Exploring self-efficacy in end-user programming
– a feminist approach

Lena Palmquist

Licentiate Thesis, August 2014

Department of Computing Science
Umeå University
SE-901 87 Umeå
Sweden

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

lenap@cs.umu.se

ISBN: 978-91-7601-103-4
ISSN: 0348-0542
UMINF 14.17

Electronic version available at http://umu.diva-portal.org/
Printed by: Print & Media, Umeå University
Umeå, Sweden 2014

To Victor

Abstract

Digital services and devices are today more spread than ever, forming a basis for new innovations, even among ordinary people. And yet, producers of such services and devices are mostly men with programming skills. Women's participation in development and design of digital products is thus not yet as influential as that of men.

An approach to this situation is to offer web-based environments for end-user development where people with no programming experiences have the opportunity to develop their own smartphone applications. The SATIN project, a collaboration between universities and IT-companies, has taken such an approach, with a focus on supporting female end-users. This project has been serving as a case in this research with the purpose of exploring and understanding end-user programming related to self-efficacy and female strategies.

Experiences from being a member of the SATIN project are accounted for as well as results from qualitative observation studies capturing subjects' reactions to the system. In the first set of observations, 9 subjects tested a mock-up version of the so-called SATIN editor, where the actual app building takes place. Later on a second set of observations with 11 subjects focused on how to support computer self-efficacy and end-user programming strategies that women prefer to a higher degree than men.

Observations indicate that the women where as positive to making use of the editor as the men. The test subjects also showed signs of motivation as well as creativity while exploring the system. An observation related to design aspects of the system was that the quality of the components that form the smartphone apps seems to be crucial if the system is expected to truly support strategies that women request.

Supporting women's own perceptions of self-efficacy related to developing computer-based systems is challenging, still indications of acceptance and enthusiasm for the system were observed.

From a design perspective, using strategies and self-efficacy sources as an evaluation framework in the development process shows potential for

improved design, and not only when designing for female users, but for diverse groups of users, hopefully paving the way for a more diverse community of producers of computer-based products.

Acknowledgements

I want to take this opportunity to thank some people who have meant a lot to me while working with this thesis.

First of all I want to thank my supervisor Anna Croon Fors for taking me under her wings. You probably have no idea how much your guidance and support have meant to me. A million thanks!

The next person who has been absolutely essential for me to finish this thesis is my co-supervisor Anders Broberg. You asked me a question with perfect timing for me. Thank you for believing that I could carry this through, and for valuable comments of an earlier draft of this thesis.

All members of the SATIN project who provided me with a most interesting case to use in this research, and especially Paula Wennberg and the design team, many thanks!

Erik Stolterman and Jürgen Börstler have both made an effort to get me going in the past. Thank you both for that.

Lars-Erik Janlert introduced me to the area of Human-Computer Interaction many years ago, and has also given me most valuable comments on a draft of the present work, thank you!

My Frigga friends—Britt-Marie and Gudrun—whom I first met at the former Centre for Women's Studies in Umeå I want to thank for giving me all kinds of support, on a personal as well as a professional level. You mean a lot to me.

My colleagues, especially Marie Nordström and Lena Kallin Westin, I thank for cooperation, community, "therapy" and encouragements.

Thanks go to my sister Agneta Halvares Palmqvist for all the roles you play in my life, such as a willing listener, an English teacher, sometimes a chauffeur, and always a friend.

My parents, Gunnel and Kjell, I thank for helping me with practical matters in everyday life whenever I have needed it, in recent years especially caring for my cats while I have been away at project meetings in Luleå. I also appreciate that you have never stopped believing that I would

finally do this. You are both amazing and great role models that I could never live up to.

Crystal, thank you so much for willingly and at very short notice posing for and adorning the cover of this thesis.

Last but not least I want to thank my son Victor who, before growing up and moving out, helped me limit my working hours and thus probably also helped me keep my health. Thank you also for being a good example to me in many situations, lately for being my own living English dictionary, for shooting the cover picture, for all the moments we have shared just laughing with each other, and by simply being there and embellishing my life. Many years ago you also gave me the best possible encouragement for me to never give up finishing my thesis by saying: *Mummy, you're so smart!*

Financial support has been provided by EU structural funds under grant number 146244, and by the Department of Computing Science, Umeå University.

Umeå in June 2014
Lena Palmquist

Table of contents

1

Past and Present

When we look at the number of students applying for computer science programmes in Sweden, the proportion of females is depressingly low. Even programmes with, what is commonly considered, a more female touch, such as interaction design programmes, almost always have a larger number of male students than female students. Several campaigns with the aim of increasing the number of female students have been conducted not only in Sweden, but in most of the Western world, and sometimes with positive results, but seldom with a permanent effect. When the campaigns are over, the number of female students seems to decrease again to the level it was before the campaign.

If the ratio between male and female students among computer science students is unbalanced in this way, an inevitable consequence is that the proportion of females among practitioners is also unbalanced. It is not an overstatement to say that the IT industry is male dominated.

The facts here stated contribute to the idea that technology in general and computer-based technology in particular is considered to be male.

Numbers are not the only sign of technology considered to be male; people in general tend to look at some technical phenomena as male and others as female. Apart from the label of male and female, we also tend to actually label the technological phenomena we consider as male as *real technology*, whereas technological phenomena considered female are usually not labelled as technology at all (Wajcman, 1991:137). This becomes an apparent sign of society regarding technology as male. Something that females engage in cannot by definition be technology, since technology is

by definition considered a male phenomenon according to this kind of view. Apart from women often feeling that technology is not their domain, this view also affects men that do not feel comfortable using technology and IT, since they may be accused of not showing signs of being male.

Another characteristic of the IT industry is that the knowledge behind computer-based products is highly valued in our society. Signs of this valuation are the relatively high salaries that people receive in this branch of industry, compared to people in other careers not considered as technical, but requiring the same amount of training (SCB, 2012).

Related to the high value of IT knowledge is the notion of power. Only the ones having this knowledge hold the power to really understand and construct the technology behind computer-based products, products that are today more widely spread than ever in society. In fact in the Western world it is almost impossible to cope without a computer and an Internet connection for handling many everyday tasks. The younger generation can hardly even endure a life without being "connected", and even though many elderly are quite concerned that everything is handled through some web-based service today, there are also many people in their eighties, and even older, that more or less willingly have embraced this technology in order to handle their everyday lives. An existence *without* the Internet and all the computer-based services and information available through it is hard to adjust to today when we have been so spoilt by the fact that "everything is out there". We can conclude that computers appear in all kinds of situations and in many different shapes and forms, the Internet is everywhere, and everything is available through the Internet. Those with the knowledge to deliver the applications and services supporting these growing needs that are now highly established are thus entrusted with great responsibilities. At the same time ordinary people do not possess the knowledge to judge whether to trust the services they use, or to avoid them because of unexpected dangers. This tendency is discussed almost daily in newspapers in Sweden, and probably worldwide.

This situation is not easy to turn around. Ordinary people are so to speak in the hands of IT-experts. People also depend on experts in other fields, such as medicine, engineering, economy and other areas. But maybe

IT is particularly difficult to understand for people in general. Even people responsible for computer support sometimes just sigh exhaustedly and admit that they do not understand why something went wrong in a certain application, printer, web service, or other computer-related phenomenon. IT products today are simply very complex.

We have a situation where a desired development is that ordinary people—of course including women—are provided with tools that support a more independent shaping of their own technological needs. One concept occurring now and then is to have systems supporting so-called end-users (with no programming knowledge) in making their own applications. There are examples of more general systems in this area, such as the App Builder[1], but also systems with a more narrow purpose such as building computer games for children, for example Kodu[2]. A final example is mashup-programming environments where users combine data, such as news, of their own choice.

The research presented in this thesis is directed to this kind of end-user programming, and especially for end-users to program their own smartphone applications, which is described in the following section.

1.1 The SATIN Project

The SATIN project[3] is an example of an attempt to support people without programming knowledge, but who still want to take charge of realizing their own application ideas (Bergvall-Kåreborn et al, 2012). One could say that in end-user programming anyone should be able to make a computer program or application even without the slightest knowledge in traditional programming. In the present case—the SATIN 2 project—we want to provide users with a product that supports making applications or apps for smartphones. The goal for the project is thus to develop a cohesive platform with a number of services. A platform in this context means a web-based portal where the main part is what we call an editor with a large number of building blocks or components, with basic functionality, that end-users choose from and assemble into smartphone apps that are then downloaded into their personal smartphones. Our

[1] http://www.theappbuilder.com/
[2] http://www.kodugamelab.com/
[3] http://www.satinproject.eu (SATIN: System för Användardriven TjänsteINnovation, English: System for user-driven service innovation)

ambition was also to provide a number of services in this platform for distributing apps to friends or the public, charging for the apps if desired, requesting components lacking in the editor from component builders, along with a number of additional services that would support users of the SATIN platform.

