Attitudes towards Computer Science-Computing Experiences as a Starting Point and Barrier to Computer Science

Carsten Schulte
Institute of Computer Science
Freie Universität Berlin
Takustr. 9
D-14195 Berlin
Germany
schulte@inf.fu-berlin.de

Maria Knobelsdorf
Institute of Computer Science
Freie Universität Berlin
Takustr. 9
D-14195 Berlin
Germany
knobelsd@inf.fu-berlin.de

ABSTRACT

In which way do computing experiences shape attitudes towards computer science? Why do they foster some students' commitment to the subject while they deter others? By using a qualitative research approach we observed that computing experiences have impacts on several dimensions: They affect the self-image with regard to computing, the world-image with regard to conceptions of the subject, and habits in computing. We determined different perceptions of computing (summarized as: use, professional use and design), which together with attitudes (attributions and concepts) of the field enable or inhibit pathways into the field.

Categories and Subject Descriptors

K3.2 [**Computers & Education**]: Computer and Information Science Education – *computer science education, information systems education*.

General Terms

Experimentation, Human Factors.

Keywords

CS, Wider Access, Gender, Computers and Society, CS Ed Research, Pedagogy, Computer Biographies, Transition, Use - Professional Use – Design, Conceptual Change, Attribution.

1. INTRODUCTION

When looking at students in general we have to admit that CS[1] has never been popular among students. Of all students beginning post high school studies in Germany in 2005, only 7.6 % decided to major in CS. With a dropout rate of 38%, CS has headed the table in Germany for years [20]. The decrease in enrolments of all CS students between 2000 and 2005 adds up to 23.5%. In the United States, CS departments also faced a dramatic decrease in enrolments of even 60 % between 2000 and 2004 [40]. This situation is even more dramatic for female students [6], [33], [35]. A great deal of research – mostly in the field of gender – has been done to understand this situation and many different reasons have been given. An overview for the US assumes the following causes: Wrong, limited or inadequate conceptions about the subject matter of CS and career opportunities, as well as the social environment and culture [9]. The German discussion on gender differences in K12 education ascribes the situation to interests, self-confidence and gender-sensibility of teaching [14]. It is therefore obvious that experiences in everyday life and learning opportunities outside formal schooling have a major impact on the current situation: People's interests, their attitudes towards, and conceptualizations of CS are already shaped when students first enter the CS-classroom – whether in school or university – and too often they refuse to enter the CS-classroom at all.

In a research project at our university we aim at understanding the role and impact of these prior experiences. Our focus lies especially on the role of (prior) computing[2] experiences. It seems that for some students computing serves as a starting point, while it is a barrier to CS for others. We want to understand the conceptualization process of CS. We assume that conceptions and beliefs about CS are developed through a learning process which – among different factors (like personal interests, peers, school settings, opportunity to use computers, media) – is influenced through computing in formal and informal settings (school and leisure).

In the next two subsections we briefly discuss the current state of research. The following sections will cover the theoretical background and research methods of our own research project as well as the results of an empirical study with undergraduate students. The paper ends with some concluding remarks.

1.1 Reasons for Low Levels of Interest

A great deal of research has been done to understand the low levels of interest in CS. This was done mainly in gender research. For example, male and female high school students conceptualize CS as a male field [27], [29], putting off women [36]. Good, Estrella and Margolis ([15], pp. 91ff) summarize the following reasons for the under representation of females in CS in US high schools: few opportunities, limited notion of relevance due to presenting a narrow view of CS, negative climate, accumulated negative experiences, and teaching concepts which fail to provide pathways into CS at postsecondary level. The authors demand to

ICER'07, September 15–16, 2007, Atlanta, Georgia, USA.

[2] With the term computing we refer to all kinds of computer usage and interaction with the computer.

foster teacher training in CS, and collaboration of university and high school in developing didactical approaches ([15], pp. 110ff). From experiences in high schools in Germany, Faulstich-Wieland [14] specifies the following reasons: interests, self-confidence and gender-sensibility. The author demands to foster gender-sensitive teaching, change in content and teaching methods, attribution training, and change in the role of teachers.

For postsecondary teaching in the United States, Cohoon and Asprey ([8], pp. 171) give the following main reasons for the gender gap: culture and image of computing experiences, entry barriers, curricula, social support (e.g. role models, peers), confidence and pedagogy. Although there are interesting approaches, they conclude that "we are closer to the beginning than to the end of research into the gender imbalance in postsecondary education" ([8], pp. 170). Similar experiences with successful approaches were made in Germany: Gender-separated courses in CS were successful in the first model-projects, but the scaling-up to a broad range of schools did not work. Faulstich-Wieland [14] believes that considering males and females as different from each other (at least concerning the subject and only due to different experiences) may simply result in stabilizing the situation (because females' attitudes towards CS as a male field are affirmed when girls need special teaching). In addition, it seems difficult to change the role of teachers. Successful programs are in danger to lose their effects when others try to teach according to the original concept.

In summary, there are some recurring topics in the discussion about increasing (female) participation in CS, there are a few successful projects [25], and there are countries – though no north-American or European one – which do not have such problems [1]. However, it is difficult to generalize successful projects. It seems to be difficult to transfer the approach of a single successful project to other institutions with the same problem. In addition, and given the overall low interest rates, we need to take into account male students, too. More theoretical understanding why single interventions work well is needed to generalize successful approaches for successful transfer.

1.2 Role of Computing Experiences

Besides the above discussed reasons for low levels in interest and high dropout rates, students' preconceptions about the subject matter of CS are further issues to be considered. Through literature review we found several hypotheses about freshmen's preconceptions of CS and the impact of prior computing experiences on these preconceptions: Greening conducted a study to explore students' preconceptions about CS and hypothesized: "Is it the case that many students who enroll for a first computer science course do so with some very limiting misconceptions of what the discipline entails?" ([17], pp. 145). He surveyed students of a secondary-level computing course and observed that 92.9% possessed and used a computer at home, but "the majority of students (over 58 percent) were unable to approximate a definition of computer science." (pp. 149). Carter mentions in her study that students who never think of majoring in CS lack subject information, and the most part of the students is unable to explain what CS topics are [7]. Beaubouef and Mason state also that "students often have misconceptions about the field of computer science. Many of them take a computer literacy course, do well in it, and believe that computer science is all about word processors, spreadsheets, or web browsers." ([3], pp. 103). The students' preconceptions that *CS is the understanding of how computer work* is a very common topic in the literature: CS freshmen often believe CS is primarily concerned with using and administrating computers, hardware and programming [24]. Prior programming experiences are often over-estimated [41]. Students who are not so interested in computers never think of studying this subject [7], while others who do well in using it believe to be able to cope with CS as well [2], [39]. Beliefs about IT-jobs and IT-careers are restricted to the cliché of a lonely male programmer at the computer [26].

In conclusion, it seems reasonable that besides gender and social aspects students' conceptualizations of CS have a major impact on the low levels of interest. For postsecondary education Cohoon and Asprey ([8], pp. 149f) conclude that there is a consensus that computing experiences (in a broad definition) have an impact on how students conceptualize CS. The same hypothesis can be found in [5] pp. 50f, [8], pp. 171 and in [17], [19], [23], [31]). However, others believe that computing experiences in programming have only minor or no impact (e.g. [30], pp. 80). Fisher and Margolis say that prior programming experience is no predictor of success; but it seems useful to explain "to entering students that prior experience is not a critical issue" ([25], pp. 80) and to build a curriculum that allows "multiple points of entry" ([25], pp. 81) into the CS-degree program.

