contribute explicitly or implicitly to the body of knowledge in CSE.

HCI and CSCL research as such is outside the scope of this thesis. One area of research that has relevance to this study, is the development of programming environments for novices (see Guzdial, 2004 for an overview) Parallel to the introduction of Object-Oriented (OO) methodology to nonprogrammers, there has been an increasing request for visual programming languages and system development environments. This has given rise to software development systems like JBuilder, Jawiz, and BlueJ, aiming to help the understanding of programming constructs and reduce the cognitive demands of the programming activity. In an evaluation of different visualisation tools, the framework of Cognitive Dimensions (Green, 1989) is used to analyse the benefits and limitations of some of the most popular programming environments (Romero, Cox, du Boulay, & Lutz, 2003).

Concluding that the difficulty of co-ordinating the different types of additional representations (e.g. control-flow vs. data structure) needs to be considered, they emphasise the need for “more theoretical knowledge about the way these systems influence the comprehension of computer programs” (Romero et al., 2003: p417). One common feature of such environments is that they offer some sort of class or object diagram visualisation. While I will not study feature of data modelling environment as such in this thesis, I will focus on the concept building in novices working with some kind of data modelling environment.

2.4. Towards a scope for this thesis

Within the landscape of CSE research, this thesis addresses the learning of data modelling, with particular focus on some aspects in which language plays an important role in this learning process. These aspects are (1) scientific concept building, (2) choice and use of natural language terms as labels for elements of the data model, and (3) discourse as a mediating tool in collaborative learning environment. A second dimension for the analysis concerns the relationship between individual and collective cognitions within each of the two first language-related aspects.

Research on the teaching and learning of system development in general, and data modelling or database design in particular, has not been particularly prominent in the literature on computer science education (McCracken, 2004). One exception is studies comparing usability, user performance and suitability of different data modelling methodologies for different tasks (e.g. Batra & Antony, 1994a; Batra, Hoffer, & Bostrom, 1990; Liao & Palvia, 2000). Such studies have focused on the differences between, and affordances of, each of the approaches, aiming to establish which one is “better”. For example, several studies comparing relational and ER[1] methodologies have concluded in favour of ER (Chan, 1998). Relational and ER models represent *logical* and *conceptual data modelling*[2] methodologies respectively. This indicates that conceptual data models (e.g. ER) are easier to use, which is also

[1] Many of the studies referred to here distinguish extended entity relationship (EER) models (Elmasri, Weeldreyer, & Hevner, 1985) from the original entity relationship (ER) models (Chen, 1976). In recent years it has become common to refer, for simplicity, to both of these versions of the methodology as ER. As I have used ER in the research papers in this thesis, I am also using that acronym here regardless of what the individual authors have used in the papers cited.

[2] See section 3.3 for a description of what is understood by *conceptual data models*.

the general conclusion made by Liao & Palvia (2000) in their review of previous results.

Studies comparing ER and OO models (i.e. two different conceptual models) have lead to less uniform results. Shoval and Shiran (1997) found ER to be superior to OO in designing unary and ternary relationships and that ER is less time-consuming and preferred by designers. Bock and Ryan (1993) also found ER to provide improved performance on selected constructs, while other studies have found better user performance or model correctness using OO methodology as compared to ER (e.g. Liao & Wang, 1997; Palvia, Liao, & To, 1992). This inconclusiveness is probably due to a lack of agreement about criteria for evaluating the methodologies, and a lack of standardised research designs for making the comparisons.

A general shortcoming of many of these studies is that they "have not explicitly addressed causes that lead to errors in conceptual data modelling" (Batra & Antony, 1994b). In general, comparison studies of different methodologies or modelling languages tend to use *modelling performance* as a measure for appropriateness of the methodology. This might be a sensible measure to use for comparing the tool support offered to modellers at any particular level. It does, however, not provide a valid measure for the learning outcome from using the language or methodology in question. Theoretical frameworks like the *Cognitive Dimensions* (see Green, 1989 for an introduction) have successfully been applied to the study of graphical system development environments (e.g. Green & Petre, 1996; Kutar, Britton, & Barker, 2002). One benefit of this approach is that it introduces a systematic analysis of cognitive usability aspects of the different methodologies. As such, it is more relevant for learning than the studies comparing user performance. However, the focus is still on the affordances of the tool or methodology for *making* data models and not the affordances for *learning* data modelling.

With the purpose of improving the learning outcome, it is called for general empirically based descriptions of the cognitive demands raised by the activity of data modelling as such irrespective of the choice of methodology. The present study therefore aims to study the learning processes of students of data modelling without focusing on the specific tool or modelling methodology used.

Batra and Antony (2001) have developed and analysed a knowledge-based consulting system for novice database designers. Their work differs from previous attempts in that it is founded on empirical studies of typical novice errors in data modelling (Batra & Antony, 1994b). They show that the success rate[3] of constructing a data model to fit a certain requirement specification is a function of the number of entities and relationships involved, while an earlier study concludes that students had little trouble with the modelling of entities, whereas the modelling of relationships was much more difficult (Batra et al., 1990). In this thesis, I will address these difficulties of modelling relationships by considering them as entities in their own right (see paper 1).

2.4.1. Scientific concept building

Within the field of psychology of programming, a number of researchers have focused on the way in which programming languages differ from natural languages and the cognitive challenges related to this distinction. The primary focus for many such studies has been on the static semantics of programming languages (i.e. mainly procedural programming languages). It is shown that students tend to confuse natural language meanings of terms with the formalised versions implemented in a programming language (Détienne, 2002; Hoc & Nguyen-Xuan, 1990; Taylor, 1990). This is in part explained by the potential mismatch between the meaning of a term in everyday language and the intended analogous meaning of the term used in the programming language. English terms used in programming languages, like **then** in the **if-then-else** construct, or **while** in the **while**-loop construct, have slightly different meanings from the everyday connotations of the corresponding *then* and *while*. These somewhat counter-intuitive implementations are shown to lead to erroneous code (Bonar & Soloway, 1985; Shackelford & Badre, 1993). In order to avoid such errors and misconceptions, Pane, Ratanamahatana and Myers (2001) conducted a study of nonprogrammers' verbal solution strategies using natural language to address typical programming problems. They found that the subjects generally produced satisfactory algorithm descriptions, but that the descriptions differed from the style that is allowed in today's programming languages. The patterns observed in these natural language

[3] Success rate is here measured by the number of discrepancies between the model produced and the requirement specifications.

algorithms have subsequently been used as input to the design process of a new programming language (Pane, Myers, & Miller, 2002).

When everyday terms are used to denote formalised concepts functioning as constructs of a programming language, these become scientific concepts of computer science. One main focus for the research presented in this thesis is the development of understanding of similar concepts within the domain of data modelling.

Others show that the most frequent bugs made by students can *not* be explained by misconceptions about language constructs, but are due to general misconceptions in the students' mental models of the computer, or *notional machine*[4], and its functioning and affordances in relation to programming (Pea, 1986; Spohrer & Soloway, 1986). "The purpose of the notional machine is to provide a foundation for understanding the behaviour of running programs." (Robins et al., 2003). Hence, the notional machine for C++ is different from that of Java. Misunderstandings that have been documented typically concern the attribution of natural language plan knowledge to programming constructs (Bonar & Soloway, 1985). Understanding of abstract concepts like the notional machine is another example of scientific concept building that is important for computer science students. du Boulay (1986: p72) observes that the students very often "form quite reasonable theories of how the system works, given their limited experience, except that their theories are incorrect.".

Reviewing literature on cognitive consequences of the OO paradigm, Détienne (1997) points out that novices tend to have misconceptions about some fundamental OO concepts like *class* and *inheritance*. For example, they tend to conceive a class as a set of objects which leads them to attribute set characteristics and properties to their classes. Similarly, students see no need to create a class or an array for holding one element only. Sets are, in their experience, used for holding multiple objects while "one item can be carried simply as is" (Hazzan, 2003: p106).

Aharoni (2000) demonstrates an interactional development process between different levels of conceptual knowledge. Students' answers to the question "What is an array?" were categorised into Programming-Language Oriented Thinking, Programming-Oriented Thinking and Programming-Free Thinking according to the level of abstraction displayed. Abstraction in this sense is understood as a process of

[4] It is common to refer to the abstraction of a computer as a *notional machine* (du Boulay, 1986; Hoc & Nguyen-Xuan, 1990).

reification where *actions* on objects at one level turns into *objects* in their own right at the next level of abstraction (Sfard, 1991). An explanation of an array as “a variable with an index in brackets behind it” is a typically example of Programming-Language Oriented Thinking, while “a set of ordered pairs, where one element of the pair has distinct values…” would be an explanation that indicates abstract Programming-Free Thinking. Identifying the students’ level of abstract thinking is essential for gaining insight into their conceptual understanding. The level of abstraction in students’ explanation of scientific concepts will therefore be studied further in this thesis.

Many of the studies mentioned above focus on the importance of sound conceptual understanding for successful programming or system design, and on particular misconceptions held by students. Such findings provide vital information for teachers by informing them of what misconceptions they should help the students to avoid. Less attention has been given to conceptual knowledge in data modelling. Since data modelling is increasingly taught, not only to computer science majors, it is important to gain similar knowledge about possible misconceptions of scientific concepts like, for instance, *connectivity*, *attributes* and different types of *keys*. In addition, it is important to study the manner in which these understandings develop.

In a study of practitioners in the commercial domain (Hitchman, 1995), it was found that modellers do not have a solid understanding of some semantic constructs. The constructs measured comprised recursion, entity sub-types, orthogonal entity sub-types and exclusivity. The study measures the subjects’ ability to apply these constructs correctly in a modelling problem, which may well be an indicator for having sufficiently grasped the function of the construct, but does not reveal qualitative information about misconceptions held. Anecdotal reasoning is offered to suggest possible reason for these problems, but no empirical evidence is provided in that respect.

One objective for the research presented in this thesis is to contribute to the knowledge of the nature of students’ understanding of scientific concepts in data modelling and of the processes that lead to this understanding.

2.4.2. Labelling

The establishment of a mapping between the problem domain to be represented and the logical/physical data structures as they are stored in the computer, is a main objective for data modelling (Peckham & Maryanski, 1988). Choosing appropriate *labels* for entities, classes, attributes or variables is an important task in this respect. In order to be able to understand the semantics of a program or data model, it is an advantage to choose intelligible terms associated with a vernacular meaning that resembles what the labelled constructs are supposed to represent. Use of natural language terms as names for variables has been found to improve understandability of code as well as programmer performance (Shneiderman, 1980). For most programming languages, this choice does only have influence on the *understanding* of the program, not on the program's *performance* on execution. It is generally assumed that using short, simple and consistent naming conventions help understandability of programs (Robins et al., 2003). In light of the discussion in section 2.4.1, however, it may be hypothesised that the use of natural language terminology could also obscure the "real" semantic meaning of the construct that it denotes in the program or data model at hand. This will be addressed in the present thesis.

Herbsleb, Klein, Olson, Brunner and Olson (1995) found that object-oriented design (OOD) seemed to help the communication between members of a design team with respect to establishment of common understanding of the semantics of the design elements. Using OOD as compared to procedural programming, the members of the design team seemed to be more elaborate, and ask each other more profound questions, enforcing more explicit definitions and explanations of the functions of features introduced to the design. Such establishing of common knowledge is indeed crucial for successful collaborative design. While Herbsleb et al. studied professional software developers, this thesis addresses common knowledge and collaborative design in novice data modellers. In doing so, the focus is on the extent to which the semantic meaning of terms used as labels is negotiated between the participants, or if it is taken for granted based on preknowledge from everyday language.

Bürkle, Gryczan and Züllinghoven, (1995) found evolutionary prototyping to be invaluable to the successful development in a large OO project in the realm of banking. Among the specific reasons for the success of the project was the enabling of communication between developers and different groups of users representing independent work cultures within the customer organisation. It appeared that the members of these different parts of the organisation had slightly different understandings of the concepts they employ. The authors emphasise the importance of basing the design on the concepts of the application domain, and of maintaining the class hierarchy model as close as possible to the model of the application domain language. To do so, the developers need to familiarise themselves with the domain specific terminology and the ways it is deployed across the enterprise. Ensuring a suitable basis for communication, the users are then able to understand and approve the data models constructed, and subsequently even contribute to the further developments of the project. What is of essence to the present thesis, is the coexistence of different understandings of concepts from the application domain, and how these are employed in a data model. I will investigate the manner in which novice data modellers are able to benefit from using everyday or problem domain terminology in labelling of entities and constructs of their data model.

2.4.3. Collaborative learning practices

The last example in the previous section points to the importance of collaboration for successful software development. Bürkle et al. explicitly state that they "view system development essentially as a learning-and-communication process." (Bürkle et al., 1995: p294). The cognitive ergonomics of programming and software development has also been studied by others as a social activity on expert and professional levels (e.g. Curtis & Walz, 1990; Détienne, 1997), and recently, the benefits of pair programming (Williams & Kessler, 2003) for professional software development has been increasingly stressed. The study by Herbsleb et al. (1995) shows that software design professionals use clarification questions extensively in order to ensure a common understanding of the implementation they are designing within a team. Williams and Kessler have also brought this discussion into the classroom, investigating the potential benefits of introducing pair programming in introductory computer science education (Williams, Wiebe, Yang, Ferzli, & Miller, 2002). They found that the students practicing pair programming have better performance on programming projects, are more self-sufficient, and demonstrate higher order thinking skills.

Dietrich and Urban (1996) also present positive performance results from an experiment involving collaborative student groups in an introductory database course. Their focus, though, is mainly on the practical aspects of organizing the course, rather than on the cognitive issues related to the students' learning outcome. This approach is characteristic of a lot of the work referred to in section 2.2. It is what Holmboe, McIver and George (2001) call “reports from the trenches”, typically focusing on the organisation of introductory courses in programming. Even though they provide a valuable resource for practitioners, such papers do not contribute to the empirically based body of knowledge about learning in computer science. This could have been achieved if the study was coupled with a discussion of the implications of collaboration for learning outcome based on theoretically founded argumentations. Such implications are discussed in this thesis.

McCracken (2004) emphasises the need for studies that take a situated perspective on learning in order to move forward in the accumulation of insight into the learning processes of system design as they take place in authentic settings. Some examples of such studies can be found. Kolikant (2004) describes *fertile zones of cultural encounter*, in which learning emerges in the meeting point between the discourses of different communities of practice (i.e. students and IT professionals). She points out that there are at least two different scientific sub-communities coexisting in a classroom. The teacher represents the academic community of computer scientists, while the students bring with them legacy from everyday computer oriented discourse and understanding. The target for vocational computer science teaching is a third community of practice – the one of IT professionals. There are in other words multiple communities of practice that all have their own ways of “doing computer science” using language in slightly different ways. Other studies also show that groups of students have their separate and distinctive ways of using scientific language in the classroom, and that these are neither adopted from the teacher nor from the textbook definitions of terms and their interrelationships (Levi & Lapidot, 2000; Taylor, 1990). This calls for further investigation of what characterizes the development of these specialised ways of using scientific language in the classroom, which will be another main concern for this thesis.

Taylor’s study furthermore describes a multi-levelled framework for analysing the different types of discourse that coexist in a programming situation. The framework comprises general problem solving discourse, formal problem solving discourse, logical discourse and mechanistic discourse. Taylor found that “students used tacit knowledge of human discourse processes both to interpret the *language* used to communicate with the computer and to interpret the *behaviour* of the machine.” (Taylor, 1990: p283) and that they did not seem to appreciate the differences between natural and formal discourse. Contributing to the knowledge of how different discourse types are handled by novices, I present a similar framework for analysing different types of discourse and their interdependencies (see paper 4).

2.5. Research questions

The discussion above leads to the formulation of the following research questions for the present thesis:

Scientific concept building

Q1: What characterizes novice data modellers' acquisition and knowledge of the scientific concepts of *keys* as used in the domain of data modelling?

Labelling

Q2a: Do novice data modellers benefit from using natural language terminology when labelling entities/classes?

Q2b: What characterizes novices' concept building processes related to labelling elements of a conceptual data model?

Collaborative learning practices

Q3a: How are the concept building processes of novice data modelling students influenced by the discursive practices of the classroom environment in which they take place?

Q3b: How do novice data modellers handle the coexistence of, and interdependencies between, different discourse types when engaged in collaborative problem solving activities in a computer science classroom?

3. Concepts and perspectives

The title of this thesis introduces three main concepts: language, learning, and data modelling. In this chapter I will discuss the roles of and interdependencies between these concepts, and establish how each of them should be understood when reading this thesis.

3.1. Language and discourse

Language and discourse can be, and have been, defined in many different ways. In the following, language should be understood as a tool for mediation of meaning mainly through talking or writing. There is a deliberate use of action-oriented terms in this description (i.e. talking and writing) because language as a tool has little interest unless it is used to perform actions. These actions occur in discourse. Discourse should accordingly be understood as "texts and talk in social practices" (Potter, 1997: p146), i.e. exchanges and development of meaning by use of language.

For the research presented in this thesis, language and discourse plays significant roles on several levels. Halliday has proposed a threefold perspective of "learning language, learning through language, learning about language." (Halliday, 1993: p113). This framework nicely illustrates the aspects elaborated in this thesis. Firstly, the research questions address the learning process related to semiotic topics like scientific concept building and categorisation, which corresponds to Halliday's perspective of "learning language". Language as a means for describing parts of the world, either scientific concepts or features of a problem domain, is thus a major part of what this research is about. In order to address this, I need to establish how language is related to the world that it describes.

Furthermore, the analyses also address language as a mediating artefact used in the discursive practices of the classroom, i.e. "learning through language". This calls for a discussion of collaborative negotiations of meaning through discursive interaction.

In the papers, I use written and spoken language as empirical data for studying learning and cognition. I therefore need to address the nature of the relationship between language and thought, both on an individual and on a socially distributed level. This latter aspect also brings me back to the initial issue of learning language, since concept building processes (i.e. attribution of meaning to terms or expressions) also concern the relationship between language and thought.

In discussing the implications of the study, I will focus on the need for metalinguistic awareness, i.e. "conscious knowledge about the use of language", as an important prerequisite for enabling the novice data modellers to handle the different discursive practices and ways of meaning through use of language that are incorporated in the practice of learning data modelling.

3.1.1. Language and thought

Taking a discursive approach to studying cognition, it is necessary to establish a theoretical rationale for linking the students' discursive behaviour to their individual as well as their distributed cognitions. This is not a trivial link, and in fact one that is still much disputed.

In traditional psychological research, language has been described as a mirror of, or a window on, the mind. As a consequence of such a view, language and discourse have been used as basis for making claims about mental activity. Coupled with a constructivist view of knowledge as individually constructed mental representations of the experiential world (see section 3.2), it is possible to study answers to structured interviews, or to use other experimental setups, to make inferences about a person's subjective understanding of some concept based on their discursive behaviour.

In the words of Vygotsky, "the meaning of a word is such a strong amalgam of thought and language that it is hard to tell whether it is a phenomenon of speech or a phenomenon of thought" (Vygotsky, 1986: p212). Meaning is an intrinsic part of both word and thought. And, what is equally important, meaning develops. Meaning is socially negotiated through discursive interaction and will therefore be altered over time as these negotiations continue. This makes it very difficult to maintain the position of discourse being a "window on the mind", since meaning in this sense would be a context-sensitive phenomenon (Edwards, 1997). However, if we bring this context into our analysis, through considering learning and discourse as situated practices, we should still be able to analyse cognition and thinking as they become visible through our discursive activities. While Vygotsky claims that meaning is equally bound to language and to thought, Wittgenstein states that the meaning of a word is defined by the way it is used (Wittgenstein, 1958). This view ties meaning more explicitly to discursive practices, which in turn makes it less problematic to use language as a means for analysing thought and cognition. Still, meaning is not seen as static. On the contrary, Wittgenstein emphasises the dynamic development of meaning in different language games. In the words of Mercer; "Words mean what humans agree together to make them mean." (Mercer, 2000: p4).

3.1.2. Concept building

The question is what it implies to know or understand a concept. The research questions of this thesis address two different types of concept building. One is the learning of the scientific concepts of computer science, which is an example of an institutional language (as defined by Drew & Heritage, 1992), while the other can be described as the redevelopment of everyday concepts that are associated with slightly new meaning content through transfer by grammatical metaphor[5] (Halliday, 1998). The scientific concept building processes are quite different from the concept building processes of everyday situations (Vygotsky, 1986). In everyday language concept building is a bottom-up process, in the sense that we first learn how to use the concepts and then later how to define them. This implies that there are communicative and bodily referents for everyday concepts like "criminal" (see paper 1) or "account" (see paper 4). In institutional languages, conceptual distinctions are developed in a different manner. Firstly, the concepts are generally dependent on explicit definitions, both of their intended meaning and of their interrelationships. Secondly, their referential function is special in the sense that their use in language most often is not based in human experiences. It is plausible to assume that this difference in the conditions for reference may cause the learner to get misguided, since he or she will be likely to use everyday meaning and experiences as their contextual frame for understanding the concepts. This distinction between scientific and spontaneous concept building was established by Vygotsky (1986), and is also briefly presented in paper 1. A further discussion of the differences and interrelationships between everyday (i.e. vernacular) and institutional (i.e. scientific) lexis can be found in paper 4.

[5] See paper 4 for a detailed discussion of grammatical metaphors and related semiotic mechanisms.

Even though Vygotsky describes the development of spontaneous and nonspontaneous concepts as two different, or even opposite, processes, he also emphasises that these two processes are related and constantly influence each other. In fact, he states that “they are part of a single process: the development of concept formation,” (Vygotsky, 1986: p157).

3.1.3. Language and the world

There is not a one-to-one correspondence between term and meaning. Vygotsky bases his theory of the relationship between thought and language on the realist view that ontologically independent objects exist. For these objects, formal expressions are introduced in the form of words that we use to represent them in oral and written language. The connection between the object and the formal expression is, however, not a direct one. Each individual ‘assigns’ a subjective content to the term, linking it to the object. This subjective content corresponds to the person’s cognitive perception of the object being referred to. The relationship between a term and the “physical” construct that it is perceived to represent is thus determined through the mental representation held by the user of the term. In the previous section, I made the claim that language can not be seen as a direct expression of mind, but rather that it is shaped, and to a certain extent made visible, through the way it is used in discourse. In a similar manner, language should not be perceived as a reflection of the world. In stead, “the world is at issue in discourse” (Edwards, 1997: p20).

In the research presented in this thesis, I take the position that an ontologically independent reality exists. The focus of interest is then on the ways in which this reality is handled through language in situated practices. One aspect which complicates things here is that in dealing with data modelling, there is more than one such referential world simultaneously involved, namely the problem domain, the conceptual domain of the data model and the logical or physical domain of the database system as implemented on some computer. To each of these domains, which should be handled as equally real and important, there is at least one set of lexical expressions potentially corresponding to one or more elements of the domain. The same terms may simultaneously be used to denote a corresponding or a different element of one of the other domains. This may appear unnecessarily complicated. However, to anticipate the results of the present study somewhat, this complexity seems to be at the heart of some of the problems faced by novice students of data modelling. This complexity also brings me back to the distinction between scientific and everyday or spontaneous concept building, as discussed in the previous subsection. In the case of data modelling it is not always evident what should be considered as scientific and what are spontaneous concepts. When an everyday term like “students” is used to denote an entity type in a data model, it takes on a highly specialized meaning that cannot be inferred from its use, but must be explicitly defined. It thus takes on the characteristics of a scientific concept, although much of the understanding of its meaning is still based on the spontaneous concept. In yet another data model, the term “student” can be used again, but this time with a third meaning. In this manner, there can be a number of sublanguages existing in parallel, that have elements of both types of concept building processes[6].

“Cognition and reality are like two sides of a coin. If we want to know about cognition, we need to take account of the world, hold reality constant, or vary it systematically, so that we can discern the workings of mind. If we want to know about reality, it is cognition and other human foibles that have to be held constant or under control.” (Edwards, 1997: p10). I have thus established that language is intrinsically related to thought through *meaning*, and similarly that thought or cognition is mutually related to reality in terms of mental representations. These two relationships seen together should in theory give us a link between language and the world. However, since both meaning and representations are dynamic and therefore change over time and between contexts, it is difficult, and outside the scope for this thesis, to describe the relationship between language and reality as such. What is of interest here is the ways in which language and reality (i.e. the problem domain for data modellers) are handled and dealt with in discursive practices.

3.1.4. Language games and common knowledge

Talking about discursive practices implies some kind of social interaction. It is therefore also necessary to address briefly the socio-cultural aspects as a fourth dimension related to the ternary relationship discussed so far (i.e. language – thought – world). When discussing the social aspects of discourse and formation of meaning, it is inevitable to touch upon issues of cognition and learning. In section 3.2, I will establish learning as a situated and socially dependent practice, which implies that

[6] Further details of spontaneous versus scientific languages in data modelling are addressed in papers 1 and 4

communication (i.e. discursive interaction) plays an important role (Edwards, 1997; Mercer, 1995; Scott, 1998). A prerequisite for successful communication – and hence for learning – is that the interacting parties find a platform of 'common knowledge' (Edwards & Mercer, 1987). In discursive interaction with other individuals, there is a need for a common frame of reference to give the sense that we understand each other. Such a common frame of reference is not automatically present. Since each person 'assigns' his or her own semantic content to the different terms, the subjective content will vary. This conceptual incompatibility is often not evident in a conversation – especially not when referring to relatively noncomplex phenomena like tables or chairs. When moving on to more abstract themes, the incompatibility will be more obvious and participants might even feel that they are not talking about the same thing (Glasersfeld, 1989). According to Mercer, "misunderstandings regularly arise, despite our best efforts, because there is rarely one unambiguous meaning to be discovered in what someone puts into words." (Mercer, 2000: p5).

Take for example the term "brother" used by Piaget in his studies of concept building in children (Piaget, 1959). When this term is used by a member of the African-American community, it should probably not be understood exclusively as a male person that has one or more siblings. In this cultural setting, the term "brother" is often used to refer to another member of the African-American community, reflecting the implicit kinship between members of a suppressed societal minority. It is thus imperative for successful communication that the participants in the discursive practice share a common frame of reference; that they have common knowledge on which to base their semantic interpretations of the utterings or speech acts made by the other parties. This common knowledge is, however, not necessarily something that can be appropriated from a given set of understandings that is accepted as valid in a particular social context. The meanings that participants attribute to the discursive acts are negotiated through the same discursive acts in social interaction between the participants.

We would expect most members of the community of English speaking African-Americans to recognise the "right" meaning of the term "brother" from the way it is used in discourse, because they do indeed have such common knowledge. The meaning of the term has thus evolved (i.e. transferred by grammatical metaphor) from its original significance, to become incorporated in a locally constituted *language game*[7] (Wittgenstein, 1958). "Every time we talk with someone, we become involved in a collaborative endeavour in which meanings are negotiated and some common knowledge is mobilised." (Mercer, 2000: p6) In this manner, locally functioning language games are developed through discursive practices in which the meanings of individual terms are negotiated and therefore may evolve or change with their use over time.

3.1.5. Metalinguistic awareness

In the following, I introduce some concepts that are used in this thesis as a means for discussing students' cognition in relation to language. By *linguistic metaknowledge* I mean knowledge about one's own knowledge of language and communication. In order to give a justified account of the linguistic aspects concerning the learning of data modelling, it will be convenient to also introduce the notion of *metalinguistic knowledge*. The latter should be understood as knowledge about the way in which language is used to describe or represent semiotic processes (i.e. meta-semiology (Andersen, 1990)).

Vygotsky uses the example of having just tied a knot, explaining that "I have done so consciously, yet I cannot explain how I did it, because my awareness was centred on the knot rather than on my own motions, the *how* of my action." (Vygotsky, 1986: p170). When we speak, we are similarly not *aware* of how we use language to do the meaning making that using language implies. In paper 1, this notion of being aware of the ways in which language is used, to do and mean different things in different contexts, is referred to as *metalinguistic consciousness*. In order to avoid the confusion potentially created from the various interpretations of the concept of *consciousness* in literature relevant to this thesis[8], I have chosen to substitute this concept by *metalinguistic awareness*. Notice the slight distinction in meaning between *metalinguistic knowledge* and *metalinguistic awareness* in that *knowledge* is concerned with what a person knows (i.e. is able to do), while *awareness* implies being consciously aware of this ability.

[7] See paper 2 for a discussion of Wittgenstein's notion of *language games* and their relevance for the learning of data modelling.

[8] The Freudian understanding of unconscious as a repression implies a late development (i.e. to follow after consciousness). This differs from the Piagetian understanding of unconscious as ”not yet conscious” (i.e. a temporary state on the way to consciousness), and from the Vygotskyan sense of consciousness as awareness of the activity of the mind (Vygotsky, 1986).

3.2. Learning and knowledge

Based on the theories of Piaget (1954), and further development by von Glasersfeld (1989), constructivism has held a strong position as the leading epistemological tradition with respect to learning until recent years. Constructivism describes learning as individual construction of knowledge, through reflection on experiences as seen against the backdrop of prior knowledge. In later years, the social aspects of this learning have been increasingly acknowledged, countering some of the criticism (Matthews, 1998) that have been raised against the purely individualist perspective of radical constructivism.

The increasing emphasis on social context as a decisive factor for learning has given rise to new strands in epistemological research. Situated cognition (Anderson, Reder, & Simon, 1996; Hennessy, 1993; Lave & Wenger, 1991), activity theory (Engeström, 1999), apprenticeship (Rogoff, 1990; Wood, Bruner, & Ross, 1976) and socio-cultural perspectives (Säljö, 1999; Wertsch, 1985) are all theoretical frameworks that place the learner in a social context. These are highly influenced by – if not directly founded on – the theories of Vygotsky (1978; 1986; Wertsch, 1985), which were made available to the international society in the late 1970s after 40 years under Soviet censorship (Kozulin, 1986). According to these theories, learning cannot be seen as independent of the context in which it occurs. The social setting is not only treated as *relevant* for the learning process (as emphasised by the social constructivists (Driver, Asoko, Leach, Mortimer, & Scott, 1994)), but it is seen as *crucial* for the learning outcome in general, and the transferability of the resulting knowledge in particular.

One main difference between social constructivism and the socio-cultural perspective is the view of what knowledge is, and accordingly how learning happens. As already mentioned, all constructivist theory is based on the key assumption that knowledge is individually constructed as mental structures or schemas. This is not to say that students are expected to construct, for example, the laws of physics for themselves from empirical observation. It is rather a statement concerned with where the knowledge resides, and what constitutes the main processes of knowledge construction. In the socio-cultural perspective, *knowledge* is described as the ability to participate in cultural practices and *learning* as the acquisition of such ability. Maintaining the different viewpoints of these theories, it is important to emphasize that they do not necessarily stand in conflict with each other. Rather, they offer contrasting approaches to the analysis and explanation of learning and knowledge, and may as such even complement each other on some occasions (Sfard, 1998).

Vygotsky introduced the notion of *Zone of Proximal Development* (ZPD) as the discrepancy between a person's individual mastery level "and the level he reaches in solving problems with assistance" (Vygotsky, 1986: p187). According to this theory, all learning takes place within the ZPD, preferably in the interaction with a *more competent peer* (Lave & Wenger, 1991). What can be immediately learned is in other words limited, and the learner will benefit from assistance or guidance in acquiring new skills and knowledge. This is predominantly an interactive process which has been described as *cognitive apprenticeship* (Collins, Brown, & Newman, 1989). Like other kinds of apprenticeship, the learning activity is based on the participants (i.e. novices) solving problems under the supervision or in collaboration with a more skilled peer (i.e. expert). This assistance (termed *scaffolding* by Wood et al. (1976)) can then be gradually removed until the learner has become a competent autonomous participant of the social practice at hand. In the literature characterising cognitive differences between novices and experts, it is indicated that "experts spend years acquiring intuitive specialist knowledge and sophisticated mental models of their domain." (Hennessy, 1993: p1). The mental models thus created are highly influenced by the social context in which this problem solving takes place. These issues are also discussed in the introductory section of paper 1.

Furthermore, "cultural transmission plays a major role in the construction of expertise." (Hennessy, 1993: p1). In order for the learner to be able to appropriate the practices inherent in a community, these practices need to be made accessible to the learner, either explicitly or through demonstration and observation (Lave & Wenger, 1991). In this way, the proponents of the socio-cultural perspective (e.g. Mercer & Wegerif, 1999; Säljö, 1998) emphasise that the transmission of meaning is mediated through tools or artefacts (including language).

3.2.1. Distributed cognition

The mediation of meaning through language can also be seen as a way of allocating knowledge by means of a contextual artefact, and thereby making it accessible to, or indeed distributed across, a whole community. This is a central aspect in the theory of *distributed cognition* which is described below.

Salomon (1993) discusses to what extent there is room for considering individual cognition within a distributed cognition perspective. I will start by asking the same question the other way around. Given that we accept the existence of an independent ontological reality, and that we acknowledge the existence of individual knowledge as mental representations of this reality, can we still take a distributed or socio-cultural perspective on knowledge? To answer this, I need to distinguish between two different understandings of the term *knowledge*. On the one hand, there is the cognitivist and individual focused understanding of knowledge as individuals' *cognitive representations* (Piaget, 1954). On the other hand, one can consider knowledge as referring to "the sum of what is known to people, the shared resources available to a community or society (as in 'all branches of knowledge')" (Mercer, 2000: p8). Knowledge in the latter sense exists mainly in the form of written or spoken language. As such, it cannot be attributed to any particular individual, nor can it be divided between the individual members of a community. A distributed view of knowledge does not, in other words, mean that cognitions are shared between the individual participants so that each member of a community holds their individual part of an aggregated body of knowledge. It is rather a question of cognitions that are "stretched" over the group, and in that sense only exist as an integrated part of the whole that cannot be divided into their individual components (see e.g. Salomon, 1993).

From a socio-cultural perspective, I have described individual knowledge as the ability to participate in cultural practices. Building on this, distributed cognitions imply that knowledge is seen as the community's ability to perform social tasks and to engage in these practices. The appropriation of competencies of a community is manifested by the ability of the participants to collaboratively utilise the tools available to them. This claim is best understood within a cultural-historical frame. At different times in history, people as members of communities of practice have gradually appropriated new skills and taken new tools into use for solving various tasks (Säljö, 2000). This has lead to a higher need for specialisation (division of labour (Engeström, 1999)), while the collected body of knowledge (i.e. accumulated set of skills and abilities) has increased immensely.

This brings me back to Salomon's question of whether there is room to consider individual cognitions within a distributed cognitions perspective. According to Salomon (1993) cognition cannot be exclusively described as being either collective or individual. Rather, the collective and individual cognitions must be understood and examined in interaction. For instance, the ability to couple a semiotic symbol (e.g. a term) to a semantic meaning must in some way be coupled with the individuals' minds as discussed in section 3.1. Speaking of collective cognition in this respect must therefore be limited to the individual members of a cultural group arriving at compatible meanings when individually interpreting a semiotic representation. This is closely related to the concept of *common knowledge* or *common frame of reference* (Edwards & Mercer, 1987).