A first phase of the project—SATIN 1—started in 2008 where the possibilities for such a platform were investigated, a proof of concept. A working system consisting of components that could be assembled in a Lego-style way was built as a first explorative step, see figure 1.1. There were however ambitions to take the project one or more steps further regarding the design and the technical implementation, as well as regarding business aspects of how to develop an eco system based on these ideas. This further development has been the focus for the next phase of the project, of SATIN 2.

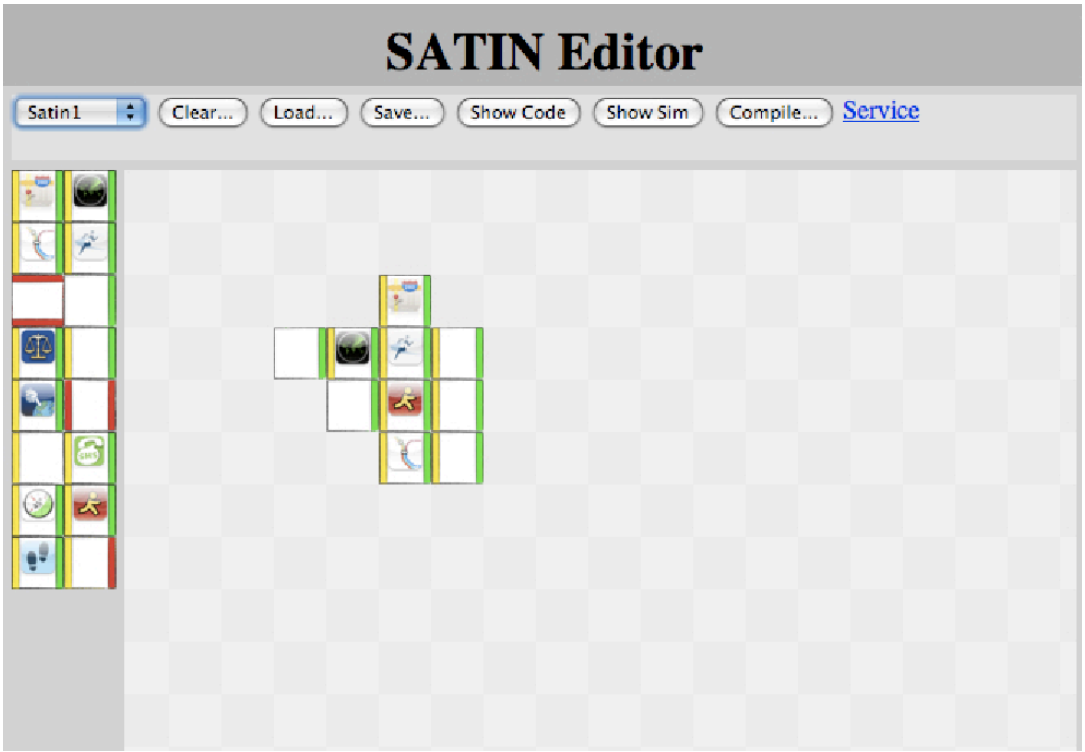

Figure 1.1: The first prototype of the SATIN editor—an important result of the SATIN 1 project—designed and implemented by Anders Broberg, Department of Computing Science, Umeå University. See Appendix F for a larger version.

The financial support that the SATIN project has received has been administered by the Swedish Agency for Economic and Regional Growth

(Tillväxtverket[4]), so there is a clearly stated goal for the project to be a kind of basis or inspiration for business ideas to grow out of the project, and especially in the most northern parts of Sweden. A characteristic of the northernmost parts of Scandinavia is a tendency for women (in particular) to leave rural areas and move to nearby cities, or even to move further south (Dahlström, 1996). Therefore, to inspire and pave the way for people in general, and women in particular, to start new businesses independent of their location, the SATIN environment could hopefully be a support for business ventures. There is thus a clear project goal of designing the system with the specific purpose to satisfy the demands of female users.

Members of the project group are people working at universities, and at small and medium-sized enterprises (SMEs). The representatives of the SMEs have been responsible for finding technically sustainable design solutions and also for implementing the system. Most of the members from universities have engaged in the conceptual design of the system, but some were also involved in technical aspects of the system. A third group has been responsible for investigating business aspects for the system. Finally a group with representatives from the other three groups has had the responsibility for making sure that gender and diversity aspects have been considered in the project, and to produce a web site where tools, methods, and models used and discussed in the project have been gathered.

1.2 Gender and Interaction Design

Having the ambition to design with the purpose of meeting the demands of women presents us with numerous challenges. There are many opinions for why this situation exists in the first place. Some claim that it is inherent in women not to engage in technological phenomena, at least not engage in the *production* of technology. One challenge to engage in is thus to sort out this aspect of gender[5] and technology. A question to ask is: If we do not believe that women are born *without* the ability to become interested in producing technological products, what are the explanations

[4] http://www.tillvaxtverket.se

[5] I have chosen to use "gender" and not "sex" in this work, in accordance with the literature referenced in this thesis.

for the underrepresentation of women in technology-producing and IT-producing industry? These aspects are reflected upon in chapter 3.

A second challenge is to determine the aspects in the actual design that could make a difference for attracting females. This challenge actually has a number of sub-challenges, and they are related to what aspects to look for, how to determine the aspects in order to succeed, and how to transform such knowledge into an actual design of interactive products.

Explanations for why women are represented to a lesser degree within the IT industry are likely to be much more complex than just finding a set of aspects to work with. This complexity needs to be explored, which is addressed in chapter 3.

1.3 Research Question and Purpose

A starting point for setting the goal for the present research is taking the overarching goal for the entire SATIN project. In the project description and application we can conclude that the goal is primarily to design and implement an app developing system that is understood and accessible for end-users who are inexperienced in the art of programming. This goal in itself is far too general and unclear to be dealt with directly.

Based on the proportion of women engaged in technology-producing activities in the Western part of the world, we could ask if there are any hands-on suggestions as to how to increase the number of females engaging in such activities. A study giving evidence of such a successful result would have to be a long-term study, which is not within the scope of this research. We thus have to settle with indications of having taken promising steps for that kind of change. Such a study would also demand a system that is officially launched and used to a significantly high degree, which is not yet the case for the product in the present project.

In order to successfully make a change, the approach also needs to be based on relevant theories and former studies relevant for the challenge at hand.

Taking what is stated above into consideration, the research question is set to be: How do we design with the purpose of including and motivating and meeting the demands of end-user programmers, from a diversity perspective, and particularly women?

In this thesis I have the ambition to share my experiences of investigating certain design suggestions, where the goal is to meet the demands of female end-users in particular. These experiences can be presented as a number of recommendations, guidelines, as well as challenges that can be used and considered when designing systems supporting end-user programming, and especially aspects relevant for females in this category of users.

What are the incentives for pursuing this research then? Based on the imbalance among the genders within the information technology (IT) sector, and the stereotypical expectations related to who are seen as competent and as experts within the IT sector, all efforts trying to make a difference are commendable.

Software and computer-based services, specifically end-user applications (software for building apps, spreadsheets, image editing programs, etc.) should be designed in such a way that marginalized groups—e.g. women—feel motivated to use them and feel included as users. This purpose motivates me to pursue the present study.

1.4 Thesis Overview

In chapter 2 I account for the methods and methodological considerations behind this research, especially the case study approach, and why a feminist approach is justified.

Feminist theory and its connections to the present work are accounted for in chapter 3. I also account for a number of dilemmas and positions taken, and justify them using the theories.

Chapter 4 is a literature study of gender and end-user programming, which the present research is based on. Its main focus is gender aspects of end-user programming and particularly how the end-user programming activity is perceived. An important focus in this chapter is self-efficacy theory and how it affects users utilizing systems with programming features suitable for end-users. Another aspect in focus is the kind of strategies that women tend to rely on in end-user programming.

In chapter 5 I give a rich description of the SATIN 2 project based on the data collection activities accounted for in chapter 2.

Chapter 6 reports the insights, experiences and remaining challenges that these activities have resulted in. It also summarizes and discusses the results from the research.