McGuffee offers a slightly different thesis for inadequate conceptualizations of the field. He questioned his CS1 and CS2 students about what CS is. While CS1 students reported "computer science is the understanding of how computers works", CS2 students defined computer science as programming ([28], pp. 75). McGuffee concluded that educators like himself may be the reason for students' narrowed view of CS. He therefore points out that "[i]t is important to remember that people who saw the computer as a new and novel experience created the definitions of computer science that exist today." ([28], pp. 75). In the didactical tradition we call this factor the effects of the "hidden curriculum". In other words, students' conceptualizations of the field may only in a minor part be shaped by prior experiences and be more influenced by the teaching approach and implicit messages about the characteristics of CS.

Overall, it is likely that there are no direct relations between prior computing experiences and participation and success in CS ([8], pp. 49ff). That is, computing experiences maybe work as a starting point for some students, while they build a barrier for others. It seems reasonable that growing-up in a technologically rich environment with computing experiences has – beside CS courses – an impact on how students understand and conceptualize CS. In addition, computing experiences of today's freshmen are different [19]. Teachers should take into account prior experiences of their students and be aware of students' preconceptions. Therefore more research is needed to understand such *experience-based entry barriers* ([8], pp. 153) and the effect of computing experiences on attitudes towards CS, including perceptions of the field as well as issues of self-confidence.

2. RESEARCH BACKGROUND

As background for our research, the biographical approach focuses on and aims at explaining how individuals encounter, interpret and incorporate experiences into the process of constructing their own individuality. Secondly, the biographical perspective, as belonging to sociology, acknowledges that the process of individualization is also a social process because individuality is socially constrained. Understanding the impact of computing experiences on enrolment choices and interests in CS in formal as well as in informal settings has two connected dimensions: On the one hand, important processes in the course of life are addressed; on the other hand, the relevance of prior experiences for teaching and learning is considered.

While the first can eventually be seen as a question of sociology, the latter is a question of educational research. From this background it is important to acknowledge experiences and preconceptions of the learners.

In the next section we discuss the theoretical background of research on life courses: biographical research. Thereafter we discuss two theoretical approaches to understand the role of prior experiences in educational research. Theses theoretical strands are then used to frame our research approach (section 3).

2.1 Social Research on Biographies

The biographical research explores how individuals interpret their experiences and social relationships in retrospect. It is a qualitative approach from social research which uses empirical data like life story, oral history, narrative, autobiography or biography. In his research about biography and education, Marotzki specify that "[t]he interplay between the individual and society is seen as an interpretative process which is played out in the medium of significant symbols (such as language). The human being becomes acquainted with the world and him/herself primarily in interpretations that are mediated by, and bound to, interaction. Qualitative biographical research accepts that the biography of an individual can always be understood as a construct, but not only as that. The main focus of its observation lies in studying individual forms of the processing of social and milieu-specific experience." ([26], pp. 102). He concludes that the process of the creation of self-image and a world-image is important for the constitution of biographies. "The *perspective of individual sense and meaning-making* leads directly to the approach of modern biographical research.[…] An understanding of learning and education […] becomes possible only when one comes to understand processes of learning and education as specific way of interpreting oneself and the world" ([26], pp. 103f). A research approach that includes biographical learning processes must therefore consider self-creation and world-making of human beings.

Based on this approach, we developed a qualitative research design to explore computing experiences and their influence on human beings' self-image and world-image in our "computerized" world. We use a coding paradigm suggested by Tiefel [37] in her work on adapting grounded theory for the analysis of biographical learning processes suggested by Marotzki. "[…] [A] modified coding paradigm is proposed, with analytical perspectives geared to the reconstruction of subjective processes of making sense and constructing coherence." ([37], pp. 66). However, learning happens through interaction. Besides trying to reconstruct sense-making and coherence-creation it is therefore important to consider strategies of action, reaction and interaction. Tiefel suggests to consider the following interwoven perspectives in the analysis process and to reconstruct the biographical learning process through them:

- Human beings' self-creation and their subjective processes of sense-making are summed up under the notion of sense perspective. From this sense perspective the self-image is reconstructed. In our case the *self-image* includes selfjudgement and attitudes of one's own computer skills and orientation in the computer world.
- Human beings' relationship to the world and their coherence-creation are summed up under the notion of structure perspective. From this structure perspective the world-image is reconstructed. In our case *world-image* comprehends personal theories and preconceptions about computing and CS.
- Finally, human beings' attitudes and contexts of interaction, reaction and strategies of action are summed up under the notion of the habits perspective. From the *habits perspective* habits or behavior are reconstructed. In our case *habits* comprehend learning strategies, typical performances with the computer and reactions to problems.

These three perspectives form an analytical point of view on the holistic biographical learning process.

2.2 Educational Research on Prior Experiences

While the above outlined biographical perspective reveal the nature of computing experiences, it does not explain their significance for learning CS, as it is done in education research. Prior experiences are commonly considered as very important. The NRC report ([12], pp. 4) mentions as the first learning principle: "new understandings are constructed on a foundation of existing understandings and experiences". Therefore, learning can be regarded as a process driven by preconceptions. Preconceptions, however, can "impose serious constraints on understanding formal disciplines" ([12], pp. 5).

Accordingly, in science teaching, theories on conceptual change play a major role in understanding and supporting learning. These theories are one background we refer to. Conceptual change is generally defined as a form of learning which changes an existing understanding. In other words, learning is based on some kind of prior understanding: students already have an understanding (conception, belief, idea, or way of thinking) about the subject, which in educational research is called pre- or misconception. Teaching with the aim of conceptual change primarily implies uncovering students' preconceptions about a particular topic or phenomenon and using various techniques to help students change their conceptual framework. According to Posner et al. (see e.g. [34] for a more recent discussion of Posner's approach), a successful conceptual change has different requirements: The learner needs to be *dissatisfied* with his or her current understanding; the new concept needs to be *understandable*, *plausible* from the beginning, and it must be *fruitful* to achieve the learners' goals. The main point is that learning problems are not always due to difficulties of understanding, but due to a kind of *unwillingness* to change the current conceptualization, caused by a lack of *meaningfulness* of the new concept for the learner. For example, why should CS be different from one's own everyday computing experiences, since teaching (at least in German high schools) often takes place in a computerized classroom and involves a great deal of time using computers?

Conceptual change theories seem to be useful to explain why it is important to take into account prior knowledge, and to incorporate everyday contexts (multimedia applications, gaming, and internet) into the teaching of programming (see e.g. [14], [19], [23], [25]). However, given this point of view, the passing of assignments and good course grades are not sufficient evidence to conclude successful conceptual change in the students' perception of the field. It could be that students understand their teachers' general assertions about CS, but do not believe them or are not able to change their own attitudes. However, such a shift seems important for long-term success in the field. For example, teaching CS in high schools should help students to make their own choices whether to enroll in CS at college with a reasonable understanding of the field. Therefore, one goal of our research is to investigate whether there are such shifts or conceptual changes and how such changes occur.

Another theoretical background for our research is the Attribution Theory. Attributions describe how a person explains (or prospects) success in a field. A person may consider his or her own abilities, difficulties of a task, already acquired skills, help from others, or luck as reasons or causes. Weiner's model ([16], pp. 6 ff.) distinguishes between: A) the source of reasons (*locus*): Internal ones like abilities or external ones like difficulty of the task. B) The *stability* of the cause: is it constant or can it varies? C) The *controllability*: can the cause be changed by the subject, like the effort put into the task, or is it uncontrollable because the task was solved by pure luck? Wilson and Shrock used attribution as one factor among others to analyze possible success factors. They found that attributions to luck were negatively correlated to midterm results ([42], pp. 187). The application of the theory for education is that persons are likely to have an "attributional style" towards a particular subject like computing. In particular, attribution is useful to explain learned helplessness. Due to computing experiences with seemingly unpredictable errors only sometimes solvable (by luck, controllability), a person might assume that computing always causes usage problems (stability) which are uncontrollable and caused by one's own lack of ability (locus), because peers are able to cope with computing and reinstall the operating system without hesitation. In the context of our research, attribution theory seems adequate to analyze more precisely the role of computing experiences.