I have already adopted the theories of Vygotsky to account for the importance of the social influence on the learning and concept building processes of children. Yet, to talk of the social interaction as an influence on knowledge construction implies the acknowledgement of such a thing as individual knowledge construction in the first place. "The development of nonspontaneous concepts must possess all the traits peculiar to the child's thought at each developmental level because these concepts are not simply acquired by rote but evolve with the aid of strenuous mental activity on the part of the child himself." (Vygotsky, 1986: p157). Vygotsky, like Salomon, thus acknowledges the mental activity of the individual as key to conceptual development. Maintaining a predominantly socio-cultural perspective, Mercer also admits to the significance of individual cognition and some form of mental representations theory. He suggests that "communicative activity, and individual thinking have continuous, dynamic influence on each other." (Mercer, 2000: p9). He argues that taking such a position invites studies of the joint creation of knowledge, as well as the interrelationships between individual and collective forms of knowledge.

In order to be able to address the research questions presented in chapter 2, I will base my discussion in this thesis on a dualistic or pragmatic view of knowledge, allowing for the consideration of both individual and collectively distributed cognitions. While some of the analyses presented are partly rooted in a cognitivist tradition, focusing on mental representations of individual students (i.e. paper 3), other discussions more clearly take a socio-cultural or situated cognition perspective as their theoretical point of departure.

3.3. Conceptual data modelling

Data are the building blocks or elements of information. A data model should accordingly be understood as a model of information elements and the interconnections between these. When talking about data models in a computer science sense, we are usually referring to database models. The piece of the "real world" that is represented in a database is commonly called an *enterprise* (Peckham & Maryanski, 1988). A data model is thus a model of entities or objects representing information elements of the enterprise and their interrelationships. The information structures of an enterprise are usually not static. Database models therefore also need structures for modelling operations used to manipulate the objects of the database schema.

A main challenge in modelling the real world structures of an enterprise is the discrepancy between human perception and the computer's need to organise information for mathematical and logical processing and storage. It is common to operate with three database modelling levels that reflect (1) the user's mental understanding of the problem domain (external level), (2) the physical model of the machine concerned with paths and storage (internal level) and (3) the mapping from one to the other of these two (conceptual level) (Peckham & Maryanski, 1988). Within this framework, most modelling methodologies can be seen as *conceptual models*.

An early contribution addressing cognitive issues of data modelling was made by Smith and Smith (1977), who introduced the notions of aggregation and generalization of abstract phenomena using one common primitive to form generic objects. Their work predates the introduction of graphical modelling methodologies, but still emphasizes the need to ease the cognitive demands of data modelling through simplification. With the introduction of graphical modelling languages like ER and later UML, it has been an aim to enable the data modellers to make representations of the enterprise that parallels the user's perception as closely as possible without concern for the physical model. A commonality of *conceptual data models* is that they enable the user to model the data in a manner similar to the human perception of the application, without having to be concerned about the details of the physical structure of the database. This ideal is also referred to as *closeness of mapping* (Green, 1989).

For the purpose of this thesis, data models will be understood as conceptual data models, referring to structural models of the concepts and constructs of a problem domain and their interrelationship including operations on these constructs.

Readers who are unfamiliar with data modelling, or with ER models, can find a general introduction on page 92 of paper 3.

4. Methodological considerations

4.1. Abduction and "Method of Science"

The research approach taken in this thesis is best described by what Fincher and Petre (2004) calls "Method of Science". To counter the scarcity of theory in computer science education as a research discipline, they propose a broader way of thinking about gaining scientific knowledge. "Method of science values description as well as hypothesis generation [and thereby] embraces both inductive and deductive reasoning." (Fincher & Petre, 2004: p11). With "Method of Science", the focus is not on generating predictive theory through testing of hypothesis generated from previous theory, but on articulating and making explicit the contributions of the research to the scientific discourse of a research field. The choice of method as such is not as important as the argument for the method chosen. This is a pragmatic approach to research that opens up for use of any research method that can be argued to contribute to the process and discourse that may or may not lead to predictive theories in the long run. Even so, rigor is demanded and the aim is still to contribute to empirically-founded theory.

Abduction is an alternative to the more widespread methods of *induction* and *deduction*. Like "method of science", abduction uses a combination of inductive and deductive reasoning. Abduction implies using existing knowledge and referential frames to find theoretical patterns that, *if they were correct*, would make sense of an empirically inductive pattern that has been found through interpretation of a single case. This abduction should then ideally be strengthened through repeated application on subsequent cases (Alvesson & Sköldberg, 1994; see Hanson, 1958 for a further discussion).

In this thesis, interpretations of observations in data (cases) are attempted explained by using established theories from linguistics, psychology, philosophy and computer science. Interpretation of further observations within the framework thus established is done only after first having identified these initial observations and their potential explanations. In this way, the work is an example of abductive reasoning, and in keeping with the standards laid out for "method of science".

4.2. Design

The underlying focus, driving the design of this PhD work, has been to investigate the concept building processes in students learning data modelling. When choosing the data collection methods for the study, it was an aim to collect as rich data as possible. Since Computer Science Education research is a theory-scarce discipline (Fincher & Petre, 2004), the work was designed to accommodate data-driven reasoning, aiming to generate, rather than to test, hypothesis. The research questions as they have been stated in this thesis were accordingly in part formulated and refined post hoc, based on the emergence of interesting observations from the analysis of the data.

4.2.1. Sampling of subjects

In Norwegian high schools some aspects of computer science have been offered as optional courses over the past couple of decades. Since 1994, the main topics covered deal with information systems and system development, with an emphasis on data modelling and implementation. These same topics are also covered in the introductory curriculum for informatics students at university level, but with a somewhat more theoretical orientation and a broader coverage in terms of methodologies. Data modelling and system development methodologies have also been a core topic for computer science courses in business schools, as well as in training programs for professionals.

High school students were chosen as the main target group, in order to have subjects of the same age and academic background, and who were ideally not too biased (from previous computer science experience). With the aim to collect as rich data as possible, I also included university students in the sampling of students for the studies presented in this thesis.

Neither demographic distribution nor representativity for a larger population were considered important for the design. It was, however, important to include samples that in some respect represented ordinary (if not necessarily typical) computer science classrooms. The subjects observed were students of two computer science teachers who were both teaching the second year of the same two-year computer science course at their respective schools, and who both had a few years of experience in doing so. One of the teachers had two classes in this subject, while the other had one. At the university, I was allowed to use the classes of four of the senior students who were involved as tutors in the course.

Data set 3 (see next section) comprised the classes already mentioned as well as six additional high school classes. These were the remaining computer science classes at the two schools that I visited – five first year classes and one second year classes.

All students were informed about their legal right to abstain from taking part in the study. No students chose to do so[9].

In terms of data modelling methodologies, the high school students were doing Entity Relationship (ER) modelling (Chen, 1976), whereas the university students were working with object-oriented (OO) modelling with the Unified Modelling Language (UML) (Booch, Jacobson, & Rumbaugh, 2001). Of the different diagram types included in UML, only class diagrams are discussed in this thesis. Class diagrams resemble ER diagrams in several ways, and it is therefore easier to compare results and observations across the different studies. Also, class diagrams are one of the most common ways of doing conceptual data modelling with UML.

4.3. Data collection

The data material mainly consists of qualitative data collected from student interactions in natural classroom settings, but also on some quantitative data, as well as a theoretical study in linguistic philosophy. In total, the material collected for the project comprises four sets of data.

1. Transcribed tape recordings and field notes from in situ observations of groups of high school students solving data modelling problems. No interference was made with classroom organization or with student tasks.
   Total material: approx 30 hours covering 10-12 pairs/groups from three different classes. Each class was visited once a week over a three-month period. Two schools and two teachers were involved. Most groups stayed unchanged for the duration of the observation period.
   Data-driven analysis was performed mainly by directly listening to the tape recordings, supported by inspection of detailed transcripts. Field notes were used to help recall the situation in which the conversation took place.

[9] The high school students were all 18 years or older.

2. Similar data collection to set 1, but with university students. Four tutoring groups (12-20 students each) were visited twice each for a 90 min session. No interference was made with the classroom organization or with the choice of exercises to be solved.
   Data-driven analysis was performed mainly by directly listening to the tape recordings, supported by inspection of detailed transcripts. Field notes were used to help recall the situation in which the conversation took place.
3. A set of five open-ended questions, given as a written questionnaire with limited time for answering[10]. The test was given to ten high school classes (107 students) and four university tutoring groups (50 students).
   Answers were rigorously coded and statistically analyzed.
4. The official specification of UML 1.4 (Booch et al., 2001) was studied and compared to the theories described in the two main books by Wittgenstein (Wittgenstein, 1958; 1961)

4.3.1. Sampling of data

During the collection of datasets 1 and 2, I used a small (5x10cm) Dictaphone with micro-tapes. The Dictaphone had a built-in microphone that provided sufficient sound quality for later transcription. The Dictaphone was normally placed on the desk in front of the students or on top of their computer screen. Only one group of students could be recorded at a time. The selection of groups to record was done ad hoc in the classroom. I would generally keep observing one group as long as their activities were focused on the problem solving. If the activity diverged towards non-curricular talk, or changed to individual work with less verbal interaction, I would move to another group. I would also leave a group if they were primarily working on layout, documentation or other tasks not related to database design or data modelling. Cues that could lead me to notice a particular group, and start recording their discursive interaction, include the following:

[10] Within a socio-cultural perspective on learning, it has been argued that it is doubtful to what extent written test items are able to test knowledge of scientific concepts (Schoultz, Säljö, & Wyndham, 2001). Still, I have chosen a special version of written test format for collecting data to analyse scientific concept building. The justification for this is discussed in detail in paper 3.

- A member of the group called for attention, or asked a question to the teacher, to me, or to students from another group.
- The group seemed vigorously engaged in some academically oriented discussion.
- The group had previously been engaged in interesting collaborative work, and I wanted to follow up the previous observation.
- I had not visited this group for a while (or at all that day), and wanted to see what they were up to.

The choices of which groups to record were partly made on impulse. But in general, since I was concerned about collecting as much and as rich data as possible, I sought to find groups with a lot of talking going on and where this talking in some way concerned data modelling or database design. This might indicate a bias towards sampling of the more active and talkative students. This potential bias was deliberately attempted avoided by occasionally approaching more quiet groups and if necessary challenge them to explain what they were up to.

While I was recording a group, I usually also observed the same group and made informal field notes. The main function of the field notes was that I transcribed selected statements and noted which student made the statement. This information was used to help identifying the individual students' voices when transcribing the data. To help in matching the field notes to the tape recordings, I also took regular time stamps from the 'counter' on the Dictaphone. In addition, notes were made about references made to visual illustrations, documents or things on the screen during the conversation, so that statements like "If you put *this one* down *there*" would make sense in the subsequent analysis. Some general, more analytical, observations that came to mind were also written down. School, class, date, time stamp (i.e. counter), and names of group members were noted on the inlay of the tape cover each time I changed group and also copied in the field notes.

4.4. Analysis

4.4.1. Discourse analysis (datasets 1 & 2)

The method employed for analysing the tape-recorded classroom interactions can be described as a kind of discourse analysis. The name 'discourse analysis' is used to describe various methodological approaches within areas like linguistics, cognitive psychology and poststructuralism and has also been associated with work in speech act theory, critical linguistics, conversation analysis etc. (Potter, 1997). Common to these are the emphasis on language function, and that they address language without focusing on the basic structures of grammar and phonetics. Having described discourse in chapter 3 as language in use, discourse analysis would be the “study of language in use” (Nunan, 1993: p7). The focus in this respect is not on the organisation of the discursive actions from a conversation analysis (Heritage, 1997) point of view, but rather on the classroom as a local sub-community with a cultural practice that is continuously shaped by the participants through use of language as a tool for collective mediation and construction of meaning. Discourse analysis is committed to studying “discourse as texts and talk in social practices.” (Potter, 1997: p146). The ultimate aim is to demonstrate and interpret how regular patterns in language use resonate with the meanings expressed and purposes served in language use (Nunan, 1993). As discussed in chapter 3, it is a part of the rationale for my study that language in use (i.e. discourse) can reveal something about the implicit knowledge and meaning of the speaker. I have therefore taken discourse analysis as my approach to analysing the data.

Several of the varieties of discourse analysis mentioned above bring with them theoretical assumptions and ontological and epistemological perspectives of their own. My intention in taking a discourse analytic approach is not to introduce further theoretical frameworks, but to use the methods developed as an approach to analysing my interactional data.

Transcription and analysis

In order not to obscure possible findings in the data, it is an ideal to make as detailed transcripts as practically possible. A common convention is the Jeffersonian system (see e.g. Potter, 1996), which is rather time-consuming if followed in full detail. Since my data material was rather extensive, it was unpractical to transcribe all the material in full detail. This was solved by doing the transcription in two phases. In the first instance, a rough transcript was made including the dialogue and a few extra features where these were easily recognisable. Pauses, were for example generally noted, but not necessarily timed. This transcript was used as support material while the analysis were based on sustained work with the tapes (which, after all, is as close as I could get to the real data).

For each observations made that in some way seemed relevant to the study of the concept building process, a timestamp (i.e. counter) and a brief note of the nature of the observation were recorded. “Part of DA may involve coding a set of materials, but this is an analytic preliminary used to make the quantity of materials more manageable rather than a procedure that performs the analysis itself. There is nothing sacred about such codings and extracts are often freely excluded and included in the course of research.” (Potter, 1997: p158). After listening to all tapes several times in this manner, both with and without the transcript as support, some passages of particular interest were identified and transcribed in more detail, using a slightly reduced version of the Jeffersonian system. Features used in the transcripts included, but were not limited to: glottal stops, repairs, overlaps, emphasis, rapid speech, pauses (timed), sighs and laughter. An example of the level of detail used can be found in the transcripts on page 96 of paper 3. Although my focus was not on details of the interactional patterns, these features were sometimes helpful – for instance, a hesitation or a prolonged pause indicating uncertainty, or an overlap indicating enthusiasm or persistence.

On preparation of the excerpts presented in the papers, much of this detail has been removed. This choice can rightfully be criticised from a reliability-point of view because it deprives the reader of access to some of the detail in the data. However, the excerpts presented already constitute a selection and hence an omission of other sequences. Similarly, the omission of transcript detail can also be seen as a part of the choices made for presentation of the data. This choice was made for two reasons. Firstly, it greatly enhances the readability of the data – especially for readers not familiar with discourse analysis and detailed transcripts. Secondly, the observations made that were selected for presentation in the papers did not depend on the information available in the features omitted. This does not mean that the information omitted did not offer interesting analytical information, but it was not relevant for the discussions and topics covered in these papers.

Unit of analysis

In their study of students’ interactions with computer representations in a science laboratory class, Kelly and Crawford (1996) set up a taxonomy of units of analysis to be used in analysis of student discourse. At the first level, they divide the transcribed discourse into *message units*, which are the smallest units of linguistic meaning. Linguistic meaning in this sense is not to be confused with semantic or semiotic meaning, which would narrow the size of the unit down even further to single words or even syllables. A message unit may well consist of a single word, but is more typically an utterance or a short sentence that would make sense in discursive interaction. The next unit level is the *action units* which are composites of one or more message units. An action unit represents an intended speech act by a member of the group. An action unit is often linked to the preceding or following action units of an interaction by being a response or inviting some kind of feedback. This brings us to the next level, which is called *interaction units*. Interaction units may also comprise of only one action unit. If, for example, I ask a question that nobody answers, my act of asking still has reference to the potential response. Some discursive analysts may even claim that the other members of the group perform a valid speech act by *not* answering, which may in some cases be an equally significant contribution to the interaction as an explicit answer would have been (Potter, 1997). Building on the interaction units, Kelly and Crawford continue by introducing *sequence units* as thematically tied interaction units. These "represent a portion of the conversation demarcated by the substance of the talk" (Kelly & Crawford, 1996: p699).

For the analyses presented in this thesis, sequence units were only used as an organizational feature helping to provide overview of the material. They were not coded into the transcripts, but only used in field notes and working documents in terms of time stamps for locating the various passages of the material. The main units of analysis used correspond to the action and interaction units and partly the message units as described in Kelly and Crawford's taxonomy. Since there was no desire to perform statistical or other quantitative measurements on the data, there was no need to explicitly code the transcribed data using these unit levels explicitly. The purpose of introducing this hierarchy here is to provide a framing for the level of granularity used in the analysis of the data.

Abductive reasoning

.Analysing the data, the individual observations were considered and attempted explained in terms of existing theory and previous findings from research in computer science education, or from general educational or semiotic theory. A few of the observed patterns were then chosen for further investigation, and additional sequences supporting or contradicting the hypothesis formed were identified. This is in line with what I have described in section 4.1 as abductive reasoning. The theories thus developed are described in detail in papers 1 and 4 supported by illustrative examples of discursive interaction.

4.4.2. Coding of written answers (dataset 3)

The coding procedures for the open-ended written questions are described in detail in paper 3. One important concern in this respect is related to the ontological status of the students' responses. The coding is to a large extent based on the terminology chosen by the students in describing the scientific concepts presented. Comparing the choice of wording in such answers across students relies on the assumption that the terms used have a semantic reference (i.e. meaning content) that is to a large extent socially shared among the students. Since I have already taken the position that each individual has their own subjective understanding of the meaning of a term, this could be somewhat problematic to justify. The point in paper 3, however, is not to discuss what these students actually mean with their explanations or how they really understand the concept they are explaining. The focus of the analysis is on the way that they use language to perform the activity of explaining a scientific concept. In that sense, it should be possible to view their choice of terminology and manner of explaining as indicative of the way they have become accustomed to use language in the social scientific practice of data modelling in a class room setting.

Unit of analysis

The unit of analysis for the coding process of dataset 3 was the individual terms and explanation techniques applied, and the way that they are interlinked in the form of thematic patterns (Lemke, 1990). Since these are not interactional data, the categories introduced by Kelly and Crawford, as described in the previous section, do not apply here. If we should keep with a similar taxonomy, however, the unit of analysis for dataset 3 would be on a lexical level of semiotic meaning – this because the coding is based on the inclusion of single terms or concepts in an explanation, with little attention to how the terms form part of a discursive act or an interaction. An important constraint in this respect was, however, that the term should be included in a manner that makes scientific sense.

4.5. Validity and reliability

The research questions posed in this thesis focus on individual and collective conceptual knowledge, and the ways they are shaped through discursive interaction. Knowledge as such is not easy to operationalize in research design. In chapter 3 I established knowledge as being situated in cultural practices and distributed through language. This opens up for language to be used as data for studying cognition, which is what I have chosen to do in this PhD project. The second main issue addressed by the research questions concern the discursive interaction as a cultural practice. This is a construct that to a certain extent can be observed directly and thus does not need to go the way via operationalization in order to be investigated empirically. The internal validity of the study is thus accounted for in the discussion of the relationship between language and thought in section 3.1.

The sampling of classes and students were made in clusters of whole classes, and no interference was made to the educational design or content of the activities that were observed. In terms of reliability, it is therefore quite likely that similar observations could be made in a subsequent study using the same research methods. Also, the fact that similar interactional patterns and subject matter problems were apparent across the different subpopulations and data sets suggests that the findings presented in this thesis are to a certain extent generalizable to a larger population. However, generalization has in no way been the aim of the study. In CSE, an effort must first be made to generate interesting and testable hypothesis in order to formulate the "right" research questions in future studies (Fincher & Petre, 2004). The results presented in this thesis should be understood as case study examples of data modelling students' discourse in collaborative problem solving environments. All analysis presented in this thesis, including the statistical analysis in paper 3, merit further investigation – either by replication of the same techniques, or through other more focused methods of research.

5. Discussion

In this chapter, I will address the research questions posed in section 2.5. I discuss the contributions offered to answering each of the questions by reviewing the results from the four papers in relation to some of the previous research findings and theories discussed in chapters 2 and 3. The chapter also includes a section with some suggestions for teaching based on the results of the different papers, and finally a discussion of limitations of the research leading to some suggestions for future work.

5.1. Scientific concept building

Q1: What characterizes novice data modellers' acquisition and knowledge of the scientific concepts of keys as used in the domain of data modelling?

The focus in this section is on the scientific concept building associated with acquiring domain-specific concepts and terminology. This links to the theoretical framework of scientific versus spontaneous concept building (Vygotsky, 1986) as presented in paper 1. Spontaneous concept building, as explained by Vygotsky, is a bottom-up process where generalizations are gradually formed on the basis of experiences with concrete cases. Scientific concepts, on the other hand, are first introduced as abstract generalized phenomena, which the learner gradually comes to understand through subsequent experience with concrete cases.

Contrary to Vygotsky's claim that concepts are learned either in the one manner or in the other (i.e. top-down or bottom-up), I am suggesting that scientific concept building does not occur on a vertical dimension. Instead the results in paper 3 indicate that the process can be described as a horizontal trajectory from initial hunches to holistic knowledge (see figure 3 on page 109 of paper 3). This trajectory process is continuously influenced in parallel from both of Vygotsky's approaches (i.e. theoretical/top-down and experiential/bottom-up).

The framework presented in paper 3 was developed in a previous study of students' and professors' conceptions of object-orientation (Holmboe, 1999) based on answers to the question "What is object-orientation?". Both this study, and the study presented in paper 3, thus employed the same method as Aharoni (2000), who asked his subjects "What is an array?". Like in Aharoni's study, my framework for the concept building process is based on the theory from mathematics of reification of actions via processes to objects in their own right (Sfard, 1991). My main conclusion, which corroborates Aharoni (2000) is that scientific concept building takes place in an interaction between practical and definitional knowledges. Aharoni (2000) elaborates on this interaction by describing it as a circular or iterative process that goes from actions on objects via processes to new reified objects that can then be the used as input to new actions on a higher level of abstraction. The horizontal trajectory described in paper 3 could, in light of Aharoni's findings, be described as a horizontal spiral movement.

A further finding of paper 3 relates to the *conceptual networks* that the students seem to be building as they develop familiarity with scientific concepts. Lemke's theory of *thematic patterns* (Lemke, 1990) stands in analogue to my conceptual networks. Yet Lemke, like Vygotsky, emphasises the ideal of one "commonly accepted" version of the scientific concept meaning, or thematic pattern, as a learning objective for the students, whereas I claim that the conceptual network is individually constructed, although highly influenced from the discursive interaction in the classroom. A main finding of paper 3 relates to the apparent impact that the collaborative nature of these classrooms has on this type of concept building process. This will be discussed further in section 5.3.

Paper 1 contains a subheading called "the name decides" in which I describe a sequence where a student assigns attributes to an entity based exclusively on its label and not on the function it has in the data model. This observation echoes the point made by Hazzan (2003) about students not being able to detach their understanding of the array construct from the everyday understanding of a set of (at least two) elements.

In section 2.4.1, I referred to a number of studies that show misconceptions held by students due to use of natural language terms as part of programming constructs. Bonar and Soloway show, for example, how the **then** of Pascal's **if-then-else** construct is interpreted to indicate a sequence in operation (e.g. "Do this first, *then* do that.") based on the "surface (lexical) link between *then* and **then**." (Bonar & Soloway, 1985: p140). My analysis of students' conceptions of *keys* did not indicate such misconceptions. However, their knowledge of the concept of *candidate key* was very limited. This is a concept that is rarely used in practical modelling activities, and is therefore something that the students can have poor understanding of and still perform well on modelling tasks. This finding corroborates Hitchman's (1995) observation about poor understanding of for example unitary relationships, which are similarly concepts that are rarely needed.

5.2. Labelling

Q2a: Do novice data modellers benefit from using natural language terminology when labelling entities/classes?

Q2b: What characterizes novices' concept building processes related to labelling elements of a conceptual data model?

Concept building, in the sense of becoming familiar with scientific terms and their established and generally accepted meaning, is probably the most obvious link between language and learning. I will now discuss a different type of concept building activity related to the labelling of data model elements.

Data modelling (as well as programming) introduces a number of technical concepts that need labelling (papers 1 and 4). Contrary to the process of scientific concept building, which is frequently described and common to most subject areas, this particular aspect of the relationship between language and learning is specific to computer science (i.e. data modelling and programming). This implies revisiting the distinction between scientific (top-down) and spontaneous (bottom-up) concept building (Vygotsky, 1986). The analysis in the present thesis shows that the concept building (paper 3) and labelling (paper 1) activities of data modelling comprise features of both these types of concept building activities simultaneously.

The spontaneous concepts in programming languages and data modelling methodologies are mainly concerned with the use of intelligible terms for denoting meaningful features of a program or a data model. This has been discussed previously in relation to naming of variables in programming (e.g. Shneiderman, 1980). The process is to a certain extent related to the learning of foreign languages (Vygotsky, 1986), in that it involves reconstructing and/or altering the relationships between terms and meanings from vernacular languages. Known *signs* are attributed new or altered meanings, and known entities or meanings are labelled with alternative terms or phrases from the more familiar ones (paper 1).

The other type of concept building (i.e. scientific concept building) takes place when new or abstract “gadgets” are introduced in the data model; for instance, when labelling relational phenomena (i.e. classes or entities that arise from objectification of a relationship between classes or entities). These are phenomena that do not have a close mapping to any everyday concepts. Hence, there is no term or expression from vernacular discourse (paper 4) that lends itself to be used as label for the phenomenon (paper 1). A “new” term must be invented or introduced, and the meaning of this term then needs to be explicitly defined. Through repeated use in the scientific discourse of the data modelling activity, the new term and its related meaning develops into a concept in the modeller’s understanding.

In the two first papers, I demonstrate the importance for novices of metalinguistic awareness[11], and the related need for explicitness in the choice and use of terms for labelling entities as well as attributes and relationships. Paper 1 illustrates this from a largely empirical point of view, whereas paper 2 takes the more theoretical perspective of linguistic philosophy. The common conclusion, which is also evident from the results in paper 4, is that it is necessary to help the students realise, and become aware of, the differences between natural language use and specialised languages like data modelling or programming. One main difference lies in the necessary levels of precision or accuracy. The meanings of natural language propositions are defined through their use in social practices (Wittgenstein, 1958). The technical language expressions introduced as labels in a data model or program, on the other hand, need to have their meanings explicitly defined in order to prevent ambiguity. Détienne (2002) describes this duality of computer programming as on the one side being represented by an unambiguous technical syntax, while simultaneously allowing for incorporation of terms from vernacular lexis as labels for variables, classes and operations. This latter aspect has been shown to help the understanding of computer programs (Shneiderman, 1980). But – as have been demonstrated in paper 1 – it also introduces problems because the students tend to confuse the artificial and the natural language domains as contextual frames when determining the meaning of a term used in a data model. It appears that the distinction between artificial and natural languages is not as clear-cut as one would like to believe, but rather that the two are intertwined. Programming language understanding is, as explained in sections 2.4.1

[11] In paper 1 this is called *metalinguistic consciousness* (see also discussion in section 3.1.5).

and 5.1, dependent upon natural language knowledge, but at the same time easily confused by it (Bonar & Soloway, 1985).

The students in paper 1 appeared to have problems with this distinction. Erroneous modelling was sometimes the result of letting an entity adopt the vernacular meaning of the term chosen as label, without the necessary transformation by grammatical metaphor (paper 4). This could have been avoided if they had been *aware* of the data model representing a different *language game*. By this, I emphasize that the students probably have the metalinguistic knowledge of this distinction, but they are not aware of it; they lack the *metalinguistic awareness*.

When making a data model for some problem domain, it is essential to maintain a closeness of mapping between the stakeholders' conceptual models of the constructs to be modelled and the representations established (Peckham & Maryanski, 1988). In the study by Bürkle et al. (1995), this was achieved by maintaining a close collaboration between user groups as experts of the domain specific language of banking, so that the concepts that were deployed in the data model were based on a sound understanding of how these concepts were generally used by the users of the system. Both the students in paper 1 and the students in paper 4 displayed problems due to lack of detailed domain familiarity, which forced them to invent meanings of concepts and their interrelationships. In addition to jeopardising the quality of their system, such inventions put an even greater demand on the students to be explicit about the intended meanings of the components of the system, as their understanding cannot rest on shared cultural-historical background knowledge.

In paper 4, I distinguish between *technological* and *scientific lexis* (White, 1998). Being unknown terms introduced as labels for new phenomena, technological concepts are clearly scientific according to Vygotsky (1986). White's *scientific expression* also conforms to the scientific concepts of Vygotsky, but paper 4 shows that the learning of these concepts does not necessarily follow the top-down patterns described by Vygotsky. It seems that they are transformed generalizations from vernacular concepts (e.g. *Blocking* as a nominalized version of the activity of blocking one's account). These concepts get their meaning through grammatical metaphor, rather than through deduction from a formal definition to specific cases. Based on the analysis of paper 4, it would therefore be appropriate to claim that the attribution of meaning to scientific concepts, in White's sense, resembles spontaneous concept building in Vygotsky's terms, rather than scientific concept building.

The complex relationships between the different semiotic systems related to the activity of data modelling are illustrated by the framework presented in paper 4. The navigation between the different metalevels, contexts and signs constitute a bridging of the gap between artificial and natural languages. To be able to handle this bridging of the gap successfully, the data modeller needs a certain level of metalinguistic awareness. This awareness seems to be particularly important for novices. Note, however, that with increasing levels of expertise (Dreyfus & Dreyfus, 1986), the difference becomes less obvious or important, and the metalinguistic knowledge is gradually less explicitly attended to in the discourse. Paper 4 furthermore introduces the notions of technical and vernacular language realms as *contextual frames*. It appears that proficiency in data modelling is characterised by the ability to seamlessly shift between these different contextual frames in discourse. By *seamless shifts* I mean that the differences of the involved language games are not attended to explicitly, but still recognised in the way the meaning of a term is determined by the contextual frame in which it is used. This finding corroborates Dreyfus and Dreyfus (1986).

5.3. Collaborative learning practices

Q3a: How are the concept building processes of novice data modelling students influenced by the discursive practices of the classroom environment in which they take place?

Q3b: How do novice data modellers handle the coexistence of, and interdependencies between, different discourse types when engaged in collaborative problem solving activities in a computer science classroom?

Learning can be described as a discursive practice that is situated in a social context. These are two of the main presuppositions that form the theoretical foundation for the research presented in this thesis. These claims are, however, closely related to each other. By discursive practices is meant any activity that in some way includes, or depends upon, the socially situated formation and mediation of meaning (see chapter 3). "Contextual features such as where and when people act, the specific contents of problems and tasks, and other elements of situated action, have all been shown to serve as resources through which people make tasks meaningful." (Schoultz et al., 2001: p214).

Adopting a socio-cultural perspective on knowledge as the ability to participate in cultural practices, learning thus emerges as an outcome of discursive interaction. Discourse is not restricted to verbal interaction, but includes a number of other semiotic processes. One of these is the reference to previous experience manifested as *encounters* (Wickman & Östman, 2002). The students described in paper 4 demonstrate such implicit reference to previous discursive experience when they base their elaborations on familiarity with the vernacular terms adopted from the world of banking. Similarly, the students in paper 1 also utilize discursive knowledge based on previous experience when they look for terms to denote their abstract relational phenomena, or when they carry a vernacular term's meaning content into the entity of the data model labelled by the same term. In this manner, the *encounters* of the participants are omnipresent in any discursive interaction through their shared contextual background. This enables them to communicate without having to explicitly define and explain every term or concept that is introduced in the conversation. It is this shared cultural frame that enables natural language use to function, in spite of the ambiguity or impreciseness of most utterances (Wittgenstein, 1958). This flexibility of language use and the contextual background is not shared by the computer. As emphasized in paper 2, this is an important reason why students experience problems when modelling the world for implementation on a computer. Similarly, students tend to take for granted the common knowledge within the group, while they in reality do not always have compatible perceptions of what an entity or class is supposed to represent (paper 1).

Given, for example, a data model that contains a class or entity type called "brother". This class or entity would have quite different connotations to a Norwegian teenager compared to those of an African-American (see discussion in section 2.1). Creating attributes or linking this class or entity to other classes or entities would accordingly be done differently depending on the frame of reference.

Claiming that learning is a discursive activity thus implies that it is founded on exactly such a cultural historical framework, which in turn implies that all learning is situated in social practices and therefore needs to be interpreted within the context where it takes place. This means that learning is largely helped by, or even dependent on, the participants having similar linguistic backgrounds.

The slightly different conceptual understandings held by professional members of a technical domain like banking is a central topic discussed by Bürkle et al. (1995). In the first phase of their study, no measures were taken to allow the developers and customers to interact and negotiate the meanings of the constructs to be modelled and their labels, in order to establish common knowledge. This led to problems of communication that actually caused the project to fail. Providing arenas for the establishment and continuous negotiation of common knowledge, however, proved to be surprisingly efficient. Herbsleb et al. (1995) similarly found that expert modellers asked each other frequent questions to clarify what was meant by some element of the model. One would expect the students to make a similar effort to establish common knowledge. The university students did this to a certain extent (paper 4), while the high school students hardly did so (paper 1).

The importance of the discursive activity as resource for learning is further emphasized by the findings presented in paper 3. Language is a social construction, and the meanings of terms are defined by the way these terms are used in social practice of a particular language game. The social development of semiotic relationships between expression and content normally takes place over an extended period of time, and is subject to gradual change over generations. A word commonly used for something today, may have carried different connotations, or may not have been a part of the everyday discourse a couple of decades ago. The activity of data modelling, however, implies a constantly ongoing formation of new semiotic relationships (paper 4), and thus also new language games (paper 2). The meaning of terms transformed from vernacular lexis to the technical discourse of a particular data model is defined explicitly or implicitly through the discursive interaction between the participants of the modelling activity. Simultaneously, the students are introduced to the technical language game of data modelling – be it with ER, UML or other methodologies. Making sense of the different scientific concepts introduced as parts of these new (to the students) language games is, as demonstrated in paper 3, a social process. In fact, the students belonging to a particular group seem to be collaboratively constructing their own locally functioning language game of meanings and relationships between the scientific concepts they are introduced to.

This simultaneous development of spontaneous and scientific concept types is to a large extent accomplished by collective exploration of various semiotic systems through discursive interaction, an activity that demands of the data modelling practitioner (or student, in my case) that he or she distinguishes between different language games (Wittgenstein, 1958). Successful participation in this cultural practice requires metalinguistic knowledge in order to separate the new artificially constructed signs representing a simplified version of a part of the world from the closely related signs from natural language representing more or less closely related meanings (paper 1).

I have described learning from a socio-cultural perspective as becoming able to participate in a cultural practice. Following this view, scientific concept building implies enculturation into the community of scientific language users. Different scientific disciplines have their own linguistic subcultures with particular concepts and customs for language use. The research presented in this thesis has shown that concept building in computer science classrooms isn't only a matter of enculturation into existing predefined linguistic practices. It appears that the students as members of the classroom community also form their own locally functioning linguistic practices endemic to the group. These collectively formed discursive practices of each classroom seem to out-compete the formal definitions provided by textbooks or teacher controlled instruction. This pattern is analogous to the one described by Bürkle et al. (1995) in which they observed differences in perception between members of different work units, even for seemingly standard concepts from the banking domain. In the data modelling classrooms of my study, such discursive processes take place both on the level of academic computer science discourse and on the level of talking about the problem domain and its representations in the data model that is being collaboratively constructed (papers 1 and 4). New language games are thus socially negotiated on several levels in these learning environments (paper 2).

5.4. Implications for teaching

The various relationships between language and the learning of data modelling that are described in this thesis have consequences for the planning of teaching sequences. Each of the four research papers offers a section on suggestions for teaching. In this section, I will summarise these points and also provide a few additional comments. The students need help in the process of realizing the different semiotic systems or language games involved when they engage in data modelling as a social activity. Even though experts do not seem to take notice of these distinctions (Dreyfus & Dreyfus, 1986), it has been demonstrated through the research presented in this thesis that novice students need to be more explicit in the way that they handle the various terms and expressions involved when solving a data modelling problem. They need to have metalinguistic awareness.