In the final chapter I account for implications for future work.

2

Methodological Considerations

This chapter accounts for methodological considerations for the present study. In this research a case study approach was chosen, with the purpose of exploring and identifying opportunities for supporting a design that includes, motivates and meets the demands of end-user programmers, with a focus on female users. Motives and characteristics for choosing a case study approach are accounted for in section 2.1

2.1 Case Studies

The main characteristic of case studies is to study a specific phenomenon, which can be a specific group, a person, a specific process, a specific program, an organization, a community or the like. In a case study the context where the phenomenon occurs is also taken into account. The context and the phenomenon studied are studied holistically, and consequently the focus is not on a single variable. The perception is that it might not even make sense to study a single variable, since it might change the conditions for this phenomenon, without really explaining it. A holistic perspective is thus taken. A situation where people interact with a computer-based system is characterized by many parameters or variables affecting the situation, which is in line with a case study approach. One purpose of case study research may be to explore and understand the principles underlying a particular phenomenon, to gain a greater understanding about the case being studied. Robert K. Yin, one of the most prominent persons in the area of case studies, puts it this way:

> A case study is an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident (Yin, 2009:18)

In summary, one can say that case studies are characterized as being particularistic, descriptive, heuristic, and that they are based on a variety of information sources (Merriam, 1994:25-27).

A case study is thus an appropriate research approach when studying a particular phenomenon, such as a system for developing apps. It is also a suitable approach in an exploratory study where there are no particular preconceptions, in contrast to having set up a hypothesis in advance.

Case studies are not defined by the methods used for data collection. Any data collection technique could be used, and one often relies on several techniques in case studies (Merriam, 1994). In fact, as many sources of information as possible, are recommended in case studies. In design projects of interactive systems, observations of test subjects are often combined with questionnaires and interviews, matching well the case study approach. Case studies are also suitable for studying processes, which is crucial for succeeding with a project where the goal is to design an interactive system.

The case study characteristics mentioned above all support using this approach in the current project. The approach matches well to design projects in general, and to the purpose of the specific design project addressed in this research. It is thus an appropriate approach while exploring the design process and more specifically the aspects to consider for supporting end-users of app developing systems, especially gender and diversity aspects.

In the present study, there is a diverse set of sources of information. The sources are experiences of quarterly meetings with all project members, phone meetings with the design team, phone meetings with the Gender and Diversity group, literature studies of related research, seminars and discussions related to this research, workshops where the system has been used, design meetings, and finally two rounds of test sessions. Since I am a member of the project team, the case is studied from within, rather than studied by someone from outside of the project.

It has also meant that I have had easy access to activities, documents, and other data of importance for gaining a greater understanding for the design process in general, and designing for diversity and end-user programming in particular. This means that some of the activities that I consider here are activities that I have initiated myself. These activities are two test rounds where I have observed test subjects using or exploring the system, and seminars based on literature studies. In the tests my interest has been on the experiences expressed in short interviews. Also, the actions that test subjects performed during the test sessions, and what they might indicate caught my attention. It is not always easy for users to be aware of what their feelings and experiences mean, and why they occur. The interpretation of what happened during observations is accounted for in chapter 6. The purpose of these tests is not only to hopefully confirm that the current design works, but also to pick up the signals that the users' behaviour, or even lack of behaviour, gives concerning how to improve the design. These signals guide a designer or a design team of how to proceed in order to reach an improved and more relevant design. Such ideas for improvement arise when users are confronted with the system, whether it is a fully working system or a prototype with limited functionality. While hearing comments and questions about the system, and seeing how the system is used, an understanding develops concerning strengths and challenges of the system.

2.2 The SATIN Project Case

The SATIN 2 project[6] has been an obvious choice for this study, since I joined this particular project in the beginning of 2011.

As mentioned in chapter 1, the first phase of the SATIN project started in 2008, with the purpose to investigate possibilities and challenges for designing a system to support user-driven application development. The results from the first phase were promising. Thus a second phase—SATIN 2—started with the ambition to manufacture a working system within the project time period. The project is a design project, where the goal is a system where users develop their own mobile services, based on a large set of components to be assembled. This kind of system is fairly new

[6] http://www.satinproject.eu/

to people in general. Even though there are a couple of similar systems, the degree of difficulty is much higher in those systems than what we aspire to accomplish in the SATIN 2 project. Existing users of those systems are not typical of the user category we want to address. The target group of our system is thus much broader. Anyone with a fairly common usage rate when it comes to computers and smartphones is a potential user of the system. But these people are seldom involved in using existing systems for developing smartphone applications. This means that there is no evident user group available today where test subjects for our project could be found, i.e. users who already have some knowledge about such systems, and still represent users with no prior programming knowledge. Instead we have had to find test subjects either with no prior knowledge of such systems, or test subjects who were more or less knowledgeable in programming. The focus of the current case study is consequently not situations that exist today, where people use a system based on their own initiative. Instead the focus is the SATIN 2 project in itself with its goals and how it is carried out related to these specific goals, representing the case.

The project members are divided into three groups, as mentioned earlier, representing three separate foci: technical considerations, design considerations and business considerations. In the present research the main focus has been on design considerations, motivated by my own participation in the design team. Technical and business considerations have had an impact on this study only through the presentations given in the quarterly meetings with all project members.

The greatest challenge of the SATIN 2 project is to find ways to simplify the rather complex task of "writing a program" into an activity of assembling components, and to make this activity understandable, enjoyable, and meaningful even for people not used to programming, and probably with the conception that ordinary programming is far too complex for them to grasp. Here the case study has the potential to bring forth aspects of the design process as well as users' experiences of being exposed to an entirely new kind of system and task, which was the case for most of the test subjects. An important aspect of this transformation of the programming activities for end-users is the level of abstraction—

manifested in the level of detail in components—to choose. Most IT design projects have the character of finding functionality and how to represent it in the system, but here we need to investigate what level to aim for in order to facilitate for end-users and at the same time to support flexibility.

2.3 Data Collection

In order to understand what the opportunities are to support end-users' motivation and inclusion, I have participated in a number of activities within the SATIN 2 project, that are accounted for in the following sections.

2.3.1 Quarterly project meetings

Since I joined the project in early 2011, I have participated in eight quarterly 2-day meetings for all participants in the project. Usually between 15 and 20 people were present in those meetings. The purpose has been to get accounts for the project progress from all project groups, since the last meeting, to learn what has happened, and to discuss if we were on the right track. A second purpose has been to plan for the next quarter related to resources and time available for project members. Members of the design group have shared information related to design in general and design for the support of female strategies in end-user programming situations in particular during several of these meetings. In the October meeting in 2011 I accounted for my findings from the literature studies I had conducted until then. The concepts of "tinkering" and "self-efficacy" that I focused on in that account seemed to be an eye-opener for members of the project. Since then these concepts have been a recurring theme during almost all of the meetings in the project, in the group meetings as well as in project meetings.

2.3.2 Design group meetings

An important part of the design group activities has been weekly or biweekly phone or computer conference meetings. Since I joined in, we have had about 30 meetings altogether. We have followed an agenda (see appendix A) for recurring topics that we established as crucial for the design part of the project. Among these topics were quality criteria for a successful design project, how to verify and test that these criteria were

met, and looking at design languages in general and quality criteria for them as inspiration. We have used a number of documents to help us work with these topics. For each meeting one of the members has taken notes and distributed them among all group members.

Apart from the phone meetings, we also arranged a number of workshops where the members of the design group met in person. Themes for the workshops were a) to characterize the product we were aiming to end up with, b) to explore aspects to consider for design languages, c) what tools and functionality the system should support, what we called the eco system. All of these topics considered the possibility to design for the inclusion and motivation of end-users, particularly women.

Working material from the workshops was scanned and kept for later access.

2.3.3 Literature studies

In an attempt to explore and identify the possibilities to support a design that includes and motivates end-user programmers, related research has been an important base to build on. Reviewing related literature is necessary for several reasons (Feak and Swales, 2009). First, it is resource saving to use research results from earlier studies relevant for present research. Otherwise there is a risk of just trying to do something that is already done. In this study the results would have been quite different, and probably with a different focus, had I not come across the studies presented in chapter 4, and specifically section 4.3 on gender differences in end-user programming. A second reason is to show how one's research is related to the area one is studying, and to the publications within the area. I have limited the literature search to texts covering gender in end-user programming, but also some clarifying texts on gender as well as on end-user programming in general. The collection of literature within this area is not very extensive, but still I have covered far from everything within the area. Finally, to account for the area studied based on what others have done earlier is a way to become established within the research community (Feak and Swales, 2009).