3. OUR RESEARCH APPROACH

We assume that computing experiences can foster interest in CS and motivation to pursue a career in this field if the experiences are rewarding and lead to a development of CS related skills and understanding. In the next section we present the research framework, data collection and analysis of our project.

3.1 Research Framework

Based on the theoretical background and coding paradigm discussed in section 2.1, we developed a model describing the effects of computing experiences on attitudes and the notion of CS (see Figure 1). It is obvious that computing experiences are individually processed within a complex framework of internal and external factors (Figure 1, row "Motivation,…, Peers"). By a process of generalization, these experiences shape the notion of CS. This process affects the world-image, the self-image and habits. The different experiences in one's course of life are stratified, so that with each new experience the three – only analytically dividable – dimensions are affected (Figure 1, middle part). Overall, we are interested in the resulting notion and attitudes towards CS (see Figure 1, part "Attitudes towards …"). We assume that these prior notions build the foundation for learning CS, and for deciding whether to enroll in a CS course.

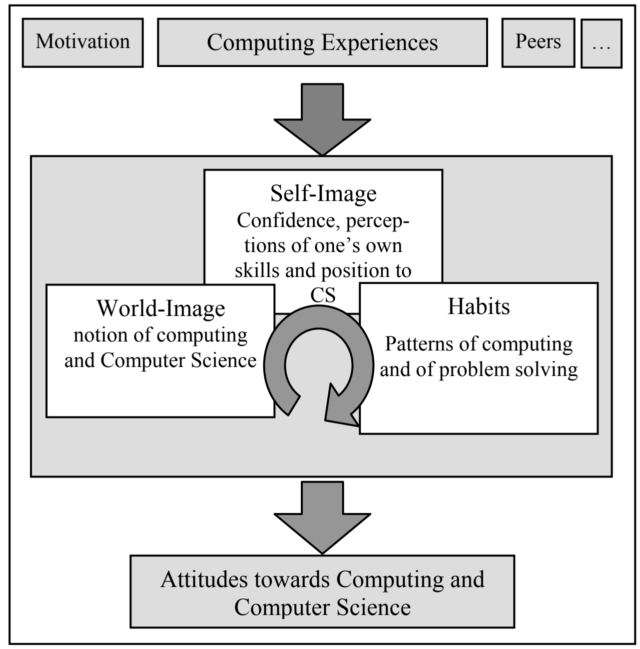

Figure 1: Model of biographical effects of computing experiences on attitudes towards computing and CS

However, this does not explain why computing experiences have such contradictory effects. A possible explanation can be derived from a model suggested by Crutzen [11]. She assumes a hierarchical opposition between different types of activities she calls *use* and *design*. "Use and design are treated as activities in different worlds; a world of senders and a world of receivers, while IT-products are seen as the exclusive links between these worlds. […] The symbolic meaning of use and design is constructed as an opposition in which *design* is active and virtuous and *use* is passive and not creative. Designers see themselves and are seen as makers of a better future and working in a straightforward line of progress. " ([11], pp. 416). *Use* can be described as a kind of computing that excludes CS, whereas *design* refers to a mode of interaction that belongs to CS (see Figure 2). Applying this opposition to the three-dimensional model given in Figure 1, computing experiences conceptualized as *use* or *design* are connected with and also are the result of a certain type of world-image, self-image and habits.

For our educational research interests it is very appealing that Crutzen deconstructs this hierarchic opposition between *use* and *design*. She demonstrates that different relations between *use* and *design* and even inverted hierarchic relations are possible. For example, one can conceptualize programming as a form of usage (of compilers and editors). On the other hand, using a word processor can be considered as designing, where the author designs a new text at the computer. That is, computing experiences can be conceptualized differently, and there is not a strict opposition between *use* and *design,* but instead there is a continuum between these poles. We think this continuum can be regarded and pedagogically used as a pathway into CS. Computing experiences conceptualized as *use* can be seen as a prerequisite or a way to get involved in tasks of creating and designing computing artifacts; thereby making it more likely that these persons are more interested in CS. On the other hand, computing experiences might build a barrier to CS, due to bad experiences or misunderstandings what CS is about.

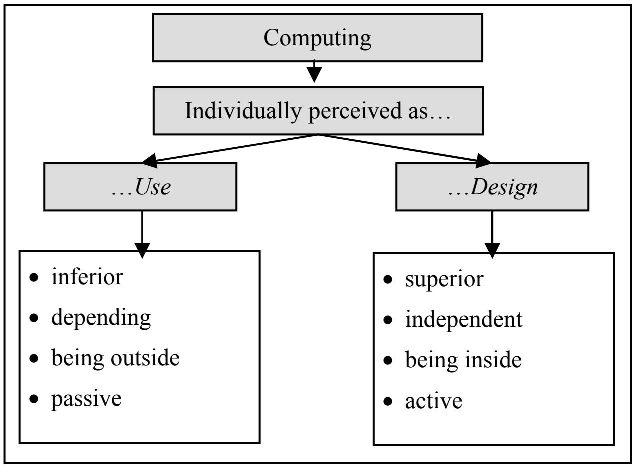

Figure 2: Definition of *use* and *design* as two opposed modes of computing.

Therefore, the research framework can be summarized as follows: Based on the theoretical background outlined above (section 2), we assume that computing experiences affect the world-image, self-image and habits of a person. That is, computing experiences influence attitudes towards and conceptualizations of CS. These prior notions include world-image, self-image and habits. We expect to find two different and opposed notions which can be labeled as *use* and *design*. The resulting general empirical research questions are:

- Are there really persons and opposed groups of persons who conceptualize their computing experiences in alignment to the notions of *use* and *design*?
- Moreover, what world-images, self-images and habits according to CS are connected to these opposed groups?
- Assuming that, at a certain point, freshmen have to think about themselves as *users*, how do they make the transition form *use* to *design*?
- What is the nature of this transitorial process, making computing a barrier for some, while serving as intriguing clue for others?

3.2 Data Collection

Our research approach concentrates on personal experiences with computers: Browsing web-pages, using computers for gaming or for doing homework, tinkering around with hard- and software as well as programming courses in high school are all biographical experiences which are likely to affect attitudes towards CS. Our data collecting method provides narrations about computing experiences in written form. We call our obtained material (and the research method) in short *computer-biographies*. In computer-biographies participants/students tell about their computing experiences in the form of a biography. Thereby, they not only describe objective events and experiences but relate them to their personality and add personal attitudes and opinions. Furthermore, no interaction with an interviewer disturbs this process of story telling.

Participants are asked to write down their biography of computing. In order to support the writing process and focus participants' thoughts, small examples from biographical texts gathered beforehand are given to the participants. These text samples – lure texts – are compiled according to special interests of the study. More precisely, the lure texts mention issues we are especially interested in. These are computing experiences at home, in school, with regard to gaming, the use of applications and the internet as well as programming experiences. In addition, the lure text samples include examples of different styles of computing which we will explain in the next section. We also tried to choose text samples without reproducing stereotypes or prejudices. In order to avoid gender-based prejudices, we provided lure texts of female and two male students. One female and one male text presented a skilled computer user with much interest in CS, while other texts presented critical or less positive attitudes towards computer technology and CS courses (see [10] for an example of a lure text). We wanted the participants to find themselves in their own computing history without influencing the biographical result. By using lure texts, participants are directed to possible interesting topics to write about without a direct request to do so, like perhaps a questionnaire would force to do. Moreover, it leaves room to write about the described aspects according to the participant's own perspective and context – not in the context given from the researchers.