The distinction between natural and artificial languages play a particularly important role in this respect, both for scientific concept building and for labelling of attributes and entities. It is therefore advisable that the teacher spends some time focusing on different types of language games, and on the meaning of utterances as defined through actual use, and not from predefined rules or by definition in a dictionary. Teaching scientific concepts to novices, it is furthermore important not only to explain how the constructs are intended to function, and what their meanings are (from an established scientific point of view). One needs also to emphasise how these understandings of the constructs differ from the natural language use of the corresponding terms. In science teaching this has been addressed by sharing with students the idea that learning science (or computing) involves learning a new language to talk and think about familiar things. In some cases the terms are familiar and carry familiar meanings; in other cases the reverse is true (Mortimer & Scott, 2003). Paper 2 offers some further explicit suggestions of possible classroom activities that may be adopted in order to improve the metalinguistic awareness of the students. These are mainly concerned with helping the students realise the distinctions.

The next step is to provide situations where the students practice formulating explicitly what they mean by the terms they choose to use as labels, and to distinguish between this meaning and their pre-knowledge from everyday language. In the related field of mathematics education, Shoenfeld (1992) describes positive results from an experiment trying to improve students' problem solving capabilities through increased metaknowledge. In the experiment, the teacher continuously moves around the classroom stopping at the table of individuals or groups and always asks the same set of questions; i.e. " What (exactly) are you doing?", "why are you doing it?" and "How does it help you?" (Shoenfeld, 1992: p356). Eventually the students got used to the questions, and were generally better prepared to give a satisfactory answer when the teacher approached them. After a while, the teacher was able to cease the questioning as the students were now asking themselves the same set of questions each time they came to a decision point. This approach can easily be adapted to a computer science classroom, and to focus on the metalinguistic awareness of the students instead of their problem solving capabilities. Questions that could be used include: “What have you labelled that entity?”, “What exactly is the meaning of the entity?” and “How does that meaning correspond to the term you have chosen?”. The answers to these questions would address both the issue of labelling and use of vernacular lexis, and the issue of creating abstract entities for relational phenomena.

The discussions presented in this thesis deal with various language systems that operate on different levels in parallel. In addition to the vernacular–technical dimension discussed in connection with everyday terms used as labels (i.e. the semiotic dimension), there is the metalanguage hierarchy of data modelling defined as a semi-graphical language for making formalized descriptions of the problem domain. It will be unnecessarily complex to introduce all of these dimensions to the students, but some of these issues should be addressed. “Learning technical discourse implies learning the lexico-grammatical language of that discourse, which, for science, implies learning to transform everyday or vernacular language into an uncommon-sense language.” (paper 4: p17). Considering the semantic framework introduced in paper 4, a set of questions that could be helpful for increased metalinguistic awareness are: “What do you mean when you use that term?” and “In which contextual frame do you understand the term when you use it like that?”.

Another significant finding of this research is the local development of language games in each particular classroom. In order to facilitate this development, it is imperative that the students are allowed to interact discursively with each other to a sufficient extent. The teacher needs to be alert and able to adapt the way he or she uses scientific language in interaction with the students. The teacher’s responsibility in this sense is twofold. First, he or she offers an invaluable benchmark or point of reference for the students as they develop their own version of the scientific discourse. Second, and equally important, the teacher needs to adapt to the discourse of the classroom as it evolves. In this way, the teacher will function as a participant in the discursive social practice and thus be able to influence the development toward fruitful viable ways of doing computer science with language. It can be advisable to even focus explicitly on this aspect of the discourse and compare the conceptual networks developed in the student group to the established text book scientific discourse. This is similar to what Lemke (1990) did in his studies of the development and use of thematic patterns in the science classroom.

5.5. Limitations, shortcomings and outstanding issues

The rich and diverse data that have been collected invite several perspectives to be taken and research questions to be asked, of which the results presented in this thesis only cover a few. The material comprises observations both of high school and university students. Differences between these two groups in terms of problem solving strategies, problem domain familiarity and metalinguistic awareness are very interesting topics for further investigation. The fact that the university students were using UML while the high school students were modelling with ER similarly invites the question of how these methodologies seem to facilitate the learning of conceptual data modelling in general.

In order to be able to address these kinds of comparative questions, the research design needs to be more focused than what was the case in the present study. The data were not sufficiently homogenous to be comparable across the subpopulations. One way of handling this would be to design particular modelling problems that the students were given to work on. Such a design would also ensure a higher density of relevant observations in the data collected. The downside is that the desired naturalness of an "undisturbed" classroom environment would be lost to some extent. Qualitative comparisons of populations furthermore presuppose that there is a well-defined set of criteria for coding and analysing the data. In order to set up such a set of criteria, as well to construct sensible modelling problems some framework is needed based on some preknowledge of what to look for, as well as along what dimensions the interesting results may emerge. The findings described in this thesis may serve to provide a starting point for building such a framework to be used in the design of subsequent studies.

Having set out with a discursive perspective of language in use as the main focus for analysis, the findings presented in the papers focus primarily on the outcome from these discursive processes, and are less explicitly concerned with the processes themselves. There is probably much to be found in the data collected that could provide valuable input to the understanding of how meaning is negotiated in the classrooms, and what discursive mechanisms are deployed in order to achieve common knowledge and build conceptual networks. One reason why this was not given more attention was that since the material collected was so extensive, it was difficult to identify the significant *interaction units* where such learning processes were displayed. It was difficult, in general, to find interaction units where the students used scientific terminology of data modelling in the first place. This observation echoes the finding of Levi and Lapidot (2000), that students tend to use their own everyday way of talking about scientific constructs, without referring explicitly to the scientific domain terminology. Levi and Lapidot used a focused teaching experiment with predesigned problems that were given to the students and that required discussion in groups. This kind of study design leads to much higher density of potentially interesting interaction units in the data, which then more easily lend themselves to analysis of the concept development "in action", so to speak.

Studying the development of conceptual understanding suggests taking a longitudinal approach, so that each student can be observed at different points in time. This allows the researcher to trace any development in the way the student uses language to "do" data modelling. Within the frames of the method employed in the present study, focusing on fewer groups of students in each class could lead to richer data within each case, and thus potentially to further insight into the learning processes of the members of the chosen groups. A longitudinal study would also be better facilitated by a research design including targeted problem-solving activities. Alternatively, the collection of data could have been made over a longer period than the three months used in this project, although not necessarily with continuous observations. As it turned out, the written questions provided some longitudinal data which it would be interesting to investigate further in a follow-up study, preferably with more than three months between observations.

References

Aharoni, D. (2000). Cogito, Ergo Sum! Cognitive Processes of Students Dealing with Data Structures. *SIGCE Bulletin, 32*(1), 26-30.

Alvesson, M., & Sköldberg, K. (1994). *Tolkning och reflection: Vetenskapsfilosofi och kvalitativ metod*. Lund: Studentlitteratur.

Andersen, P. B. (1990). *A Theory of Computer Semiotics*. Cambridge: Cambridge University Press.

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

Batra, D., & Antony, S. R. (1994a). Effects of Data Model and Task Characteristics on Designer Performance - a Laboratory Study. *International Journal of Human-Computer Studies, 41*(4), 481-508.

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

Batra, D., & Antony, S. R. (2001). Consulting support during conceptual database design in the presence of redundancy in requirements specifications: an empirical study. *International Journal of Human-Computer Studies, 54*(1), 25-51.

Batra, D., Hoffer, J. A., & Bostrom, R. P. (1990). Comparing representations with the relational and EER models. *Communications of the ACM, 33*, 126-139.

Batra, D., & Sein, M. K. (1994). Improving Conceptual Database Design through Feedback. *International Journal of Human-Computer Studies, 40*(4), 653-676.

Bock, D. B., & Ryan, T. (1993). Accuracy in modeling with extended entity relationship and object oriented data models. *Journal of Database Management, 4*, 30-39.

Bonar, J., & Soloway, E. (1985). Preprogramming knowledge: A major source of misconceptions in novice programmers. *Human-Computer Interaction, 1*(2), 133-161.

Booch, G., Jacobson, I., & Rumbaugh, J. (2001). *OMG - Unified Modelling Language Specification v1.4*. Needham, MA: OMG Object Management Group.

Brooks, R. (1977). Towards a theory of the cognitive processes in computer programming. *International Journal of Man-Machine Studies, 9*, 737-751.

Bürkle, U., Gryczan, G., & Züllinghoven, H. (1995). Object-oriented system development in a banking project: Methodology, experience, and conclusions. *Human-Computer Interaction, 10*(2&3), 293-336.

Chan, H. C. (1998). User performance differences between relational and entity relationship models: a summary review of the literature. *Behaviour & Information Technology, 17*(1), 59-61.

Chen, P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9-36.

Collins, A. M., Brown, J. S., & Newman, S. E. (1989). Cognitive Apprenticeship: Teaching the craft of reading, writing and mathematics. In L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in honor of Robert Glaser*. Hillsdale, NJ: Erlbaum.

Curtis, B., & Walz, D. (1990). The Psychology of Programming in the Large: Team and Organizational Behaviour. In J.-M. Hoc, T. R. G. Green, R. Samurcay & D. J. Gilmore (Eds.), *Psychology of Programming* (pp. 253-270). London: Academic Press.

Détienne, F. (1997). Assessing the cognitive consequences of the object-oriented approach: a survey of empirical research on object-oriented design by individuals and teams. *Interacting with Computers, 9*, 47-72.

Détienne, F. (2002). *Software Design - Cognitive Aspects* (F. Bott, Trans.). London: Springer.

Dietrich, S. W., & Urban, S. D. (1996). Database theory in practice: Learning from cooperative group projects. *SIGCSE Bulletin, 28*(2), 112-116.

Drew, P., & Heritage, J. (Eds.). (1992). *Talk at Work*. Cambridge, UK: University Press.

Dreyfus, H., & Dreyfus, S. (1986). *Mind over Machine*. Glasgow: Basil Blackwell.

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

du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research, 2*(1), 57-73.

Edwards, D. (1997). *Discourse and Cognition*. London: Sage Publication.

Edwards, D., & Mercer, N. (1987). *Common Knowledge: the Development of Understanding in the Classroom.* London: Routledge.

Elmasri, R., Weeldreyer, J., & Hevner, A. (1985). The category concept: An extension to the Entity-Relationship model. *Data & Knowledge Engineering, 1*(11), 75-116.

Engeström, Y. (1999). Communication, discourse, and activity. *The Communication Review, 3*(1-2), 165-185.

Fincher, S., & Petre, M. (2004). Part One: the field and the endeavor. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 1-82). London: Taylor & Francis Group plc.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns*: Addison-Wesley Professional.

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

Green, T. R. G. (1989). Cognitive dimensions of notations. In A. Sutcliffe & L. Mmacaulay (Eds.), *People and Computers*. Cambridge: V. Cambridge University Press.

Green, T. R. G., & Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages and Computing, 7*, 131-174.

Guzdial, M. (2004). Programming Environments for Novices. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 127-154). London: Taylor & Francis Group plc.

Halliday, M. A. K. (1993). Towards a Language-Based Theory of Learning. *Linguistics and Education, 5*, 93-116.

Halliday, M. A. K. (1998). Things and relations: Regrammaticing experience as technical knowledge. In J. R. Martin & R. Veel (Eds.), *Reading Science. Critical and Functional Perspectives on Discourses of Science* (pp. 185-235). London and New York: Routledge.

Hanson, N. R. (1958). *Patterns of Discovery. An inquiry into the foundations of science*. Cambridge: Cambridge University Press.

Hazzan, O. (2003). How Students Attempt to Reduce Abstraction in the Learning of Mathematics and in the Learning of Computer Science. *Computer Science Education, 13*(2), 95-122.

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

Herbsleb, J. D., Klein, H., Olson, G. M., Brunner, H., & Olson, J. S. (1995). Object-oriented analysis and design in software project teams. *Human-Computer Interaction, 10*(2&3), 249-292.

Heritage, J. (1997). Conversation Analysis and Institutional Talk: Analysing Data. In D. Silverman (Ed.), *Qualitative Research: Theory, Method and Practice* (pp. 161-182). London: Sage Publications.

Hitchman, S. (1995). Practitioner Perceptions on the Use of Some Semantic Concepts in the Entity-Relationship Model. *European Journal of Information Systems, 4*(1), 31-40.

Hoc, J.-M., Green, T. R. G., Samurcay, R., & Gilmore, D. J. (Eds.). (1990). *Psychology of Programming*. London: Academic Press.

Hoc, J.-M., & Nguyen-Xuan, A. (1990). Language Semantics, Mental Models and Anology. In J.-M. Hoc, T. R. G. Green, R. Samurcay & D. J. Gilmore (Eds.), *Psychology of Programming* (pp. 139-152). London: Academic Press.

Holmboe, C. (1999). A Cognitive Framework for Knowledge in Informatics: The Case of Object-Orientation. *ACM SIGCSE Bulletin (Proceedings of ITiCSE), 4*, 17-21.

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

Kelly, G. J., & Crawford, T. (1996). Students' interaction with computer representations: Analysis of discourse in laboratory groups. *JOURNAL OF RESEARCH IN SCIENCE TEACHING, 33*(7), 693-707.

Kolikant, Y. B.-D. (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.

Kozulin, A. (1986). Vygotsky in Context. In L. Vygotsky (Ed.), *Thought and Language* (pp. xi-lvi). Camebridge, MA: MIT Press.

Kutar, M., Britton, C., & Barker, T. (2002). A Comparison of Empirical Study and Cognitive Dimensions Analysis in the Evaluation of UML Diagrams. In J. Kuljis, L. Baldwin & R. Scoble (Eds.), *Proceedings of the 14th annual workshop of Psychology of Programming Interest Group* (pp. 1-14). London, UK.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge, UK: Cambridge University Press.

Lemke, J. L. (1990). *Talking Science: Language, learning and values.* Norwood, NJ: Ablex.

Levi, D., & Lapidot, T. (2000). Recursively Speaking: Analyzing Students' Discourse of Recursive Phenomena. *ACM SIGCSE Bulletin (Proceedings of SIGCSE Technical symposium), 32*(1), 315-319.

Liao, C. C., & Palvia, P. C. (2000). The impact of data models and task complexity on end-user performance: an experimental investigation. *International Journal of Human-Computer Studies, 52*(5), 831-845.

Liao, C. C., & Wang, L. (1997). The comparison of EERM and OOM on novice user performance. *Information Management Research, 1*, 25-48.

Matthews, M. R. (Ed.). (1998). *Constructivism in Science Education*. Dordrecht: Kluwer Academic Publishers.

McCracken, W. M. (2004). Research on Learning to Design Software. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 155-174). London: Taylor & Francis Group plc.

Mercer, N. (1995). *The guided construction of knowledge. Talk amongst teachers and learners*. Clevedon: Multilingual Matters Ltd.

Mercer, N. (2000). *Words & Minds: How we use language to think together*. London: Routledge.

Mercer, N., & Wegerif, R. (1999). Is 'exploratory talk' productive talk? In K. Littleton & P. Light (Eds.), *Learning with Computers. Analyzing productive interaction* (pp. 79-101). London: Routledge.

Mortimer, E. F., & Scott, P. H. (2003). *Meaning Making in Secondary Science Classrooms*. Buckingham, UK: Open University Press.

Nunan, D. (1993). *Introducing Discourse Analysis*. London: Penguin books.

Palvia, P. C., Liao, C. C., & To, P. (1992). The impact of conceptual data models on end-user performance. *Journal of Database Management, 3*, 4-16.

Pane, J. F., Myers, B. A., & Miller, L. B. (2002). *Using HCI Techniques to Design a More Usable Programming System.* Paper presented at the Symposium on Empirical Studies of Programmers, Arlington, VA.

Pane, J. F., Ratanamahatana, C., & Myers, B. A. (2001). Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human-Computer Studies, 54*(2), 237-264.

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

Peckham, J., & Maryanski, F. (1988). Semantic Data Models. *ACM Computing Surveys, 20*(3), 153-189.

Petre, M. (1990). Expert Programmers and Programming Languages. In J.-M. Hoc, T. R. G. Green, R. Samurcay & D. J. Gilmore (Eds.), *Psychology of Programming* (pp. 103-116). London: Academic Press.

Piaget, J. (1954). *The construction of reality in the child*. New York: Basic books.

Piaget, J. (1959). *The Language and Thought of the Child*. London: Routledge and Kegan Paul.

Potter, J. (1996). *Representing Reality; Discourse, Rhetoric and Social Construction*. London: Sage Publications.

Potter, J. (1997). Discourse Analysis as a Way of Analysing Naturally Occurring Talk. In D. Silverman (Ed.), *Qualitative Research: Theory, Method and Practice* (pp. 144-160). London: Sage Publications.

Rist, R. S. (2004). Learning to Program: Schema creation, application and evaluation. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 175-198). London: Taylor & Francis Group plc.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education, 13*(2), 137-172.

Rogoff, B. (1990). *Apprenticeship in Thinking: Cognitive Development in Social Context*. NY: Oxford University Press.

Romero, P., Cox, R., du Boulay, B., & Lutz, R. (2003). A survey of external representations employed in object-oriented programming environments. *Journal of Visual Languages and Computing, 14*, 387-419.

Salomon, G. (1993). No distribution without individuals' cognition: a dynamic interactional view. In G. Salomon (Ed.), *Distributed Cognition* (pp. 111-138). Cambridge, NY: Cambridge University Press.

Schoultz, J., Säljö, R., & Wyndham, J. (2001). Conceptual knowledge in talk and text: What does it take to understand a science question? *Instructional Science, 29*, 213-236.

Scott, P. (1998). Teacher talk and meaning making in science classrooms: a Vygotskian analysis and review. *Studies in Science Education, 32*, 45-80.

Sfard, A. (1991). On the Dual Nature of Mathematical Conceptions: Reflections on Processes and Objects as Different Sides of the Same Coin. *Educational Studies in Mathematics, 22*, 1-36.

Sfard, A. (1998). On Two Metaphors for learning and the Dangers of Choosing Just One. *Educational Researcher, 27*(2), 4-13.

Shackelford, R. L., & Badre, A. N. (1993). Why can't smart students solve simple programming problems? *International Journal of Man-Machine Studies, 38*, 985-997.

Shneiderman, B. (1980). *Software Psychology: Human Factors in Computer and Information Systems*. Cambridge: M.A:Winthrop.

Shoenfeld, A. H. (1992). Learning to Think Mathematically: Problem Solving, Metacognition, and Sense Making in Mathematics. In D. A. Grouws (Ed.), *Handbook of Research on Mathematics Teaching and Learning*: Macmillan Education Ltd.

Shoval, P., & Shiran, S. (1997). Entity-relationship and object-oriented data modeling - An experimental comparison of design quality. *Data & Knowledge Engineering, 21*(3), 297-315.

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

Smith, J. M., & Smith, D. C. P. (1977). Database Abstractions: Aggregation and Generalisation. *ACM Transactions on Database Systems, 2*(2), 105-133.

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

Soloway, E., & Ehrlich, K. (1984). Empirical Studies of Programming Knowledge. *Transactions on Software Engeneering, SE-10*(5), 595-609.

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

Srinivasan, A., & Teeni, D. (1995). Modeling as Constrained Problem-Solving - an Empirical-Study of the Data Modeling Process. *Management Science, 41*(3), 419-434.

Säljö, R. (1998). Learning as the use of tools: A sociocultural perspective on the human-technology link. In K. Littleton & P. Light (Eds.), *Learning with Computers. Analyzing productive interaction* (pp. 144-161). London: Routledge.

Säljö, R. (1999). Concepts, Cognition and Discourse: From Mental Structures to Discursive Tools. In W. Schnotz, S. Vosniadou & M. Carretero (Eds.), *New Perspectives on Conceptual Change* (pp. 81-90). Amsterdam: Pergamon.

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

Taylor, J. (1990). Analysing Novices Analysing Prolog - What Stories Do Novices Tell Themselves About Prolog. *Instructional Science, 19*(4-5), 283-309.

Visser, W., & Hoc, J.-M. (1990). Expert Software Design Strategies. In J.-M. Hoc, T. R. G. Green, R. Samurcay & D. J. Gilmore (Eds.), *Psychology of Programming* (pp. 235-250). London: Academic Press.

Vygotsky, L. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press.

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

Weinberg, G. (1971). Psychology of Computer Programming.

Wertsch, J. V. (Ed.). (1985). *Vygotsky and the social formation of mind.* Cambridge, MA: Harvard Univeristy Press.

White, P. R. R. (1998). Extended reality, proto-nouns and the vernacular. Distinguishing the technological from the scientific. In J. R. Martin & R. Veel (Eds.), *Reading Science. Critical and Functional Perspectives on Discourses of Science.* (pp. 266-296). London and New York: Routledge.

Wickman, P.-O., & Östman, L. (2002). Learning as Discourse Change: A Sociocultural Mechanism. *Science Education, 86*, 601-623.

Williams, L. A., & Kessler, R. R. (2003). *Pair Programming Illuminated*. Boston: Addison-Wesley.

Williams, L. A., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In Support of Pair Programming in the Introductory Computer Science Course. *Computer Science Education, 12*(3), 197-212.

Wittgenstein, L. (1958). *Philosophical Investigations* (G. E. M. Anscombe, Trans. 2nd ed.). Oxford: Basil Blackwell.

Wittgenstein, L. (1961). *Tractatus Logico-Philosophicus* (D. F. Pears & B. F. McGuinness, Trans.). London: Routledge & Kegan Paul.

Wood, D. J., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Psychology and Psychiatry, 17*, 89-100.

Computer Science Education
Vol. 15, No. 2, June 2005, pp. 143–161

Conceptualization and Labelling as Cognitive Challenges for Students of Data Modelling

Christian Holmboe*
University of Oslo, Norway

Constructing a data model for a problem area requires identifying and formulating some symbolic representation of the concepts involved, their characteristics, and the relationships between them. Taking a socio-cultural perspective on learning, analysis of classroom dialog is used to identify cognitive challenges met by novice students of data modelling. This paper shows how Vygotskyan theory of concept building sheds light on some psycholinguistic aspects of data modelling. The high-school students in the study displayed a lack of what will be called *metalinguistic consciousness*. Many of their problems were related to the conceptualisation process of forming entities and assigning appropriate labels to them. In teaching data modelling, there seems to be a need to focus more explicitly on the four-way relationship between (1) concrete or abstract objects of the world, (2) the terms denoting these objects, (3) the related subjective meaning and (4) the symbolic representation in a data model.

1. INTRODUCTION

More and more people need to be familiar with data modelling and system development (Mcfadden, Hoffer, & Prescott, 1998). A better knowledge of the cognitive challenges involved in database thinking will ensure more accurate training and possibly lead to more skilled practitioners. In order to facilitate better pedagogical content knowledge (Laurillard, 1993) for teachers and course designers of data modelling courses, this paper aims to identify and describe the nature of some of the conceptual challenges met by novice students of ER (Entity Relationship) data modelling.

Previous research has paid little attention to linguistic issues involved in data modelling. Since data modelling as a cultural practice is closely related to language, it is plausible that such issues play an important role as a cognitive challenge to data modellers – especially novice students. Data modelling is a discursive activity in at

*Corresponding author. Christian Holmboe, Department of Teacher training and School development, University of Oslo, Norway. E-mail: christian.holmboe@ils.uio.no

ISSN 0899-3408 (print)/ISSN 1744-5175 (online)/05/020143-19 © 2005 Taylor & Francis Group Ltd
DOI: 10.1080/08993400500150796

least two senses. First, it is discursive in the sense that a data model often is a product of verbal interaction between two or more people. Second, data modelling is discursive because it describes a chosen part of the world, using a specialised kind of symbolic language in which terms from everyday discourse are used as labels for groups of objects or abstract phenomena. Through analysis of classroom conversation transcripts, I will use the first of these discursive qualities to discuss the importance of the second aspect for successful data modelling. In this way I will try to pinpoint some of the difficulties experienced by novices when learning (i.e. becoming participants in the cultural practice of) data modelling. In doing so, I will concentrate on how a number of the students' problems are related to (1) the *conceptualisation* process of forming and labelling entities and (2) the apparent lack of *metalinguistic consciousness*. Conceptualisation in this context concerns the ways in which terms are attributed meaning, and vice versa how different meanings, as represented by relationships or entities, are assigned more or less suitable terms as labels. By metalinguistic consciousness I mean the students' awareness of the ways in which language is used in these processes.

2. BACKGROUND

Although several noteworthy contributions have been made to the understanding of the cognitive features of programming (for an overview, see e.g. Clancy, Stasko, Guzdial, Fincher, & Dale, 2001; Robins, Rountree, & Rountree, 2003), much of the research in *computer science education* has focused on failure rates of first year programming courses, and has presented different tools or teaching techniques for solving this problem (Holmboe, McIver, & George, 2001). Research on the teaching and learning of system development in general, and data modelling or database design in particular, has been less visible. A series of studies has compared usability, user performance or suitability of different visual modelling systems like the relational model, ER and UML (e.g. Chan, 1998; Peckham & Maryanski, 1988). These studies focus on methodology, or on language specific affordances and limitations, in order to identify qualitative differences between the systems. Equally important from a teaching perspective is insight into more general cognitive challenges inherent in the activity, regardless of the choice of design methodology or tool. In this respect, some influential contributions have been made by Batra and associates. They emphasise the complexity of the relationships between entities as a main obstacle to successful modelling (Batra, Hoffer, & Bostrom, 1990), and have also provided descriptions of different heuristic approaches to problem solving taken by novice students (Batra & Antony, 1994; Srinivasan & Teeni, 1995). These observations resemble the ones made with respect to *plan knowledge* in expert programming (Ehrlich & Soloway, 1984; Soloway, 1985). While the plan knowledge theory describes programming plans as a desired quality of expert programmers, Batra and associates demonstrate that novices tend to misapply their heuristic approaches due to restricted knowledge, either of data modelling as such, or of the problem domain to be modelled. Within the field of psychology of programming, a number of researchers have focused on the relationship between the constructs of natural language and those of programming languages (Murnane, 1993; Pane, Chotirat, & Myers, 2001; Taylor, 1990). A similar focus has been set by Chan and Goldstein (1997), exploring a new approach to allow formulation of relational database queries that are based on the user's knowledge of the real world. Common to most such studies is that they compare programming performance with or without the use of natural language constructs – either in terms of correctness of code, or of understandability of finnished code. The general findings support the intuitive assumption that closeness to natural language leads to improved performance. The present paper aims to shed further light on the relationship between natural language and the learning of data modeling as a collaborative activity.

2.1. Sociocultural Perspective on Learning

Groups of students in a classroom, collaborating on solving a problem, constitute examples of *intersubjective thinking* (Vygotsky, 1986). This implies that two or more members of a community collectively work on a cognitive activity. This collective cognitive process is enabled by the use of certain social *tools* (Säljö, 1998), the most important being language and discourse. *Learning* in this setting occurs whenever one member of the community appropriates some piece of information and makes it part of his or her own reasoning. This paper is written from the perspective that learning takes place through interaction with the environment, including fellow members of the community (Driver, Asoko, Leach, Mortimer, & Scott, 1994; Rappaport, 1998). The learner will gradually adopt existing, as well as develop his or her own, ways of handling the situations and resources available. This means that learning is seen as the development of skills for participation in different communities of practice (Wenger, 1998).

The classroom discourse also entails intersubjective thinking between student and teacher (i.e. a more skilled member of the social community). A teacher can, by using certain strategies (Lemke, 1990; Mercer, 1995), help a student solve a problem he or she would otherwise not be able to solve. Gradually this aid can be removed, until the student can master the problem himself. This strategy is called *scaffolding* (Wood, Bruner, & Ross, 1976), and works when the knowledge required to solve the task at hand lies within the student's *Zone of Proximal Development* (ZPD). Briefly, ZPD is described as the discrepancy between a person's individual mastery level "and the level he reaches in solving problems with assistance" (Vygotsky, 1986, p. 187).

2.2. Language and Concept Formation

According to Vygotsky, language is adopted by children and the meanings of words are inherited from adult language. It is, though, necessary to distinguish between the development of spontaneous concepts (i.e. everyday language) and scientific concepts (i.e. institutionalised or specialised language).

Starting out as syncretic images (visually based bonds between objects), spontaneous concepts subsequently take the form of complex thinking, where words adopted from adults refer to corresponding physical objects or groups of objects based on concrete and factual bonds. Through increasing levels of abstraction, genuine concepts gradually develop as logical, abstract generalisations of such groups of objects. The child's development of spontaneous concepts thus moves from the concrete and specific to the general and abstract.

The child's learning of scientific concepts, on the other hand, proceeds the other way around. Based on the formal definitions delivered from a teacher or textbook, scientific concepts are initially highly generalised and abstract. Through systematic and repeated application of the corresponding terms, they then gradually develop towards concrete phenomena.

In this paper I illustrate how data modelling as a collaborative activity incorporates each of these two very different psychological activities.

3. RESEARCH METHOD

3.1. Methodology

The rationale of most quantitative research is to provide empirical data to support or falsify one or more predefined hypotheses. In an emerging research field, there may not be a sufficient knowledge base for the formation of such research hypothesis. A qualitative research methodology is therefore "concerned with inducing hypotheses from field research" (Silverman, 1993, p. 2) rather than testing predefined hypotheses. Only few qualitative studies have been published on learning in computer science (e.g. Booth, 1992; Kolikant, 2004; Kolikant, Ben-Ari, & Pollack, 2000; Taylor, 1990). These have in different ways demonstrated the strengths and values of such approaches to the research field. The present paper represents a further contribution to this latter type of research.

Discourse and cognition is strongly interlinked and can not be seen independently from one another (Edwards, 1997). One approach to studying cognition is therefore to study the educational discourse that brings about learning. Such talk data can be handled quantitatively through coding and subsequent statistical analysis. Naturally occuring classroom interactions, however, do not easily lend themselves to rigourus coding, because they are highly contextual (Littleton, 1999). The interpretation of an utterance is dependent both on the immediately preceding and the subsequent turns in the interaction (Mercer & Wegerif, 1999), and on the larger context (e.g. the classroom culture and the academic and social history of the participants).

3.2. Material and Data Collection

Three different classes (10–14 students each) of senior high school students (age 18) were visited once a week for 3–4 months. The classes were from two different schools, and two different teachers were involved. Both classrooms were organized as problem oriented workshops with pairs or small groups of students solving problems in front of a shared computer. In general these pairs or groups were the same for the duration of the period. The visits took place during the final part of a specialised course in system development running over 5 lessons per week for two years. The course curriculum covers most areas of system development and analysis, including data modelling in the Entity Relationship (ER) modelling language.

Each visit took place in the ordinary classroom, and usually lasted for a double period (i.e. 2 × 45mins). The researcher remained in the classroom throughout the visits. One small dictaphone was used to record the conversations among the students in one pair or group at a time. The dictaphone was discretely placed on the desk or screen in front of the students, while the researcher usually observed from behind taking notes. These notes later made it possible to identify the individual participants voices. After the first visit, the students did not seem to pay any attention to the presence of the dictaphone or the researcher. The material also contains several conversations between the teacher and one or more students. With varying intervals, the researcher would move the dictaphone to a different group. The sequences of continous recording in one group vary from 5 mins to more than half an hour with an average of 15–20 mins. A 90 min visit usually resulted in 3–6 interactional sequences. The total material thus comprises 100+ sequences.

Full detailed transcripts were made of all sequences according to the Jefferson convention, as described in Potter (1996, pp. 233–234). Longer periods of non-audible talk or non-academic chatting were omitted from the transcripts.

3.3. Method and Data Analysis

The proponents of discursive psychology have coined the ideal of unmotivated looking as a counterpart to coding into researcher's predefined categories (Potter, 1997). The analysis undertaken in this paper have been based on this ideal, meaning that no predefined hypothesis were set out, neither for the data collection nor for the analysis. It was furthermore not an aim to provide generalised accounts of the students' learning, but rather to offer descriptions of cognitive challenges illustrated by excerpts of interaction transcripts.

The material was inspected and reinspected repeatedly, searching for any discursive patterns or events that would stand out in some manner. For this inspection, both tape recordings and transcripts were used extensively and interchangably. The analysis was data-driven in that observations were registered sequentially as they were made in the analysis of the data. Not being restricted by a predefined hypothesis, the observations varied a lot in type and scope from purely interactional events through tool-specific references to sequences illustrating conceptual understanding. This provided a large and varied set of obervations. The material included a number of different observations related to conceptualisation and the use of natural language constructs in the data modelling activity. These topics were chosen for further analyses in the present paper. The excerpts presented below illustrate the observations made that are relevant to the issues of conceptualisation and metalinguistic consciousness. They have been translated from Norwegian and some of the transcriptional detail that was included for analytical purposes have been removed to improve readability of the excerpts where this detail is not explisitly attended to in the discussion. As already mentioned, no claim is made about the genralisability of these observations. They demonstrate conceptual challenges faced by students of data modeling that should be recognised and considered by teachers of this topic.

The examples presented concern two of the different problems that the students worked on during this period. Problem 1: ‘‘Build a data model for a system to keep track of which student had which *form master* at what time’’. (In Norwegian schools, each class (i.e. group of 15 – 27 students) is assigned a dedicated teacher, referred to as a form master, with certain administrative responsibilities in addition to the teaching. The class can have the same form master for several years). Problem 2: ‘‘Build a *crime registry* for the police to hold data about crimes, people involved, and other relevant information’’. Problem 1 was given orally as an ‘‘exercise of the moment’’, whereas problem 2 was addressed by all students in two of the classes for the duration of the three-month observation period. Other problems addressed were of similar nature and complexity.

4. ANALYTIC FRAMEWORK

Whereas *phenomena* are things and situations in the real world, *entities* are the corresponding representations of these phenomena by means of a data model. The analysis will show that it is fruitful, for the understanding of students' problems with data modelling, to differentiate between the types of entities that the students are supposed to construct by levels of abstraction based on the phenomena that the entities represent. The formation of entities to represent different types of phenomena occurs in two fundamentally different ways, corresponding to the directions already described for children's development of spontaneous and scientific concepts. Two main types of phenomena are outlined and illustrated by examples of entities from a suggested data model for problem 2. The entire data model (see Appendix) represents a cross-section of the ones made by the different groups of students.

4.1. Instances or Types

Independently of the levels of abstraction, there are two semantically different ways of modelling a given phenomenon; (1) by making entities that represent lists of individual objects and (2) by making entities that represent lists of types of objects – more or less generalised. The resulting kinds of entities will be called instance-oriented and type-oriented respectively. A car salesman would probably use an instance-oriented entity for modelling his merchandise (e.g. **car**(carnumber, make, colour, etc. . .)), whereas a furniture store would be better off with a type-oriented entity for their merchandise (e.g. **commodity**(identifier, make, #items_in_stock, etc...)). Both of these types of entities may occur within each of the abstraction levels presented below.

4.2. Spontaneous Concepts

The child develops its spontaneous concepts by sorting the objects experienced in the world into categories that are labelled by some term. These categories need not be exclusive or consistent, but make sense in discourse with others at a given point in time and in a given setting. There is a trajectory from the concrete phenomena in the experiential world to the abstraction represented by the concept. The process for data modellers is similar, but moves faster through these two stages and adds a third stage where the concept is modelled by an entity. This implies the following schematic development:

World - > Language - > Model

4.2.1. Concrete and semiconcrete phenomena. The world may be seen as consisting of a number of different types of physical objects visible to the eye and touchable by the hand. In our everyday language, these objects are categorised into more or less generalised concepts labelled by some term. Such *concrete phenomena* correspond to Vygotsky's spontaneous concepts. It is easy to see whether one person's concept matches that of another person (i.e. refers to the same set of physical objects or concrete phenomena in the referential world). It is furthermore easy to produce sensible sets of attributes to describe such phenomena with entities in a data model (Batra et al., 1990).