The kind of literature study I have conducted is driven by a wish to understand challenges related to gender and end-user programming. It is thus a kind of narrative that is presented. I have merely wanted to find

information and then use it in the conduct of our project. Systematic, meta-analysis or focused literature reviews are not used in the present study (Feak and Swales, 2009).

The keywords that I used when I searched for relevant literature were simply "gender" and "end-user programming" to start with, which resulted in a large number of papers mainly from a research group based at the Oregon State University, with Margaret Burnett and Laura Beckwith as two main researchers (Beckwith & Burnett, 2004; Beckwith et al., 2006). This research group has produced a large number of papers, and other keywords that appear in some of them are "mash-up programming" (Cao et al. 2010), "female strategies in end-user programming" (Subrahmaniyan et al., 2008) among others, and these keywords formed the basis for a broadened literature search.

2.3.4 Seminars

Based on my initial literature studies I presented some findings in a seminar with the purpose of discussing possibilities to design for supporting a design that includes and motivates female end-users. The seminar was given at four occasions: for project members and other people interested in these issues in Umeå, later on in Luleå for the same kind of audience, for students taking a computer science design course where they specifically designed for the SATIN 2 project, and finally for informatics students taking a course in innovation. During these seminars I took notes of comments made related to the challenges for the SATIN 2 project.

2.3.5 Semiotic inspection

In February 2012 I conducted a so-called semiotic inspection of the project application in accordance with the first step of the semiotic engineering method for evaluating suggestions for interaction design (de Souza & Leitão, 2009). The inspection uses five questions to which answers are looked for in the documents inspected. The questions are:

- Who are the users?
- What do the users want or need?
- How do the users want to do things?
- Why do the users want to do what they do?
- Why do the users want to do what they do in a certain way?

I found answers to the first four questions when I inspected the project application, but the last one was left unanswered. Many of the items in the list of answers to the questions are in accordance with the research focus of the present study, further described in section 5.1.1.

2.3.6 Testing and observing

Two main test occasions were conducted during the project. The first test was a quick-and-dirty kind of testing when we only had access to a computer-based mock-up of the system. This test took place in December 2011, and was conducted by three of the members of the design group. All three followed a simple protocol (se Appendix B), where we first described the purpose of the system, encouraged the test subjects to explore the system, and tried to say as little as possible during the actual test. There were around 20 test subjects in total, and one test session took about ten minutes—not including the introduction—when the test subjects explored the system. The main purpose of this first test round was to explore the system's availability, to see if the interaction concept was understandable and intuitive, or if it contributed to excluding and rejecting the kind of users we wanted to address. The test sessions were recorded using Camtasia (in my case) or some other screencast software, saving everything that happened on the screen, and everything that was said. For each test subject, information about their former programming skills and smartphone experiences were also noted. In the present research I have only considered the nine test subjects that participated in the tests I was responsible for.

The second test occasion had a much more specific purpose. The idea was to try to incorporate the strategies and theories found in the literature study (see sections 4.1-4.3), into the tests in order to see if these strategies could support inclusion and motivation among the test subjects. In total, eleven test subjects participated in one session each. The tests took place during September, October and November 2012. Five women and six men participated as test subjects, and their programming experiences varied from none to very experienced. Each session took about one hour, where the first step was to give the test subjects a questionnaire to fill in (see appendix C). This was the same questionnaire that was used in one of the Oregon studies (Compeau & Higgins, 1995). The next step was to

carefully introduce the system to the subjects, going through all features, and showing a relatively complex example of an app, consisting of six components. I also encouraged the subjects to ask questions whenever they wanted to, to criticize the system if they wanted to, and to suggest other ways of interacting if they had any ideas. Then the subjects were encouraged to build apps on their own, but based on an idea from me, and I reminded them that they could ask me anything anytime. The system that was used in this test was a fully working prototype, with actual apps as a result, that could be downloaded to a smartphone. The resulting apps could also be tested in a web browser.

If the first app to build did not take too long, I suggested a second app to build, but there was no expressed time pressure on the test subjects. After the subjects had built one or two apps they were again given the same questionnaire as before. I also asked them a number of questions to clarify how they experienced using the system, and finally they filled in five more questions (see appendix D) related to the kind of help they had access to during the test session.

2.4 Aspects of the Analysis of Data

In this study there is a focus on gender, and gender differences. The literature studies show that women and men experience end-user programming situations slightly differently. There are also indications that women and men have different strategies for how to address tasks in such situations. The observation studies that I have been part of have had both women and men as test subjects. But the results mainly consider women's behaviour and experiences. This approach could certainly be questioned, but there are theories claiming that such an approach is not only acceptable, but sometimes even preferred, see section 3.5.

2.5 Critical Reflections on Method and Execution

When we choose a certain research method, it should be the one most suitable for a certain research project. Even though one is confident that case studies is the best way to go, there could still be hazards to consider or avoid. Merriam (1994) mentions five challenges to overcome in case studies. First there is the extent of the description. Either the resources available are limited, preventing the description to be thick enough, or the

description becomes too vast, which results in difficulties for the reader to grasp the case and draw conclusions based on the case. A second challenge is related to the risk of oversimplifying or exaggerating observations and what they can tell us. Thirdly, what is presented also depends on the skills of the researcher for conducting this kind of research. A fourth risk has to do with the ethical stance of the researcher, or at least the researcher's awareness of certain risks related to choosing among available data. The risk is to choose data that support the researcher's preconceptions. Finally, the fifth challenge of case studies is how to relate to criteria for judging research design, which usually are said to be validity, reliability, and generalizability. There is no agreed upon practice for how to deal with these criteria within case study research according to Merriam (1994).

It is close at hand to assume that the differences that we study have to do with essential characteristics of the genders. The consequences of such assumptions are stereotypes. A stereotype is a preconception of a category of humans. It means that people holding a stereotype have the conviction that they know a lot about members of this category without actually knowing the individuals of the category. Often this conviction is explained by inherent characteristics that all individuals of the specific category exhibit. Typical categories exposed to stereotypical ideas are gender, ethnicity, sexual orientation, and many others. My belief is that these stereotypes do not reflect how reality is actually shaped. I also claim that the signs of these stereotypes, that people claim to see, are first of all not as spread as people believe, people tend to see what they expect so see, and ignore the signs that do not follow the stereotypical view. Secondly, differences *between* categories are shown to vary less than differences *within* categories, which is a sign of stereotypical perceptions not being reliable (Fausto-Sterling, 1992). Thirdly, the way individuals develop is affected by the stereotypes related to the categories they belong to. To summarize, an individual should not be identified with stereotypical preconceptions related to the categories he or she belongs to. Still, in this study we look at differences between categories, more based on the belief that stereotypes have a tendency to influence people, than on a belief that possible differences can be explained by essential and inherent characteristics. And

the goal of identifying these differences is that they will disappear in a future utopian society.

There is thus a dilemma related to studying a particular category, such as gender, that should be taken seriously, the risk to preserve stereotypes of women and men when we study the differences between the genders. At the same time, not studying differences might have the consequence that the existing situation, such as women not engaging in producing computer-based products, does not change. And to have more people of diverse backgrounds engaging in such activities might be an important goal to strive for. Part of solving this dilemma is thus to try to understand why there are differences, which is elaborated in section 3.2, where Yvonnes Hirdman's gender system is explained (Hirdman, 1988).

A clear difference between the Oregon studies and the present study is that in Oregon they studied spreadsheet applications, which is a kind of application that many people have prior experience of using, whereas in the present study, the application studied is supporting users in building smartphone applications, which is something that a limited number of people have experienced earlier, and only one of the test subjects in this study had experienced before. This fact is likely to contribute to the very different results that the questionnaires showed. It is probably much more likely for regular people familiar with computer usage, to estimate their skills of using a spreadsheet application as quite extensive, than the skills of using an app building application.

Worth mentioning is that in the observation studies some of the test subjects were people of my acquaintance, and in a few cases they were even close acquaintances. The consequences of this fact are not easy to know and account for. The way I experienced their participation was that they were very honest in how they reacted during the observation sessions. I emphasized that any thoughts and complaints were very welcome, and such comments had nothing to do with me personally. Therefor I consider their comments to be as valuable as the comments from the test subjects that I did not know, or did not know that well.