The instrument itself should prevent the story-tellers from omitting relevant aspects. It has been argued that the literature form as narration has a compelling effect on the author, so that the writing process itself forces the participant to a certain extent to deliver a *full story* with beginning, main part and (happy?) end. In addition, as people think about their experiences, they write about those issues they spontaneously and actively remember and assess as important enough to be included in the story. Therefore, the computer biographies are never complete, but are likely to contain only the important issues. Based on this argument we can assume that e.g. a lack of hints on early use experience is mostly due to the fact that it has not occurred or is now irrelevant to the biographer's story of his experiences.

However, while the instrument abandons interaction between the researcher (interviewer) and participant, it allows efficient collection of multiple biographies in parallel – but it prevents the researcher from being able to ask additional questions in order to fill in gaps or clarify ambiguous statements. In addition, we have seen that persons need different amounts of time to write, and the texts differ in length – while overall being quite short. Hence, we assume the text does not give a full account. However, as computer-biographies can be collected more easily on a larger basis, we believe it is a useful instrument to understand *typical* computing experiences, see how they are interwoven and in which way they affect the biographer's attitudes. In an advanced stage of research we intend to supplement data gathering with narrative or semi-structured interviews for more detailed results.

3.3 Sample

In this study we explore whether we can confirm the opposition of *use* and *design* presented in section 3.1, according to which students develop different ideas about their computing experiences. We therefore aimed at finding two opposed groups of students. First we chose a group of CS-affiliated students to confirm a correlation between *design* and computing in their biographies. More precisely, we asked university freshmen starting to major in CS, because we wanted to prevent any prior influence of the university on their attitudes about CS. For the *user*-group we needed CS-unaffiliated majors whose subject is opposed to CS. We also needed persons with a substantial background of computing experiences and decided to ask students majoring in psychology. Psychology lies between natural and human sciences, yet the overall conception of the subject of study is likely to be seen as opposed to CS. This offers an opportunity to contrast with CS. At the same time psychology majors have computing experiences as they have to attend courses in computer supported analysis of empirical studies. This is why we did not ask psychology freshmen, but students in their 3rd year.

We collected 44 biographies of psychology students (39 of them female) and 89 of CS freshmen (17 of them female). We let the participants write down their computer-biography during their usual university courses. The participants wrote about 30-50 minutes and produced between one sentence, one paragraph or one page of text. Despite the varying text length, most biographies are quite short and of the same tenor, particularly the CS freshmen's biographies. The number of the obtained biographies may seem quite high for a qualitative research approach, in fact only a limited number of biographies provided enough data to be useful for the theory-generating process. On the other hand, less useful biographies were not completely useless. Therefore it is difficult in the end to divide the biographies into helpful and useless data. We read and coded them all, so that all influenced the results, some more then others.

However, it might be that the results apply only to a comparison of students majoring in psychology against students majoring in CS, and at one university only. Given the fact that the general direction of results found in the computer biographies fit into the general picture which is discussed in section 1, we think the results can be generalized to a certain extent. In future, it is interesting to widen the scope of collected biographies and take into account students from other universities, subject areas or countries, in order to get a better understanding of the typical biographical processes.

3.4 Data Analysis

Theoretical coding, the text analysis method in grounded theory [4], provides the basis for the data analysis in our project. We will briefly outline this approach here and explain afterwards at which stage of theoretical sampling the results we present in this paper ranges. In theoretical coding, three types of coding may be distinguished: open, axial and selective coding. "In open coding data are 'broken down' analytically and […] from the text, a succession of concepts is developed that may ultimately be used as building blocks for the model." ([4], pp. 271). In axial coding, one of the concepts developed through open coding "is located at the centre and a network of relationships is developed around. […] [W]hat is of particular importance is the development of relationships between the axial categories and the concepts that are related to them in terms of their formal and content aspects." ([4], pp. 271). Finally, in selective coding the main phenomenon of the analysis is established through the axial categories which were elaborated in the earlier coding process. At the beginning of our project we conducted a preliminary study and sampled biographies from CS majors and from German language and literature majors [21], [22]. We did open coding and some rudimental axial coding developing first concepts. With these concepts we then did open coding and especially axial coding, using the coding software MaxQDA. Eventually, we will approach the selective coding stage in further studies.

Nevertheless, it is recommended to adapt the pure theoretical coding to particular questions and circumstances as we did with the coding paradigm suggested by Tiefel (see sections 2.1 and 3.1). For our research project this means that we reconstruct biographies by reconstructing the world-image, the self-image and the habits of an individual with a focus on biographical learning processes.

4. RESULTS

In this section we present our three analysis questions followed by the results.

4.1 Analysis Questions

For data sampling two different groups of students were chosen with the aim to investigate the correlation between *use* and *design* and computing experiences in their biographies. The most important question concerning the CS-affiliated students was, whether they see themselves as designers and what this means. With regard to the CS-unaffiliated students majoring in psychology we were particularly interested in whether they see themselves as users and what consequences this has. The important terms used in the analysis process are summarized in Table 1.

Unaffiliated students	Students who do not pursue a career in CS. Here: psychology students in their 3rd year.
Affiliated students	Students who do pursue a career in CS. Here: CS majors at the beginning of their first semester at the university
Outsiders	Students who perceive themselves outside according to their world-image of CS and computing.
Insiders	Students who perceive themselves inside according to their world-image of CS and computing.
To use/ users	*Use* is a conceptualization of one's own computing experiences as inferior, feeling to be excluded from the real thing, passive and depending on given possibilities. A *User* is a person who sees most of his/her computing as *use*.
To design/ designers	*Design* is a conceptualization of one's own computing experiences as superior, feeling to be participating in the real thing, active and widening given possibilities. A *Designer* is a person who sees most of his/her computing as *design*.
Transition/ transitorial process	A biographical learning process in which the conceptualization of the types and / or of computing experiences change from *use* to *design*. A transition therefore can be seen from *use* to *design*, and from outsider to insider. The transition can be implicit or explicit: The person is aware or unaware of a change.

Table 1: Description of important concepts used in the study

With the special focus on the opposition of *use* and *design* we reconstructed the world-image, the self-image and the habits of the students regarding the impact of computing. Three analysis-questions served us as guidelines in the analysis process:

1. Reconstruction of affiliated and unaffiliated students' biographies: How do students of both groups perceive themselves according to the oppositions of *use* ↔ *design* and *outsider* ↔ *insider?*
2. Reconstruction of affiliated and unaffiliated students' conceptual changes: Are there transitorial processes according to *use* ↔ *design* and *outsider* ↔ *insider*?
3. Reconstruction of pathways into and barriers to CS: What are the pathways into and barriers to CS?

These three questions lead to the final question: Are there any hints in the biographies that can be used to develop didactical interventions in order to support a transition? Note, however, that we do not believe that everybody should be a computer scientist, nor that everybody has the abilities, but given the discussion above, it seems there are many people who are deterred from CS due to misconceptions of their own abilities and of the field.

In the following subsections we discuss the results of the study according to the three research questions.

4.2 Reconstruction of Students' Biographies

In order to answer the first question, we reconstruct for both student groups the self-image, the world-image and habits. With respect to the world-image we reconstruct how students see and understand computing and CS. With respect to the habits we reconstruct typical usage patterns, learning strategies and reactions to problems. With respect to the self-image we reconstruct how students perceive themselves according to the oppositions of *use* ↔ *design* and *outsider* ↔ *insider* according to their world-image. We start the reconstruction of CS-unaffiliated students' biographies by examining their world-image about computers, then turn to their computing habits and their self-image and finally reexamine their world-image.