In our crime registry, examples of instance-oriented and type-oriented entities respectively may be **criminal** – where one would need information about each individual person separately, and **commodity** – where it might be sufficient to register that 15 computers and 3 photocopiers were stolen from an office.

The semiconcrete phenomena include all natural categories (Rosch, 1978; Roth, 1995) that are not concrete phenomena.. A natural category is a set of phenomena in the world referred to by a spontaneous concept such that it is intuitively evident what the corresponding term means in everyday language.

From the crime registry, we can identify **crimetype** and **denouncement** as examples of semiconcrete phenomena. Of these, the last is by necessity an instance-oriented entity, since a given **denouncement** will always be unique, whereas **crimetype** may be thought of as a type-oriented entity.

4.3. Scientific Concepts

A person's development of scientific concepts (Vygotsky, 1986) resembles what happens in data modelling when relational phenomena are introduced. The ER modelling technique introduces entities that are not a priori related to concepts in the surrounding world. Later some of these may be assigned a meaning corresponding to phenomena of the experiential world. In the same schematics that were used for spontaneous concepts above, this could be illustrated as follows:

Model (mechanical) - > Language - > World

4.3.1. Relational Phenomena. A given **crime** may include several different **crimetypes**, like robbery, violence or murder. Similarly, each different **crimetype** may occur as part of a **crime** on more than one occasion. To avoid ambiguity of the data model, the relationship must be reified as an entity of its own (Figure 1).

I call the product of such an entitisation a *relational phenomenon*. In this example, the relational phenomenon is **crimepart**, which is not a common everyday word, but still makes sense. The connotations to the term **crimepart** may imply that this too is a type-oriented semiconcrete phenomenon just like **crimetype**. The intended meaning of **crimepart** hence differs from at least one possible conception of it. **A_particular_part_of_a_specific_crime** would perhaps be a more precise choice. The challenge, for students as well as for professional data modellers, lies in understanding what the meaning of such an entity is, what information it holds, and how the corresponding phenomenon relates to the other parts of the scope for the data system. (The issue of labelling relational entities will be revisited in a subsequent section of the paper.) The relational entities inherit the *primary keys* from each of the original entities as *foreign keys*. The combination of these foreign keys may be used as a combined primary key for the relational entity (see Appendix). Since these two foreign keys often are the only two attributes of a relational entity, the content of the corresponding table in the database will be a list of ordered pairs. (In the case of the **crimepart,** this means a list of pairs of crime-ID and crimetype.)

Through entitisation of a many-to-many relationship where one or both of the related entities are relational, the level of abstraction increases further. Whereas first order relational phenomena gave a list of ordered pairs of concretes or semiconcretes, higher order relational phenomena accordingly gives pairs of such abstract pair

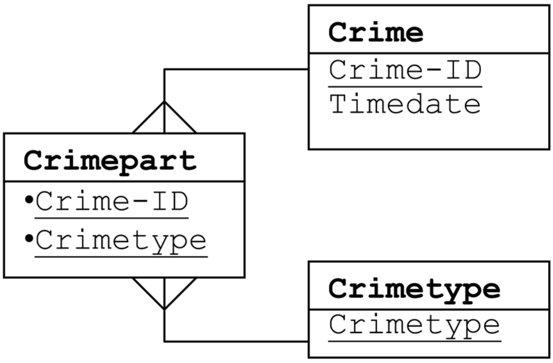

Figure 1. Entitisation of the relationship between **Crime** and **Crimetype**

combinations. The number of foreign keys jointly constituting a candidate key increases, making it more difficult to grasp the meaning of the entity.

Sometimes a relational phenomenon may correspond to a spontaneous concept. If realised by the data modeller, this simplifies the attribution of meaning to the entity. An example of such an entity from our problem is **sentence**, formed from the many-to-many relationship between **judgement** and **sentencetype**. Despite being expected to be rather abstract, the concept of **sentence** is quite familiar in terms of label, meaning and relevant attributes. It is thus also a semiconcrete entity corresponding to a spontaneous concept.

5. FITTING THE WORLD INTO BOXES

The essence of data modelling is to identify phenomena or objects within the Universe of Discourse (UoD) that constitute suitable entities in the data system. The students need to fit their understanding of the "world" into "boxes" (i.e. entities) and label them using "words" from their spontaneous language.

5.1. Entitisation

Entitisation by the reification of a many-to-many relationship into an entity is usually not technically problematic for the students. They appear to be highly skilled in entitisation using the software tool. They have learned to replace any many-to-many relationship that might occur with a new entity regardless of what the relationship implies in terms of meaning. The problem arises when it is time to name this new abstract entity. It is often difficult to grasp the meaning of a relational phenomenon, due to the fact that the instances of the entity are "no longer rooted in the original situation and must be formulated on a purely abstract plane, without reference to any concrete situation or impressions." (Vygotsky, 1986, p141).

Students S and T in Excerpt 1 are working on the crime registry. They have introduced a relational entity between **denouncer** and **denouncement**, which they have called **denouncementregistration**. They continue by considering the cardinality restrictions of the relationship between **denouncer** and **denouncementregistration** (lines 105 ff).

Failing to recognise the meaning of the new abstract relational phenomenon, they conclude correctly that each **denouncer** can give many **registrations** (line 105), but also that a **denouncementregistration** can be made by several **denouncers**. This makes T conclude that they have yet another many-to-many relationship (line 120), which is supported by S (line 121). The two students do not seem to realise the difference between the semiconcrete phenomenon of **denouncement** and the relational phenomenon of **denouncementregistration**, and end up recursively generating relational phenomena from many-to-many relationships. The students do not seem to conceive the instances of **denouncementregistration** as physically existing objects. Hence this entity remains a generic concept that will relate to other entities in a generalised manner. It appears that the students do not consider each

Excerpt 1:

(Students S, T)	
101	T: we must have a relationship between denouncer and
102	Denouncementregistration and
103	S: yes
104	T: that is to say that one denouncer
105	Can have several denoucementregistrations?
106	S: eh hn
107	T: like this, look now
108	((T working on the computer))
109	Yes
110	S: and then you have
111	Eh on the bottom there then you have one and many because
112	The denouncer he if he
113	He gives
114	If he is a denouncer then he must reasonably have given one
115	Denoucement, right
116	T: or he could give many.
117	The denouncementregistration, it can have
118	One
119	But zero
120	Or wait a minute then we have many-to-many
121	S: yes it ends up with many-to-many then.

single registration (since they have probably never seen one), but rather the idea of a registration in general.

In the crime registry, it may be desirable to store information about which **criminal** stole what **commodities** on which **crime**. Consider a relational phenomenon, **participation**, containing pairs of **criminal** and **crime**. We need to make a relationship between **commodity** and this relational phenomenon. Simply relating **commodity** to the **crime** or to the **criminal** (Figure 2) would not suffice, as we would lose part of the information we want to preserve.

The students in Excerpt 2 have not yet labelled the relational entity between **criminal** and **crime**. They have simply created it because they discovered a many-to-many relationship. In their model, they have related **commodity** to the **criminal**. The teacher, R, gives a rather long explanation leading up to the conclusion that the relationship from **commodity** should be made to the relational entity (lines 201–212). In line 214, the teacher checks that the students have understood, by pausing for a moment to let one of them finish the sentence (Lemke, 1990). Instead of confirming that the relationship should be dragged into the relational entity, S suggests 'the **crime**' as alternative to 'the **thief**' (line 216).

It seems that connecting an entity to a relational phenomenon that has not yet got a name, let alone any meaning, is beyond the ZPD for these students. Even with a rather thorough scaffolding (Wood et al., 1976) from the teacher, student S lands on

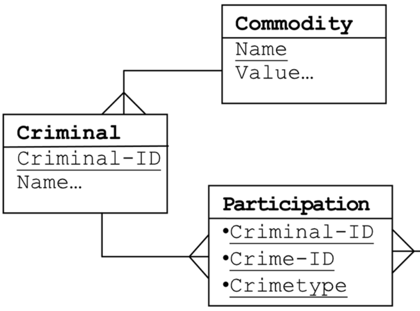

Figure 2. **Commodity** should be related to **Participation**.

Excerpt 2:

(Teacher R; Students S, C)	
201	R: here youhave said that a thief can participate in many
202	crimes
203	C: mm
204	R: and then you say that one crime can uh:
205	Can be do- carried out by several thieves together. And
206	Then you say that for each time you register that a thief
207	Participates in a crime then you must go into this table
208	S: mm
209	R: then information is stored there
210	C: yes
211	R: for each time you have connected a thief to a crime, then
212	You can also say what commodities he has taken
213	C: mm
214	R: and then you drag that relationship not into the thief,
215	But into ((small pause))
216	S: the crime

the wrong answer or simply makes a wild guess based on elimination. Note also the inconsistency in use of terms between the data model and the dialogue. Whereas the entity is labelled **criminal**, both the teacher and the student use the term *thief* when discussing it.

5.2. Labelling

A symbolic representation (i.e. a word) can be paired with a corresponding meaning either by choosing a term to label a given meaning, or by finding a meaning to suit a given term (i.e. world-to-concept or concept-to-world). The most obvious task for the student of data modelling is to recognise phenomena within the UoD and assign an appropriate label to it. But the challenge is just as great the other way around. During the activity of data modelling, several entities occur (mainly from entitisations) that do not have an a priori sensible meaning to the students. The software used in this course invites them to label the entity by some term before it occurs on the drawing or is assigned any attributes. The term chosen then needs to be associated with some meaning, for the students to be able to use the entity appropriately in their further work. Hence, the second aspect of the challenge of labelling, is the assigning of meaning to constructed terms denoting abstract relational phenomena. This subactivity has bearing on the preconceptual thinking of *complexes as collections* (Vygotsky, 1986, p. 114). In collections the nature of a concept is revealed through the attribute(s) that differ from one instance to the next (i.e. the primary key).

The teacher in Excerpt 3 is aware of this fact when he asks M to provide some attributes to his entities as an aid for understanding what information the relational entity he has just created will contain.

In order to envision what the entities represent and how they relate to each other, the students need to form and assign "new" *subjective contents* to the formal descriptions (i.e. terms) they use to denote the phenomena represented by the entities in the data model. This subjective content or *meaning* often differs from the preconception the student might have of the term from everyday language. Sometimes the "new" understanding represents a more precise conception, whereas the former one might have been somewhat misguided or vague. This attribution of new meaning to a formerly known term also resembles concept building in the learning of a foreign language (Vygotsky, 1986, p. 197).

Searching for a term to label an entity, many students tend to browse their vocabulary for any topic-relevant word that may be associated with the phenomenon

Excerpt 3:

(Teacher R; Student M)	
301	R: Have you entered any attributes there?
302	M: No
303	R: No
304	You get a little help from that if you
305	If you enter attributes first
306	Say to that one and that one
307	And then you see what happens when you create eh when
308	You entitsize
309	M: mm
310	But I don't know what to call it
311	R: no, but ehm that you can change the name
312	That is later
313	Cause it isn't that easy to call it something when you don't
314	Know what it will contain
315	M: No that's right

at hand. This might be a sensible strategy, but sometimes it fails. Two students were working on problem 1. They both chose **class** as label for the relational entity between **student** and **form_master**. There are two possible reasons why this label may have been chosen. Firstly, the strategy outlined above would naturally bring **class** as an alternative for a data model for **students** and **form_masters**. **Class** is a semiconcrete phenomenon that the students have a reasonable conceptual understanding of, and that has not yet been used in the data model. Furthermore, a **class** is what connects a **student** to his or her **form_master**. The choice of the term **class** as label is made without considering the meaning of the relational entity as defined by its attributes. This shows that the students tend not to think in terms of attributes but refer to their conceptual understanding of the world.

5.2.1. The Name Decides. "The word, to the child, is an integral part of the object it denotes" (Vygotsky, 1986, p. 222). A group of children were told that "in a game a dog would be called 'cow'". When subsequently asked if a 'cow' has horns, the children answered "yes [...] if it's called a cow, it has horns. That kind of dog has got to have little horns." Sometimes this manner of thinking can be recognised in the students as well. Emphasis is put on finding the "correct" label or name for an entity. Once the choice is made, the name has a major influence on the further modelling.

Working on problem 1, the two students have suggested that the abstract entity could be labelled **class**. This might have worked, had they not taken the choice of name literally. In Excerpt 4 the students go on to discuss the cardinality restrictions of the relationship between **class** and **student**. Based on everyday meanings of these terms, this is a one-to-many relationship, which is the opposite of the cardinality for the relationship they have actually created between **form_master/student-pair** and **student** (see Appendix). Excerpt 5 provides a further illustration of how the name decides the meaning of an entity for these students.

Teacher C directs student J2's attention towards the information stored in the combined key-attributes which have appeared automatically after the entitisation (lines 501–505). J2 misinterprets the question, and starts to assign new attributes to the entity. Since the entity is called **class**, the attributes chosen are ones that are relevant for a **class**. C tries again to bring attention to the attributes already present

Excerpt 4:

(Students J, J2)	
401	J2: Well I'm thinking such that e
402	Cause I think that
403	That a class should have a form master
404	J: a class must have a form master
405	J2: and the fact that a class, it should at least consist of
406	One student, but then ...

Excerpt 5:

(Teacher C; Student J2)	
501	C: Yes, what kind of information do you have then?
502	For each single like that down in that
503	In that table
504	For each line.
505	What pieces of information is it that you have there?
506	J2: in class you mean?
507	C: yes in class
508	In the class-table
509	J2: eeh classcode
510	C: yes
511	J2: and then ehm
512	Possibly which school it is in,
513	With which track
514	C: yest but the way it
515	The way that it stands now then
516	So far?
517	J2: no:w?
518	C: yes
519	J2: classcode anyhow
520	C: no, you don't have attribute which is called that
521	J2: no I haven't added it
522	But e: it
523	C: well well, okey so you want to have a classcode and then

(line 514). If J2 had managed to follow this line of reasoning, she might have discovered that the instances of the entity are **student/form_master** pairs, whereas an instance of a **class** should have more than one **student**. Unfortunately this does not happen. Neither the origin nor the content of the entity seems to have any impact on J2's interpretation of its meaning. The meaning assigned by J2 to the entity is exclusively generated from the chosen label, **class**.

5.2.2. Relationship to Everyday Language. Being a discursive activity, data modelling is initially rooted in the students' everyday language. The terms used are found among their spontaneous concepts and carry a predefined set of connotations. This connection to everyday semantics is imperative for the common understanding of the data model. Simultaneously, this connection may provide a false sense of familiarity (Holmboe, 2004). The meaning of a term in a data model may need to be detached from the meaning of the same term in everyday language. In Excerpt 6 student N uses the term **sentence**, while **punishment** might have been a more precise choice.

Student N's utterance is part of a longer discussion where the students alternately let the terms **sentence** and **sentencetype** refer to instance- or type-oriented entities. (In Excerpt 6 **sentence** is described as a type-oriented entity). In the same sequence, Excerpt 6.

(Student N)	
601	N: A crime can receive different sentences, right, cause it can
602	Get both a fine and prison.

the term **sentence** is also used with meanings resembling the phenomena that are modelled as **judgement** and **sentenceserving** (see Appendix). These students obviously do not have a mutual understanding of the meaning of each of these terms, or of the difference between the instance-oriented and type-oriented ways of modelling a phenomenon.

Lack of mutual understanding has significant influence on the way the data model is constructed. In everyday use, the meaning of the term **sentence** would most likely be understood without ambiguity in a statement like the one in Excerpt 6. In a data model, however, such distinctions are crucial. The students have to choose the words carefully and use more precise formulations when discussing the data model than they normally would in everyday discourse. Unfortunately, it doesn't seem that they are able, or sufficiently aware of the need, to do so.

6. CONCLUSION

I have illustrated how a lack of common understanding, or a lack of detachment from everyday use of terms, can cause difficulties for an otherwise fruitful attempt at solving a data modelling problem. Hazzan points out that "when students meet objects which are abstract for them, students rely on their previous experience, [assign] the unknown object familiar properties, and thus, reduce the level of abstraction." (Hazzan, 2003). This resembles closely the observed problems with a label determining the meaning of an entity.

The challenge for a student of system development is to translate the understanding he or she has of the world into a consistent description, that is understandable and non-ambiguous to the computer system, to others involved in the project, and indeed to him or her self. The different activities and challenges involved include labelling phenomena, forming and assigning "new" meaning both to existing and constructed terms, and abstracting and generalising objects into groups. The challenges furthermore include reifying relational phenomena for further handling as entities in their own right. Such relational entities may need additional attributes and may have relationships connected to them. When well formed and accounted for, they can then be included in the intersubjective as well as the intrasubjective discourses of the classroom. Finally, it is necessary to backtrack in order to recognise when constructed relational phenomena may correspond to a familiar meaning, which will, in turn, simplify the understanding of the data model. In all these subactivities, it is obvious that the community of practice (i.e. the classroom as well as the surrounding semiotic environment) plays an important role. The choices and specifications are made by groups of students in discursive interaction with each other as well as the teacher.

It seems that learning data modelling and learning to program have a lot in common. The present study confirms the findings of Taylor (1990) that the students tend to confuse natural language and formal elements of the programming language. The main challenge for the students is to use language to describe the world in a context-independent and stringent manner so that a computer can "understand" it. This may apply to most areas of computer science. Using terms and phrases from their everyday language, the computer science students have to define exactly what is meant by each expression – or it is defined for them (e.g. in a programming language). These definitions will often differ from the students' prior understanding of the terms. The question of whether or not this aspect proves problematic for students when learning to program needs to be explored further.

6.1. Implications for Teaching

Teachers and teacher trainers in computer science need to pay more attention to the linguistic or semiotic aspects of the subject. Concerning data modelling in particular, we have seen that the students show a lack of *metalinguistic consciousness*. They seem to be unfamiliar with the thought that a term does not have a predefined meaning which is uniform to all users of a given natural language. In the teaching of data modelling there is a need to focus more explicitly on the four-way relationship between (1) concrete or abstract objects of the world, (2) the terms denoting these objects, (3) the related subjective meaning and (4) the symbolic representation in a data model. Furthermore, a focus on the descriptive aspects of the activity may prove helpful. It might be fruitful to have the students explain to each other, rather than to the computer, how the phenomena should be related and why.

REFERENCES

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

Batra, D., Hoffer, J.A., & Bostrom, R.P. (1990). Comparing representations with the relational and EER models. *Communications of the ACM*, *33*, 126 – 139.

Booth, S. (1992). *Learning to Program: A phenomenographic perspective* (Vol. 89). Gothenbourg: University of Göteborg.

Chan, H.C. (1998). User performance differences between relational and entity relationship models: a summary review of the literature. *Behaviour & Information Technology*, *17*(1), 59 – 61.

Chan, H.C., & Goldstein, R.C. (1997). User-database interaction at the knowledge level of abstraction. *Information and Software Technology*, *39*(10), 657 – 668.

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.

Driver, R., Asoko, H., Leach, J., Mortimer, E., & Scott, P. (1994). Constructing scientific knowledge in the classroom. *Educational Researcher*, *23*(7), 5 – 12.

Edwards, D. (1997). *Discourse and Cognition*. London: Sage Publication.

Ehrlich, K., & Soloway, E. (1984). An empirical investigation of the tacit plan knowledge in programming. In J.C. Thomas & M.L. Schneider (Eds.), *Human factors in computer systems* (pp. 113 – 133). Norwood, New Jersey: Ablex Publishing Corporation.

Hazzan, O. (2003). How students attempt to reduce abstraction in the learning of mathematics and in the learning of computer science. *Computer Science Education*, *13*(2), 95 – 122.

Holmboe, C. (2004). A Wittgenstein approach to the learning of OO modelling. *Computer Science Education*, *14*(4), 275 – 294.

Holmboe, C., McIver, L., & George, C. (2001). Research Agenda for Computer Science Education. In G. Khadoda (Ed.), *Proceedings of the 13th annual workshop of the Psychology of Programming Interest Group* (pp. 207 – 223). Bournemouth, UK.

Kolikant, Y.B.-D. (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.

Kolikant, Y.B.-D., Ben-Ari, M., & Pollack, S. (2000). The Anthropology of Semaphores. *ACM SIGCSE Bulletin (Proceedings of SIGCSE Technical symposium)*, *32*(3), 21 – 24.

Laurillard, D. (1993). *Rethinking university teaching: A framework for the effective use of educational technology*. London: Routledge.

Lemke, J.L. (1990). *Talking science: Language, learning and values*. Norwood, NJ: Ablex.

Littleton, K. (1999). Productivity through interaction: An overview. In K. Littleton & P. Light (Eds.), *Learning with Computers; Analysing productive interaction* (pp. 179 – 194). London: Routledge.

Mcfadden, F.R., Hoffer, J.A., & Prescott, M.B. (1998). *Modern database management* (5th ed.). Reading, MA: Addison Wesley.

Mercer, N. (1995). *The guided construction of knowledge. Talk amongst teachers and learners*. Clevedon: Multilingual Matters Ltd.

Mercer, N., & Wegerif, R. (1999). Is 'exploratory talk' productive talk? In K. Littleton & P. Light (Eds.), *Learning with computers. Analyzing productive interaction* (pp. 79 – 101). London: Routledge.

Murnane, J. (1993). The psychology of computer languages for introductory programming courses. *New Ideas in Psychology*, *11*(2), 213 – 228.

Pane, J.F., Chotirat, R., & Myers, B.A. (2001). Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human-Computer Studies*, *54*(2), 237 – 264.

Peckham, J., & Maryanski, F. (1988). Semantic data models. *ACM Computing Surveys*, *20*(3), 153 – 189.

Potter, J. (1996). *Representing reality; discourse, rhetoric and social construction*. London: Sage Publications.

Potter, J. (1997). Discourse analysis as a way of analysing naturally occurring talk. In D. Silverman (Ed.), *Qualitative research: Theory, method and practice* (pp. 144 – 160). London: Sage Publications.

Rappaport, A.T. (1998). Constructive cognition in a situated background. *International Journal of Human-Computer Studies*, *49*(6), 927 – 933.

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

Rosch, E.R. (1978). Principles of categorisation. In E.R. Rosch & B. Lloyd (Eds.), *Cognition and categorisation*. Hillsdale, NJ: Laurence Erlbaum Associates.

Roth, I. (1995). Part I: Conceptual categories. In V. Bruce (Ed.), *Perception and representation: current issues*. Buckingham: Open University Press.

Silverman, D. (1993). *Interpreting qualitative data: Methods for analysing talk, text and interaction*. (1st ed.). London: Sage Publications.

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

Srinivasan, A., & Teeni, D. (1995). Modeling as constrained problem-solving - an empirical-study of the data modeling process. *Management Science*, *41*(3), 419 – 434.

Säljö, R. (1998). Learning as the use of tools: A sociocultural perspective on the human-technology link. In K. Littleton & P. Light (Eds.), *Learning with computers. Analyzing productive interaction* (pp. 144 – 161). London: Routledge.

Taylor, J. (1990). Analyzing novices analyzing prolog - What stories do novices tell themselves about prolog. *Instructional Science*, *19*(4 – 5), 283 – 309.

Vygotsky, L. (1986). *Thought and language* (A. Kozulin, Trans.). Cambridge, MA: MIT Press.

Wenger, E. (1998). *Communities of practice. Learning, meaning and identity*. Cambridge: Cambridge University Press.

Wood, D.J., Bruner, J.S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Psychology and Psychiatry*, *17*, 89 – 100.

APPENDIX

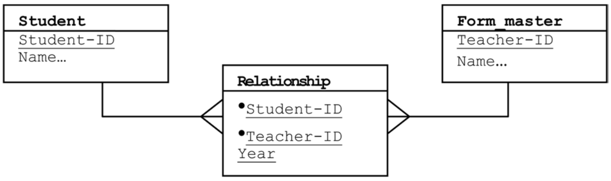

Figure 3. Sample data model for problem 1 (student / form_master)

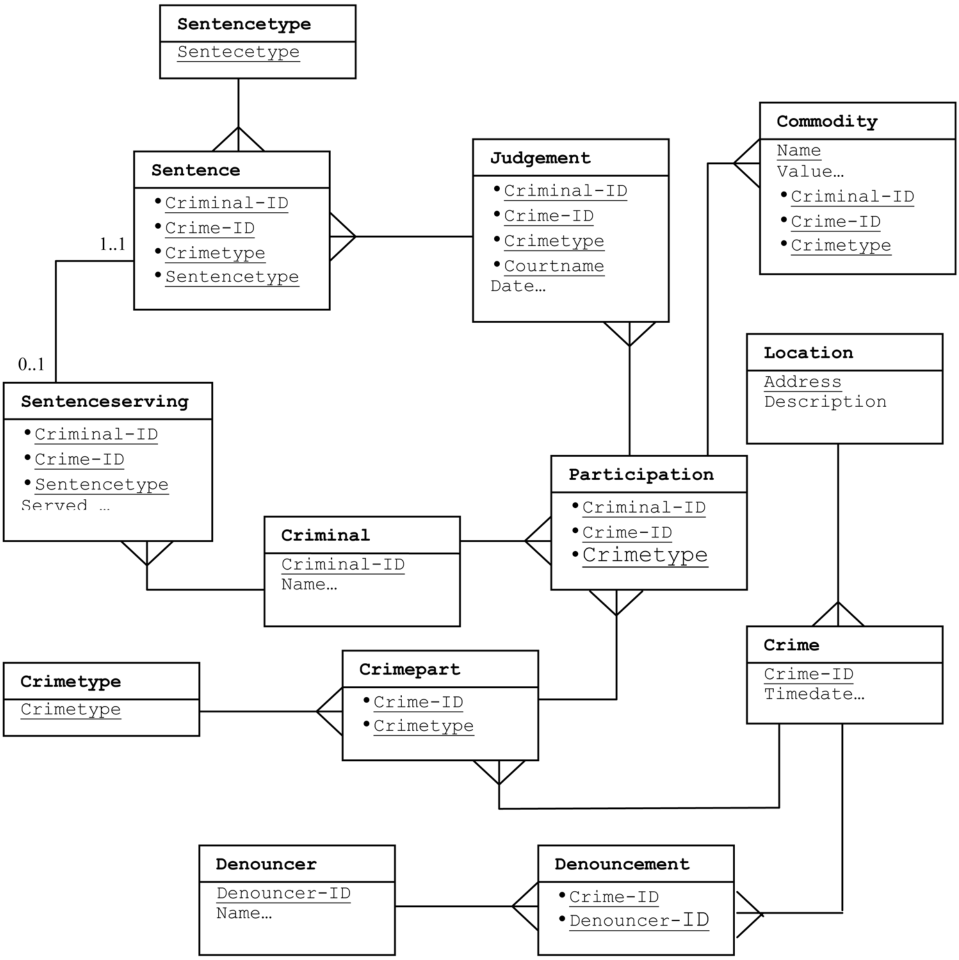

Figure 4. Sample data model for problem 2 (the crime-registry)

Computer Science Education
2004, Vol. 14, No. 4, pp. 277–296

Taylor & Francis
Taylor & Francis Group

A Wittgenstein Approach to the Learning of OO-modeling

Christian Holmboe
Department of Teacher Education and School Development,
University of Oslo, Oslo, Norway

ABSTRACT

The paper uses Ludwig Wittgenstein's theories about the relationship between thought, language, and objects of the world to explore the assumption that OO-thinking resembles natural thinking. The paper imports from research in linguistic philosophy to computer science education research. I show how UML class diagrams (i.e., an artificial context-free language) correspond to the logically perfect languages described in *Tractatus Logico-Philosophicus*. In *Philosophical Investigations* Wittgenstein disputes his previous theories by showing that natural languages are not constructed by rules of mathematical logic, but are language games where the meaning of a word is constructed through its use in social contexts. Contradicting the claim that OO-thinking is easy to learn because of its similarity to natural thinking, I claim that OO-thinking is difficult to learn because of its differences from natural thinking. The nature of these differences is not currently well known or appreciated. I suggest how explicit attention to the nature and implications of different language games may improve the teaching and learning of OO-modeling as well as programming.

1. INTRODUCTION

When making a data model or writing a program, the system developer, whether expert or novice, always relies on his or her underlying understanding of the problem domain to be modeled or represented. Probably one of the most common mistakes CS lecturers make is to assume that students have appropriate understandings of the problem domain (and of the computer's capabilities). Quite often students' understandings are limited or even erroneous, especially since, due to their relative youth, students are usually

Address correspondence to: Christian Holmboe, Department of Teacher Education and School Development, University of Oslo, Oslo, Norway. E-mail: christho@ils.uio.no

10.1080/08993400420003034447$16.00 © Taylor & Francis Ltd.

less experienced or knowledgeable in several of the domains involved in the given tasks. Even when students have the appropriate domain knowledge, the depth of knowledge normally differs significantly from one person to the next. Thus, two students who collaborate on a project could have very different understandings of what is meant by the relevant terms and phrases. In fact, the appropriate or determined meanings of the terms involved in a system description or the relationships among them are not given a priori, but are always open to interpretation. This paper describes the theoretical foundations for these problems and make suggestions for how to overcome some of them.

Many authors explain that object orientation has as its main aim to enable system developers to model the world in the same manner that they envision it in a natural setting. For example, Coad and Yourdon (1991) explain:

> OOA – Object Oriented Analysis – is based on concepts that we first learned in kindergarten: objects and attributes, wholes and parts, classes and members.

The truth of this claim is generally taken for granted and left undisputed. Nevertheless, several researchers have shown that it is difficult to learn OO-modeling and design (Andersen, 1997; Shoval & Shiran, 1997; Tegarden & Sheetz, 2001). In the late 1980s, a number of studies examined the difficulties students have learning procedural programming (some significant contributions can be found in Hoc, Green, Samurcay, & Gilmore, 1990; Soloway & Sleeman, 1986; Soloway & Spohrer, 1989). One main finding in several of these studies is that novices tend to attribute human interpretation skills to the computer (e.g., Pea, 1986), both on a logical and on a linguistic level. du Boulay (1986), in turn, focused on the problems of having to learn not only the programming language, but a number of other systems or languages like the editor, the debugger, and so forth. The latter point is particularly relevant to the work presented in this paper.

A number of studies have shown relationships between learning to program and natural language use. Shneiderman (1980) demonstrates how programmers benefit from using everyday terms when labeling variables and procedures, which supports the claim that closeness to natural language and thinking makes modeling, as well as programming, easier to learn. Similarly, Petre (1990) and Pane, Chotirat, and Myers (2001) show how techniques of pseudocode and natural language algorithm descriptions help students understand or produce programs. Other researchers have focused on the differences between programming languages and natural languages, demonstrating language-related difficulties that students may face (du Boulay, 1986; Holmboe, 2005; Taylor, 1990).

A different strand of research has shown how the activity of programmers relies on special problem-solving strategies, referred to as *programming plans* (Soloway, 1985) or *schemas* (Détienne, 1990). A computer programmer constructs a program by assembling such plans in the appropriate order. The problems that novices experience in trying to assemble plans have been attributed to their misunderstanding or misapplication of programming plans (Ehrlich & Soloway, 1984). In the area of data modeling, Dinesh Batra and associates have performed a number of studies comparing novice and expert behavior (Batra, 1993; Srinivasan & Teeni, 1995). They describe problems such as those related to programming plans or schemas as *misapplied heuristics* (Batra & Antony, 1994). They discovered that their students show a tendency to adopt intuitive problem-solving techniques (i.e., heuristics) that do not take into account the fact that the translation problem at hand is supposed to be "understood" by a computer, in addition to being readable and intelligible to a thinking person.

While several of these studies address linguistic issues of learning to program in general, little has been done to help understand the apparent discrepancy between the alleged naturalness of OO-thinking and the evident problems faced by students of OO-modeling in particular. The present paper will contribute by focusing on this discrepancy, using the two major theories of Ludwig Wittgenstein as a point of departure. In doing so, the paper demonstrates how importing theories from linguistic philosophy can provide useful insight to the field of computer science education research.

Ludwig Wittgenstein's first book, *Tractatus Logico Philosophicus*, was originally published in 1921. In this book, Wittgenstein (1961) outlines a set of postulates about the use of language to describe the world. From a mathematical point of view, he gives an idealized description of a *logically perfect language*. In this paper I demonstrate how object-oriented modeling can be seen as an example of such a logically perfect language. Wittgenstein's second major publication is the *Philosophical Investigations* (Wittgenstein, 1958). The more pragmatic view of language presented in the latter book helps explain the apparent contradiction between the assumed naturalness of OO-thinking and problems documented in the process of learning this activity. From a sociocultural perspective, I discuss the implications of these two theories for learning OO-modeling. With its basis in the works of Lev Vygotsky (1986), the sociocultural perspective on learning (Anderson, Reder, & Simon, 1996; Säljö, 1998; Wenger, 1998) is currently one of the leading theories in educational research. It is an epistemological theory that emphasizes the importance of social interaction and context for learning.

Within research in science education, the concept of *learning demands* has been used to appraise the differences between a piece of subject knowledge to be taught and the corresponding (and sometimes disturbing) everyday conceptual understanding that the students bring into the classroom (Leach & Scott, 2002). Drawing on this concept, I suggest alternative approaches to teaching and learning data modeling as well as programming. These methodological aspects are additional import features of this paper (i.e., from general educational psychology and from research in science education).

Since the 1960s, when object orientation was introduced as a paradigm for computer programming (Dahl & Nygaard, 1966), a large body of techniques and modeling languages have emerged. Of these, UML (OMG, 2001) is currently becoming a de facto standard in corporations worldwide and has been chosen as the basis for the discussion in this paper. A wide variety of tools are available to support UML development; these tools enable modelers to describe and design different aspects of a computerized information system. This paper concentrates on *structural modeling* and, in particular, on *class diagrams*.

2. TRACTATUS LOGICO-PHILOSOPHICUS

The next portion of this paper introduces several clauses from the *Tractatus Logico-Philosophicus* (Wittgenstein, 1961), hereafter referred to as *Tractatus*. The *Tractatus* is organized as numbered postulates, with seven main postulates, 1 through 7, each with several hierarchically numbered sub- and subsub- postulates. Hence T4.5 indicates the fifth main comment to postulate 4 of the *Tractatus*. To avoid cluttering the presentation, these numbered clauses are listed without further citation.

Wittgenstein's aim in the *Tractatus* is to describe the conditions that would have to be fulfilled by a *logically perfect language*. His concern in this endeavor is not with the psychological issues of using language with the intention of conveying some content with a specific meaning. Nor does he discuss the epistemological issue of the relationship between thought and language on the one hand and to what it refers on the other. What is of interest to Wittgenstein is the relationship that "one fact must have to another in order to be *capable* of being a symbol for that other" (Russell, 1922). This kind of symbolism presupposes the idea of a unique meaning or reference for each symbol or combination of symbols. Thus, a perfect language has a one-to-one correspondence between simple facts and symbols, or between combinations of facts and the related combinations of symbols.

> T4.5: It now seems possible to give the most general propositional form: that is, to give a description of the propositions of *any* sign-language *whatsoever* in such a way that every possible sense can be expressed by a symbol satisfying the description, and every symbol satisfying the description can express a sense, provided that the meanings of names are suitably chosen.