In fact, the test subjects that I knew only superficially might have felt obligated to be more positive than they actually were, as a courtesy to me,

in spite of the information I gave them at the beginning of the test sessions.

Altogether the results from the observation sessions could not be used as a proof of how people in general experience the SATIN editor. Rather the results of the observations can serve as examples informing us of possible reactions to the editor.

3

A Feminist Research Approach

Generally feminist research is motivated by a biased perspective in traditional research, which is claimed to take a man's or men's perspectives. This means, for example, that research questions are mostly addressing men's conditions and problems. Even studying women might have the purpose of supporting men's conditions. This traditional research approach might not be a conscious choice in traditional research, but rather a consequence of neglecting a perspective that does not seem to be evident to consider.

Based on these claims, one might conclude that men are a homogeneous group, and that the same goes for women. This is evidently not true. There is not one single female personality type, and another single male personality type. What females have in common is not common personality types, competences, talents, preferences; there is something totally different that females—and males—have in common, and it has to do with the *expectations* society relates to females and males, *expectations* of personality types, *expectations* of competence, *expectations* of talents, *expectations* of what they prefer. That goes for males as well, of course, but often affects them in a different way.

These stereotypical expectations influence society in a number of ways. An individual female or male might be quite unaffected by these stereotypical expectations due to a certain upbringing they have experienced, how their parents value things and so on. But on a general level these expectations, or sometimes lack of expectations, might influence individuals to a great deal. For instance the choices an individual

makes for a career, and other paths in life, are done related to these stereotypes, either in line with them, or contrary to them. But choices are seldom totally independent of the stereotypes.

The reason why stereotypes influence individuals in our society to a high degree is that we are repeatedly and constantly exposed to them. We find them in books, in movies, in TV shows, in song lyrics, in jokes, in how people comment each other's behaviour etc. So people express their stereotypical opinions, and other people adjust to them or are at least affected by them. And I dare say that even the most insightful feminist is also likely to be affected by these stereotypes, as well as sometimes expressing them or at least silently thinking them.

This account might sound like a situation in balance from a gender perspective, affecting the genders reciprocally. There is, however, an imbalance between the genders inherent in these stereotypical expectations as to how they affect males and females. Studies show that most features associated with males—the stereotypical expectations of male characteristics and talents—are more highly valued than stereotypical expectations of female characteristics and talents (Hirdman, 2001). For example being reasonable is often considered to be a male characteristic, while emotionality is considered to be female (Hirdman, 2001:48). It is also notable that the characteristics that are considered to be male, is often the opposite of what is considered to be female. Being a male is thus *not* being a female. These differences in values are shown in for example salaries that occupations associated with such characteristics are paid, where traditionally male occupations receive higher salaries than traditionally female occupations. Another example of how this imbalance affects females and males differently is when the kinds of characteristics or personality features associated with males are more highly appreciated when new employees are hired. They might even be explicitly spelled out in advertisements.

In the present study, we look at gender differences within the IT area, and particularly in end-user programming. This area is described in chapter 4. And in chapter 5 the SATIN 2 project case illustrating end-user programming is accounted for with some feminist theories in mind. These theories are described in the following sections. In section 3.2 I account

for the gender system as described by Yvonne Hirdman, and Sandra Harding's gender levels. Then gender and technology theories are covered in section 3.3. Section 3.4 takes Feminist Technoscience as a framework that explains the unstable character of observations of the world, and thus why we should not see the present work as an eternal truth, but rather a cut in time. Feminist standpoint theory justifies studies where only marginalized groups are studied, which is described and related to the present study in section 3.5. Section 3.6 accounts for an escape from the stereotypical dilemma of preserving stereotypes when looking at differences between the genders. Finally implications for research in end-user programming are covered in section 3.7.

First a number of definitions of what feminist research is considered to be among certain feminist researchers, is accounted for in section 3.1

3.1 What is Feminist Research?

In the present research my ambition has been to have a feminist approach. Among some people such an approach might seem biased, interpreting this to be taking sides against men, and for women. So, what does it mean to have a feminist research approach? First, feminism has many orientations, but most of them still claim that our world is not equal, the living conditions for women as a group are in general not as good as for men as a group. Also it is claimed within most feminist orientations that these differences should be erased, that they are constructed (and thus possible to affect) and unfair. Just to clarify, these claims do not mean that *all* men are better off in our world than *all* women. But if we look at statistics describing fortunes and power e.g., it is easy to see that men seem to be the "winning" group. There is thus an agenda in feminist research for achieving changes aiming at a more equal world.

Secondly, there are no specified methods that feminist research prescribes. Rather, it is the chosen epistemology or epistemologies that provide directions for how to choose the methods to be used (Webb, 1993). A common idea in feminist epistemology is *situated knowledge*, a concept coined by Donna Haraway, to acknowledge that knowledge has dependencies and is not neutral and independent of situational factors (Haraway, 1988). How a person reacts to using a computer, for example, depends not only on their former experiences or lack of experiences of

using similar programs, but also on their gender and even the expectations and prejudice of gender and computers that are common in society in general. Also the notion of *Strong objectivity* (Harding, 1993) is a common idea in feminist epistemology, which is described in section 3.4.

Feminist research does not settle with descriptions and explanations of our world, it also aims at improving the world, to find paths for a better world. Consequently a feminist research approach has the clear aim of transformations (Björkman, 2005). Often this transformation is to achieve an improved situation for women in particular.

3.2 Models of Gender Systems

In an early feminist effort to understand what was going on between females and males, one talked about gender roles. Observations showed that females and males were involved in different practices, in different places. Accounts of these differences are legion, and can be found in philosophical as well as psychological and many other areas, and not necessarily in feminist accounts. Using the concept of a *role* might, however, lead us in a false direction. A role is most often consciously played, and can rather easily be replaced by another role, almost like a garment that is easy to change. What females and males experience, is not that easy to just leave behind and exchange for a more suitable garment. In a seminal text Yvonne Hirdman gives a brilliant account of what is going on between the genders (Hirdman, 1988). She touches upon using gender instead of sex, where gender has traditionally been explained as something that is socially constructed, as if we were dressing the biological sex in something that was not there at birth (Hirdman, ibid). But this gender identity that many experience is not something that is easy to take off, and replace by another garment. It is a lot more complex than that.

It is quite obvious that there is a division between the genders in our society, even today; a division manifested in several areas of life, such as occupation, fortunes, responsibilities in ordinary housework etc. This is related to gender patterns in society, where males far more often than females have power, fortunes and a voice in the public place. These patterns are salient if we look at the gender categories in general, but many exceptions exist within the genders. There are fathers being the parent

mainly caring for the children, and there are women who are prime ministers and presidents.

To explain the patterns that so many people experience—some embrace them, and others loathe them—Hirdman (ibid) talks about the *gender system*. Characteristic of the gender system are hierarchies and keeping the genders apart (figure 3.1).

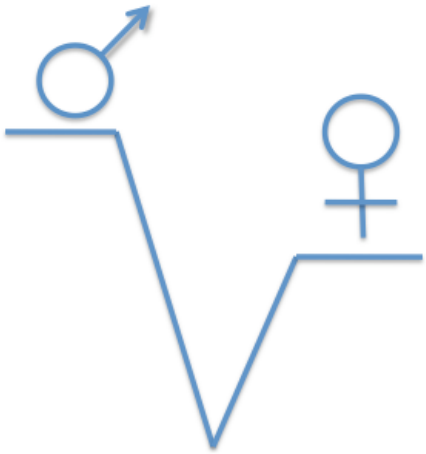

Figure 3.1: An illustration of the Gender System, made by Katarina Olsson, and shown at an equity workshop in Luleå, 2012.

There is no person or group of people consciously guarding and trying to preserve this situation, even though such attitudes exist within debates on feminism. Instead there is a constant renegotiation often prevailing in our (Western) society. This system does not blame males for the existing situation, neither does it say that only males benefit from it. What it does say is that we have a complex system of practices, attitudes, prejudices, etc. that influences most of us more or less everywhere and all the time. The system does not give a general description of every location on earth, at any time. Rather it is a tool for understanding what is going on at a certain place, during a certain period of time. Practices and characteristics associated with the genders vary over time and place, but the gender system helps us to analyse what the *gender contract* (Hirdman, 1988) for a certain place and point in time might be. A gender contract could be described as the result of negotiations in society as well as between individuals, regarding gender. This negotiation is in general not something

happening between equals, since there seems to be a male norm that dominates society.

Another aspect of the gender system is the notion of opposites, where maleness is considered to be what femaleness is not, and the other way around.