4.2.1 Unaffiliated students

The unaffiliated students' world-image can be understood from the significance the students assign to a computer. First of all, the computer is seen as a tool for work. This can include leisure activities like web-surfing or e-mailing as well as working with MS-Office applications. Due to the common perception of computers as tools to achieve a goal (mostly to accomplish a homework-task) interest for computers, computer technology and finally CS arises quite late – when computing becomes more important. A biography sample may illustrate the "discovery" of computers as a work tool: *"The principle of 'working' at the computer I discovered many years later, shortly before graduating from high school."* [P13f1977] [3]. In school they choose CS-courses to improve their computer using skills and their understanding of the background. We illustrate again: *"In grade 11 we could attend a computer science class, something I did because the general interest in computers and internet had grown by then. I had to realize, however, that we were taught something completely different as I had expected. We were allowed to learn programming. How extraordinary! Actually I wanted to learn how to use the different kinds of applications, etc."* [P10f1982]. More abstract issues like algorithms or programming are refused and estimated as being absurd, useless or stupid.

These attitudes towards computers are connected with certain habits: Unaffiliated students only learn something about computing if needs be. This learning process is mostly started by external needs (for homework assignments in school or university, or requests in a job). We rarely find text samples describing such learning as extending knowledge. Only seldom we find statements in which exploration of usage possibilities are described and even more rarely these experiences are considered as positive. In summary, we interpret such an approach to learning as *defensive learning* (cp. [18]).

Interestingly, most students describe their computing-skills as being adequate and good enough, but as soon as skills are needed to solve computing-problems, they typically express the following problem solving strategy: *"Nowadays I use it primarily for text processing and internet. I don't know anything about its inner life, in case of problems my boyfriend helps me (he's a computer scientist☺)"* [P03f1980]. *"Even today I only know the bare essentials – and these I have basically copied from others. In case of questions/problems I have to ask a friend who studies computer science! ☺"* [P04f1982]. Unaffiliated students usually ask for help when faced with a problem. These two habits, defensive learning and asking for help, match each other, as both can be qualified as passive and relying on external incentives.

In addition, negative emotions are connected to problem-solving experiences: Unaffiliated students feel stupid and exposed when computing-problems occur. In this context, often mixed emotions are mentioned: *"By now I possess a notebook [...] and I see-saw between enthusiasm about the practical aspects and panic that it will do things I don't understand and where I have to beg for help like a stranded little girl."* [P12f80]. Usually, these experiences and feelings are not attributed to laziness or lack of interest. On the contrary, unaffiliated students are very embarrassed to feel helpless and to be forced to ask for help: *"On my own I tried to install a new program or modem, but I always felt very unconfident. Apart from that, in 80% of the cases, it did not work as described in the manual. Then my prejudices concerning computers, which complicate matters instead of facilitating them, were confirmed, I felt helpless and then always had to ask the boyfriend of my flatmate for help. He was a real 'nerd' who could help me very quickly in most cases - but I was always embarrassed, first because I felt so 'stupid', and then because I barely knew that guy (after all computer nerds are boring most of the time) and I thought I would take advantage of his kindness."* [P07f1979]. We can analyze this example with attribution theory (locus, stability, controllability, see section 2.2): Although the person engaged in problem solving, and e.g. read the manual (learning), her usual (in 80 % of the cases) experience is, that engagement and learning in general do not help (stability). It is not entirely obvious, but it seems the source (locus) of this is due to her capabilities (therefore she feels "helpless", "embarrassed" and "stupid"); note, that in the example the person does not say that e.g. the computer was defect, or the manual wrong. The "80%" however, indicate that sometimes things work, so in addition it seems that it is not controllable. We find this or similar pattern quite often. It seems that as a consequence unaffiliated students try to avoid such situations instead of gaining more competence in order to be able to help

[3] This code refers to a biography and stands for: P=psychology major, I=computer science freshmen, f=female, m=male. First number occurrence means biography's number, the second one is the person's year of birth.

themselves. Again, this is defensive learning, opposed to expansive learning (cp. [18]).

With the knowledge about habits and the self-image let us return to the world-image: The discussed text samples also illustrate how unaffiliated students conceptualize their world-image. Their *asking for help* problem-solving strategy relies on persons who are more competent then they see themselves. In the examples above such persons are sometimes called "nerds", but more often computer scientists. So, while the unaffiliated students perceive themselves as persons with "ordinary computer skills" they distinguish between themselves and others, often computer scientists, who can use the computer in a professional way. These *professional users* are able to cope with problems and other tasks like installing and uninstalling programs, performing administration tasks, and especially solving usage problems (due to malfunction). Usually, such persons are studying the subject CS and/or can be considered as computer scientists. Regarding world-image, the conclusion is that unaffiliated acknowledge something they call computer science, they apparently see it as the domain of solving computing problems. Due to their own lack of such problem solving skills concerning computing, they are not part of this world, they are outside, while professional users and computer scientists are inside. In summary, they just *use* the computer in the above defined meaning. Consequently they perceive themselves as *users* and *outsiders*.

However, the above discussed reconstruction of unaffiliated biographical pattern is a generalized picture. Therefore we want to present the singular exception we found in the computer biographies. The following biography of a psychology major is exceptional as it describes several successful problem-solving experiences and activities in exploring the computer in order to gain deeper understanding: *"1998 first PC, chose it with books like 'PC for dummies'. […] I was the first person in my family who took interest in PCs. Computer games sparked my interest. From the beginning, I used the PC or the notebook as a tool for writing (school, university) and for investigations in the internet or mails and games. To this day, I can't program (I am probably missing a gene or so), but I have the highest respect for people who can do it. Some experiences with viruses deepened the connection between me and my drudge (after it, rebuilding the system – what is a 'partition'? was no problem anymore). Today I work on my PC with programs like Matlab – the same fascination and exaltation like in the old days with my playstation + 'WipeOut'...In general: The possibilities computer development gives are overwhelming and perspectives like AI are just fascinating!"* [P23f1982]. From en external point of view, this student can be considered as a professional user: She demonstrates knowledge of hardware, about different applications and about system internals like partitions. However, she states very clearly that, in her world-image, she is a *user* because she misses the ability and skills to program. She seems to believe that such abilities are not *learnable* because one needs to possess extra skills like a special "gene". Although she is a skilled user, she perceives herself as *user* and *outsider*.

4.2.2 Affiliated Students

The affiliated students' first contact with the computer is initiated by family members (mostly the father or older brother/sister), peers or coincidental individual situations. Games are usually the favorite starting application. The students depict their fascination for the machine in great detail. The passages in the biographies about the first contact with the computer are often described as the beginning of a preordained path. According to the world-image of affiliated students, the computer is a very unusual and special "toy" or fascinating "miracle" a person can spend countless interesting, rewarding and inspiring hours with.

These kinds of attitudes towards the computer are connected with habits: Games, the favorite starting application, sparking interest and fascination for the computer. Up to here the computer-biographies of affiliated and unaffiliated students are quite similar. That is, from an external point of view, the first computing experiences of the affiliated can be described as *use*. However, from then on an opposed development of habits begins. Affiliated students engage in expansive learning: *"I made my first hesitant contact with a computer at the age of three when my parents bought an Amiga. Of course it was exiting to have an occupation, but the diversity a computer offered was more interesting. Of course, at the age of three you are interested in many funny games, this did not change for quite a while. Encouraged by school and internet, I realized how much more was possible. (...) despite horror stories how difficult math is for computer scientists I dared to jump into computer science studies and to learn more about computers."* [I50m1986] Many different applications are successfully explored, like image editing, writing, spreadsheets and internet services. From this "playful" starting point affiliated students develop interest and motivation to learn more about computers. With learning by doing they get to know the computers' functionality [21]. We summarize this as *expansive learning*. In case of computing problems they are intrigued to find a solution. They often test different possibilities until they find a solution. We summarize this as *exploratory problem solving*.