In T1.2, Wittgenstein establishes the world as divided into facts. The objects (facts) can occur in combinations (states of affairs), and these are depicted in thoughts and propositions that have something (form) in common with the real state of affairs.

> T2.04: The totality of existing states of affairs is the world.
> T2.06: The existence and non-existence of states of affairs is reality. [. . .]
> T3: A logical picture of facts is a thought.
> T3.1: In a proposition a thought finds an expression that can be perceived by the senses.

3. OO-MODELING AS A LOGICALLY PERFECT LANGUAGE

This paper assumes the reader understands the basic ideas and features of object-oriented modeling. For readers seeking an introduction to UML, see, for example, Fowler and Kendall (2000). The definitions at the beginning of each subsection are drawn from the UML specification issued by the Object Management Group (OMG, 2001). Below I present definitions of the main features of UML class diagrams and focus on the philosophical aspects of each, particularly their relationship to Wittgenstein's theories.

3.1. Class Diagram

> *A diagram that shows a collection of declarative (static) model elements, such as classes, types, and their contents and relationships. (OMG, 2001)*

A class diagram is intended to give a static description of a portion of the world (the Universe of Discourse) suitable for implementation in an object-oriented programming language. The different objects of this subworld are grouped into classes that are given a set of attributes as well as operations, and the classes are connected to one another as a graph to illustrate the corresponding connections between the respective objects of the real world.

T4.0311: One name stands for one thing, another for another thing, and they are combined with one another. In this way the whole group – like a *tableau vivant* – presents a state of affairs.

The class diagram represents a picture of a part of the world in the same manner as Wittgenstein claims that the world consists of facts that may be coupled together in “states of affairs”, and that propositions in the language are images of these facts and states of affairs. The class diagram is a structural model representing a structure of the corresponding phenomena in the referential scope of the world. For Wittgenstein, this scope was the whole world, or more precisely that of which we can speak and think. Of the unthinkable one cannot speak.

T4.001: The totality of propositions is language.
T4.01: A proposition is a picture of reality.
A proposition is a model of reality as we imagine it.
T5.6: *The limits of my language* mean the limits of my world.

In the same manner that the scope of a class diagram is limited to what can explicitly be captured by the elements of it, the world of Wittgenstein is also limited by what can be described through language.

3.2. Class

A description of a set of objects that share the same attributes, operations, methods, relationships, and semantics. (OMG, 2001)

Object-oriented modeling involves different abstraction mechanisms used to describe the common structure of similar phenomena. The declaration of a class represents an abstraction of substance (Nygaard, 1986).

T2.021: Objects make up the substance of the world.
T3.344: What signifies in a symbol is what is in common to all the symbols that the rules of logical syntax allow us to substitute for it.

The main element of an object-oriented model is the *class*, which represents an abstract collection of objects with some (for the purpose suitable) common set of properties. *Components* of a *system* (i.e., part of the world) are modeled into *objects* that are in turn classified as members of a *class*. Of each *class* we can instantiate *objects* that represent a ‘‘simulation’’ of the *components* in the *system* that we have modeled (Andersen, 1997).

When building a class diagram we are initially looking for phenomena with some common set of properties that together form a generalized description of the phenomenon in question. What qualifies as a class within a given system can be seen in light of the following proposition:

> T2.02331: Either a thing has properties that nothing else has, in which case we can immediately use a description to distinguish it from the others and refer to it; or, on the other hand, there are several things that have the whole set of their properties in common, in which case it is quite impossible to indicate one of them.

The depictions of these things with their common sets of properties constitute members (i.e., objects or instances) of a class. The different values of each of these properties (i.e., attributes) are discussed in the following section.

3.3. Attribute

A feature within a classifier that describes a range of values that instances of the classifier may hold. (OMG, 2001)

Wittgenstein described the issue of value-scope as different spaces within which each object must find its position.

> T2.0131: A spatial object must be situated in infinite space. (A spatial point is an argument-place.) A speck in the visual field, though it need not be red, must have some color: it is, so to speak, surrounded by color-space. Tones must have *some* pitch, objects of the sense of touch *some* degree of hardness, and so on.

In the language of object orientation we may say that each object has a set of attributes, and that each attribute may hold a value within a given range or value-space defined by the variable-type. The values of the attributes may change over time, like a person’s age or the size of a bank account changes in the real world.

T2.0271: Objects are what is unalterable and subsistent; their configuration is what is changing and unstable.

The configuration in Wittgenstein's postulate represents the attributes and their changing values. In object-oriented theory, this configuration is usually referred to as the *state* of an object or a system. What brings the system or the objects from one state to another are *operations* (or *methods* in other methodologies).

3.4. Operation

A service that can be requested from an object to effect behavior. An operation has a signature which may restrict the actual parameters that are possible. (OMG, 2001)

All objects of the world have certain affordances (i.e., abilities to act upon or be used by other elements of the surrounding world). In object orientation, these special properties are represented by operations.

T5.24: An operation manifests itself in a variable; [...]
It gives expression to the difference between the forms.
T5.25: The occurrence of an operation does not characterize the sense of a proposition. [...]

Since it does not hold a value, an operation does not alter the meaning of an object. It merely represents a disposition for behavior or interaction between objects or propositions. One may claim, though, that the operations add to the characteristics of the group of objects instantiated from that particular class.

It should be noted that the concept of *operations* in the *Tractatus* is limited to mathematical truth-operations on simple propositions that produce new non-simple propositions without altering the initial sense or form. This only represents a small subset of the types of operations available in OO-design. Operations as affordances of objects should therefore be considered a special case of the semantic content of *facts* or *states of affairs* in Wittgenstein's terms. When we consider *associations*, the final of the main elements discussed in this paper, the analogy between UML class diagrams and the *Tractatus* is more obvious.

3.5. Association

The semantic relationship between two or more classifiers that specifies connections among their instances. (OMG, 2001)

Two or more classes that are associated indicate that the objects of these classes stand in a certain relation to one another.

T2.0272: The configuration of objects produces states of affairs.
T2.031: In a state of affairs objects stand in a determinate relation to one another.
T4.1: Propositions represent the existence and non-existence of states of affairs.

In a class diagram, each association is usually accompanied by role-names enabling the modeler to "read" the association as a proposition about the relationship between the classes involved. This is very much the same function that Wittgenstein assigns to the elementary proposition.

T3.203: A name means an object. The object is its meaning.
T4.22: An elementary proposition consists of names. It is a nexus, a concatenation, of names.

Finally, Wittgenstein is concerned with the relationship between the propositions as expressions for our thoughts and reality as reference for establishing the truth of a proposition.

T4.06: A proposition can be true or false only in virtue of being a picture of reality.

In data modeling one is not restricted to making true statements (or associations), since these are primarily intended for implementation on the computer. Most of the time, however, an information system is supposed to represent or depict some part of the real world, and will be evaluated according to its correspondence with the real world counterparts.

4. PHILOSOPHICAL INVESTIGATIONS

So far, I have shown that OO-modeling as defined for UML class diagrams corresponds remarkably well to the properties of a logical perfect language as described by Wittgenstein in the *Tractatus*. To shed further light on the implications this has for teaching and learning OO-modeling, I turn to Wittgenstein's later work, which introduces a different perspective on language. Indeed, some scholars refer to Ludwig Wittgenstein as *two* of the most important philosophers of the 20th century. This somewhat satirical comment is based on the fact that in his second major publication, *Philosophical Investigations* (Wittgenstein, 1958), Wittgenstein rejects many of the fundamental claims that he made in the *Tractatus* and presents a more pragmatic view of language and meaning. The first part of *Philosophical Investigations*, which is also the main part of the work, is written as a series of remarks or paragraphs that are numbered chronologically. In the remainder of this paper, paragraphs drawn from *Philosophical Investigations* are given as a paragraph number prefaced by the letter P.

> P23 [...] It is interesting to compare the multiplicity of the tools in language and of the ways they are used, the multiplicity of kinds of word and sentence, with what logicians have said about the structure of language. (Including the author of the *Tractatus Logico-Philosophicus.*)

The main idea of the *Philosophical Investigations* is that "the meaning of a word is its use in language" (P43). Hence, a word or phrase does not automatically carry a given meaning. This meaning is associated with the word only through the use of the word in language, that is in the way it is intended or understood. In turn, the way a word is intended and the way it is understood are not necessarily the same thing.

> P508 I say the sentence: "The weather is fine"; but the words are after all arbitrary signs – so let's put "a b c d" in their place. But now when I read this, I can't connect it straight away with the above sense. – I am not used, I might say, to saying "a" instead of "the", "b" instead of "weather", etc. But I don't mean by that that I am not used to making an immediate association between the word "the" and "a", but that I am not used to using "a" *in the place* of "the" – and therefore in the sense of "the". (I have not mastered this language.)
>
> (I am not used to measuring temperatures on the Fahrenheit scale. Hence such a measure of temperature 's*ays*' nothing to me.)

Wittgenstein introduces the concept of *language games* as dynamic sets of unwritten rules for the use of language features within different settings. Through a series of rhetorical questions and examples, he shows that, given a different set of rules or a different setting, much of what we take for granted about the relationship between language and meaning could be entirely altered, as if we were playing a different language game.

5. UML AND OO-MODELS AS DIFFERENT LANGUAGE GAMES

Any set of propositional signs used by a particular group of people for the purpose of communicating or conveying meaning will count as a language game. Data modeling and programming should therefore be considered as language games, though rather specialized ones with quite rigid sets of rules to control the use of the languages. They belong to a group of language games that we call *artificial languages*. The main focus for Wittgenstein was the group of language games referred to as *natural languages*. This latter group of language games was his primary reason for rejecting the *Tractatus* as a theory for describing languages.

OO-modeling introduces the student to new language games on two different levels: UML as a language for making data models, and each data model as a language for describing a given Universe of Discourse. The artificial language of UML makes up a grammar in terms of boxes, lines, and other tools for designing a data model. This is a highly specialized artificial language game that belongs to the community of System Engineers. Students have to learn the game by first being told its ground rules and then gradually internalizing the needed skills through practicing the language game in interaction with other practitioners (e.g., teacher and fellow students) (Lave & Wenger, 1991).

The idea that data modeling methodologies and programming languages are substantially different from natural languages is usually covered well in the traditional CS classroom. What tends to be overlooked, and what is probably not sufficiently appreciated even by skilled practitioners, is the parallel issue of particular data models as locally functioning languages for the problem at hand. This means that for each new data model, a new language game is generated in which the meaning of the different terms or labels must be defined. At this level, these language games are more closely related to, or influenced by, natural languages. The participants' prior linguistic knowledge has a strong impact on the terms they choose to denote the different phenomena and associations in the data model. While an OO-model is, strictly speaking, an artificial language, it also has features of and uses concepts from natural languages. This can be illustrated in terms of a Venn diagram by

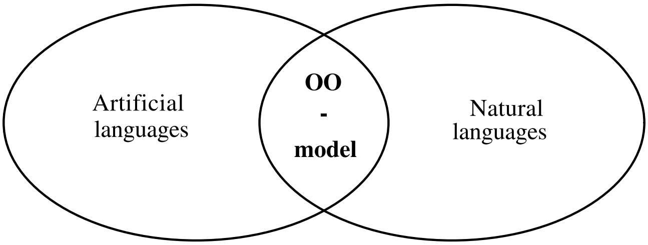

Fig. 1. A particular OO-model represents an artificial language game, but simultaneously incorporates several features of natural language games.

recognizing that the language game of a particular data model is located in the overlapping section between artificial and natural languages (Fig. 1).

6. IMPLICATIONS FOR TEACHING

In this section I discuss the learning demands (Leach & Scott, 2002) inherent from the preceding analysis and focus on the teachers' role in helping students to meet these demands.

6.1. Relationship Between Term and Meaning

A key principle in OO-modeling as well as for other programming languages is that each element of the model must be given a name or label and that the meaning of this labeling term should be uniform and unambiguous. When we use natural language, however, the recipient interprets our utterances from a given context. We can therefore use different words for the same meaning or can have the same word mean different things. Our use of words and the corresponding meanings depend on the language game within which we operate. According to the *Philosophical Investigations* there is no one-to-one correspondence between term and meaning, regardless of context. Instead, this is a question of automatic connections that are made unconsciously by each individual. Thus, mutual understanding is based on the context of the conversation and on each participant's experience in using language within the given language game.

> P601 When I talk about this table, – am I *remembering* that this object is called a “table”?

The use of words within a familiar language game is highly automatic. A skilled participant of a language game does not explicitly make the association between term and meaning. This is true for the person forming a proposition as well as for the person perceiving it, whether or not they have the same understanding of the meaning. In contrast, with OO-modeling there is continuous generation of new language games, each with a new set of correspondences between term and meaning. This results in the constant need for adaptation and learning. As already mentioned, "the meaning of a word is its use in language" (P43). For data models this implies that the terms derive their meanings from the way they are used in relationships to other phenomena that are labeled by other terms. Thus, the context for this game is given by the relationships between the different classes or objects of the data model.

Usually terms are adopted from natural language and used as far as possible to denote objects or groups of objects that resemble the meaning of the term in a natural language game. However, the sense-impression connected with a given term used in a data model does not always correspond to the meaning defined for this term in the language game that the model represents.

> P355 The point here is not that our sense-impressions can lie, but that we understand their language. (And that language like any other is founded on convention.)

Recall the claim that one strength of OO-modeling is the closeness of OO-thinking to natural thinking. This claim bears a remarkable resemblance to the relationship between thought and propositional signs presented in the following postulate of the *Tractatus*:

> T3.2: In a proposition a thought can be expressed in such a way that elements of the propositional sign correspond to the objects of the thought.

If this was the case, we would have the one-to-one correspondence between term and meaning that enables us to describe a part of the world in data modeling. Unfortunately this does not hold.

6.2. The Problem of Ambiguity in Natural Languages

As Wittgenstein himself pointed out (Wittgenstein, 1958), the idealized postulates from the *Tractatus* do not hold for natural languages. Language in itself can never provide a one-to-one correspondence, neither with the referential world nor with the human thought of this world. This contradicts the traditional view of the relationship between language and meaning. In fact, Wittgenstein only realized this discrepancy several years after writing the *Tractatus*, which led him to dispute his original theory in *Philosophical Investigations*. This radical change in his thinking over time suggests how difficult it is to understand that natural language and natural thinking are neither logically perfect nor unambiguous. This underscores the need for careful attention while teaching the concepts of OO-modeling and programming so students come to understand the differing natures of the language games they are learning to use and construct.

6.3. Learning UML

From a sociocultural perspective, knowledge is described as the ability to participate in different practices. The rules of language games must be learned through observation of, and participation with, other individuals engaged in these language games, especially more skilled practitioners (Hennessy, 1993; Wenger, 1998; Wood, Bruner, & Ross, 1976). In our case, this means internalizing or appropriating the rules of the language games of data modeling or computer programming.

> P54 [...] One learns the game by watching how others play. [...]
>
> P340 One cannot guess how a word functions. One has to *look at* its use and learn from that. But the difficulty is to remove the prejudice, which stands in the way of doing this. It is not a *stupid* prejudice.

In the context of OO-modeling and programming, these prejudices refer to students' prior understandings of the general relationship between language and meaning as well as the connotations of single terms.

Learning the language game of a particular OO-model can be described as familiarizing oneself with the meanings of the terms in question. Intuition alone is not sufficient for determining these meanings. The conventions of a given data model may even be counterintuitive. The connection between term and meaning does not function automatically in such a setting. To agree upon meaning, participants must offer one another explicit definitions and explanations. This is a much different scenario than that encountered while learning and using natural language, where mutual understanding is based on intuition and experience.

6.4. Suggestions for Teaching

As presented in this paper, the learning demand of data modeling is epistemological in nature. Students must first realize that language as they know it is ambiguous by nature and that what they mean by a single word or a detached phrase may well be misunderstood. Next, students must become aware of the difference between their everyday use of language and the way they need to use it when applying the same terms to phenomena or relationships in a data model. A pedagogical approach that can help students achieve this is to focus on the pragmatics of natural language. A learning sequence designed to bring forth these ideas could proceed as follows:

1. Each student develops several propositions or pieces of text that are ambiguous and that they predict will be misunderstood by a recipient.
2. Next, each student develops several propositions or pieces of text that they predict are so clear that no one can misinterpret them.
3. When tasks 1 and 2 are complete, organize the students in small groups to critique one another's ambiguous and unambiguous descriptions.
4. To end the exercise, ask each group to share their observations with everyone in the class. This should lead the students to the conclusion that in order to avoid misunderstanding, terms and sentences must be explicitly explained and defined.

Later in the teaching period, when students have some modeling experience, the instructor can ask them to solve a modeling problem without using any terms that they can find in a dictionary. This restriction would apply to the names they can choose for classes as well as the names for attributes and associations. The aim of such an exercise would be to demonstrate to what extent we depend on our preknowledge of the meaning of the terms we use in a data model. It should serve nicely as backdrop for a subsequent discussion about the implications of such dependence on natural language knowledge.

Another useful activity is to have students interchange names between classes of objects and then assign attributes. An example that can serve as background for the discussion is Vygotsky's statement about "whether a cow still has horns if we call dogs for 'cows'" (Vygotsky, 1986: 222–223). This example can help students think about how a term's connotations influence the attributes and associations assigned to a class in a data model.

Further inspiration for the design of teaching sequences can be found in the material presented in *Philosophical Investigations* (Wittgenstein, 1958).

Several passages could be directly introduced as learning material, at least for students at college level. The interested reader could consider paragraphs P47, P48, P73, P74, and P508 of *Philosophical Investigations* for ideas. Such passages challenge students' existing conceptions of language and provide excellent starting points for discussion.

7. WIDENING THE PERSPECTIVE

I have shown how UML class diagrams can fulfill the general criteria for a *logically perfect language* as described in Wittgenstein's *Tractatus*. The analogy also seems relevant in working with OO-programming languages since these are designed to implement OO-models. This relationship probably also holds for other artificial context-free languages within Computer Science as well as in other domains. Although later criticized both by himself and others, the *Tractatus*, published in 1921, presented not only a philosophical treatise, but also contributed to what was at that time the unknown field of Computer Science. The ideas in *Tractatus* provided a generalized set of criteria for the mathematical analysis of stringent, context-free languages. Today, the use of logic and mathematical reasoning to decompose and prove computer programs and programming language features is part of formal methods, an important field of research within Computer Science.

7.1. Relations to Previous Work and Thoughts for Future Research

In a recent study of procedural versus OO strategies, Corritore and Wiedenbeck (2001) show that while both procedural and OO-programmers use bottom-up strategies when coding, only OO-modelers use a top-down problem-solving approach. This, coupled with the common assumption that top-down problem-solving strategies are more natural (Détienne, 2002), could explain why OO-thinking is often viewed as being more natural.

Further support for this view comes from studies that focus on the benefits of having programming languages closely match natural language (Pane et al., 2001; Petre, 1990; Shneiderman, 1980). At the same time, other researchers have shown that closeness to natural language can create problems, especially for novices (Bonar & Soloway, 1985; Holmboe, 2005). When using a context-free language like UML, or indeed any programming language, we can no longer depend on others to interpret what we mean based on experience and context. The ultimate recipient is a computer, which has no innate interpretive skills. Indeed, novices in particular may be so accustomed to the pragmatism of natural language communication that they forget this limitation and expect the computer to "think" and "interpret" what they "say" in the same way that people do. This is what Pea has called the superbug of novice programming (Pea, 1986). These results point out the linguistic confusion that can occur when users of natural languages attempt to learn artificial context-free languages such as programming languages or data modeling methodologies. A plausible explanation is that the participants (i.e., the students or novices) do not realize that they must play different language games from the ones they use in everyday discourse. The participants are probably not even aware that different language games exist. They might furthermore be misled by the fact that the artificial languages they are learning use English terms that they already know from natural language. All of these factors contribute to the difficulty of finding the balance between the stringent demands of a *logically perfect language* and the use of natural language.

The points in this paper suggest several paths for further investigation. One obvious approach would be to study the effect of applying some of the strategies introduced earlier while teaching OO concepts. A careful qualitative analysis of student interactions would provide insights into how and why they use language and terminology when modeling or programming. The main focus for such an analysis could be the linguistic consciousness revealed by the discursive interactions of students engaging in natural problem-solving activities. It would also be informative to see how professional software and system developers handle the issues discussed in this paper. It seems likely that professionals have already overcome the linguistic obstacles described earlier, making them better equipped to distinguish between a variable name and the everyday meaning of the term chosen to denote the variable. Discovering when and how this cognitive development occurs would provide useful insights into the learning processes involved and might suggest teaching approaches that can facilitate such development at an earlier stage.

8. CONCLUSIONS

This paper has demonstrated how ideas from the field of linguistic philosophy can provide valuable insights for the computer science education research community. I have also shown that these ideas can provide valuable insights for how to teach object-oriented thinking and modeling as well as programming in general.

A key point of this paper is that in his later work (Wittgenstein, 1958), Wittgenstein became one of the main critics of the theories he presented in his earlier writings, the *Tractatus Logico-Philosophicus* (Wittgenstein, 1961). In the years between 1921 and 1958, Wittgenstein came to realize that language and meaning are constructed in social practices rather than from mathematical logical reasoning. He had come to see that there are an uncountable number of coexisting language games in which words and propositions may carry different meanings. This view of language as different games reveals dissimilarities between natural languages and artificial context-free languages such as UML and programming languages. The sum of these differences leads to the conclusion that OO-thinking and modeling are quite different from natural thinking and language use. At the same time, while each particular data model represents an artificial language game, it is informed by and uses concepts from natural language. The failure to acknowledge this duality might be one reason why evidence has shown that OO-modeling appears to be more difficult to learn than might be expected.

The ideas presented earlier have led me to believe that a focus on the learning demands identified in this paper would alleviate some difficulties described in previous research on teaching and learning computer science concepts. This would aid students in acquiring sufficient linguistic consciousness to realize that different sets of rules apply to the new language games in which they are about to become players.

ACKNOWLEDGEMENTS

I wish to thank Jens Kaasbøll along with members of the PPIG-community for their encouraging comments on early versions of these thoughts. For the final preparations, valuable contributions were offered by Andreas Quale as well as Dr. Phil Scott. Most of all I wish to thank the anonymous referees and the guest editors of this issue for impressively thorough and helpful reviewing and advice for final improvements of the paper.

REFERENCES

Andersen, E.P. (1997). *Conceptual modeling of objects: A role modeling approach.* Unpublished Ph.D., University of Oslo, Oslo.

Anderson, J.R., Reder, L.M., & Simon, H.A. (1996). Situated learning and education. *Educational Researcher*, *25*(4), 5–11.

Batra, D. (1993). A Framework for studying human error behavior in conceptual database modeling. *Information and Management*, *25*(3), 121–131.

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

Bonar, J., & Soloway, E. (1985). Preprogramming knowledge: A major source of misconceptions in novice programmers. *Human–Computer Interaction*, *1*(2), 133–161.

Coad, P., & Yourdon, E. (1991). *Object-oriented analysis* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Corritore, C.L., & Wiedenbeck, S. (2001). An exploratory study of program comprehension strategies of procedural and object-oriented programmers. *International Journal of Human–Computer Studies*, *54*(1), 1–23.

Dahl, O.-J., & Nygaard, K. (1966). SIMULA – an ALGOL-based simulation language. *Communications of the ACM*, *9*(9), 671–678.

Détienne, F. (1990). Expert programming knowledge: A schema-based approach. In J.-M. Hoc, T.R.G. Green, R. Samurcay, & D.J. Gilmore (Eds.), *Psychology of programming* (pp. 205–222). London: Academic Press.

Détienne, F. (2002). *Software design – cognitive aspects* (F. Bott, Trans.). London: Springer.

du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, *2*(1), 57–73.

Ehrlich, K., & Soloway, E. (1984). An empirical investigation of the tacit plan knowledge in programming. In J.C. Thomas & M.L. Schneider (Eds.), *Human factors in computer systems* (pp. 113–133). Norwood, NJ: Ablex Publishing Corporation.

Fowler, M., & Kendall, S. (2000). *UML distilled: A brief guide to the standard object oriented modelling language* (2nd ed.). Reading, MA: Addison-Wesley.

Hennessy, S. (1993). Situated cognition and cognitive apprenticeship: Implications for classroom learning. *Studies in Science Education*, *22*, 1–41.

Hoc, J.-M., Green, T.R.G., Samurcay, R., & Gilmore, D.J. (Eds.). (1990). *Psychology of programming*. London: Academic Press.

Holmboe, C. (2005). Conceptualisation and labelling as linguistic challenges for students of data modelling. *Computer Science Education*, *15*(2).

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge, UK: Cambridge University Press.

Leach, J., & Scott, P. (2002). Designing and evaluating science teaching sequences: An approach drawing upon the concept of learning demand and a social constructivist perspective on learning. *Studies in Science Education*, *38*, 115–142.

Nygaard, K. (1986). Basic conceps in object oriented programming. *SIGPLAN-Notices*, *21*(10), 128–132.

OMG. (2001). *Unified modelling language specification v1.4*. Needham, MA: OMG Object Management Group.

Pane, J.F., Chotirat, R., & Myers, B.A. (2001). Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human–Computer Studies*, *54*(2), 237–264.

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

Petre, M. (1990). Expert programmers and programming languages. In J.-M. Hoc, T.R.G. Green, R. Samurcay, & D.J. Gilmore (Eds.), *Psychology of programming* (pp. 103–116). London: Academic Press.

Russell, B. (1922). Introduction. In L. Wittgenstein (Ed.), *Tractatus Logico-Philosophicus* (1st English ed.). London: Routledge & Kegan Paul.

Shneiderman, B. (1980). *Software psychology: Human factors in computer and information systems*. Cambridge, MA: Winthrop.

Shoval, P., & Shiran, S. (1997). Entity-relationship and object-oriented data modeling – An experimental comparison of design quality. *Data and Knowledge Engineering*, *21*(3), 297–315.

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

Soloway, E., & Sleeman, D. (1986). Special issue: Novice programming. *Journal of Educational Computing Research*, *2*(1).

Soloway, E., & Spohrer, J. (Eds.). (1989). *Studying the novice programmer.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Srinivasan, A., & Teeni, D. (1995). Modeling as constrained problem-solving – an empirical-study of the data modeling process. *Management Science*, *41*(3), 419–434.

Säljö, R. (1998). Learning as the use of tools: A sociocultural perspective on the human-technology link. In K. Littleton & P. Light (Eds.), *Learning with computers. Analyzing productive interaction* (pp. 144–161). London: Routledge.

Taylor, J. (1990). Analyzing novices analyzing Prolog – What stories do novices tell themselves about Prolog. *Instructional Science*, *19*(4–5), 283–309.

Tegarden, D.P., & Sheetz, S.D. (2001). Cognitive activities in OO development. *International Journal of Human-Computer Studies*, *54*(6), 779–798.

Vygotsky, L. (1986). *Thought and language* (A. Kozulin, Trans.). Cambridge, MA: MIT Press.

Wenger, E. (1998). *Communities of practice, learning, meaning and identity.* Cambridge, UK: Cambridge University Press.

Wittgenstein, L. (1958). *Philosophical investigations* (G.E.M. Anscombe, Trans., 2nd ed.). Oxford: Basil Blackwell.

Wittgenstein, L. (1961). *Tractatus Logico-Philosophicus* (D.F. Pears & B.F. McGuinness, Trans.). London: Routledge & Kegan Paul.

Wood, D.J., Bruner, J.S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Psychology and Psychiatry*, *17*, 89–100.

Jl. of Computers in Mathematics and Science Teaching (2005) **24**(1), 89-115

Characterising Individual and Social Concept Development in Collaborative Computer Science Classrooms

CHRISTIAN HOLMBOE
University of Oslo, Norway
christho@ifi.uio.no

PHIL H. SCOTT
CSSME, Leeds University, UK
p.h.scott@education.leeds.ac.uk

Within-group similarities and between-group differences are used to illustrate the socio-cultural nature of the concept-building process in highly collaborative computer science classrooms. Simultaneously, a social constructivist perspective is used to describe the individual aspects of this development. The study uses written explanations from high school students as well as novice university students to illustrate the cognitive trajectory from initial hunches to a holistic knowledge of the concepts of keys in database modelling. The main findings of the study, however, are of a general epistemological nature, as they enlighten and exemplify the social processes of these classrooms as seen from a perspective of situated cognition. Based on these findings, the paper finally addresses implications for teachers. In particular, it is emphasised that teachers need to pay careful attention to their own use of language in discursive interaction with students.

Describing knowledge as individually constructed mental representations of the experiential world (Glasersfeld, 1989), the constructivist theory of learning still has a lot to offer in terms of understanding the learning processes going on in classrooms. Over the past 15 years, however, increased emphasis has been given to the importance of the attendant social processes for this learning. The focus of epistemological research has thus tended to shift from a constructivist to a situated view of learning (Sfard, 1998). According to the perspective of situated cognition, knowledge is anchored (Vanderbilt, 1990) in particular cultural practices. Learning is, in turn, described as a process of entering a particular community of practice (Wenger, 1998) and can, therefore, not be seen as independent of social context. The present work is influenced by the theories of situated cognition (Lave & Wenger, 1991) and the concept of cognitive apprenticeship (Brown, Collins, & Duguid, 1989; Hennessy, 1993). Based in situated cognition theory, the latter describes how learning activities should be organised to resemble real life situations in order to enhance the potential transfer value of the learning outcome.

The aim of this paper is to offer an example of scientific concept development (Vygotsky, 1986) in computer science. Concept development, as it seems to evolve in naturally occurring collaborative computer-based classrooms, can be described as both an individual and a social process. Based on this duality, the paper will explore to what extent aspects of social constructivism (Driver, Asoko, Leach, Mortimer, & Scott, 1994), as well as the perspective of situated cognition, can be utilised in analysing examples of such concept development. Simultaneously both theories should benefit from the empirical illustrations provided.

Acknowledging that there have been fierce discussions between constructivists and anti-constructivists (Matthews, 1998), as well as between cognitive and situated perspectives on learning (J. R. Anderson, Reder, & Simon, 1996, 1997; Greeno, 1997), the analysis presented in this paper will serve to illustrate that these differing epistemological paradigms may have complementary explanatory qualities.

Situated cognition research has been criticised for lack of empirical evidence in terms of knowledge outcomes for students from collaborative or computer-based learning activities (Anderson et al., 1997). In fact, much of the empirical work informed by situated cognition perspectives has been based either on studies in artificial intelligence (Clancey, 1997) or on in-vitro studies of artificially constructed group-based teaching sequences. Reviewing literature on situated cognition, Hennessy (1993) concludes by addressing the need to "seek to ground theories of action in empirical evidence, generalising from records of particular, naturally occurring activities" (p.34).

In the areas of computer programming and system development, increasing emphasis is being placed on the value of collaborative work for the production outcome. Studies document significant improvement in productivity and accuracy from collaborative methods such as ‘pair programming’ or ‘extreme programming (XP)’ (Anderson, Beattie, & Beck, 1998; Nosek, 1998; Williams & Kessler, 2000). However, little evidence beyond subjective satisfaction (and in some cases general test performance) is provided in terms of learning outcomes from such activities.

High school (HS) students, and novice university (UNI) students (included for purposes of comparison), were asked to explain a few database-related concepts in their own words. These explanations were then used to analyse the concept building process in collaborative classrooms as an interrelationship between individual and social developments of scientific terminology. Computer science in Norwegian high schools is normally taught by having the students collaborate in project-based workgroups of 2-4 students, where the teacher mainly acts as a supervisor. This mode of work, which often covers 80-90% of the time spent in the classrooms, fosters extensive interaction between students who are solving problems in front of a shared computer or solving problems collaboratively, in parallel, at separate computers. The present study thus conforms to Hennessy’s request for research based on ‘records of naturally occurring activity.’

DATABASE TERMINOLOGY

A large number of introductory books have been published on relational databases and data modelling. Correspondingly, there are numerous sets of terms being used, and different ways of defining their semantic content. The terms used in this paper are translations of the definitions provided in the textbooks used by the HS-students (Kolderup & Bostrøm, 1998).

A data model consists of *relation types* between different *entity types*. Each entity type has a set of *attributes*, of which one (or a subset) is chosen as identifier (*primary key)*. The primary key uniquely determines the value of the remaining attributes in a given record. Two related entity types are ‘linked’ by introducing the primary key from one as an extra attribute (*foreign key*) in the other. An entity type with a set of records forms a *table*, which is displayed as a *scheme* in MS Access.

For simplicity, the students use *entity* and *relation* for *entity type* and *relation type*, respectively; therefore, the same terms are used in this paper.

Example

For readers who are not familiar with data modelling, the following section gives a brief example illustrating the meaning of each of these terms.

The information in a database is stored in tables. When making a database of cars, their owners, and insurance companies, one would normally need three different tables, one for each of these entities. Each column in a table represents an attribute. One or more attributes have to be unique in order to be able to identify a particular row in the table. Such an attribute, or combination of attributes, is called a *candidate key*. One of the candidate keys is chosen as a *primary key* and used as *identifier* for the table.

CarOwner:

SocialSecurityNumber	Name	DriversLicence	InsuranceCompany
1234567	Pete McGordon	09876	123 Motor Insurance
7654321	Pete McGordon	45678	Admiral
2345678	Emma Thompson	87654	Admiral
…			

DriversLicence and SocialSecurityNumber (SSN) are both candidate keys for CarOwner. If SSN is chosen as primary key, it must be included as a *foreign key* in the Car-table in order to link a car to its owner.

Car:

LicencePlate	Make	Year	OwnerSSN
G8 AAT	Volvo	1997	1234567
G13 PET	Nissan	1995	1234567
P4 ZED	Audi	2000	2345678
…			

One can now see from the Car-table that car P4 ZED belongs to the person with SSN 2345678, and then, by referring to the CarOwner-table, confirm that this owner is Emma Thompson, who is insured through Admiral. Assuming that insurance companies have unique names, this name can be used as primary key in a table of insurance companies, which, in turn, makes it a foreign key in the CarOwner-table.

MATERIAL AND METHODS

The Course(s) and the Students

In the two final years of Norwegian High School (HS2 and HS3), students may choose to follow a course in system development for five lessons per week in each of the two years. The course curriculum covers most areas of system development and analysis. During the first year (HS2), the curriculum for system development and databases is limited to making simple data models with up to five entities, using the ER (Entity Relationship) modelling notation. The implementation is done using MS Access. The second year (HS3) is entirely devoted to system development; the data models are more complex, and emphasis is put on project planning and management, as well as documentation (e.g., various analyses and reports). The tools used are still ER and MS Access as well as MS Project and other MS Office applications.

At the university level, a course is given with similar subject matter content. This course is designed to occupy 50% of the total study time during one term, and there are no prerequisites. Some students have completed the HS2 and HS3 courses prior to the University course, although the majority have not. These students will be referred to as UNIexp and UNInoexp respectively. The university students are introduced to relational data modelling using NIAM as well as object-oriented modelling with UML. The use of computer-based modelling and implementation tools is less prominent than in the HS-courses.

The Collaborative Classroom

In the HS2 and HS3 classrooms, the students mainly work in pairs or small groups, while the teacher functions as a supervisor. The classes are normally quite small (10-15 students) and there is substantial between-group interaction in addition to the obvious within-group discourse (e.g., "How is your group doing?" "What did you guys do about this or that…" etc…). The university course is based on weekly plenary lectures in an auditorium. In addition, the students attend two weekly tutoring groups led by a senior student. The tutoring groups are held in a computer lab, and resemble the HS2 and HS3 classrooms.