3.2.1 Harding's three gender levels

Sandra Harding, a renowned professor in social sciences, has made influential contributions to understanding the processes that preserve traditional gender stereotypes. These processes, she claims, simultaneously take place at three levels related to gender (Harding, 1986). The levels are the *structured level*, the *symbolic level* and the *individual level*. The structural level is what can be described in numbers and percentages. The individual level obviously has to do with personality, interests and behaviour of the individual. And finally there is a symbolic level representing norms and stereotypes associated with the genders, it is thus associated with beliefs and conceptions of gender rather than actual facts. An example of how symbolic gender is manifested is when someone watches the small child playing, and if it conforms to stereotypical playing, it is commented with a "boys will be boys" if the child is a boy and he is playing in a way considered typical of boys. These stereotypes are not only something uttered in a particular moment with no consequences; it also affects the small child, in one way or another. It might strengthen an already boyish or girlish behaviour, or it might reduce what the child feels is accepted to do as a boy or a girl, leading to a frustration in the child.

When we grow up and make lots of decisions about what, where, with whom to play or otherwise engage, we are often affected by those comments from when we were small, from books and films and the way gender is depicted in society as a whole. Not everyone adjusts to these stereotypes, but unless one has somehow missed the "gendered message", going against the stereotypes is likely to provoke or at least make people react to the individual's unusual choices. That is why newspapers write articles about girls training to become car mechanics. Taking such an "extraordinary" step in life is considered to be a story well worth telling by journalists.

3.2.2 Sameness, difference and equality

An issue that I have left out so far in this account of gender and what I claim to be stereotypes, is the question whether it really has to do with stereotypes, or if there is an essential difference to be found between the genders. This central issue has often been discussed in feminism (e.g. Evans, 1995). A consequence of essential differences between the genders is to re-evaluate the female characteristics in order to reach equality. If we believe that caring is a typically female occupation, while using and producing technology are typically male occupations, a logic consequence would be that these different occupations should be equally "rewarded" within working life, with similar salaries. That would also mean that campaigns with the purpose of attracting young women into technological working areas, and attracting young men into childcare and other traditionally female working areas, are a big waste of time and resources. Why would we want to change a behaviour that is essential and not constructed? At the same time we know of individuals going against those traditional divisions of labour. Are they going against their true nature? Or are there rare, but still existing exceptions of individuals who have somehow gained a nature beside "the natural"? To my perception, there are far too many exceptions to the stereotypes for them to be laws of nature (Fausto-Sterling, 1992).

3.3 Feminism and Technology

One of the areas where there is an obvious inequality is technology. It is clearly manifested in for example the number of females applying for educations in computer science, where the proportion of women is often as low as 5-10 per cent of the total amount of students.

One of the explanations of this situation in the Western society is in line with Hirdman's gender system, described in section 3.2. In short, technology is considered to be a male phenomenon. At the same time phenomena considered as female are not considered to be technology. We have a clear example of distinctive separation between maleness and femaleness. What is considered male and what is considered female is clearly separated in our Western minds. Another angle of how technology is perceived is that technology is highly valued. In summary these observations indicate that technology is considered to be male and also

highly valued, whereas what women do and use is neither considered to be technology nor highly valued.

There are several explanations for this situation. Essentialists claim that such differences are inherited, and should simply be accepted. A "proof" that this situation has essential explanations—i.e. that it is inherited in men to be interested in technology, and the opposite is true of women—is what we see when we look at what many men do (they are engaged in technological activities) and what women do (they are not engaged in activities considered as technological). But this is a closed-circle argument.

An alternative explanation is that girls and women just do not have relevant information, and therefor do not realize that technology suits them to the same degree as it suits men and boys. This is also the strategy that many campaigns for recruiting girls to technical education use. Such a strategy is based on the idea that all technology is gender neutral, and apart from giving relevant information, providing a female-friendly atmosphere is the best way to address this problem (Henwood et al., 2000). But is technology really neutral?

Judy Wajcman is one of the feminist researchers who have been criticizing the traditional definition of technology and how it is perceived in the Western world (Wajcman, 1991). In a more recent text (Wajcman, 2010), she accounts for the complexity of how gender and technology is constructed. She emphasizes that a challenge for feminism is to show that an interest in technology among men is not explained in men's biology, but rather in the social construction of gender and technology, also touched upon in Croon and Palmquist (2001). One way to approach this is to challenge traditional definitions of what constitutes technology. The activities and artefacts that women traditionally have been involved in have not been looked upon as technology, simply because of their gender.

But a feminist approach to technology is not restricted to redefining traditional definitions of technology, and encouraging women to use traditionally male technologies. Sandra Harding's classical wording "The science question in feminism" can inspire us to instead use technology as a tool for emancipation (Harding, 1986). One possible approach could be to provide environments and eco systems where women to the same degree as men not only use technology, but also actively participate in *producing*

technology, based on their own preferences and needs. Women can thus be regarded as active agents in relation to information technology (Wajcman 2010).

In Wajcman's and Henwood's work we find explanations to the currant situation of women as a group not feeling altogether comfortable with seeing themselves as producers of technology. This lack of self-efficacy can also be observed in end-user programming contexts (Beckwith et al., 2005).

3.4 Feminist Technoscience

To study gender differences of various kinds may be a necessary step to take in order to achieve a more equal society. As long as we do not know what the potential differences are, there is a risk that what we consider normal in society is actually mostly male. A consequence of not knowing about differences is that decisions are based more on men's experiences and behaviour than on women's experiences and behaviour. At the same time it is dangerous to focus on differences. In various debates about gender, strict differences between the genders are often taken for granted. People defend everything from differences in salary to who is considered the most appropriate to take care of children as having to do with inherent characteristics of the genders. My opinion on this, which is justified by numerous studies, is that such differences are often greater within each gender than between genders, and can seldom be explained by nature-given characteristics for the sexes. Research on gender differences thus creates a dilemma, namely the risk of preserving gender stereotypes rather than paving the way for a more equal society.

Feminist Technoscience has provided me with a framework where I can still justify investigating gender differences and thus overcome the dilemma mentioned above. This dilemma is further addressed in section 3.6. Below I explain the concepts that help support this kind of research.

A position, expressed by Barad (2003) among others, is that our reality—whether we refer to what is found in nature, what people have made, or the people themselves—cannot be described in eternal truths or facts. Rather, what we see and perceive are merely snapshots, which appear at a certain time, but maybe not forever, and are also related to a certain situation or context. The concept of *agential realism* helps us

understand this position. It can be explained as what we perceive in everyday life and in scientific experiments is always a kind of agency or in the words of Barad, *intra-action*. This contrasts to *interaction*—that might be how we often describe how phenomena occur in the world—in that something *emerges* from an agency involving a number of entities, rather than there being a distinct order of distinct agencies. It could be some substance or phenomenon that reacts to some other substance or phenomenon. What we normally call the characteristics or attributes of a thing or phenomenon is better explained as a reaction that this phenomenon shows under certain circumstances. A rock does not interact in the same way with water as it does with e.g. a piece of glass. The consequences of such an interaction are not the same if it only happens once as when it happens repeatedly. In fact, while measuring or observing phenomena in the world for scientific (and other) purposes, what happens is actually an action occurring, an interaction between the measuring or observing tool and the phenomenon being measured or observed. Barad (2010) illustrates such phenomena and the activities involved by describing Niels Bohr's experiments within quantum physics where a more traditional view of things and objects, and the states they represent is overthrown. One is forced to conclude that the interaction between energy and matter behaves like waves *as well as* particles, simultaneously, since observations and experiments bring forward both of these results. We experience something that can be described as "agencies of observation" (Barad, 2010).

Also people and people's reactions in a certain situation is likely to vary based on a variety of interacting factors that are linked both to the individual and the categories the individual belongs to, even though it is not possible to totally predict an individual's response to a given situation based on such factors.

3.4.1 Design for end-user programming

In a number of studies on the topic of end-user programming (e.g. Burnett et al., 2011) statistically significant differences between the genders based on a number of aspects have been found. These differences, I believe should be prioritized when we design systems for such applications. Needs and patterns of behaviours of females as well as

males should be equally met. At the same time a risk with this approach appears, that these needs and behaviour are seen as inherent and forever present. One particular attribution that easily gets stuck on females is technology aversion, which probably is not productive in the long run. If instead we consider these gender differences as the consequence of a collection of past events and experiences that the women and men in the study have encountered, then these results are not carved in stone, but rather an "agential cut" in history (Barad, 2010). Consequently, the results on gender differences can only be regarded as valid at the particular point in time of the study, and related to the particular zeitgeist and culture that shaped the results. At another place in the world at a different time, the results could have been very different.