Affiliated students' habits shape their self-image. With their interest in computers and the positive experiences, they have the impression that they can master the computer, an impression which of course makes them confident and provides further motivation for computing. This stands in opposition to the unaffiliated students who feel exposed and stupid. Because of their expansive learning and successful exploratory problem solving they perceive themselves as being part of the computer community, as *insider*.

With the knowledge about habits and the self-image we take up the world-image again. Affiliated perceive the computer as fascinating tool. But this perception means: the computer is a tool they can use and design or reshape. To become a computer scientist means to learn to be a designer (this transitorial process is examined in more detail in the section 4.3). In their world-image computing includes many different activities and possibilities. CS is sometimes seen as a mixture of *design* and *professional use*, while sometimes CS is seen as *design* in opposition to *professional use*.

In general, affiliated are optimistic. Like in the prior subsection we present an unusual biography for this group. Here it is the biography of an affiliated student, which from an external point of view can be seen as a *user*: *"As a child I very often sat in front of a computer and mainly started to edit images. In high school* […], *I had to take a computer science course. It was very disappointing and boring. I learnt how to use Excel and a little HTML. My father is a computer scientist and therefore I know that this was no real computer science. I haven't learnt a programming language until now, and my computer skills are limited to a little image editing. However, abstract thinking attracts me and I wish I could program. I suppose I inherited the enthusiasm for computer sci-*

ence from my father..." [I84f1987]. It seems that her computing skills are inferior to those of the psychology student quoted at the end of the last subsection. Although she seems to be a less skilled user, she perceives herself as (future) *designer* and *insider*.

4.2.3 Summary of the Reconstruction

With respect to their self-image unaffiliated students perceive themselves as *outsiders* and concerning their computing experiences as *users*. Their world-image conceptualizes CS as a field where professionals have extraordinary computing skills and can solve usage problems (*professional use*). Typical habits we found are *defensive learning* and *ask-for-help problem solving*.

With respect to their self-image affiliated students perceive themselves as *insiders* and as *designers* when it comes to their computing experiences. Their world-image conceptualizes CS as a field deeply connected to *design*. For some, d*esign* includes *professional use*, while for others design is opposed to *professional use*. Typical computing habits are expansive learning and exploratory problem solving. Skills in administrating the computer as well as general problem solving strategies are developed, which could explain why affiliated (seen as computer scientists) are perceived as *professional users* by unaffiliated students.

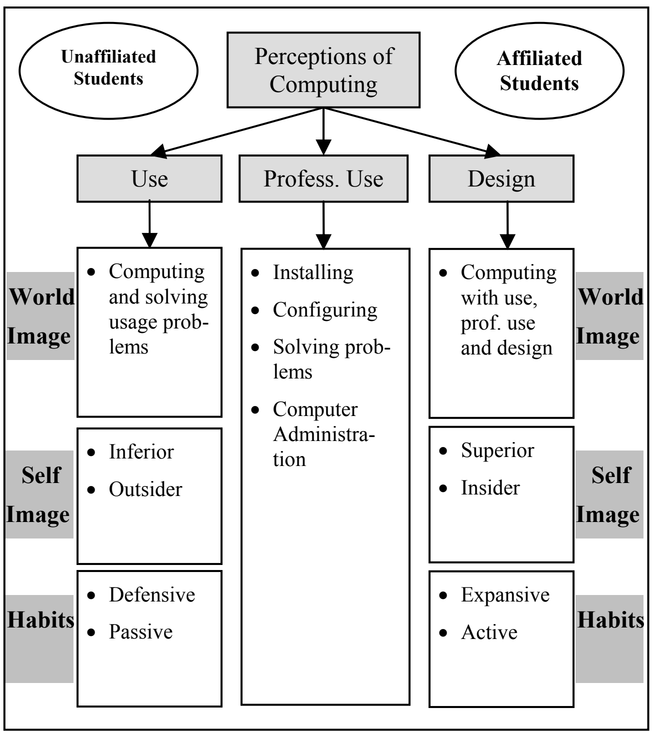

Figure 3: Comparing *use, professional use* and *design* as different perceptions of computing

Unaffiliated as well as affiliated students write about professional use. The former consider *professional use* as problem solving skills, so that usage problems are no more obstacles in the daily interaction with the computer. As a consequence, CS is perceived as the science of professional computer use. Professional use is what they are excluded from. Therefore they believe that CS is linked to computer usage and especially to solving usage problems. For affiliated, *professional use* is what they are capable of. It is seen as part of their computing experiences. Sometimes it is considered as belonging to *design* and as belonging to CS, too. However, more often *professional use* is seen as opposed to *design*, and as being outside CS. Note, however, that we did not find any student who perceived him- or herself as a professional user. In

Figure 3 we summarize the perceptions of computing of the two groups. In this illustration, *professional use* is a perception which is used by the students to distinguish themselves from the opposed group. It seems *professional use* is a barrier for the unaffiliated, and a pathway for the affiliated.

4.3 Reconstruction of Transitorial Processes

The reconstruction of transitorial processes of unaffiliated students is quite short: There are no transitions. With respect to their world-image a transition from *use* to *professional us*e should be expected. Unfortunately the unaffiliated students believe that special skills are needed for this transition. We examine this barrier in more detail in section 4.4.

In contrast, affiliated students start very early to request to change the so far used applications and to design new ones; they start to develop *design* activities. At the beginning, *design* means to write simple web-pages in a WYSIWYG-editor or simple Visual-Basic-applications in Excel or Word. These first steps start the process of a smooth transition from *user* to *designer*. In several biographies we found hints on why the transition is often experienced as a smooth process: *"A one-week java class which covered only the theoretical aspects of OOP raised the feeling in me that the computer would do anything I tell him."* [I28m1986]. *"My interest started in school. With a voluntary class. I was immediately fascinated. I couldn't explain the reason but it diverted me a lot to create something on the PC even if it was just in BASIC."* [I34m1984]. Both persons feel that *design* is another style of computing which widens their possibilities.

With respect to this transitorial process we observed that affiliated students can be conscious or unaware of the transition. Those students who are conscious about it very often believe that they have undergone it already (even if from an outside point of view we would describe their computing experiences as *use*). There is also a small group which describes from the beginning activities which can be classified as *design*.

Very often affiliated students are unaware of a transition and explicitly unaware of the transition in their biography. The analysis of the next biography may illustrate a typical unconscious transition: *"I had my first contact with computers when I was five when my parents bought a computer with 10 MHZ speed and my father taught me how to use Paint. Later, a computer with 100MHZ and Windows95 followed, where I developed an intuitive understanding with learning by doing. Particularly the Windows-games and different fonts intrigued me.*

Then my parents bought a computer with 700MHZ and Windows98. There I got to know Excel and soon discovered the possibility to write small programs with VBA. The most successful creation was a kind of 'Who Wants to Be a Millionaire?' in Excel. By the way, I studied and tried to understand HTML-Code generated by Frontpage. With my first own computer with 1.4GHZ and the discovery of Linux at school, my interest for further possibilities grew. A homepage together with the driving creativity of two friends got me to study PHP together with MySQL, database theory, normal forms, etc. Today we conduct some kind of company

for all multimedia services and offer webdesign for small and medium sized enterprises among other things.
The school indeed didn't influence me much in web programming, teaching me Miranda, Modula-2 and Java, but it increased my interest for a deeper understanding of computers and their structures and functionality." [I52m1988]. In this biography, a transition can be seen, it was experienced unconsciously: The list of computers used is split into a new paragraph precisely at the point which marks the transition from *use* to *design*. Design-oriented activities are perceived as just "further possibilities". Single events like programming the game "Who wants to be a Millionaire?", or conducting a company mark events along the way of widening one's own skills and exploring more computing possibilities.