Thus, the courses represent a unique type of learning environment. This is not because project-based collaborative problem-solving is novel, but because it covers such a large proportion of the total teaching time. It represents what is normal rather than being an odd exception from traditional teaching. The material, therefore, provides a valuable opportunity to study the impact from naturally occurring collaborative classroom activities.

DATA COLLECTION

The study was conducted using straightforward "What is…"-questions[1]:

Q1. What is a *candidate key*?
Q2. What is an *entitisation*?
Q3. What is a *primary key*?
Q4. What is a *query*?
Q5. What is a *foreign key*?

From a design perspective, the intention of this approach was to collect individual accounts of students' conceptual understandings for later analysis and comparison. The social dimension of the analysis was not considered when designing the study. The techniques chosen for the data collection obviously have clear limitations, one being the apparent lack of richness considering the briefness of the explanations. It was, however, an aim to get accounts that reflected the students' immediate connotations to the different concepts. Choosing an open-ended question format also facilitated qualitative analysis and/or diagnostic coding of the answers. The format of the questions may be criticised for triggering formal, definition-like answers. The didactical contract (Brousseau, 1997) implicit in a test-situation would imply that the students would feel expected to provide as precise formal definitions as possible to the kinds of questions presented here. As the analysis will show, however, this did not seem to discourage more informal practice-related explanations from the students.

Short Interviews versus Written Questionnaire

International comparative studies in education have traditionally measured knowledge by students' performances on standardised tests, mainly consisting of multiple-choice items. More recent studies have demonstrated the advantages of open-ended items, providing the possibility for diagnostic coding and analysis (Lie, Taylor, & Harmon, 1996). A problem with both of these types of written tests is the students' failure to understand the problem when it is presented in writing. Students may also have problems articulating their knowledge in writing, which may give biased results to the latter type. Schoultz and associates (2001) showed that the average performance of 15-year-olds on a given problem improved from 19% to 90% by changing the format from a written MC-item to an oral interview setting. In contrast, the problem of biases and lack of objectivity in conducting interviews has also been documented (Lang & Lang, 1991; Mercer, 1995).

Ideally, the questions should have been administered as structured mini-interviews, since one would expect this to optimise the immediateness of the unprepared answers and allow for detailed analysis of the transcribed answers. Written questionnaires, on the other hand, allow for greater material that opens up the opportunity for quantitative analysis. To explore the relationship between oral interviews and a written "test," five HS3-students (from two of the 10 classes included in the written study) were interviewed individually and given two of the questions orally prior to taking the written test. The oral answers were recorded, transcribed, and compared with the same students' written answers as a validation. It may be seen as problematic that the interviewed students were familiar with two of the questions beforehand. The oral interviews were extremely brief and without any form of follow-up or confirmation from the interviewer. It is, therefore, not likely that this familiarity should have substantial effect on the performance of these students in the written study, which was conducted a couple of days later with no pre-warning. Therefore, all students were included in the analysis on equal terms.

The written questionnaire with all five open-ended items (including the two from the interviews) was administered to all students. In keeping with the aim to provoke immediate "off the top of your head" types of answers, the students were only allowed five minutes to complete the written questionnaire. Most students finished their questionnaire well within the time limit. They were informed that the results would neither be evaluated by their teacher, nor influence their grades.

The final choice of what data material to use in the analysis followed from the comparison of the oral and written responses of the five high school students who were interviewed. The analysis in the following paragraphs should be read as a methodological discussion addressing the accuracy and validity of the different methods of data collection.

The general impression is that the level of accuracy of the answers is approximately the same in the oral and written formats. The written responses are generally shorter and more precise than in the interviews. The latter, however, can be more elaborate. To put in writing an articulated answer containing relevant arguments is difficult (Schoultz, Säljö, & Wyndhamn, 2001). High school students in general have little experience in formulating precise explanations. This makes their accounts appear arbitrary with regard to which aspects of a phenomenon are included or excluded in their explanations. In the oral answers, this problem is sometimes 'solved' by making multiple elaborations or referring to examples as part of the explanation. As an example, the responses from one of the students are examined more closely in Table 1.

Table 1

Transcription and translation[2] of one student's oral and written answers respectively to the same two questions

ORAL INTERVIEW	WRITTEN TEST
I: What is a foreign key? S: Foreign key, it is e:h what y'use i:: (0.2) n'a data model (.) to con:nect (.) e:ntities, (0.2) or (.) to eh to: (0.1) >get the connection between the entities of the relation< (.) decide it.	Q5: What is a foreign key? S: A foreign key is necessary to get the relations between the entities correct, such that there is a connection between the entities that are connected.
I: And then one more question, What is a query? S: A query is a scheme >or not a scheme< a:: e:h thing you make in Access to:: (0.2) make schemes out of it. (0.1) or (0.4) >my God ho'do'I explain that?< (0.3) You: e: and choose e: cert- s:ome certain inf'mation from (0.3) ↑schemes (0.2) a:nd tables (.) for then to: (0.1) make a: (.) new scheme out of what you: ask for in the query= I: Uhm S: =You: (.) define what you want to have in the scheme in the query.	Q4: What is a query? S: In a query one can define the different instances one wants to have in a scheme. One takes the instances from tables or schemes and set the conditions you want there to be.

The two explanations of *foreign key* are quite similar, the oral account being a grammatically less structured statement. This indicates that the format of the test is of little significance for the outcome in this case. However, through a careful transcription of the oral answer, it is evident that it carries quite a bit of additional information. One example is the hesitation initiated by the ‘or’ in the fifth line of the response, indicating an uncertainty and a wish to clarify the explanation further. This is followed-up by repeating the initial explanation in the following two lines. Hence, what in the written response seems to be a firm mental representation of a foreign key, may not be so certain after all.

In the two accounts of what a *query* is, a similar pattern is found. The two responses are more or less based on the same components and the same means of explanation, except for the reference to setting conditions, which is only made in the written answer. Notice that the student initially makes a wrong statement and then immediately corrects the response. Such spontaneous mechanisms would hardly appear in a written reply. Then again, there is a line starting with ‘or’ followed by a more explicit statement of uncertainty. The second explanation (lines 7-12) is an elaboration and clarification of the initial attempt at answering. Then, finally, in the last two lines a third version is provided. This one is shorter and the key words are all emphasised. It may seem that this last part is initiated by the interviewer’s ‘Uhm,’ but notice the immediateness of the transfer from the word ‘query’ to ‘you.’ It is therefore more likely that the clarification is motivated by the student’s own need to sum up rather than an invitation from the teacher.

The Choice of Instrument

The comparison between the oral and written responses from one of the students indicates that there is little difference in performance level on these kinds of direct questions. The limited additional information available in the oral answers can be valuable, whereas the written version enables collection of a larger body of empirical data. Considering the relatively minor differences in content between the answers in the two formats, the written test was considered to provide sufficiently detailed and accurate accounts of the students’ conceptual understanding. Since the written test also allows for greater material and, consequently, for statistical as well as qualitative analysis, it was chosen as the source of data for further investigation.

Acknowledging that oral and written language may represent different forms of knowledge (Halliday, 1993), the written responses were treated in the analysis as if they were oral. This choice could be justified since focus was on the semantic content of the answers rather than traditional criteria for written scientific accounts, such as clarity or linguistic precision level.

The Sampling

For the main study, the written questionnaire was administered to all students in several classes/groups (see Table 2). In HS2 and HS3 the test was first given in February, and then again in May to a smaller sample. Eight HS2 and eight HS3 students completed the test twice and their answers have been analysed qualitatively to identify patterns of development. Due to administrational problems, the university students only received the test once, in March (i.e., half way through their term). They were asked to indicate whether they had previously completed one or both of the HS courses (UNIexp) or not (UNInoexp).

Table 2
Number of groups and students participating in the written test.
(Number in parenthesis indicates that 16 of the students taking the test in May also participated in February)

	HS2feb	HS2may	HS3feb	HS3may	UNInoexp	UNIexp
Number of groups	5	2	4	2	4	
Number of students	57	23 (8)	33	10 (8)	40	10

THE PROCESSING OF DATA

For the analysis of the written responses, different characteristics and/or terms that might appear in an answer were developed as dichotomous coding variables. This means that each variable has two possible values, ‘1’ for the occurrence of this feature in the explanation, and ‘0’ for no such occurrence. Each question had a separate set of variables that expanded when new features emerged during the coding of the questionnaires.

Different terms, which may be used to denote similar semantic meanings, were coded into separate, term-specific variables (i.e., entity/table or field/attribute/variable etc.). Additional variables were included to capture particular characteristics of an explanation. These include references to the software tool, inclusion of an example, the occurrence of an error, imprecise use of language, responding with nonsense, or leaving the answer blank. On analysing the material, several of the term-specific variables were joined into semantic or conceptual variables. In this process, synonyms and/or semantically related variables were merged, signifying that a particular concept or aspect was referred to regardless of which terms were used to do so.

Such merged variables are referred to as explanation *features*.

In addition, each answer was assigned score points according to the general level of accuracy. The score (0, 1, or 2 points for each question) was based on an overall evaluation. Points were awarded in such a way that two quite different answers could both be given two points as long as the features mentioned provided a reasonable account of the concept. Any answer that included some level of correctness was given at least one point. Thus, only completely wrong or missing answers would receive no points. The following example of an answer to Q1 (foreign key), given by a UNInoexp-student, was awarded two points, despite the slightly misguided claim that candidate key is a synonym for primary key. The underlined words indicate terms/variables that were registered for this particular answer.

> *Student 1419: Q1 "candidate key"*
>
> "Candidate key is also called primary key. It is unique for every line in the table and is not repeated. It is kind of an identity for each line or attribute in the line in a table."

For each answer, the number of different registered variables were summarised as well as the number of correct features used. The terms 'primary key' and 'identity' belong to the same feature, as does 'unique' and 'not repeated,' leaving this example with seven terms and five features registered. This gives a measure of the complexity of the answer in terms of the number of different scientific terms or features included.

STATISTICAL ANALYSES

All statistical analyses were performed with SPSS 11.0. The analyses for the first set of results are all simple frequency and mean calculations using independent samples T-Test to measure possible significant differences between groups.

To ensure uniformity of the compared groups, only HS students from the February test were included in the discriminant analysis (see detailed explanation below). This means that data was available from five HS2 groups and four HS3 groups. A stepwise forward method was performed based on smallest Wilks' lambda values. Single variables were added to or deleted from the model in 23 iterations using $P(F) < 0.5$ to include and $P(F) > 0.10$ to exclude a variable from the model. Significance of the outcome was assured by omitting functions with Eigenvalues < 1.0. Equal group sizes were assumed and the classification was based on pooled within-group variance. The resulting classification was finally validated both by cross validation (i.e., classifying each case by the functions derived from all cases other than that case) and by running the same analysis after randomly assigning all cases to one of nine imagined groups.

INDIVIDUAL AND COLLABORATIVE CONCEPT BUILDING

A central part of the analysis of the data was to ascertain the dynamics of the concept development processes both on an individual and on a social level. This was achieved by analysing the written responses to questions Q1 (candidate key), Q3 (primary key), and Q5 (foreign key).

The findings are presented in three sections. First the individual concept development is described as construction of ‘conceptual networks’ or ‘thematic patterns’ (Lemke, 1990). In the second section, discriminant analysis is used to demonstrate ’within-group similarities’ and ‘between-group differences’ supporting the theories of the development of ’common knowledge’ (Edwards & Mercer, 1987) and emphasising the situatedness of knowledge in social practices. Finally, the results are used to suggest that concept development constitutes a trajectory from what will be referred to as ‘initial hunches’ towards ‘holistic concept knowledge’ (Holmboe, 1999).

Only the nine HS classes from February were included in the analysis of within-group similarities and between-group differences. This was done in order to have as uniform and comparable results as possible (i.e., only full groups of students answering the questions for the first time and only students with similar classroom experiences).

Conceptual Network

Across all the different groups of students, there was a significant difference in their ability to accurately explain the three different types of keys. The primary key was the concept about which the students had the best knowledge, followed by the foreign key, and finally the candidate key. The mean scores were (Q1=0.48±0.12, Q3=1.19±0.12, Q5=1.05±0.12). Accordingly, among the 172 students, most answered Q3 (N_{Q3}=140), followed by Q5 (N_{Q5}=129), while quite few attempted to answer Q1 (N_{Q1}=54). However, when only including the students who actually attempted to answer each question, the mean scores were largely the same across the three questions (Q1=1.52±0.19, Q3=1.46±0.10, Q5=1.40±0.10).

Figure 1 shows the mean scores as well as the average number of variables and average number of features registered for each group of students on each question. The first graph displays the average total for all three questions, whereas the three other graphs display the results for one question each. In the three latter ones, only students who actually answered the question are included. The number of students for each group/bar is indicated inside the variables-bar.

The students with prior experience with the subject matter (i.e., HS3 and UNIexp) scored significantly higher (P_{171}<0,001) than the students without such experience (HS2 and UNInoexp) when tested by means of independent samples T-test (SPSS 11.0). This pattern holds not only for the overall score, but also for each single question—even when only considering the students who actually answered a particular question (Q1: P_{53}>0,093, Q3: P_{138}>0,060, Q5: P_{134}>0,036). This observation indicates a higher level of sophistication in answering this type of question as experience increases.

A very similar pattern was found when measuring the number of different registered variables or features used in the explanations given. Considering that this number only covered correct aspects of an answer, it should be expected to correlate highly with the score. Still, the score was based on an overall evaluation of the answer. Therefore, it would be possible to get a high score with a low number of scientific terms, whereas the opposite would be rarer. The two measures were used for different purposes in the analysis, but could also be used to validate each other in the sense that they did in fact correlate.

HS3 students and UNIexp students used a significantly larger number of concepts in their answers than the HS2 and UNInoexp students did. The pattern for the score and for the number of scientific terms was repeated for each single question, as well as for the three questions together. These patterns illustrate that the explanations given by the more experienced students were more extensive as well as more accurate.

This study shows that the vocabulary available to and actively used by the students accumulated over time. It is worth noticing that the students tended to engage a variety of explanative-approaches, even within the same answer. The complexity of the concepts probably made the students feel a need to elaborate on their explanations, in order to include different aspects with which they were familiar and found relevant to the question at hand.

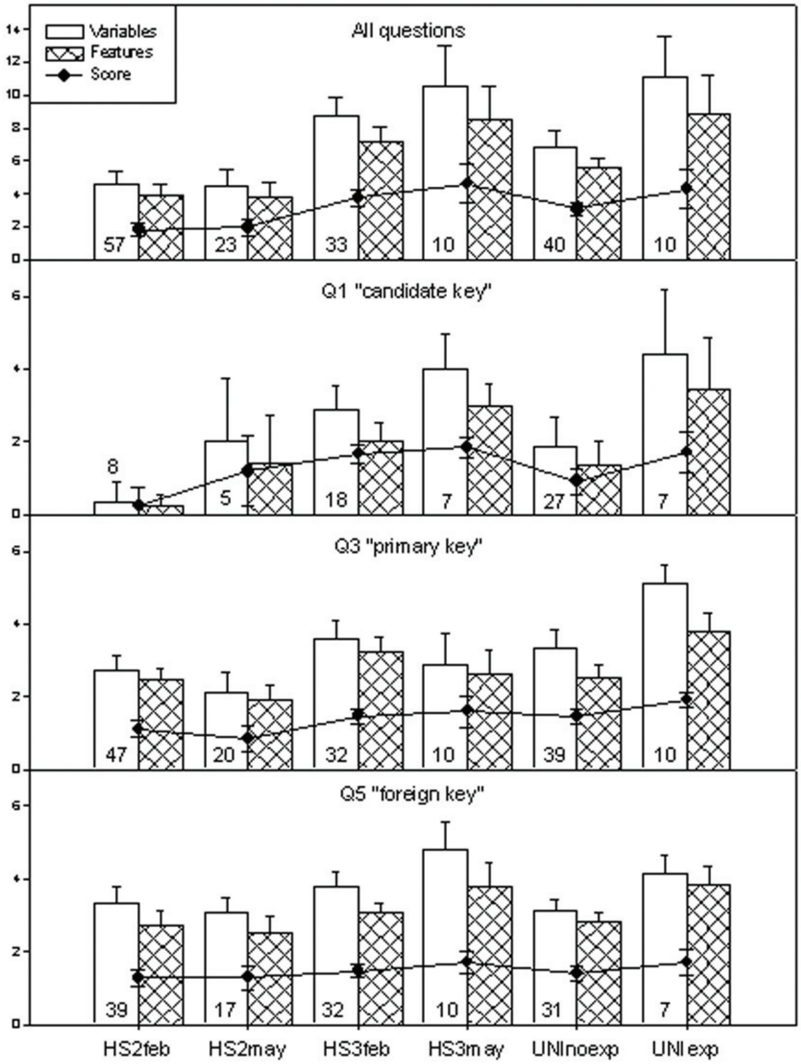

Figure 1. Mean score (line) and average number of correct variables registered and explanation-features used respectively (columns). Question-specific values are calculated only from the students who answered the question (number of students indicated inside the bars).

These findings suggest that the students were in a process of developing a network of scientific concepts. A number of related and interlinked terms, features, and constructs were developing as part of their subject knowledge. Lemke (1990) describes *thematic patterns* as "a way of picturing the network of relationships among the meanings of key terms in the language of a particular subject" (p.98). The results from this part of the current study indicate (in conformity with Lemke's theories) that this network gradually grows both in size and in complexity. At a given point during this development, the student may be aware of a number of semantic relationships without being able to see the whole picture. At this level, an account for the meaning of a concept would possibly appear fragmented as a number of single detached statements. The following excerpt is an example of such an explanation containing at least three separate statements related to a primary key.

> *Student 3107 (Feb) Q3 "primary key"*
>
> "It is a field that counts as identifier for an entity. i.e., non-ambiguous -> not possible to register more equal primary keys. Most often, it is just one primary key, but it can be up to three in an entity. The primary key becomes a foreign key in a new entity."

First, the primary key is explained as an identifier. Then the uniqueness is brought in, both in terms of non-ambiguity and further explained by no-equal keys. Furthermore, the number of attributes involved is discussed before finally focusing on the link to a foreign key.

Economy of Expression

The gradually more complex and fragmented explanations of scientific concepts reflect the increase in complexity of the students' conceptual networks (thematic patterns). At a later point, when the concept is understood more fully, the students may feel confident enough to isolate one or more central features and provide an account, which is more typical for a person with holistic knowledge. Holistic knowledge is apparent when the fragmented descriptions come together to form a whole. This can be illustrated by looking at the answer given in May by the same student as in the example above.

Student 3107 (May) Q3 "primary key"

"The attribute that decides the other attributes."

This development is also evident in Figure 1. Notice that the number of variables and number of features did not increase, but rather decreased from HS3feb to HS3may. This is contrary to the development seen on the other questions, and it is contrary to the development in score.

Use of examples

Some students included examples as part of their explanations. The percentage of students who did so is displayed in Figure 2. Recalling that the students found Q1 (candidate key) to be the hardest question, followed by Q5 and Q3, we see that this interrelationship is also reflected in the number of students feeling confident enough to include an example as part of their answer.

The HS3may students did not feel the need to include an example in order to make themselves understood on the primary key question, whereas 20-30% of the other students answering this question did include an example (Figure 2). This, together with the lack of increase in number of scientific terms from HS3feb to HS3may (see Figure 1), indicates that the students seem to be closer to a holistic knowledge of what a primary key is at this stage. The other two key-types follow the familiar pattern of an increasing number of scientific terms, suggesting that the students were still at an earlier stage of their concept-building trajectory for these two concepts.

In fact, only two students included an example in their explanation of candidate key. This may indicate that a certain level of confidence is required for the students to be able to (or choose to) illustrate the explanation using an example. Referring to the trajectory model, it may seem that during the trajectory, reference to practical examples can be frequently observed as part of an account for a concept, while this will be less common both on the hunch level and on the holistic level.

Further hypotheses may be made concerning aspects of metaknowledge. It is possible that the scientifically more mature students also are better at recognising the didactical contract (Brousseau, 1997) implicit in the question format. Having the metacognitive ability to distinguish between theoretical and practical means of explanation, they would then be able to provide shorter and more precise explanations. Further investigation is recommended to address such issues.

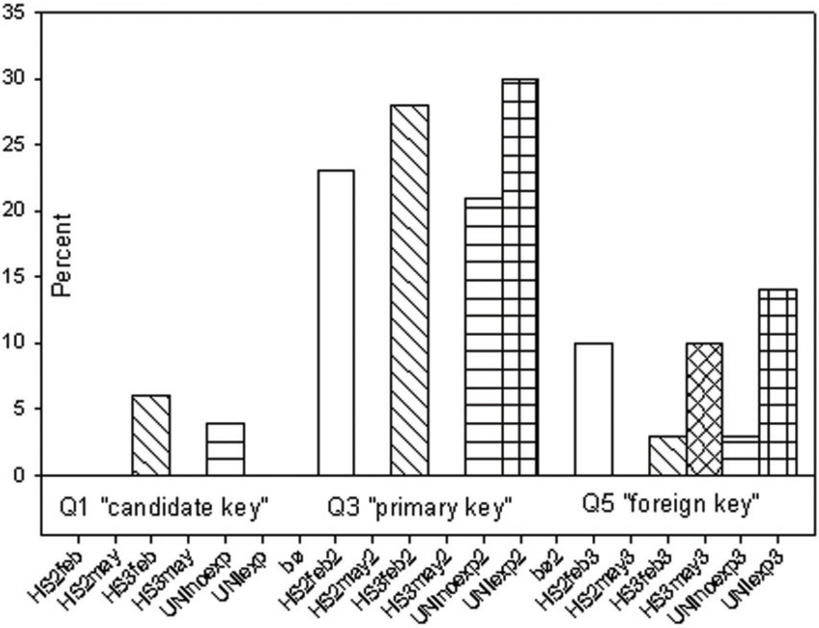

Figure 2. Percentage of the students answering a question who included an example in their explanation.

Collaborative Processes

Collective Development of Conceptual Networks

In order to measure the socially constructed common knowledge within groups and possible between-group differences, discriminant analysis was used to build a predictive model of group membership based on observed characteristics of each student's answers. The procedure generates a set of discriminant functions based on linear combinations of the predictor variables that provide the best discrimination between the groups. The functions are generated from a sample of students for which group membership is known; the functions can then be applied to new students with measurements for the predictor variables but unknown group membership. In this case, the known groups were the classes to which the students belonged (original group), and the discriminant analysis was used to predict to which class a student belonged based on the characteristics of his or her answers to the three questions. From the total of 110 variables, 23 were used in the five canonical functions. The set of 23 variables used in the final analysis included, for instance, Q1 (whether the explanation for candidate key included the term primary key), Q3 (whether the primary key was said to be the main attribute), and Q5 (whether an example was included in the explanation of foreign key).

Notice that score was not included in these predicting characteristics. In fact, there was little difference in score between the groups within each age level. The between-group differences were thus found in how (i.e., in what manner), and not in how well the students explained the different concepts.

Table 3 shows the predicted distribution generated by the discriminant analysis as explained at the end of the "Material and Methods" section. Each row represents a real group of students (i.e., class). The number of students predicted by discriminant analysis to belong to each group is indicated in the columns. Of the eight students belonging to Group 5 for example, seven were correctly predicted to belong to Group 4 while one was erroneously predicted to belong to Group 14.

Table 3

Predicted distribution of number of students in each group generated by discriminant analysis (Prediction accuracy 75.6%). Groups 1-5 are HS2-classes; Groups 11-14 are HS3-classes.

		Predicted group number									
		1	**2**	**3**	**4**	**5**	**11**	**12**	**13**	**14**	**Total**
	1	**9**		3	1		1				**14**
	2		**9**	3						2	**14**
	3		2	**9**	1					2	**14**
Original group number	**4**				7					1	**8**
	5					**6**				1	**7**
	11		1				7				**8**
	12		1					**8**			**9**
	13								7		**7**
	14			1		1			1	**6**	**9**

Since the prediction of group membership for a case is based on functions calculated partly from that same case, there is a tendency to capitalise on chance so that the prediction accuracy will be somewhat misleading. To get a better estimate of the prediction accuracy, a cross validation was performed, meaning that each case was classified based on functions computed from all cases except that one. With this method, the prediction accuracy was still as high as 48.9%. For comparison, the analysis was performed with nine randomly constructed groups. The cross validation prediction accuracy for the random sample was 11.1%, which is the same as chance for nine groups.

The discriminant analysis thus gives remarkably high prediction accuracy. This indicates that there are substantial qualitative differences in the way students in different groups used language to explain scientific concepts. One may argue that this could just as well be a result of influence from the teacher. The lessons in these classrooms were, however, mainly not controlled by the teacher, but by the verbal interaction between students. Furthermore, Groups 12 and 13 had the same teacher. Still, the discriminant analysis did not predict any students from Group 12 to come from Group 13 or vice versa. These findings conform well with previous research documenting the social construction of meaning between people in collaborative classroom practices (Lemke, 1990; Mercer, 1995; Scott, 1998). While these emphasise the student or class/teacher interaction as essential, the present results indicate that the student-student interaction is also an important factor in forming the common understanding (Edwards & Mercer, 1987) or common thematic patterns (Lemke, 1990). Assuming that "learning is a process of enculturation," Brown and associates (1989: p.40) point out that "groups of practitioners are particularly important."

In agreement with constructivist theory, Hennessy claims that "children's prior conceptual knowledge significantly affects their predictions, explanations, and perceptions of novel phenomena" (1993: p.10). The data presented here suggest that the social setting, in which these novel phenomena are introduced and used, plays at least as important a role. Hennessy also emphasizes the importance of social interaction as a contributing factor for children's cognitive development. The importance of social and discursive interaction has also been emphasised by others (e.g., Brown et al., 1989; Edwards, 1990). What is seen in the present study is empirical evidence of the nature of this contribution to the scientific concept-building process.

Collective shift of focus

The discriminant analysis presented above introduces the idea of between-group differences and within-group similarities in terms of the students' explanations to the three questions. A study of the answers from some of the repeat students gives further support for the influence of social processes on the cognitive development. Of the eight HS3 students who took the test twice, none mentioned determining or deciding as a feature for the primary key on the first occasion. Three months later, however, six of these eight included this aspect. Five of them even included this aspect as the only one the second time. Clearly, there had been a shift in focus on the description of what a primary key is over these few months. This observation illustrates that the students develop a common way of using language to describe a given concept. Such tacit establishment of common conceptual knowledge and language use has previously been described, for instance, by Goodwin (1997) studying chemistry students' perception of shifts in the colour of a solution. Such development is dependent on discursive interaction and is therefore probably more prominent in a collaborative classroom.

Practical Versus Definitional Knowledge

So far, the individual development of conceptual networks, as well as the social influence on this development, has been discussed. In this section, the concept development is considered in terms of different types of knowledge. The distinction between practical and theoretical knowledge is used as a starting point for the development of a framework for describing the trajectory from initial hunches to holistic knowledge.

Despite the format of the questions possibly inviting definition-like answers, the students' accounts held elements of both practical and definitional nature. Some of the explanations were mainly related to a practical context (e.g., an example of the functionality of the software tool), whereas others resembled a definitional statement.

Sfard (1991) has described cognitive development as iteratively moving from operational to structural knowledge. According to this theory, the first encounter with a new concept will be in terms of operations on known objects that fall under this concept. Through practising the skills by applying operations to different objects, an understanding of the concept, as such, is developed. The operation, or process, is then experienced as an object of its own. The *operational knowledge* is *reified* (made into an object for further treatment on a higher level) (Sfard, 1991). What is known as operations on one cognitive level, return as objects on a higher level. This description of the learning process is similar to what Halliday (1993) calls a reinterpretation of the world from the spoken knowledge mode (reflecting common sense) to the written, educational mode of scientific knowledge. Still, this is not necessarily a dichotomy. There is a synthesis between the development of everyday and scientific concepts in adolescence (Vygotsky, 1986). "Any particular **instance**, of any kind of phenomenon may be interpreted as some product of the two—once the adolescent has transcended the semiotic barrier between them." (Halliday, 1993).

Following this, the dichotomy or hierarchical division between operational and structural knowledge (Sfard, 1991) appears not to capture the process of scientific concept development in adolescents in sufficient detail. Rather, it seems sensible to place the two types of knowledge as parallel influences to a continuous development from *initial hunches* to *holistic knowledge*. Concept development may thus be described as a cognitive trajectory (Figure 3) that is influenced by both operational (practical/context dependent) knowledge and structural (definitional) knowledge. Roth and Lawless (2002) demonstrate how "beginning with initially almost incomprehensible talk, students developed observational and theoretical language for the phenomenon at hand" (p.375). Described as a parallel process constituting the development from what they call *initial "muddle"* to *mature science talk*, this provides a clear parallel to the trajectory outlined in Figure 3.

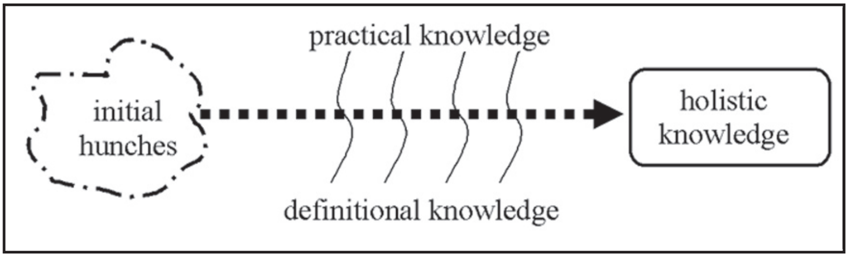

Figure 3. The knowledge trajectory for concept-building

It can be hypothesised that from initially only having vague hunches of what a concept means, the students gradually improve their understanding towards holistic knowledge influenced by both practical experience and theoretical input from textbooks or teachers. Holistic knowledge is, in other words, reached through an interaction between skills and understanding. A student needs knowledge of the process as well as the concept on which the process operates (Gray & Tall, 1994). Mere understanding has no value without the skills to implement it, and the skills alone, though useful in many situations, cannot be seen as knowledge unless accompanied by a mental understanding of the concepts at hand.

The extent to which different types of sub-statements are prominent in a student's answer is influenced by the exposure to which the student has been subject. The problem-oriented classroom organisation is a likely reason for the practical aspects being as prominent as they are in the students' explanations in this study, despite the question style possibly inviting more definitional answers.

CONCLUSIONS

From a social constructivist perspective, concept building has been described in this paper as a trajectory from initial hunches towards holistic knowledge influenced by both practical experiences and theoretical input. During this trajectory, the students seem to gradually expand the network of related concepts available for use in a verbal account of the concept at hand. At an intermediate level of familiarity, the students feel confident enough to include concrete examples as part of their explanations. When the level of understanding is closer to holistic knowledge, however, seeing the whole picture enables the student to extract a few central features and still feel confident that the account is sufficiently accurate. At this level, the example may also be omitted. This general description of concept development is not necessarily related to the educational setting in which this study was undertaken. It rather provides a picture of the outcome from individual construction of conceptual knowledge in any domain as influenced from the surrounding recourses (i.e., hands-on experience, textbook definitions, and fellow students).

The second main finding of this study, however, addresses the situatedness of this learning process. The identification of strong within-group similarities and between-group differences provides a means of coming to grips with the practical implications of the situated perspective on learning. "Prevalent school practices assume […] that schools are neutral with respect to what is learned, that concepts are abstract, relatively fixed, and unaffected by the activity through which they are acquired and used" (Brown et al., 1989: p.37). Contrary to this assumption, the situated perspective on learning claims that context is not only relevant, but also crucial for the learning outcome. Indeed, the way in which the students in this study use language to explain scientific concepts clearly reflects the context in which these concepts have been learned and dealt with. In collaborative learning activities, the need for *grounding* (Baker, Hansen, Joiner, & Traum, 1999) or *common knowledge* (Edwards & Mercer, 1987) has been thoroughly documented. For the present study, this is relevant because the concept development is not only influenced by the social context in which it occurs, but it further seems to be a collective process toward a common way of using language

– a socially constructed language game (Wittgenstein, 1958).

The classrooms of this study are characterised by a particular work style, where the computer functions as a natural cognitive assembly point for subject-related discourse. This is an aspect of computer-supported collaborative learning that has, thus far, not had much focus. Roth (Roth, 1995) shows how an interactive computer tool can support the development of canonical use of scientific language, if appropriately utilised by the teacher in dialogue with the student. Others have provided qualitative descriptions of the computer as a tool facilitating the development of mutual understanding between students (Littleton & Light, 1998). This paper adds quantitative evidence of the significance of these learning processes.

These findings should be applicable to most subject areas provided that the educational setting is replicated. Organizing a physics or chemistry classroom with a group-based and problem-oriented teaching approach as the predominant activity type would probably foster the same kind of collaborative scientific concept development as the one that has been documented here, especially if the computer as mediating artefact (Säljö, 1998) is introduced as an assembly point for the exploratory talk among the learners (Mercer & Wegerif, 1998). This, however, needs further investigation.

The results presented here are based on transverse and not on longitudinal data. The description of the individual development of conceptual networks (or thematic patterns (Lemke, 1990)) in a social setting would benefit from collecting data from the same individuals at different points in time—as has only been done in part for this study. Furthermore, assumptions about the learning process are based on a comparison of the outcome of learning from a particular type of classroom with the expected outcome from a more traditional classroom. The interactional processes that lead to this outcome are only implied. It would be valuable to expand the data with tape recordings or video documentation of the discursive practices constituting this collective concept development.

Implications for Teaching

From a constructivist perspective (Glasersfeld, 1989), the experiences to which students are exposed indirectly contribute to forming their understanding of scientific concepts. Students in different classrooms will obviously have different experiences on which to base their construction of knowledge. Lemke (1990) uses the notion of *thematic patterns* as a means for analysing the way language is used in a classroom setting for describing relationships between scientific concepts. Thematic patterns resemble the conceptual networks described in this paper as an individual's understanding of the meanings of and interrelationships between scientific concepts. The point is that the way in which language is used in the scientific discourse of the classroom seems highly decisive for the students' concept development, and should, therefore, not be underestimated. The teacher has a clear role in offering unifying interventions in order to prevent the group at large from developing unwanted ways of using scientific language. Meanwhile, the teacher needs to adapt to the different linguistic mini-cultures that emerge in these classrooms. Teaching two different classes simultaneously, the teacher needs to be able to adapt his or her use of language to fit the linguistic cultures of each of the two classrooms.

Teachers also might find it useful to have in the back of their minds the stages, which students tend to pass through as they develop their conceptual understanding. The apparent interdependency of skills and understanding for the development of holistic knowledge places a challenge on the teacher to carefully balance the practical experiences of problem-solving activities with theoretical (or definitional) input.

References

Anderson, A., Beattie, R., & Beck, K. (1998). Chrysler goes to "Extremes". *Distributed Computing*, 24-28.

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

Anderson, J. R., Reder, L. M., & Simon, H. A. (1997). Situative Versus Cognitive Perspectives: Form Versus Substance. *Educational Researcher, 26*(1), 18-21.

Baker, M., Hansen, T., Joiner, R., & Traum, D. (1999). The Role of Grounding in Collaborative Learning Tasks. In P. Dillenbourg (Ed.), *Collaborative Learning: Cognitive and Computational Approaches* (pp. 31-63). Amsterdam: Pergamon.