If everything is considered to be just temporary results, one might question why bother carry through empirical studies at all? An interpretation of the sayings of Barad (2010) implies that it becomes important to identify as much as possible of factors affecting an empirical result. These factors are to my understanding what Barad refers to as an *apparatus* of a certain phenomenon (Barad, 2003). The more we understand about these affecting factors, the more we understand results stemming from empirical studies, and thus reality. Based on this reasoning one should not conclude that empirical results are totally random, but rather that they depend on and are related to more than just one single factor. There are most likely a large number of interacting factors that are entangled in a way that might not be possible to dissolve and study separately, contrary to the simple truths that are so much more convenient to handle, if they only were relevant. Observing one factor at a time might not show the same results as studying the factors entangled. And this urge to find the simple and easy solutions to unwanted phenomena and incidents seems to be desirable not only among researchers, but also among people in general. A frustrating consequence of this is that we might have to surrender and accept that not all explanations can be guaranteed to be complete. Instead of aiming for the ultimate (and simple) truth behind a certain phenomenon, recurrent research of this phenomenon could be seen as steps taken towards an improved understanding of that phenomenon.

Applied to the example above, this means that it is important to find explanations for why gender differences have appeared, what can explain why it looks the way it does at a certain point in time, at a certain place, in a certain context. By identifying causes for a phenomenon, we can also pave the way for finding actions for how to improve this phenomenon or situation. Using the terminology of Barad (2003), identifying these causes and explanations is what identifying the *apparatus* of a certain phenomenon, in this case gender differences in end-user programming, is all about.

3.4.2 Design and agential realism

The view of research and reality that Barad (2003) calls *agential realism* I claim fits like a glove to *design development*. Characteristic of design activities is a *direction* that designers follow in their design work. Designing interaction is an iterative process where the first design proposal is successively improved based on problems occurring, and challenges encountered while people test and interact with a certain system. The designer gains knowledge about the direction the design should take with each iteration, testing and evaluating the design, whether it takes the form of a simple mock-up or a fully working system. Developing the design becomes a step-by-step activity where design, testing, and gaining insights are intertwined. Having a total overview of all factors that need to be considered in advance is not possible for an interaction designer. A large amount of contributing factors are searched for while testing and using the system. This iterative approach results in a stepwise improvement of the product, with potential drawbacks to be considered in the process as well.

Following a work process like the one described above consequently means to step-by-step discern factors to consider with the purpose to produce a well-functioning product. This could be regarded as identifying the *apparatus* of the phenomenon of a certain interactive system, certain users, and a certain context. This process of identifying factors must inherently be conducted in this stepwise fashion. It is impossible to study each factor separated from the others, since we need a system built up and based on some basic factors of the apparatus. Thus these factors must be studied entangled, and are not possible to separate from the system and

are thus not possible to study separated from the system. Actually, the most difficult factors to find will probably be easier to trace with an increasing number of factors supported by the system in a good enough way, every improvement of the system means clarifying the remaining shortcomings of the system.

In the SATIN project this has become very obvious. In every interaction situation, whether using or just exploring the system, more and more factors have emerged that need to be improved in order to support users with no or little programming experience. Some of these deficiencies have been corrected in the system, while others remain, but could be overcome with information about how the system works. Eventually, all deficiencies should of course be fixed, but in the meantime, users have to deal with them by knowing how to get around them.

3.4.3 Design, responsibility and accountability

When working with interaction design, a system is produced that is probably used by people who would never be able to produce anything similar themselves. This means that users might find it difficult to criticize and suggest improvements for the system, since they do not have the knowledge of how to actually implement such improvements. There has however in recent years been a trend where users are invited to give feedback to web sites and web-based services. This trend could be interpreted as recognizing that all opinions about and reactions to a site or system can be important in the step-by-step process that designing websites as well as other types of applications involves. This possibility to give feedback in turn affects the users' views of their role in the process. Consequently giving feedback and having the chance to influence is becoming a more integral role for users. A question to ask related to this is who should be responsible for an interactive product to function in the best possible way. The notion of responsibility and accountability related to technology is well covered in feminist technoscience literature, and is accounted for in an early paper by Donna Haraway (1988). A question emerging here is what to be responsible and accountable for. Factors such as functionality, security, usefulness, simplicity, etc., are common factors to consider. Again we can understand these responsibilities as an entanglement where designers collect reactions and opinions and adjust

the design accordingly, while users provide their opinions and experiences to the designers. Another aspect of responsibility is that once a person is aware of something—whether it is sexism, racism, or a malfunctioning system—he or she could be ascribed responsibility, and even accountability for what might happen while using a system. The important insight here, I believe, is that technology is not neutral, it does not just appear from nowhere in a perfect form. People are involved in the process of designing and implementing interactive systems, and a responsible approach is recommended, even though it is probably not possible to guarantee that no problems will ever occur.

3.4.4 Feminist technoscience and feminist HCI

My area of research is concerned with challenges related to designing computer systems for ordinary people, an area within the Human-Computer Interaction (HCI) discipline. There is a growing interest in investigating what a feminist HCI would look like. Protagonists of this orientation are Shaowen and Jeffrey Bardzell (Bardzell & Bardzell, 2011). Their paper clearly sorts out the "feminist" in feminist HCI as well as in feminist technoscience, even though they only mention HCI in their paper. Their focus is actually on social science and taking a feminist stance in that field, and how such an approach could benefit HCI, in practice as well as in research.

As a starting point they set the key levels of gathering data in research as being epistemology, methodology, and methods, going from an abstract to a concrete level, in accordance with Harding (1987). The values, positions, etc. of an epistemology is implemented through a methodology that directs how to choose certain data gathering methods. We can conclude that there is no unique set of methods in feminist research (Harding, 1987), it rather has to do with how we use them and why, which is what a feminist methodology is about.

What is the relation between feminist technoscience and feminist HCI? Actually, the "feminist" of both of them has the same meaning, to my understanding. Bardzell and Bardzell (2011) clarify a way of interpreting how feminism contributes to any scientific discipline, or at least the discipline of social science.

In line with the view of feminist technoscience, they problematize traditional scientific objectivity. A theory where objectivity has been discussed and dealt with is Feminist Standpoint Theory with Sandra Harding as one of its proponents (Harding, 1993). Harding talks about a "strong objectivity", a way of leaving behind the "God trick" (Haraway, 1988), meaning a view from nowhere, as if it would be possible to study anything with absolutely no agenda or values. In strong objectivity the marginalized get a voice, heterogeneity is preferred over homogeneity. This has strong connotations to HCI practice, where we want to get away from the view of technicians, and instead broaden the perspective as much as possible, and hopefully meet the needs and wishes from a truly heterogeneous group of people.

3.5 Feminist Standpoint Theory

In Feminist Standpoint Theory marginalized groups are said to have double perspectives or competences (Harding, 1993), whereas people representing the norm only have the perspective of "being normal", and they are even claimed to be "poorly equipped" to be representatives for an entire population (Harding & Norberg, 2005). People from marginalized groups constantly have to adjust to what is considered normal as well as dealing with being marginalized. This also explains why women are often told to behave more like men in order to be successful in their careers. They are forced to understand the traditionally male way of pursuing a successful career, and at the same time they experience the consequences of being women, the expectations related to being a woman, the competences women are expected to have and not have etc. In the present research we are mostly interested in seeing and understanding experiences and challenges related to the use of an end-user programming product. Focusing on women's experiences is also in line with the project goal.

The idea is that, being a woman one is likely to be a member of a marginalized group. Still, women have to adapt to "the norm", which could be the way men in general act or perceive the world. Belonging to a marginalized group—regardless of which marginalized group or category one belongs to—means that one is likely to master not just one (the normative) but several perspectives, or one has experienced the world (or

a certain situation) in an alternative way compared to the dominant or normative experience.

Having this as a starting point, it becomes quite obvious that the more perspectives or experiences that inform a certain project, the better. It does not mean choosing one single perspective that a small minority has, but rather that a project—or a product—is more likely to become relevant and correct with as complete a background as possible.

3.5.1 Critique

Standpoint theory has been criticized for having an essentialist view of males and females. It is easy to assume that when women are said to have a special kind of perspective or experience—that needs to be considered—that this perspective must be inherited in all women. Based on this assumption, the conclusion made by its critics becomes that if all women have the same perspective and experience, then it has to be in their genes, hence an essentialist approach. There are several accounts of this critique, and one description is by Alison Wylie (2004) and a second by Sandra Harding (2004).