4.4 Reconstruction of Pathways and Barriers

In this section we discuss pathways and barriers to CS.

Unaffiliated students sometimes explicitly state that they have no interest in learning CS (and apparently they neither have desire to overcome the barrier): *"In school, I consciously decided not to attend computer science classes. Wasn't interested in it.[…] although I managed to deal with it, I wish I could better understand the phenomenon computer."* [P9f83]. The student's interest for computing was low in high school and rose during college. However, in the above given example the person does not mention regretting not to have taken CS class. On the contrary, she does not connect it with her lack of knowledge. All biographies we have collected until now indicate the following: unaffiliated students generally seem to be convinced that computer science class cover "strange" and "very abstract" topics, and though it appears that future professionals (computer scientists) benefit from these topics unaffiliated students will not. Other studies have already revealed this problem and teachers tried to take care of these aspects by teaching use skills (like e. g. Word or Excel) or computer science topics in a more applied way (e.g. teaching HTML programming instead of Pascal or Java). The problem is that these approaches do not necessarily change the world- and self-image of the students. Teaching Word or Excel helps to use Word or Excel in a better way, but as soon as computer problems occur, the habits of the unaffiliated occur again: *ask-for-help problem solving* and *defensive learning*.

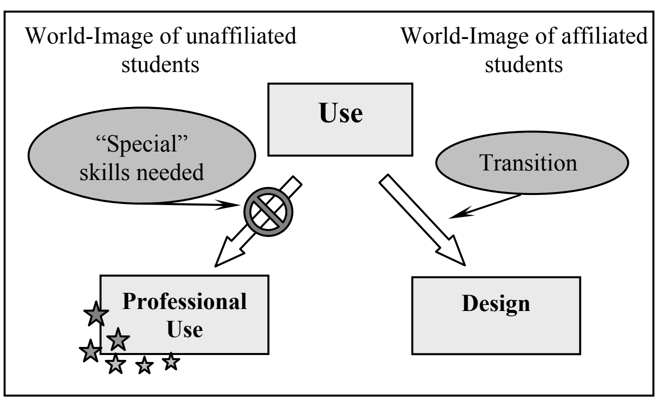

Figure 4: Worldview and transition process of unaffiliated and affiliated students

These habits form a barrier to CS. Moreover, the self-image makes unaffiliated students believe that they can not learn CS. They are convinced they can not become an insider. As outsiders they speak about computer's functionality in terms of "mystery" and "unresolved secret". They have absolutely no idea how computers work, but, and this is the crucial point, they believe they will never learn it, because to understand CS a person has to be "different"! To draw an illustrating comparison: Similar to muggles in the world of Harry Potter [32] who can never learn to do magic, "normal people" in the world-image of outsiders can never learn CS and become insiders. In addition, unaffiliated students conceptualize a transition from being outside CS to being inside not as a transition from *use* to *design,* but as a development from *use* to *professional use*. This world-image of CS is another barrier (see Figure 4).

In contrast, the affiliated students perceive themselves as being insiders "by birth", see e. g. the following example: *"It was the year 1995, in the tender age of 9 when I got my first PC. Already when I saw the metallic shell and the black screen, I knew the computer would become my element. No sooner said than done. Not even one year later I started with Q-Basic, and W3.11* [Windows 3.11] *was not an issue anymore."* [I14m1986]. The student does not describe what happened in this first year of computer usage. But he states clearly that at the end he felt as a *designer* and an *insider* from the beginning onwards.

The following table (Table 2) summarizes and contrasts barriers and pathways into the field. They are interpreted within the theoretical background outlined in section 2.2. Therefore we refer to attribution theory, conceptual change approaches and to a certain learning style we found.

Barriers for unaffiliated students	Pathways of affiliated students
attribution	
• Perceiving one's own skills as insufficient for learning CS • Perceiving CS as field which can not be learned like other subjects	• Perceiving one's own skills as sufficient for learning CS
Habits ("learning mode")	
• Defensive learning • "Ask for help" problem solving	• Expansive learning • Exploratory problem solving
Preconceptions	
• Perceiving CS as domain of professional use of computers • Problems in understanding the role of programming: what programming is, why it is needed and how it is useful	• Do not see boundaries between use, design and professional use. • If boundaries are seen, they are estimated as unimportant or easy to cross • Experiencing programming as another type of interacting with the computer

Table 2: Summary of barriers and pathways

Analyzing barriers with attributional theory, the picture is seen as follows: Outsiders have developed a negative attributional style as they feel not being able to cope with errors due to lack of ability. However, aiming at supporting self-confidence still is not enough due to the overall conception of CS as being the subject of professional computer usage. In addition, persons with such conceptions would have to change them. Hence, conceptual change approaches are needed, too. Analyzing the attributional style of unaffiliated with regard to locus, stability and control (see section 2.2) reveals that such attributions are hard to change: Lack of ability is an internal, stable and uncontrollable reason for avoiding engagement in CS. Such attribution towards computing must be judged even as learned helplessness. Attribution training for people with such a negative attributional style would need to work at all three dimensions: teachers would have to redirect the focus to external reasons like task difficulty, demonstrate that the person is able to solve at least some usage problems (stability) and can become better due to effort and training (control). In addition, teachers should aim at changing the defensive learning into expansive learning [18]. Aiming at conceptual change, the teacher needs to take into account that unaffiliated students see computer scientists as *professional users*. They want to make a transition from *use* to *professional use*, and not from *use* to *design*.

5. Conclusion

At the end of section 3.1 we stated our research questions (Q). See Figure 3, Figure 4, and Table 2 for a summary. We now give some pointed answers (A) to the questions:

Q: Are there really persons and opposed groups of persons who conceptualize their computing experiences in alignment to the notions of *use* and *design*? **A:** Yes there are, we can roughly categorize unaffiliated as *user*, and affiliated as *designer*.

Q: Moreover, what world-images, self-images and habits according to CS are connected to these opposed groups? **A:** We found *defensive* and *expansive learning* (habits), the notion of *professional use* as barrier or boundary (world-image), and self-images as unsure outsider or confident insider.

Q: Assuming that, at a certain point, freshmen have to think about themselves as *users*, how do they make the transition form *use* to *design*? **A:** Transitions are hard to find: While the unaffiliated do not make a transition, the affiliated do not perceive a transition.

Q: What is the nature of this transitorial process, making computing a barrier for some, while serving as intriguing clue for others? **A:** Maybe the transition can be seen as a threshold (cp. [13]): The transition seems insurmountable when not undergone, but afterwards one can hardly remember to have ever seen the world and oneself differently.

What is the significance of these results? The model of transition, including aspects of attribution theory, conceptual change and the notion of different pathways into CS, leads to some conclusions: Firstly, it is important to aim at combining aspects of *professional use* with *design*, so that students see how *design* can support *use*. Educational concepts should intertwine introduction to CS (e.g. learning programming - a *design* activity) with learning professional use. Given the above discussed prerequisites for conceptual change, a major problem is to teach another world-image of CS. Undoubtedly, a lack of correct information may be an issue too, but trying to explain to freshmen and unaffiliated students what CS "really" is would not help – a world-image is deeply connected with self-image and habits. In addition, in some cases reattribution training is needed.

Finally, it should be noted that a general understanding of computer science makes it possible to understand the information-technological world and particularly *design*. This does not mean that each person should study CS, but everybody should be able to experience today's "computerized" world as *professional user* or *designer*, but always as *insider*.

6. REFERENCES

[1] Adams, J. and Bauer, V. and Baichoo, S. (2003). *An expanding pipeline: gender in Mauritius*. In Proceedings of the 34th SIGCSE technical symposium on Computer science education, 59-63.