Brousseau, G. (1997). *Theory of didactical situations in mathematics*. Dordrecht: Kluwer.

Brown, J. S., Collins, A. M., & Duguid, P. (1989). Situated Cognition and the Culture of Learning. *Educational Researcher, 18*(1), 32-42.

Clancey, W. J. (1997). *Situated cognition : on human knowledge and computer representations*. Cambridge: Cambridge University Press.

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

Edwards, D. (1990). Discourse and the Development of Understanding in the Classroom. In O. Boyd-Barrett & E. Scanlon (Eds.), *Computers and Learning* (pp. 186-204). Wokingham, UK: Addison-Wesley.

Edwards, D., & Mercer, N. (1987). *Common Knowledge: the Development of Understanding in the Classroom.* London: Routledge.

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

Goodwin, C. (1997). The Blackness of Black. In L. B. Resnick, R. Säljö, C. Pontocorvo & B. Burge (Eds.), *Discourse, tools and reasoning. Essays on situated cognition* (pp. 111-142): Springer-Verlag.

Gray, E., & Tall, D. (1994). Duality, ambiguity and flexibility: a proceptual view of simple arithmetic. *Journal for Research in Mathematics Education, 25*, 116-140.

Greeno, J. G. (1997). On Claims That Answer the Wrong Questions. *Educational Researcher, 26*(1), 5-17.

Halliday, M. A. K. (1993). Towards a Language-Based Theory of Learning. *Linguistics and Education, 5*, 93-116.

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

Holmboe, C. (1999). A Cognitive Framework for Knowledge in Informatics: The Case of Object-Orientation. *ACM SIGCSE Bulletin (Proceedings of IT-iCSE), 4*, 17-21.

Kolderup, E., & Bostrøm, E. (1998). *Systemutvikling. Informasjonsteknologi modul 2a*. Otta: Gyldendal Undervisning.

Lang, K., & Lang, G. E. (1991). Studying events in their natural settings. In K. B. Jensen & N. W. Jankowski (Eds.), *A Handbook of Qualitative Methodologies for Mass Communication Research* (pp. 193-215). London: Routledge.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge, UK: Cambridge University Press.

Lemke, J. L. (1990). *Talking Science: Language, learning and values.* Norwood, NJ: Ablex.

Lie, S., Taylor, A., & Harmon, M. (1996). Scoring Techniques and Criteria. In M. O. Martin & D. L. Kelly (Eds.), *Third International Mathematics and Science Study, Technical Report* (Vol. 1). Chestnut Hill, MA: Boston College.

Littleton, K., & Light, P. (Eds.). (1998). *Learning with Computers; Analysing productive interaction*. London: Routledge.

Matthews, M. R. (Ed.). (1998). *Constructivism in Science Education*. Dordrecht: Kluwer Academic Publishers.

Mercer, N. (1995). *The guided construction of knowledge. Talk amongst teachers and learners*. Clevedon: Multilingual Matters Ltd.

Mercer, N., & Wegerif, R. (1998). Is ‚exploratory talk‘ productive talk? In K. Littleton & P. Light (Eds.), *Learning with Computers. Analyzing productive interaction* (pp. 79-101). London: Routledge.

Nosek, J. T. (1998). The case for collaborative programming. *Communications of the ACM, 41*(3), 105-108.

Potter, J. (1996). *Representing Reality; Discourse, Rhetoric and Social Construction*. London: Sage Publications.

Roth, W.-M. (1995). Affordances of computers in teacher student interactions - the case for interactive physics (TM). *Journal of research in science teaching, 32*(4), 329-347.

Roth, W.-M., & Lawless, D. (2002). Science, Culture, and the Emergence of Language. *Science Education, 86*(3), 368-385.

Schoultz, J., Säljö, R., & Wyndhamn, J. (2001). Conceptual knowledge in talk and text: What does it take to understand a science question? *Instructional Science, 29*, 213-236.

Scott, P. (1998). Teacher talk and meaning making in science classrooms: a Vygotskian analysis and review. *Studies in Science Education, 32*, 45-80.

Sfard, A. (1991). On the Dual Nature of Mathematical Conceptions: Reflections on Processes and Objects as Different Sides of the Same Coin. *Educational Studies in Mathematics, 22*, 1-36.

Sfard, A. (1998). On Two Metaphors for learning and the Dangers of Choosing Just One. *Educational Researcher, 27*(2), 4-13.

Säljö, R. (1998). Learning as the use of tools: A sociocultural perspective on the human-technology link. In K. Littleton & P. Light (Eds.), *Learning with Computers. Analyzing productive interaction* (pp. 144-161). London: Routledge.

Vanderbilt, The Cognition and Technology Group at (1990). Anchored Instruction and Its Relationship to Situated Cognition. *Educational Researcher, 19*(6), 2-10.

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

Wenger, E. (1998). *Communities of Practice. Learning, Meaning and Identity*. Cambridge: Cambridge University Press.

Widdicombe, S., & Wooffitt, R. (1995). *The Languages of Youth Subcultures*. London: Harvester Wheatsheaf.

Williams, L. A., & Kessler, R. R. (2000). All I Really Need to Know About Pair Programming I Learned in Kindergarden. *Communications of the ACM, 43*(5), 108-114.

Wittgenstein, L. (1958). *Philosophical Investigations* (G. E. M. Anscombe, Trans. 2nd ed.). Oxford: Basil Blackwell.

Notes

[1] Widdicombe and Wooffitt (Widdicombe & Wooffitt, 1995) have successfully used a similar method of short direct questions.

[2] The interviews were transcribed in Norwegian using the conventions developed by Jefferson as described in Potter (1996). The analysis of the material was completed prior to the translation to English.

A semiotic framework for learning UML class diagrams as technical discourse

Christian Holmboe[1] & Erik Knain[2]
[1]Department of Informatics
University of Oslo, Norway
[2]Department of Teacher Education and School Development
University of Oslo, Norway
christian.holmboe@ifi.uio.no and erik.knain@ils.uio.no

Abstract

In this paper we study the *semiotic demands* placed upon the students by UML class diagrams as a means for making descriptions of a subset of the world using technical language. Examples of interactions between students who are solving a data modelling problem also illustrate the large variety of *semiotic resources* available to the students when faced with a modelling task. Part of this complexity is attempted captured by a three-dimensional framework developed by drawing on Hjelmslev's semiotic theories of metalanguage and the notion of grammatical metaphor from Halliday. Our analysis interprets discourse as a movement between different metalinguistic layers, between expression and content, and between technical and everyday contextual frames.

By our approach we find that concept building in a technical language and acquisition of metalinguistic knowledge cannot be understood solely as the appropriation of a predefined conceptual system. For instance, we find two different approaches to data modelling, distinguished by the contextual frame of the activity: i.e. data modelling as (1) a condensed image of the world, and (2) as a schematic representation of an information system.

It is important that learning activities make explicit how language is used in different ways within the different discourses operating simultaneously in a data modelling situation. An awareness of these aspects constitutes an important metacognitive competency for novice data modellers.

Prologue

A main rationale for OO modelling is to enable system developers to model a part of the world in the same manner that they envision it in a natural setting (Coad & Yourdon, 1991). Data modelling methodologies are tools for representing and transforming key aspects of the world in the form of coherent and logically consistent descriptions. In the following excerpt, three undergraduate university students are pondering their first modelling assignment using UML class diagrams, after first having spent a few weeks of an introductory course in system development on relational databases. They have previously made a relational data model for the same problem (i.e. a bank system), but are now asked to start from scratch using object-oriented (OO) methodology instead.

Excerpt 1: (Students P, S & D)[1]

011[2]	P:	Yea, but what kind of diagram are we actually going to make
012		here?
013	S:	Class diagram I think
014	D:	Think () we () are () going () to () make () class-
015	P:	Yea, but is that the class diagram?
016		((points to a choice on the screen))
017		Is it, does it look like that, or does it look like that, or does it
018		look like that?
019	S:	What are you saying?
020	D:	I think it looks like that one, with these ()
021		eh those boxes to put it like that
022	P:	Is that a class diagram as well, then?
023	D:	Yes
024	S:	think so
025		[…]
026	P:	We are going to make this kind of class diagram that looks
027		like that
028	S:	Yes more or less
029		huh
030		I feel so extremely sure about this ((ironically)).

The students immediately focus on figuring out what they are expected to do, rather than on how to actually solve the problem. It is a matter of fitting in and doing what is appropriate; of fusing previous experiences with meaning making, perceived and tacit expectations in the situation, and their motivated interest in the act of sign making.

[1] The transcripts have been made in significant detail, using the Jefferson system (as described in Potter, 1996: p233-34). On completion of the analysis, the selected excerpts have been translated from Norwegian and simplified for clarity.

[2] The line numbers run continuously through the original transcript. Thus, excerpt 2 follows directly after excerpt 1, while excerpts 2 and 3 are separated by aprox. 150 lines of transcription (or 3.5 minutes of discourse).

They quickly agree on what they think is expected of them (i.e. making a class diagram), even though they are not quite sure what a class diagram is.

Once the students have established this expected goal of the activity, the focus is changed to solving the problem at hand, making the most of their initial understanding and the tools available to them. In excerpt 2, they are playing around with the UML concept of *class*, trying to identify types of objects in the everyday world that lend themselves to being included as generalized and simplified versions in terms of classes in their OO model.

Excerpt 2:

031	P:	Uhm
032		What kind of boxes do we need, then?
033		We wan'a have those with three on them
034	S:	Erase everything, then
035		pull a box around uhm
036	P:	Yes, one of those
037	S:	There, now what classes are we going to include?
038	D:	Uh, haha
039	P:	Well we must have a-
040		we must have one of accounts
041		empl- cust-
042	S:	Maybe a class account or something
043	P:	Yes, we need to have account
044		and () or persons
045		juridical entities that too- shou-
046		should also been a class there
047	D:	Yes

Through exercises such as exemplified above, the students are about to become participants in the social practice of object-oriented data modelling. They are faced with a whole new set of textual tools and conventions. This situation will be the focus of investigation in the present paper.

Scope and research method

Problem description

From a socio-cultural perspective on learning (Säljö, 1998; Wertsch, 1985), the basic challenge confronting these students is to become practitioners of modelling with UML class diagrams, which will in turn make them members of the community of practice in this field. In this paper we want to study the semiotic demands that class diagrams, as a tool for making unambigous descriptions of a subset of the world, put upon students. In doing so, we focus on the semiotic aspects of technical vs vernacular discourse related to the activity of object-oriented conceptual data modelling. This study was designed to answer the following questions through discourse analysis (Edwards, 1997) of naturally occurring interactions between students who are solving a data modelling problem.

- What semiotic challenges are introduced by the activity of OO modelling with UML class diagrams?
- How do the students handle shifts between contextual frames when making a UML class diagram?
- What type of discursive interactions are characteristic for successful novice data modellers?

The goal of this paper is to develop, and demonstrate the usefulness of, a semiotic framework to improve the understanding of the activity in which the students are engaged (i.e. OO modelling with UML class diagrams). It is therefore not intended as an exhaustive description of the discourse. In particular, as we focus on key events such as shifts of contextual frame, the social interaction per se is not very prominent in our analysis of the discourse. In Halliday's framework, language comprises three metafunctions (the ideational, the interpersonal, and the textual) that are always simultaneously present in language use (Halliday, 1994). We focus in this paper on grammatical metaphor in the ideational metafunction, the aspect of language dealing with the referential world; what language is about. A full analysis in Halliday's social semiotic framework should also include the interpersonal metafunction related to who takes part in the discourse and the relations between them. This dimension is obviously paramount to understanding students' interactions, but we find it to be outside the scope of the present analysis. The textual metafunction (what kind of activity are the participants engaged in) is not seen as very important for our analysis, given our focus on ideational meaning. However, the textual metafunction would become important if the dynamics of students' moves during the discourse were the research focus, and further studies should include this aspect.

Data collection

The data presented in this paper are taken from a larger set of observations of different groups of students in a 2nd term university computer science class covering basic general system design issues. Four groups of approx 20 students each were observed during 2 sessions of 90 minutes, in which groups of 2-4 students were collaborating to solve different UML modelling problems. The tasks comprised use case, sequence and class diagrams. During the sessions, one researcher was present as an observer, and a dictaphone was used to record the interactions of one group of students at a time.

No case is made for representativity of the observations discussed in the analysis of this paper. It is rather the underlying process and the analytical model that we claim is of general interest. We have therefore, for increased readability, chosen to use only one interactional sequence to illustrate the type of discourse that the complete material comprises. Similar interactional patterns were, however, also observed in other groups and on other modelling tasks. The sequence discussed in this paper concerns class diagram modelling, and lasted approx 9.5 minutes (i.e. 400 lines of transcription).

Terminology and Background

An essential notion in our semiotic approach is that language is fundamentally metaphorical. Following this, “reality” is neither a term that can be considered independent of context, nor can it be accessed in any direct way. Our language provides us with an everyday, common sense “theory” of the world around us through its grammar. However, this everyday language can be developed into language practices associated with a more sophisticated, rigorous, and specialized knowledge about certain aspects of the world. Historically, specialized languages have developed as subsets of everyday common-sense or vernacular language to serve functional needs in institutionalized professional communities. These language subsets differ not only in what kinds of objects populate a given version of reality, but also in what kind of processes goes on.

Basically, language use is in this article understood as interplay between *text* and *context* (Halliday & Hasan, 1989). In agreement with Halliday and Hasan, we take *context* to mean any textual or non-textual resources that are drawn upon to associate meaning with a text or an utterance, including knowledge about the situation. The context is not restricted to the close physical environment or immediately preceding discourse, but are, in psychological terms, a “subset of the hearer’s beliefs and assumptions about the world” (Blakemore, 1992: p18). However, in principle virtually everything could be drawn on as a resource. We will refer to the resources (e.g. situational and linguistic) associated with certain events and types of situations as *contextual frames*. Among the contextual resources are memories of particular occasions and individuals. Wickman and Östman (2002) have developed a conceptual tool for capturing the social processes of negotiating meaning. With reference to

Wittgenstein (1958), they claim that the basis for all learning is what is *standing fast*[3]. If what is standing fast does not suffice for explaining an experience, a *gap* may be identified. Such a gap will often lead the students to bring other contextual resources into the discourse. In discourse, a gap is often materialized in a question (see e.g P's statements in lines 011 and 015 of excerpt 1, asking for the meaning of a term). The aim of the discursive activity that follows will be to establish a modified or extended version of what is standing fast. This is usually accomplished through the identification of similarities and differences, as they are revealed through *encounters* with the surrounding world, including other participants as well as memories of events from the past and knowledge from related areas. Excerpt 1 shows that through encounters from a previous lecture, P and D attempt to establish similarities between class diagrams and the graphic types of diagrams available in the menu of their modelling software (lines 015 & 020). This helps them establish a suggestion for what a class diagram is (lines 026-028), even though this new knowledge admittedly isn't standing very fast yet (line 030).

In the following analysis, we will use the notions of gap, encounter and standing fast as a means for identifying key events in the flow of the discourse. These events are then subject to analysis based on the semiotic framework developed in this paper.

The three-dimensional framework

What part of the referential world should be represented in a data model, and in what way? To answer this question, we need to look at a data model in semiotic terms and try to illustrate how there are several coexisting semiotic systems that influence the data modelling process in different ways. We will develop a three-dimensional framework, which we will thereafter use as a tool for analyzing the learning of UML as a technical discourse.

The UML metamodel architecture

As will be further discussed below, UML class diagrams are defined in a four-layer metamodel architecture which regulates the interface between the field of interest and the OO model (see table 1).

[3] The notion of *standing fast* is introduced by Wickman and Östman (2002) meaning something along the lines of "for an individual perceived as an established fact". Despite the expression being a bit at odds with standard English, we have chosen to stay with the expression as the authors have used it.

Table 1: The four-layer metamodel architecture adapted from "OMG – Unified Modelling Language" (Booch, Jacobson, & Rumbaugh, 2001)[4].

Layer	**Description**	**Example**
Meta-metamodel	The infrastructure for a metamodelling architecture. Defines the language for specifying metamodels	Meta-Class; Meta-Attribute; Meta-Operation
Metamodel	An instance of a meta-metamodel. Defines the language for specifying a model.	Class; Attribute; Operation
Model	An instance of a metamodel. Defines a language to describe an information domain.	Customer; interestRate; withdrawMoney()
user objects (user data)	An instance of a model. Defines a specific information domain.	<Paul_Johnsen>; 0.024; withdrawmoney()

For the purpose of this paper, we will leave the top layer alone, and focus on the lower three layers. It will be demonstrated how this structure can be seen in semiotic terms as a taxonomy of *metalanguages*. In the further discussion, we will refer to the different metalayers using the numbers 1, 2 and 3, with 1 being the layer of user objects.

The semiotic dimension

A *sign* is composed of a *signifier* and a *signified*. These are, in a sense, arbitrarily connected; but through regularities in usage, they may become stable by convention. In this way, the signifiers do not directly determine the semantic or conceptual meaning of their signifieds, which enables people to reconstrue their theory of experience (i.e. alter their understanding of the world in terms of what is standing fast).

Following Hjelmslev, a *plane of expression* (E) for the signifiers is associated with a *plane of content* (C) for the signified (Barthes, 1967). According to Barthes, Hjelmslev portrays the process of signification, an act of binding the signifier and the signified, as a relation (R) between E and C. Halliday (1994) has taken a further step of stratifying the content plane, C, into a lexico-grammatical level and a level of semantics. Thibault elaborates on this separation between a lexico-grammatical level and a level of semantics:

[4] Use of symbols and punctuation is copied from the OGM specifications and will be carried through in the subsequent tables as well.

> Linguistic […] signifiers are phonological or graphological patterns which construe […] the lexico-grammatical forms – morphemes, words, phrases, etc – which in turn construe the semantic or conceptual meanings of these lexicogrammatical categories. The signifier signifies the word – the lexicogrammatical form – not the meaning in any direct way (Thibault, 1998: p4).

Following this, we would say that a set of customers as they are represented in the implemented information system (C), correspond to a class labelled Customer (E) by a relation R that is determined by how the expression is used to denote the content. Table 2 shows the expressions at the different metalayers of a data model with suggestions for corresponding referential contents.

Table 2: The expressions (E) at the different metalayers of a data model with their corresponding referential contents (C).

	Expression (E)	**Content (C)**
3	Class; Attribute; Operation	A category as a set of objects sharing a collection of features; A quality or feature describing a range of values that a classifier may hold; A service that can be requested from an object to effect behaviour

2	Customer;
	interestRate;
	withdrawMoney()
		A 'customer' as described in the information system; a variable connected to an account that can hold a real number; description of the actions required for registering a withdrawal of money from an account

1	<Paul_Johnsen>;
	0.024;
	withdrawmoney()
		The information system representation of 'Paul Johnsen'; the digitalized number '0.024'; the execution of the actions described for registering a 'withdrawal of money from a given account'

Connotation

A system E R C can be imbricated into another system in such a way that the two are out of joint with each other. This can be achieved in two different ways depending on the point of intersection of the first system into the second. "In the first case, the first system E R C become the plane of expression, or signifier, of the second system" (Barthes, 1967: p89), (E R C) R C. This is what Hjelmslev calls *connotative semiotics* (see figure 1). For instance, the content for the expression *customer* may be reified on a higher level as a representation (E) of a formalized interpersonal relationship (C), signifying trust, responsibility, security, etc. which in turn could become a signifier (E) for a new system, the banking industry (C).

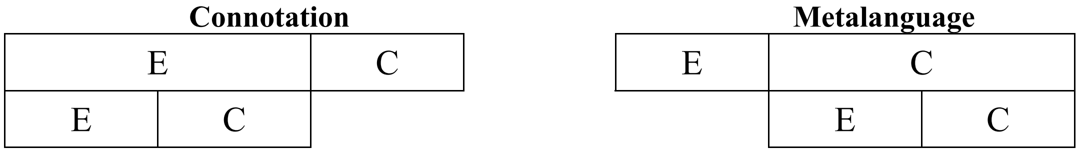

Figure 1: Illustration of connotation and metalanguage as illustrated in Barthes (1967)[5]*.*

Large fragments of the denoted system can constitute a single unit of the connoted system, for instance the tone of a text which is made out of numerous words. The signified of connotation are by character "general, global and diffuse; it is, if you like, a fragment of ideology" (Barthes, 1967: p91).

Metalanguage

Our primary concern in this paper will be the second case (see figure 1), where the second system E R C (i.e. *metalanguage*) is developed from the plane of content of the first (*object language* in Hjelmslev's terms (1984: p128)), as in E R (E R C). "This is the case with all *metalanguages: a metalanguage is a system whose plane of content is itself constituted by a signifying system*" (Barthes, 1967: p90). Being a language that is used to describe another language, a metalanguage may in turn be the object language of a new metalanguage, thus constituting a taxonomic hierarchy of terms in order to account for how specialized discourses operate in particular context-types (Thibault, 1998). This theory of a metalanguage hierarchy is an immediate analogy to the metamodel architecture of UML class diagrams (see tables 1 and 2). The expression *Class* as a metamodel phenomenon represents the idea of the world being dividable into generic classifications of groups of "things" with common attributes, operations, and associations. It is this technical meaning that defines the generation of specific classes at the model-layer.

The third dimension; technical vs. vernacular discourse

We have now established two of the dimensions of our semiotic framework, one along the 3 metalayers of UML, and the other consisting of the semiotic relationship between expression and content (E R C). Having come this far, we need to emphasise the distinction between the expression <Paul_Johnsen> (E) as a data model object and the expression 'Paul Johnsen' (E) from vernacular language. Their respective referential phenomena (C) are accordingly not identical either. <Paul_Johnsen>

[5] Sr and Sd in Barthes' illustration have been substituted by E and C respectively.

represented as an instance of the class Customer is a simplified and altered version of the physical person ‘Paul Johnsen’ who is a live customer of the bank. Similarly, on metalayer 3, the technical expression *class* is a technical term with a transformed meaning from an everyday understanding of the term *class*. A term usually associated with everyday situations is in other words put into a specialized language designed for handling technical representations. In table 3, this change of contextual frame represents a move from right to left. In this manner, we can describe the data model representation and its object of reference as a transformation of their vernacular parallels. What has changed is the contextual frame. On one level (i.e. the right hand side), we have the semiotics of a vernacular lexicogrammar related to everyday life phenomena, and on the other (i.e. the left hand side), the semiotics of a specialized technical lexis that by design is supposed to serve certain purposes for OO modellers. The designed metalanguage hierarchy “feeds on” the vernacular metalanguage hierarchy. The technical-vernacular dimension thus completes our three-dimensional framework.

Table 3: The three-dimensional framework for the semiotics of data modelling. (E.g. The shaded cell will be referred to as TC2 indicating the Content of a Technical term on metalayer 2.)

Technical language (T)			Vernacular (everyday) language (V)	
	Expression (E)	Content (C)	Expression (E)	Content (C)
3	Class; Attribute; Operation	A category as a set of objects sharing a collection of features; A quality or feature describing a range of values that a classifier may hold; A service that can be requested from an object to effect behaviour	class; attribute; operation	A categorization of the world into concepts with common features; A quality or feature of a member of a class; a nominalised (see below) version of an action or process
2	Customer; interestRate; withdrawMoney()	A ‘customer’ as described in the information system; a variable connected to an account that can hold a real number; description of the actions required for registering a withdrawal of money from an account	‘customer’; ‘interest rate’; ‘the withdrawal of money’	The generalized typical ‘customer’; the quantification of the interest rate quality of an account; the physical process of withdrawing money from an account
1	<Paul_Johnsen> ; 0.024; withdrawmoney()	The information system representation of ‘Paul Johnsen’; the digitalized number ‘0.024’; the execution of the actions described for registering a ‘withdrawal of money from a given account’	‘Paul Johnsen’; ‘2,4 %’; ‘a withdrawal’ of money	‘Paul Johnsen’ in the flesh; the annual interest of 2,4 %; a particular ‘withdrawal of money’ from a given account

Looking at the right hand column in table 3, there is some technicality involved in this language too. It is in a sense a metalanguage taxonomy for banking. It is however *vernacular* (considered from the perspective of OO-modelling) in the sense that it is associated with the everyday business of arranging one's financial affairs[6]. So, there are two metalanguage taxonomies involved, one technical and one vernacular. The vernacular metalanguage is defined through the use of the object language in social practices. T3 however represents a normative metalanguage, in that the rules for the use of language at the T2 layer are predefined and thus not influenced by the actual use in the same manner.

Finally, it is important to keep in mind that there is no exact correspondence between E and C. Language does not simply reflect an independent external reality. "Instead, each level in the hierarchy recursively re-construes the others" (Thibault, 1998: p9), while the context constrain the possible relationships.

The dynamics of metalanguage

Nominalization as grammatical metaphor

Halliday has pointed out the importance of what he labels grammatical metaphor as a kind of coupling and decoupling between a plane of semantics interfacing with the world of human experience, and a grammatical plane as a conceptual system for that experience. Scientific knowledge has evolved as a very particular kind of metaphorical re-construal of experience, in which *nominalizations* are important in expanding, transcategorising, compacting, distilling and theorizing (Halliday, 1998). Nominalizations are therefore useful both for argumentation, and for building theory.

Nominalizations are meanings condensed from a full process, lacking not only *participants*, but also *action* (i.e. the verb). For instance, "I want to block the account" represents a full process that is realized by a verb (to block) and contains participants (i.e. *actor* - I, and *goal* - account). In our framework (table 3) this will be a V1 statement. The details of this process (time, participants, responsibility) is gradually lost by way of passive voice "The account was blocked" into a noun in form of the nominalised concept of *a blocking*, which implies a shift to V2. Excerpt 3, below, serves to illustrate this mechanism.

[6] The bottom right cell with 'Paul Johnsen in the flesh' is perhaps the only 'true' vernacular context.

Excerpt 3:

192	S:	No wait
193		a blocking that can just be some field inside the accounts
194		() right?
195	P:	Yes
196	D:	Blocked ((inaudible)) then we take a print-out ((inaudible))
197	S:	Yes
198	P:	Yes
199	S:	For example
200		if you want to get a list of all those blockings then
201	P:	Do you have to be able to get a list of all the blockings that
202		have come on ones account?
203		()
204		Then you can store blockings in a hash map
205	S:	Yes and that- then it becomes blocking-objects
206		because they are
207	P:	Yes
208	D:	Unless we should just get out a list of it
209	P:	Don't need a list of all blockings do we?
210	S:	No, I have no idea about that
211	P:	I have never tried to block my account
212	D:	Let's try that and we'll see

The shift from the activity of blocking an account to the nominalised *blocking* implies a transport (Greek: *metaphoros*) of meaning from the category of actions to the category of things. Note that some of the action that would be lost in the transformation from process (verb) to nominalization (noun) may be put back into the clause, as when a costumer in a bank says "I want to make a blocking[7]". The lost action is in this way re-created in a V1 situation.

By nominalization as grammatical metaphor, a vernacular process is made into an object labelled by a noun (nomen). Through a shift of contextual frame from the vernacular to the technical, this nominalised expression is transformed into a technical sign with a meaning derived from the original vernacular one. Thereafter, one may need to unwrap the grammatical metaphor of the technical sign in order to gain further understanding of its content, or one may indeed return to the vernacular context and use the technical sign there in order to gain a further understanding of it. The technical-vernacular dimension is thus a dynamic link which enables the modeller to shift back and forth between the different contextual frames, as alternate resources for coming to grips with the content aspect of the sign in the situation at hand.

[7] In the participants' native tongue, the term "sperring" functions well as a nominalised term also in V1, while such use may be less plausible in English.

Within the field of mathematics, Sfard (1991) advocates the view that the process by which mathematical reality is constructed in the image of physical reality involves a metaphorical projection in which the virtual reality discourse of mathematics is built in the image of actual reality discourse (i.e. a transformation from vernacular to technical discourse). Sfard furthermore distinguishes between treating mathematical notions as referring to abstract objects in the form of *structural* conceptions, and *operational* conceptions of a notion such as processes, algorithms and actions. The transition from operational to structural conception is called *reification* and corresponds to the generalization of moving from T1 to T2 in our framework. Experienced objects come into being when they are attributed existence and properties in discourse (Dörfler, 1999; 1999).

In more recent work, Sfard describes mathematical (i.e. technical) lexis as defined through, and simultaneously constitutive for, mathematical discursive practices (Sfard, 1999). Applied to our framework, the relationship between TE and TC is defined through discursive interaction within the contextual frame of data modelling; and at the same time, this discursive practice is influenced by the technical lexis and its meaning as it is used in the same discourse.

The fact that there are alternative expressions and alternative ways of expressing, means that the students are requested to master metalanguage and "intervene with this hierarchy so as to adjust it to suit the agent's own purposes" […] "the possibilities of higher- or metalevel norms entails possibilities of choice from among alternatives" (Thibault, 1998: p5). Students can do this in many ways because the relation between expression and content is a flexible one, and also because the relation between a given denotative language and its metalanguage is a priori arbitrary[8]. This implies that the same vernacular E R C may give rise to various metalanguage systems depending on the context of specialization.

Technical vs. vernacular discourse revisited

White (1998) distinguishes between *scientific* and *technological* discourses. Typically, scientific discourse reconstructs commonsense reality, whereas technological discourse also seeks to *extend* reality by developing "new categories

[8] It may, however, become stable by convention in practice, by way of example and authority.

and new names for these categories" (ibid. p267). Thus, while scientific language tends to transform a common-sense language into an uncommon-sense one, technological language tends to populate an everyday world with artefacts. This is commonly achieved by the use of acronyms, which may take an independent status from what they originally stand for. We see an example of this phenomenon when our students experience the need for an attribute of a person to tell whether this person is an employee or a customer. For labelling this feature, they introduce the technological acronym-like term *EC-code* (see excerpt 6). However, since the distinction between scientific and technological language isn't crucial for the argument in this paper, we will use technical language as a notion comprising both scientific and technological language with their combined features and properties.

Transformations and shifts of contextual frame

As emphasized by Halliday, the lexico-grammar of scientific language has developed into what it is because science is fundamentally an activity where everyday concepts are transformed into something less familiar and more formal and systematized. This provides an iconic strangeness that "serves as a signal that the version of reality which these terms construe is "alien" to the version of reality construed by the familiar, typically native or nativised forms of vernacular discourse" (White, 1998: p290).

In each of the following two short excerpts (4 & 5), we see a shift from the everyday use of the terms *account* and *country* to the technical versions of these same terms as they are used to denote components of a data model.

Excerpt 4:

037	S:	There, now what classes are we going to include?
038	D:	Uh, haha
039	P:	Well we must have a-
040		we must have one of accounts
041		empl- cust-
042	S:	Maybe a class account or something

Excerpt 5:

126	P:	country
127		no that is not a
128		[class, is it?
129	D:	[The object country
130	P:	No
131	D:	haha

The terms *class* and *object* respectively function as signals that 'Account[9]' and 'country' are used as labels for data model components, whereas the same two terms in the initial statements of these two excerpts are more likely to refer to the vernacular understanding. The phrasing "one of accounts" in line 040 indicates that P here refers to the class (one) as representing (of) a vernacular phenomenon (accounts).

The interaction in excerpt 3 starts in a technical contextual frame with the expression *Blocking* (TE2) referring to the class of the data model (line 193). Further along the same line, the implementation is mentioned as a field of one particular customer's account, indicating that the contextual frame has shifted to TC1. While D (line 196) is probably still referring to *blocked* as a state value in the implemented system, S subsequently initiates a shift towards the world of banking (VE2) followed up by P (lines 201-202) focusing on the everyday customer's ability to get a list of blockings on request (V1). Line 204 marks an important shift by introducing the feature of hash-maps as a concluding remark to the preceding discussion. The conclusion is in part based on the reference to their knowledge of everyday banking (V1 & V2), but simultaneously draws on knowledge of the metalanguage of OO (T3), describing the available features of a data model and their use in creating a description of a domain. Next, S comes to the conclusion that they then need *blocking-objects*, as opposed to the initial suggestion to settle for an *attribute* of the class account. The phrasing "…then it becomes…" (line 205) indicates that this conclusion is also derived from knowledge on T3 level. At this point there is, however, still some confusion concerning the necessity of being able to produce lists of blockings. The students therefore return to the vernacular realm of banking for further reference (line 211). Everyday language is always there to be drawn on, and in general, language use tends to gravitate towards its everyday, vernacular state, especially when users of technical language run into difficulties. The statement in line 211 furthermore supports the intuitive claim that knowledge of the problem domain is important for successful modelling.

Language users will move between everyday language, metalanguage and technical language in complex ways. As an aid for keeping track of the shifts and

[9] The capital 'A' indicates that the term is a class name, which by convention is capitalized in many OO programming languages.

transitions discussed in this paper, figure 2 graphically illustrates of some key movements in the three-dimensional framework discussed.

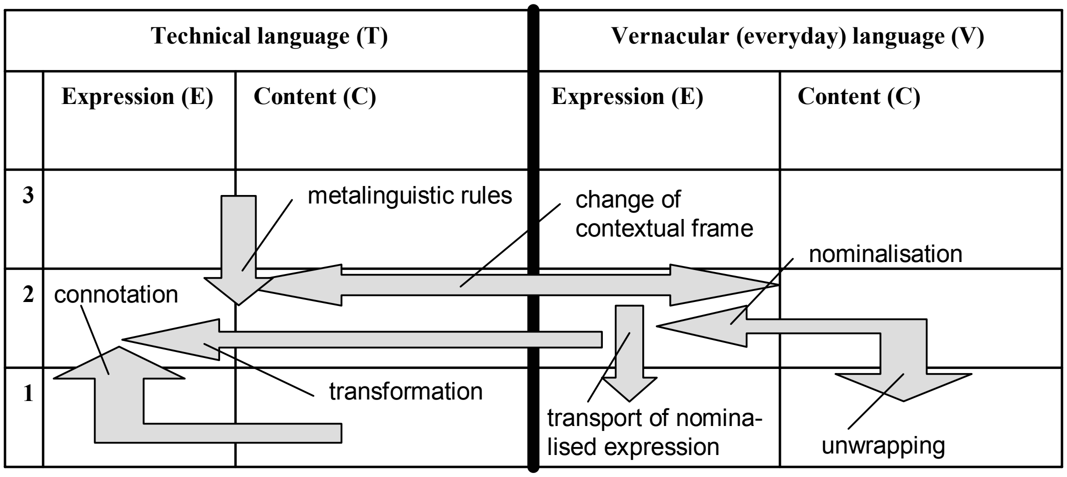

Figure 2: Examples of students' movements within the framework as they are discussed in the analyses.

Sometimes, students will alternate between answering the question of what an expression means and the oppositely directed question of how to label an arised meaning content (Holmboe, 2005). In excerpt 6, below, the students seem to have grasped the content (TC2), but need a technical expression (TE2). They are accustomed to look for objects in the world (VC1) that can be generalized into categories (VC2) with a corresponding label (VE2), which can in turn be "borrowed" (i.e. transformed) in order to function as TE2 (confer arrow labelled 'transformation' in figure 2). After brief suggestions by P (lines 287-288), this strategy is abandoned, and reference is made to a previous solution to a similar problem. They end up by creating an acronym *EC-code* (line 292) which is technological in White's sense, in that it does not bring about a shift in content C once identified (as one would expect a scientific term to do), but rather simply put a name tag on something. The name implies that it is a dichotomous variable, E for employee, C for customer.

Excerpt 6:

285	S:	Should we have an employed/unemployed or something like
286		that?
287	P:	Yes, function ehr ()
288		employed question mark hehe ()
289		We could just call it that then
290	S:	Yes, but didn't we called it C-code here
291		or something here then
292		EC-code or something
293	P:	EC-code
294	S:	EC-code, yes that's right

Learning technical discourse implies learning the lexico-grammatical language of that discourse, which, for science, implies learning to transform everyday or vernacular language into an uncommon-sense language. This implies a change of contextual frame from V to T (again, confer figure 2). A semiotic relationship E R C from one metalayer is given a new meaning when it is re-contextualised into the next metalayer. R is likewise formed according to certain metalinguistic rules (described in the metalayer above). Following Thibault, the metalinguistic principles that are part of learning OO modelling "enable language users to contextualize the relationship between object and system of interpretation" (Thibault, 1998: p3). In excerpt 3, we observed how students construed *blocking* by moving from a technical contextual frame into a more vernacular frame; from *Blocking* in an OO model into *blocking* as experienced in everyday life of banking. This constitute a move in the opposite direction from the one just suggested for scientific discourse learning (i.e. from T back to V), demonstrating the reciprocal nature of the relationship between vernacular and technical languages. This shift can subsequently be followed by an unfolding of the nominalised vernacular concept, to better understand the meaning of it as it is inherited from the initial processes.

Two approaches to data modelling

Data modelling can be described as a semi-graphical representation in two different respects; (1) as a simplified image of the world, and (2) as a schematic representation of an information system. These correspond to two different approaches to data modelling distinguished by the contextual frame of the activity. In (1) the contextual frame is on the real world domain to be modelled (i.e. V1 and V2), whereas (2) focuses on the use of the finished information system.

We have so far maintained the position that TE corresponds (albeit in a nondeterministic way) to TC. This semiotic relationship is one between the data model and the implemented information system. Yet, when making a data model, the establishment of classes, attributes and relationships can, as we have demonstrated, be based on knowledge of the problem domain (i.e. the vernacular realm). In a sense, the expressions of the data model are then construed, by the data modeller, with reference to a simplified version of the real world phenomena that they correspond to. This implies that there is a kind of direct relationship between TE and VC. Each of these two approaches will be exemplified and their implications briefly discussed in the following two subsections, still with a clear focus on the students' contextual shifts within the three-dimensional framework.

Data model as a simplified image of a subset of the world

An important aspect of constructing a class diagram is identifying the categories of phenomena from the problem domain that are to be represented by classes in the data system. As already described, this activity will usually include generalizing from objects of VC1 to a nominalized expression in VE2, subsequently transformed to TE2. The generalization from metalayer 1 to metalayer 2 in the vernacular plane operates in parallel with the dictation from metalayer 3 describing the rules for the transformation from the generalized VE2-VC2 relationship to the modelled TE2-TC2 relationship. In more basic terms: The formation of a class is influenced or controlled from two different directions. One is the technical-vernacular dimension where what is to be modelled is located in the vernacular and transformed to a technical expression by way of grammatical metaphor. The specifications of the UML metamodel (T3) constitutes the other direction. Being of a normative nature, T3 sets the rules for how the aforementioned transformation may be carried out.

As illustrated in excerpt 7, these processes are not necessarily referred to explicitly in the discourse. One can, however, pinpoint where attention is given to the different contexts involved. The students agree that they need to include Account as well as Person as classes. These both represent natural categories (Rosch, 1978), in the form of semiconcrete and concrete phenomena (Holmboe, 2005), respectively. Such categories are easily transferred to components of the data model at the T2-layer, given that the students have a reasonable understanding of the different constructs available to them and their implications as inherited from the T3 layer. They are, however, uncertain as to the significance of a class Person in relationship to JuridicalEntity (lines 045-046) and whether or not employees and customers should be represented by different classes (lines 048-049), be subclasses of Person/JuridicalEntity (line 050), or just be identified through an attribute of JuridicalEntity that holds a value telling whether the person is a customer or an employee (lines 053-054). One reason for this confusion is the fact that they here encounter two different metalanguages for the same base language (i.e. the juridical and the everyday versions of bank-language). The interaction in excerpt 7 operates in the intersection between two parallelly functioning metalanguage systems. After jointly listing their options, P asks "what do we choose?" (line 056). The decision is experienced as a choice where several options may be correct, but the choice will have impact on the further modelling and future implementation of the system.

Excerpt 7:

043	P:	Yes, we need to have account
044		and () or persons
045		juridical entities that too- shou-
046		should also have been a class there
047	D:	Yes
048	P:	or do you want to search for employees and juridi-
049		or customers in two classes?
050	D:	That becomes subclasses, then
051	P:	Yes
052	S:	Yes or either subclasses or that it just
053		that we shove a field into that one which says eh:m
054	P:	whether it is employee or customer?
055	S:	Yes
056	P:	What do we choose?

Data model as a schematic representation of an implemented information system

The sequence in excerpt 7 is not totally independent from the implementation and use of the data system. By referring to "search for employees" (line 048), P takes a different perspective than the one described in the previous section. The contextual frame is no longer a focus on the real world domain to be modelled (i.e. V1 and V2), but has shifted to a focus on the use of the finished information system. The possible requirements of a future user of the system suggest that customer and employee should be made into separate subclasses of JuridicalEntity, in order to facilitate faster and easier search in the database.

Excerpt 8 provides another example where the use of the implemented system is in focus. Both S and D acknowledge that in order to be able to store certain information, "you get" a class in the model (lines 133 & 137), in this case the class Country.

Excerpt 8:

133	S:	Well if we shall store that, then you get a class country
134	P:	Time date becomes something like that
135		((referring to something different on the screen))
136		maybe something like that
137	D:	well we do get a class country then
138		if we want to store that at all.

The interaction in excerpt 3 is also interesting seen in light of the two different approaches to modelling just described. As already mentioned, the initial reference to Blocking (line 193) is as a nominalised term of TE. Reference is then made to activities of using the database (i.e. requesting a print-out), which brings P to focus on the customer wanting such a print-out (lines 201-202). Recognising the need for this functionality, P immediately shifts back to the implementation level (line 204). This is picked up by S in the subsequent turn, concluding that objects are needed (line 205) in order to get a hash-map. Having reached this conclusion, the conversation shifts back again to the everyday setting of the real bank and the customers' need (lines 208-212). This conforms to the fact that people seek ways to use language that resemble natural language as much as possible. Further support for this is found in the tendency of people to ease the cognitive burden by finding ways to minimize the level of abstraction in problem solving (Hazzan, 2003).

Through the sequence of turns in excerpt 3, the students work out what they need to include in their model, and how to implement the blockings. In addition to this, they use the technique of unwrapping to consolidate the meaning of the nominalised term *blocking* through constant comparison and reference to the everyday processes that initially were reified to construe the object.

Other contextual shifts

The examples in the previous section demonstrate how the students' contextual frames shift discursively between modelling a part of the world and modelling the use of an implemented system. The following excerpt provides a slightly different example of meaning making through shift of contextual frame.

So far, we have mainly demonstrated how the metalanguage part described in Barthes' semiological framework (see figure 1) has been used by the students as a resource for developing the relationships between expression (E) and content (C). The way that the signifier and the signified are frequently fused in the experience of the symbolizer (Nemirovsky & Monk, 1999), provides a premise for the connotative shift, where the E R C of a language may become the E of a semiotic system on the next level (confer section 3.2.1). This is a further resource for the students' meaning-making. In excerpt 9 the E R C of an implementation in Java (T1) becomes a sign for a range of associations that could be drawn on for program practices and ways of structuring the field of interest (T2).

Excerpt 9:

141	P:	I have some problems with transferring this stuff from those
142		tables to objects
143		() class country?
144	S:	Okey, just like-
145		forg- forget that it is a table
146	P:	Yes
147	S:	Now you are in a Java-
148		you are making a Java program
149		And there you are only going to store addresses
150	P:	Yes
151	S:	and then you want to have inside here some kind of option
152	P:	Hash-
153		can't you just make a hash-map of all the countries for that?
154	S:	Yes, but what does it contain?
155		() When we have two things
156	P:	Yes
157	S:	we want to store about each country
158	P:	Yes, then it will go to a country-object
159	S:	Yes, haha

In order to help P understand how the problem should be implemented (i.e. that they need a class Country), S chooses to initiate a shift of contextual frame to the activity of programming in Java (lines 144-148). The conclusion of the reasoning (line 158) can then be carried back to the initial context of the data modelling problem.

This particular connotative shift of contextual frame furthermore resembles the tendency to seek towards more familiar semiotic systems. These students are clearly more familiar with programming in Java than with UML modelling. Design choices made for an implementation in Java therefore function as useful references for an improved understanding of how to model the problem at hand with a UML class diagram.

Discussion

The concept of metalanguage is described by Thibault as

> a taxonomic hierarchy of terms, either folk or scientific, when what is needed is an account of the ways in which metalinguistic discourses are themselves operative in particular context-types.

He furthermore calls for attention to

> the need for a more dynamic, praxis-oriented approach, rather than a static, taxonomizing one. That is, we need to investigate the relations among the local interactional context, the metasemiotic consciousness of the interactants, their always partial viewpoints, and the ways in which the interaction of all of these perspectives serves to bring into or out of focus particular metalinguistic forces of a given utterance, as construed from some social viewpoint. (Thibault, 1998: p6).

In this, we find support for our claim that concept building in a technical language and acquisition of metalinguistic knowledge is something different from the appropriation of a predefined conceptual system. Learning to handle these kinds of discursive activities offers opportunities for the students to draw on a variety of previous knowledge and experiences, as well as the resources present in the form of encounters (Wickman & Östman, 2002) in the situated context of the learning activity.

Navigating in this complex network of contextual frames of reference, the students tend to strive for *optimal relevance* (Blakemore, 1992). When a gap is identified, they will use experiences and encounters as cognitive tools leading them into different contextual frames of reference, until a sufficiently adequate solution is reached and further efforts are no longer worth while. In other words, ”the utterance will have adequate contextual effects for the minimum necessary processing” (Blakemore, 1992: p36). The theory of optimal relevance depends on the fact that concepts do not have a fixed predefined meaning independent from their use in discursive practices (Wittgenstein, 1958). However, further studies are needed in order to map this complexity, including the interpersonal and textual metafuncions in Halliday’s theory.

Data modelling implies the construction of new technical semiotic systems or language games (Holmboe, 2004). The meaning of the terms used in a data model will therefore be defined by the way they are used in that particular setting. Sometimes vernacular terms are transferred by grammatical metaphor into *scientific expressions*. Such concepts do not always differ much from the vernacular meaning of the same terms. But the artificially constructed language game of the data model also incorporates technological terms representing more abstract phenomena (Holmboe, 2005), and these have a less familiar meaning or content.

Proficiency

The findings of this study applies to novices of data modelling with UML (and probably with other methodologies too, as well as with programming). We claim that to be successful in these activities, novices must rely on the ability to operate across different metalinguistic layers and with different semiotic systems in parallel. Indeed, several previous studies have pointed to similar kinds of flexibility as a characteristic of expertise (e.g. Bonar & Soloway, 1985). An overall pattern for expert behaviour, as documented in these studies, seems to be that they are able to handle information at different levels in paralell (Petre, 1990). In conformity with the main findings of our paper, Détienne emphasises that designers (and thus data modellers) "use knowledge from at least two different domains, the application (or problem) domain and the computing domain, between which they establish a mapping" (Détienne, 2002: p22). This notion has also been emphasised by others (e.g. du Boulay, 1989) and it parallells insights in science education, where being able to express meaning across different semiotic systems is seen as essential "in learning science and learning how to think and act scientifically" (Wallace, Hand, & Prain, 2004: p43). In accordance with previous findings (e.g. Visser & Hoc, 1990), Détienne (2002) furthermore describes the seemingly unstructured behaviour of experts as *opportunistic design*, with emphasis on the multi-dimensional nature of program design. A brief overview of studies describing behaviour of expert programmers and designers can be found in Robins, Rountree and Rountree (2003).

The distinction between a data model as an image of the problem domain, and as a representation of the information system, can be said to correspond to *analysis* and *design* respectively. Hitchman (2003) found that expert system data modellers working with ER claimed not to focus on the analysis part. His subjects did, however, consistently check their model against their knowledge of the problem domain. The novices observed by Berge et al. (2003) did not seem to do so. This is in agreement with the point made in the present paper that experts are distinguished from novices by having the ability to shift between contextual frames without explicitly having to focus on the fact that they do so. Since Hitchman's experts do not separate between analysis and design models, Berge et al. (2003) claim that there is no need to introduce this distinction for introductory students. Contrary to this claim, the analysis presented in our study indicates that even if such distinction is not made explicitly by expert practitioners, it needs to be focused on explicitly in teaching. Further support for this position is found in the theory of skill levels outlined by Dreyfus and Dreyfus (1986). According to their theory, an expert is characterised by knowing the task and the rules for performing it so well that he or she does not have to consciously use these rules any longer, whereas the less experienced practitioner needs to concentrate on the rules for the activity in order to manage the task.

The excerpts presented in this paper are from one group of three students who cope fairly well with this aspect of the data modelling activity. We have seen that frequent contextual shifts seemed to be valuable for reaching common understanding of the problem and of the implications for modelling the problem domain. As mentioned, our example is taken from a larger set of observations of several similar groups of students. Many of the other groups demonstrated less ability or willingness to shift as frequently as S, P, and D did. After shifting from the technical domain their discussion could typically dwell in the vernacular realm for as much as 10 to 15 minutes before they eventually managed (sometimes aided by the tutor) to shift back to focusing on the data model. On other occasions they got stuck in technicalities of the data modelling environment without referring to the problem domain to check the relevance of the issue they were pondering. The transcripts of these interactions are unfortunately less suitable for rendering here, due to considerations of length. The tendency of *anchoring* – i.e. that novices get stuck with an initial approach to solving a problem – has been noted by others as well (Batra & Antony, 1994; Schoenfeld, 1992)

Implications for teaching

Seen from a socio-cultural perspective, the aim for learning is to become proficient participants of a socially constituted practice. The practice of OO modelling with UML class diagrams incorporates different semiotic systems, operates across a variety of metalinguistic layers, and draws on a multitude of discursive resources. Being able to manoeuvre in this complex semiotic network (here described in terms of the three-dimensional framework) is thus a crucial part of the skills associated with proficiency in the activity of data modelling. So is the ability to draw on additional information available from previous experiences and encounters. We have demonstrated how the framework can be used as an analytical tool for identifying the students' use of grammatical metaphor to transfer terms and meaning from one semiotic system to another, in addition to other movements between different contextual frames. However, future work encompassing larger and varied groups of students in various educational contexts is needed in order to corroborate and refine the model suggested above.

In light of the results indicated in this paper, and previous studies suggesting similar patterns of expert behaviour, the organization of teaching and learning of data modelling in general, and UML class diagrams in particular, should make room for the students to practice formulating meaning in different ways. Shifting between vernacular and technical language, and applying previous linguistic knowledge and experience to new situations, requires practice. It is furthermore important to make explicit the fact that language is used in different ways within the different discourses operating in parallel in a data modelling situation. Explicit awareness of these aspects constitutes an important metacognitive competency, which students should be helped to attain.

It is outside the scope of this paper to suggest specific ways to organise learning environments in order to facilitate this. The present study is an explorative study describing the discursive behaviour of students. Through this description, we aim to introduce a mindset for teachers rather that to prescribe a certain set of classroom activities. More explicit implications and suggestions will hopefully follow as outcomes from subsequent experiments applying the proposed framework as an analytical tool.

Acknowledgements

We would like to thank our collegue Karl Henrik Flyum for providing valuable insight into the mazes of semiotic theory.

References

Barthes, R. (1967). *Elements of Semiology* (A. Lavers & C. Smith, Trans.). New York: Hill and Wang.

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

Berge, O., Borge, R. E., Fjuk, A., Kaasbøll, J., & Samuelsen, T. (2003). *Learning Object-Oriented Programming.* Paper presented at the Norsk Informatikkkonferanse (Norwegian Informatics Conference).

Blakemore, D. (1992). *Understanding Utterances. An Introduction to Pragmatics*. Oxford and Massachusetts: Blackwell Publishers.

Bonar, J., & Soloway, E. (1985). Preprogramming knowledge: A major source of misconceptions in novice programmers. *Human-Computer Interaction, 1*(2), 133-161.

Booch, G., Jacobson, I., & Rumbaugh, J. (2001). *OMG - Unified Modelling Language Specification v1.4*. Needham, MA: OMG Object Management Group.

Coad, P., & Yourdon, E. (1991). *Object-Oriented Analysis* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Détienne, F. (2002). *Software Design - Cognitive Aspects* (F. Bott, Trans.). London: Springer.

Dreyfus, H., & Dreyfus, S. (1986). *Mind over Machine*. Glasgow: Basil Blackwell.

du Boulay, B. (1989). Some difficulties of learning to program. In E. Soloway & J. Spohrer (Eds.), *Studying the novice programmer* (pp. 283-299). Hillsdale, NJ: Lawrence Erlbaum.

Dörfler, W. (1999). Means for Meaning. In P. Cobb, E. Yackel & K. McClain (Eds.), *Symbolizing and Communicating in Mathematics Classrooms: Perspectives on Dicourse, Tools and Instructional Design* (pp. 99-132). Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

Edwards, D. (1997). *Discourse and Cognition*. London: Sage Publication.

Halliday, M. A. K. (1994). *Introduction to Functional Grammar* (Second ed.). London & Melbourne: Edward Arnold.

Halliday, M. A. K. (1998). Things and relations: Regrammaticing experience as technical knowledge. In J. R. Martin & R. Veel (Eds.), *Reading Science. Critical and Functional Perspectives on Discourses of Science* (pp. 185-235). London and New York: Routledge.

Halliday, M. A. K., & Hasan, R. (1989). *Language, context, and text: aspects of language in a socialsemiotic perspective*. Oxford: Oxford University Press.

Hazzan, O. (2003). How Students Attempt to Reduce Abstraction in the Learning of Mathematics and in the Learning of Computer Science. *Computer Science Education, 13*(2), 95-122.

Hitchman, S. (2003). An interpretive study of how practitioners use entity-relationship modelling in a ternary relationship situation. *Communications of the Association of Information Systems, 11*, 451-485.

Hjelmslev, L. (1984). *Sproget. En introduktion* (2. ed.). København: Museum Tusculanums Forlag.

Holmboe, C. (1999). A Cognitive Framework for Knowledge in Informatics: The Case of Object-Orientation. *ACM SIGCSE Bulletin (Proceedings of ITiCSE), 4*, 17-21.

Holmboe, C. (2004). A Wittgenstein Approach to the Learning of OO modelling. *Computer Science Education, 14*(4), 275-294.

Holmboe, C. (2005). Conceptualisation and Labelling as Linguistic Challenges for Students of Data Modelling. *Computer Science Education, 15*(2).

Nemirovsky, R., & Monk, S. (1999). "If You Look at it the Other Way ...": An Exploration Into the Nature of Symbolizing. In P. Cobb, E. Yackel & K. McClain (Eds.), *Symbolizing and Communicating in Mathematics Classrooms: Perspectives on Dicourse, Tools and Instructional Design* (pp. 177-223). Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

Petre, M. (1990). Expert Programmers and Programming Languages. In J.-M. Hoc, T. R. G. Green, R. Samurcay & D. J. Gilmore (Eds.), *Psychology of Programming* (pp. 103-116). London: Academic Press.

Potter, J. (1996). *Representing Reality; Discourse, Rhetoric and Social Construction*. London: Sage Publications.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education, 13*(2), 137-172.

Rosch, E. R. (1978). Principles of Categorisation. In B. Lloyd (Ed.), *Cognition and Categorisation*. Hillsdale, NJ: Erlbaum.

Schoenfeld, A. H. (1992). Learning to Think Mathematically: Problem Solving, Metacognition, and Sense Making in Mathematics. In D. A. Grouws (Ed.), *Handbook of Research in Mathematics Teaching and Learning*. New York: Macmillan.

Sfard, A. (1991). On the Dual Nature of Mathematical Conceptions: Reflections on Processes and Objects as Different Sides of the Same Coin. *Educational Studies in Mathematics, 22*, 1-36.

Sfard, A. (1999). Symbolizing Mathematical Reality Into Being - Or How Mathematical Discourse and Mathematical Objects Create Each Other. In P. Cobb, E. Yackel & K. McClain (Eds.), *Symbolizing and Communicating in Mathematics Classrooms: Perspectives on Dicourse, Tools and Instructional Design* (pp. 37-98). Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

Säljö, R. (1998). Learning as the use of tools: A sociocultural perspective on the human-technology link. In K. Littleton & P. Light (Eds.), *Learning with Computers. Analyzing productive interaction* (pp. 144-161). London: Routledge.

Thibault, P. J. (1998). *SRB Insights: Metasemiosis*. Retrieved 09.10.03, 2003, from http://www.chass.utoronto.ca/epc/srb/srb/metasemiosis.html

Visser, W., & Hoc, J.-M. (1990). Expert Software Design Strategies. In J.-M. Hoc, T. R. G. Green, R. Samurcay & D. J. Gilmore (Eds.), *Psychology of Programming* (pp. 235-250). London: Academic Press.

Wallace, C. S., Hand, B., & Prain, V. (2004). *Writing and Lerning in the Science Classroom* (Vol. 23). Dortrecht and Boston: Kluwer Academic Publishers.

Wertsch, J. V. (Ed.). (1985). *Vygotsky and the social formation of mind.* Cambridge, MA: Harvard Univeristy Press.

White, P. R. R. (1998). Extended reality, proto-nouns and the vernacular. Distinguishing the technological from the scientific. In J. R. Martin & R. Veel (Eds.), *Reading Science. Critical and Functional Perspectives on Discourses of Science.* (pp. 266-296). London and New York: Routledge.

Wickman, P.-O., & Östman, L. (2002). Learning as Discourse Change: A Sociocultural Mechanism. *Science Education, 86*, 601-623.

Wittgenstein, L. (1958). *Philosophical Investigations* (G. E. M. Anscombe, Trans. 2nd ed.). Oxford: Basil Blackwell.

Appendix A: The original transcripts in Norwegian for the excerpts presented in paper 1

Excerpt 1:

(Students S, T)			
101	T:	we must have a relationship between denouncer and	vi må ha en relasjon mellom anmelder og
102		denouncementregistration and	anmeldelseregistrering og
103	S:	yes	ja (5.0)
104	T:	that is to say that one denouncer	det vil si at en anmelder (1.0)
105		can have several denouncementregistrations?	kan (.) ha (.2) flere anmeldelseregistreringer? (3.0)
106	S:	eh hn	E:h (.5) hn:
107	T:	like this, look now	sånn, se nå
108		((T working on the computer))	(5.0) ((T arbeidende på PC'en))
109		yes	(7.0) Ja (1.0)
110	S:	and then you have	også har du::
111		eh on the bottom there then you have one and many	Eh nederst der så har du en og mange fordi (.3)
112		because the denouncer he if he	anmelderen han hvis (.) han (.2)
113		he gives	han gi:r (.4)
114		if he is a denouncer then he must reasonably have	hvis han er en anmelder så må han jo rimeligvis ha gitt
115		given one denouncement, right	en anmeldelse ikke sant (2.0) ø:h
116	T:	or he could give many.	eller han kan gi mange.
117		the denouncementregistration, it can have	Anmeldelsesregistreringen, den kan ha
118		one	(2.0) e:n (.3)
119		but zero	men null (.5)
120		or wait a minute then we have many-to-many.	eller vent litt da har vi mange til mange da.
121	S:	yes it ends up with many-to-many then.	ja det blir jo mange til mange da

Excerpt 2:

(Teacher R; Students S, C)			
201	R:	here you have said that a thief can participate in many	her har dere sagt at en tyv kan være med på mange
202		crimes	forbrytelser
203	C:	mm	[mm]
204	R:	and then you say that one crime can uh: can be do-	og så sier dere at en forbrytelse kan ø:
205		carried out by several thieves together. and then you	kan gjø- utføres av flere tyver i felleskap. Og så sier
206		say that for each time you register that a thief	dere at for hver gang dere registrerer at en tyv er med
207		participates in a crime then you must go into this table	på en forbrytelse da må dere inn i denne tabellen
208	S:	mm	Mm
209	R:	then information is stored there	da blir det lagret informasjon der
210	C:	yes	Ja
211	R:	for each time you have connected a thief to a crime,	for hver gang dere har knyttet en tyv til en forbrytelse,
212		then you can also say what commodities he has taken.	så kan dere også si hvilken vare han har tatt.
213	C:	mm	Mm
214	R:	and then you drag that relationship not into the thief,	Og da drar dere den relasjonen der ikke inn til tyven,
215		but into ((small pause))	men inn til (.2)
216	S:	the crime	forbrytelsen

Excerpt 3:

(Teacher R; Student M)			
301	R:	Have you entered any attributes there?	har du lagt inn noen attributter der?
302	M:	no	nei
303	R:	no	nei ()
304		you get a little help from that if you-	du får litt hjelp av det da hvis du,
305		if you enter attributes first	hvis du legger inn attributter først
306		say to that one and that one	skal vi si til den og den
307		and then you see what happens when you create eh	og ser vi hva som skjer når du oppretter en
308		when you entitisize	entitetisering
309	M:	mm	med mer
310		but I don't know what to call it	men jeg vet ikke hva jeg skal kalle den
311	R:	no, but ehm that you can change the name	nei, men e:h det kan du jo endre på navnet også
312		that is later	senere
313		cause it isn't that easy to call it something when you	for det ække så lett å kalle den for noe når du ikke vet
314		don't know what it will contain	hva den skal inneholde
315	M:	No that's right	nei riktig.

Excerpt 4:

(Students J, J2)		
401	J2: Well I'm thinking such that e	Assa jeg tenker det atte: ()
402	cause I think that-	for jeg tenker atte- ()
403	that a class should have a form master	at en klasse burde ha en klasseforstander ()
404	J: a class must have a form master	en [klasse må ha en klasseforstander
405	J2: and the fact that a class, it should at least consist of	[og det a- og det atte en klasse, det burde hvertfall
406	one student, but then ...	bestå av en elev, men så ...

Excerpt 5:

(Teacher C; Student J2)		
501	C: Yes, what kind of information do you have then?	ja, hva slags opplysninger har du da?
502	for each single like that down in that-	Fo- for hver enkelt sånn nedover i den-
503	in that table	i den tabellen (.)
504	for each line.	for hver linje.
505	What pieces of information is it that you have there?	Hvilke opplysninger er det du har der?
506	J2: in class you mean?	i klasse mener du? [eller-
507	C: yes in class	[ja (.) i klasse (.3)
508	in the class-table	i klassetabellen
509	J2: eeh classcode?	e::h klassekode?
510	C: yes	ja
511	J2: and then ehm	også: e:hm (1.0)
512	possibly which school it is in,	eventuelt hvilken skole det er på,
513	with which track	med [hvilken linje
514	C: yes but the way it-	[jo men sånn som det-
515	the way that it stands now then	sånn som det står per nå da (.3)
516	so far?	foreløbig? (.4)
517	J2: no:w?	nå:?
518	C: yes	ja (.5)
519	J2: classcode anyhow	klassekode hvertfall (1.2)
520	C: no, you don't have any attribute which is called that	nei, du ha'kke no attributt som heter det du
521	J2: no I haven't added it	nei jeg har ikke føyd det til
522	but e: it	men [e de:t
523	C: well well okey so you want to have a classcode and then	[jaja, okei så du vil ha en klassekode og [så

Excerpt 6:

(Student N)		
601	N: A crime can receive different sentences, right, cause it	En forbrytelse kan få forskjellige dommer ikke sant for
602	can get both a fine and prison.	den kan få både bot og fengsel.

Appendix B: The original transcripts in Norwegian for the excerpts presented in paper 4

Excerpt 1: (Students P, S & D)			
011	P:	Yea, but what kind of diagram are we actually going to	Jamen, hva slags diagram er det vi skal lage her
012		make here?	egentlig?
013	S:	Class diagram I think	Klassediagram, tror jeg
014	D:	Think () we () are () going () to () make () class-	Tror () vi () skal () lage () klasse-
015	P:	Yea, but is that the class diagram?	Jamen er det klassediagramet?
016		((points to a choice on the screen))	((peker på et valg på skjermen))
017		Is it, does it look like that, or does it look like that, or	det, ser det sånn ut, eller ser det sånn ut eller ser det
018		does it look like that?	sånn ut? Er
019	S:	What are you saying?	Hva sier du nå
020	D:	I think it looks like that one, with these ()	Jeg tror det ser sånn ut, med de her ()
021		eh those boxes to put it like that	eh sånne bokser for å si det sånnt
022	P:	Is that a class diagram as well, then?	Er det også klassediagram, da?
023	D:	Yes	Ja
024	S:	think so	Tror det
025		[…]	[…]
026	P:	We are going to make this kind of class diagram that	Vi skal lage et sånt klassediagram som ser
027		looks like that	sånn ut
028	S:	Yes more or less	Ja omtrent
029		huh	heh
030		I feel so extremely sure about this ((ironically)).	Jeg føler meg så innmari sikker på det her (ironi).

Excerpt 2:			
031	P:	Uhm	Ehm
032		What kind of boxes do we need, then?	Hva slags bokser må vi ha da
033		We wan'a have those with three on them	Vi skal ha sånne med tre på
034	S:	Erase everything, then	Slette alt sammen da
035		pull a box around ehr	trekke en boks rundt eh.
036	P:	Yes, one of those	Ja, en sånn
037	S:	There, now what classes are we going to include?	Sånn, ja hvilke klasser er det vi skal være med?
038	D:	Uh, haha	Eh, hehe
039	P:	Well we must have a-	Ja vi må ha en-
040		we must have one of accounts	vi må ha en over kontoer
041		empl- cust-	ans- kund-
042	S:	Maybe a class account or something	Kanksje en klasse konto eller no sånn
043	P:	Yes, we need to have account	Ja konto må vi ha
044		and () or persons	også () eller personer
045		juridical entities that too- shou-	juridiske enheter det ogs- bur-
046		should also have been a class there	burde også vært en klasse der
047	D:	Yes	Ja

Excerpt 3:			
192	S:	No wait	Nei vent, da
193		a blocking that can just be some field inside the accounts	en sperring det kan bare være no felt inni eh kontoer ()
194		() right?	vel
195	P:	Yes	Ja
196	D:	Blocked ((inaudible)) then we take a print-out ((inaudible))	Sperret ((uhørbart)) så tar vi en utskrift ((uhørbart))
197	S:	Yes	Ja
198	P:	Yes	Ja
199	S:	For example	For eksempel
200		if you want a list of all those blockings then	hvis man skal ha liste over alle de sperringene da
201	P:	Do you have to be able to get a list of all the blockings	Må man kunne få ut en liste over alle sperringene som
202		that have come on ones account?	er kommet på kontoen sin?
203		()	()
204		Then you can store blockings in a hash map	Da kan man lagre sperringer i en hash map
205	S:	Yes and that- then it becomes blocking-objects	Ja og den da blir det sperringsobjekter
206		because they are	på grunn av at de er
207	P:	Yes	Ja
208	D:	Unless we should just get out a list of it	Hvis ikke vi bare skulle få ut en liste over det
209	P:	Don't need a list of all blockings do we?	Tren'ke liste over alle sperringer gjør vi det?
210	S:	No, I have no idea about that	Nei det aner jeg ikke
211	P:	I have never tried to block my account	Jeg har aldri prøvd å sperre kontoen min, jeg
212	D:	Let's try that and we'll see	Vi prøver det og så ser vi

Excerpt 4:

037	S:	There, now what classes are we going to include?	Sånn, ja hvilke klasser er det vi skal være med?
038	D:	Uh, haha	Eh Hehe
039	P:	Well we must have a-	Ja vi må ha en-
040		we must have one of accounts	vi må ha en over kontoer
041		empl- cust-	ans- kund-
042	S:	Maybe a class account or something	Kanskje en klasse konto eller no sånn ja

Excerpt 5:

126	P:	country	Land
127		no that is not a	nei det er ikke noe
128		[class, is it?	[klasse er det det 'a
129	D:	[The object country	[Objektet land
130	P:	No	Nei
131	D:	haha	hehe

Excerpt 6:

285	S:	Should we have an employed/unemployed or something	Skal vi ha en ansatt, ikke ansatt eller
286		like that?	no sånn
287	P:	Yes, function ehr ()	Ja, funksjon eh ()
288		employed question mark hehe ()	ansatt spørsmålstegn hehe ()
289		We could just call it that then	Vi kan bare kalle den det da
290	S:	Yes, but didn't we called it C-code here	Ja, men har vi ikke kalt den K-kode her eller no sånn
291		or something here then	her'a
292		EC-code or something	AK-kode eller no sånn
293	P:	EC-code	AK-kode
294	S:	EC-code, yes that's right	AK-kode, ja stemmer det

Excerpt 7:

043	P:	Yes, we need to have account	Ja konto må vi ha
044		and () or persons	også () eller personer
045		juridical entities that too- shou-	juridiske enheter det ogs- burd-
046		should also have been a class there	burde også vært en klasse der
047	D:	Yes	Ja
048	P:	or do you want to search for employees and juridi-	eller vil man lete etter ansatte og juridi-
049		or customers in two classes?	eller kunder i to klasser
050	D:	That becomes subclasses, then	Det blir jo til subklasser, da
051	P:	Yes	Ja
052	S:	Yes or either subclasses or that it just	Ja eller enten subklasser eller at det bare
053		that we shove a field into that one which says eh:m	at vi kjører et felt i den som sier e:h
054	P:	whether it is employee or customer?	Om det er ansatt eller kunde?
055	S:	Yes	Ja
056	P:	What do we choose?	Hva velger vi?

Excerpt 8:

133	S:	Well if we shall store that, then you get a class country	Altså hvis vi skal lagre det så får du en klasse land
134	P:	Time date becomes something like that	Tidspunkt dato blir vel noe sånn
135		((referring to something different on the screen))	((peker på noe annet på skjermen))
136		maybe something like that	kanskje no sånn
137	D:	well we do get a class country then	Altså vi får jo en klasse land da hvis vi skal lagre det i
138		if we want to store that at all.	det hele tatt.

Excerpt 9:			
141	P:	I have some problems with transferring this stuff from	Jeg har litt problem med å overføre dette her fra de
142		those tables to objects	tabellene til eh objekter jeg
143		() class country?	() klasse land
144	S:	Okey, just like-	Okei, bare sånn
145		forg- forget that it is a table	glem glem at det er en tabell
146	P:	Yes	Ja
147	S:	Now you are in a Java-	Nå er du i et java-
148		you are making a Java program	du skal lage et javaprogram
149		And there you are only going to store addresses	og her skal du bare lagre adresser
150	P:	Yes	Ja
151	S:	and then you want to have inside here some kind of option	Og så vil du inni her gjerne ha en eller annen mulighet
152	P:	Hash-	Hash
153		can’t you just make a hash-map of all the countries for that?	kan du ikke bare lage en hashmap over alle land da ’a
154	S:	Yes, but what does it contain?	Jo, men hva ligger det i den
155		() When we have two things	() Når vi har to ting
156	P:	Yes	Ja
157	S:	we want to store about each country	vi skal lagre om hvert land
158	P:	Yes, then it will go to a country-object	Ja da går den til et land-objekt
159	S:	Yes, haha	Ja hehe