[2] Beaubouef, T. (2003). *Why computer science students need language.* In ACM SIGCSE Bull. 35(4): 51–54.

[3] Beaubouef, T. and Mason, J. (2005). *Why the high attrition rate for computer science students: some thoughts and observations.* In ACM SIGCSE Bull. 37(2): 103–106.

[4] Böhm, A. (2004). *Theoretical Coding: Text Analysis in Grounded Theory.* In Flick, U., von Kardoff, E. and Steinke, I. (edit.): A Companion to Qualitative Research. Sage Publications Ltd, 270 -275.

[5] Byrne, P. and Lyons, G. (2001). *The effect of student attributes on success in programming.* In Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 49–52.

[6] Camp, T. (2002). *The incredible shrinking pipeline*. In ACM SIGCSE Bull. 34(2): 129–134.

[7] Carter, L. (2006). *Why students with an apparent aptitude for computer science don't choose to major in computer science.* In Proceedings of the 37th SIGCSE technical symposium on Computer science education, 27–31.

[8] Cohoon, J. McGrath and Aspray, W. (2006). *A Critical Review of the Research on Women's Participation in Postsecondary Computing Education.* In Cohoon, J. McGrath and Aspray, W. (edit.): Women and Information Technology: Research on Underrepresentation. MIT Press, 137-180.

[9] Cohoon, J. McGrath and Aspray, W. (2006). *Women and Information Technology: Research on Underrepresentation.* MIT Press.

[10] Computer-Biographies. Webpage of the research project: https://www.inf.fu-berlin.de/w/DDI/ComputerBiographies.

[11] Crutzen, C. K. M. (2000). *Interaction, a world of differences. A vision on informatics from the perspective of gender studies,* http://www.cecile-crutzen.de/Downloads/ 2000-Dissertatie-English-Summary.pdf, Open Universiteit Nederland Heerlen.

[12] Donovan, M. S. (2005). *How students learn. History, mathematics, and science in the classroom*. Washington, DC: National Academy Press.

[13] Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., and Zander, C. (2006). *Putting threshold concepts into context in computer science education*. In Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 103-107.

[14] Faulstich-Wieland, H. (2004). *Mädchen und Naturwissenschaften in der Schule, Expertise für das Landesinstitut für Lehrerbildung und Schulentwicklung*. Hamburg.

[15] Goode, J., Estrella, R., Margolis, J. (2006). *Lost in translation: Gender and High School Computer Science.* In Cohoon, J. McGrath and Aspray, W. (edit.): Women and Information Technology: Research on Underrepresentation. MIT Press, 89-114.

[16] Graham, S. (1991). *A review of attribution theory in achievement contexts.* In Educational Psychology Review 3(1): 5-39.

[17] Greening, T. (1998). *Computer science: through the eyes of potential students.* In Proceedings of the 3rd Australasian Conference on Computer Science Education, ACSE, 145-154.

[18] Grotlueschen, A. and Root, M. (2003). *Klaus Holzkamp's Learning Theory & School Criticism*. www.erzwiss.uni-hamburg.de/personal/grotlueschen/2004/pdf/AG_Holzkamp_englisch.pdf

[19] Guzdial, M. and Soloway, E. (2002). *Teaching the Nintendo generation to program*. In Commun. ACM 45(4): 17-21.

[20] Heublein, U., Schmelzer, R., Sommer, D. et. al. (2005). *Studienabbruchstudie 2005.* HIS: Hochschul-Informations-System (The Higher Education Information System), http://www.bmbf.de/pub/studienabbruchstudie_2005.pdf.

[21] Knobelsdorf, M. and Schulte, C. (2005). *Computer Biographies - A Biographical Research Perspective on Computer Usage and Attitudes Toward Informatics.* In Proceedings of the 5th Annual Finnish / Baltic Sea Conference on Computer Science Education, Koli.

[22] Knobelsdorf, M. and Schulte, C. (2007). *Das informatische Weltbild von Studierenden.* In Proceedings of the 12th GI-Fachtagung Informatik und Schule, INFOS 2007.

[23] Ladd, B. C. (2006). *The curse of Monkey Island: holding the attention of students weaned on computer games.* J. In Comput. Small Coll. 21(6): 162-174.

[24] Maaß, S. and Wiesner, H. (2006). *Programmieren, Mathe und ein bisschen Hardware...Wen lockt dies Bild der Informatik?* In Informatik Spektrum 29(1): 125-132.

[25] Margolis, J. and Fisher, A. (2002). *Unlocking the Clubhouse: Women in Computing.* Cambridge, MIT Press.

[26] Marotzki, W. (2004). *Qualitative Biographical Research.* In Flick, U., von Kardoff, E. and Steinke, I. (edit.): A Companion to Qualitative Research. Sage Publications Ltd, 101-107.

[27] Martin, C. D. (2004) *Draw a computer scientist.* In Proceedings of the 9th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 11-12.

[28] McGuffee, J.W. (2000) *Defining Computer Science.* In ACM SIGCSE Bull. 32(2): 74-76.

[29] Moorman, P. and Johnson, E. (2003). *Still a stranger here: attitudes among secondary school students towards computer science* In Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 193-197.

[30] Murphy, L., Richards, B., McCauley, R., Morrison, B. B., Westbrook, S., and Fossum, T. (2006). *Women catch up: gender differences in learning programming concepts.* In Proceedings of the 37th SIGCSE technical symposium on Computer science education, 17-21.

[31] Romeike, R. and Schwill, A. (2005). *Das Studium könnte zu schwierig für mich sein - Langzeitbefragung zur Studienwahl Informatik am Institut für Informatik der Universität Potsdam.* Institut für Informatik der Universität Potsdam.

[32] Rowling, J. K. (1999). *Harry Potter and the Philosopher's Stone.* Bloomsbury Publishing.

[33] Schinzel, B. (1993). *Zur Gleichstellung von Frauen und Männern in der Informatik: Curriculare Vorschläge.* In Infotech (5)4.

[34] Sinatra, G. M. (2005). *The "Warming Trend" in Conceptual Change Research: The Legacy of Paul R. Pintrich.* In EDUCATIONAL PSYCHOLOGIST 40(2): 107–115.

[35] Statistisches Bundesamt Deutschland (Federal Statistical Office Germany). (2005). *Hochschulstandort Deutschland 2005.* http://www.destatis.de/presse/deutsch/pk/2005/aktuelle_ergebnisse.pdf.

[36] Teague, J. (2002). *Women in computing: what brings them to it, what keeps them in it?* In ACM SIGCSE Bull. 34(2): 147–158.

[37] Tiefel, S. (2005). *Coding in terms of Grounded Theory: modifying coding guidelines for the analysis of biographical learning within a theoretical framework of learning and education.* In ZBBS: Zeitschrift für qualitative Bildungs-, Beratungs- und Sozialforschung 6(1): 65-84.

[38] Turkle, S. (2005). *The second Self: Computers and the Human Spirit.* The MIT Press.

[39] Turner, E. H. and Turner, R. M. (2005). *Teaching entering students to think like computer scientists.* In Proceedings of the 36th SIGCSE technical symposium on Computer science education, 307-311.

[40] Vegso, J. (2005). *Interest in cs as a major drop among incoming freshmen.* In Computing Research News 17(3): 126-140.

[41] Wiedenbeck, S. (2005). *Factors affecting the success of non-majors in learning to program.* In Proceedings of the 2005 international Workshop on Computing Education Research, ICER, 13-24.

[42] Wilson, B. C. and Shrock, S. (2001). *Contributing to success in an introductory computer science course: a study of twelve factors.* In Proceedings of the 32th SIGCSE technical symposium on Computer science education, 184-188.