My interpretation of this theory is not according to the critique above. The female category is the only category (I think) that is a majority and marginalized at the same time (this statement depends on from where we look, in Europe Chinese people are marginalized, but they are still a majority in the world, but let's say that my statement holds if we look at the world as a whole). In spite of women being a majority in most of the world, they still are not the normative group. Many women learn that if they want to be successful, they need to act more like men. This advice is common in politics as well as in business and research.

There is a big dilemma in this situation, related to the essentialist aspect. It seems to build on women being a category with their specific ways of being, totally separated from men's ways of being. But this is not the point. Instead it has to do with the *dominant way* of being, which happens to overlap with the way the majority (or a large enough amount) of men in official contexts behave. In our society the dominant way of behaving is—I claim—considered the *right* way of behaving or the *norm*.

3.5.2 Implications for a design project

My interpretation of listening to the message of standpoint theory is that it is crucial to have a group based on diversity when we test a product with users. If the test subjects are too homogenous we will simply not get enough information. Also if we test the product with those who already have an interest in the product from the start, we will miss out on results from more marginalized groups that could be potential users in the future. Designers want to find experiences and ideas that contribute to an improved product, and just turning to the already converted seems like a bad idea.

3.6 The Stereotypical Dilemma

A clear dilemma while talking about gender is the fear of preserving a stereotypical view of female and male characteristics, behaviour and such. There is obviously something that makes a larger proportion of males than females choose computer related careers. While investigating reasons for this state the message might be interpreted as: this is how things are.

At the same time, what feminism is about is change. We want females and males to feel free to choose the career they want independently of stereotypical opinions of what is female and what is male. Today I believe that women as well as men are influenced by gender stereotypes in different ways throughout their lives.

Even if we have clear results from research studies showing that there are certain differences between females and males, we still have a dilemma. These results could be seen as eternals truths, this is how it is right now, and this is what will always be the case, might be considered the message of the study.

But don't we have a movement going on related to gender? We know that characteristics and activities that have been considered gendered have changed throughout history. A typical male during the 18th century would not hesitate to cry in public for example. Knitting was a typical male activity in Scotland some hundred years ago (and still is in certain places). Today pink is very much considered to be a girlish colour in the Western world, but before the women's liberation movement, only males could wear such a bright and bold colour. Females were considered far too unimportant for wearing such a bright colour.

There are plenty of examples of how characteristics and activities have changed from being related to males to now being related to females and vice versa. Hence we can never ascertain that a certain observation of the state right now is an eternal state of fact.

There are however observations that have been valid for a long time. Females have been considered "the second sex" in most cultures of the world for as long as we know. This unfortunate condition manifests itself in different ways. Lena Abrahamsson (2002) shows in a study of industrial workers that males who were introduced to new machinery had no complaints as long as they were not aware that females used the machinery as well. Once they learned that, some of the males refused to continue using it. Some of them even quit their employment, or at least moved to another department. They explained their behaviour by saying that they did not want to do "women's" jobs. Other signs of this view is that girls are said to be tough - in a positive way - when they play with boys, while boys are said to be sissy - in a negative way - when they play with girls.

So, how do we address this dilemma of preserving stereotypes while investigating gender differences? Schirmer et al (2011) present an approach that helps us to avoid that. Their study has to do with software migration processes, but the model they relied on is useful for any situation where differences are studied, and preserving stereotypes is a risk to be avoided. Instead of looking at categories and specific needs associated with each studied category, they focused on particular needs independent of categories, and how to support those needs.

3.7 Implications for Researching End-User Programming

In the SATIN project there is a clearly pronounced goal stating that the system should appeal to female users in particular. The reason behind this is to encourage females in the northern part of Sweden to take their ideas for smartphone apps and hopefully build a business plan around their apps. This approach motivates looking at gender differences, which is also mentioned by Mörtberg (1999) where she argues that women's experiences and knowledge are worth studying as a part of a process for change.

In order to pave the way for a change in gender distribution among end-user developers, we thus need to be more aware of design aspects and strategies that appeal to women to a higher degree than is the case in most end-user development systems today.

4

Gender and End-User Programming

To take steps towards figuring out how to approach designing for end-user development with the aim of supporting the demands of women in particular, there are a number of areas and theories to consider. In this chapter I therefor account for literature related to end-user programming to start with. The SATIN editor can be seen as an example of end-user programming, a tool to be used in end-user development. The purpose is to understand the challenges and possibilities related to this area.

Next I introduce technological self-efficacy with the purpose of clarifying a certain problem area related to people's beliefs in their own ability to master a certain technology. This is also related to self-efficacy theory in general. Self-efficacy theory is also part of the basis for a number of studies illustrating gender differences in end-user programming.

Then I account for strategies in end-user programming situations supporting the demands of female users in particular.

Finally I introduce some implications for the challenge of designing for design activities.

4.1 What is end-user programming?

First we need to establish what end-user programming really means. Normally interaction with computers has to do with specific *functionality* that has a certain representation in the user interface. The challenge here is to find conceptual models, terminology, structure, and actual representations to depict on a screen that are in accordance with how users normally perceive the task to manage. In end-user programming it is

not only a question of finding and representing functionality, it is also a question of supporting the modification of some kind of application. This modification activity is more or less in line with regular programming, but is in one way or another simplified compared to regular programming. Margaret Burnett expresses very simply that "End user programming enables end users to create their own programs" (Burnett, 2009:15). Examples of applications or programs that represent what end-users might accomplish are spreadsheets (Beckwith et al., 2006), computer games, web services, and customized information on the web or so called mashup programming (Cao et al., 2010). From the research on end-user programming, a related concept has occurred, which is *End-User Development (EUD)*, with the purpose of describing activities, techniques, etc. that support users in modifying applications (Burnett, 2009). Lieberman et al. define it as follows:

> "EUD can be defined as a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact." (Lieberman et al., 2006:2)

Yet another related research topic is End-User Software Engineering, which is basically EUD with the addition of establishing quality aspects (Burnett, 2009).

4.1.1 End-users and their needs

What characterizes end-users is that they express a need to change and modify the systems they use, to gain more control over their computers and computer applications (Lieberman et al., 2006; Repenning & Ioannidou, 2006). They each might have individual preferences and requirements that they want their systems to fulfil. These expressed needs can be interpreted as signs of innovation and creativity among ordinary people (Lieberman et al., 2006). We can thus identify needs such as taking control over existing software applications through possibilities to modify, expressing one's creative side, and being innovative.

4.1.2 Purpose of EUD

EUD is a research area with the purpose of finding methods, tools, and strategies that meet end-users' needs, not only actual end-users, but

potential end-users as well. In an application area that is very new, or relatively new, existing and active end-users are not always available for testing products within the observed area. This is particularly relevant for products in end-user programming, where new ideas are tested quite frequently. This situation complicates developing support for users of such products.

There are also political reasons for this approach in software development. As many people as possible should have the opportunity to actively participate in the growing information society—users as well as developers—is a wish expressed in Lieberman et al. (2006:2).

The kind of support that the EUD community wants to provide to users ranges from being able to make simple modifications of existing software, to producing new software products. Examples of modifications are filters in e-mail programs and introducing one's own character in computer games. An example of producing entirely new products could be creating a computer game using Kodu[7].

4.1.3 Support for EUD

There are a number of different ways to support end-user development, ranging from programming approaches to rather specific guidelines. First we have programming approaches, starting with domain specific languages, where the language is accommodated to a certain application domain, such as medical applications, database applications or spreadsheet applications (Mernik et al., 2005; Spinellis, 2001; Nardi, 1993). Another approach is programming by example, where users show examples of what to do, and the system then infers the "program" from that (Lieberman, 2001). Thirdly we have visual programming languages such as AgentSheets (Repenning et al., 2000) and Forms/3 that simplify the programming activities for the end-users (Burnett et al., 2001). Apart from programming approaches, there are also guidelines to consider for the design of end-user programming systems, which I account for in the following section.

[7] http://www.kodugamelab.com

4.1.4 EUD guidelines

To meet some of the challenges that we face when we design software for end-user development, useful guidelines could come handy. Repenning & Ioannidou (2006) have gathered their experiences in the field in three types of guidelines, namely *syntactic guidelines*, *semantic guidelines* and *pragmatic guidelines*. Their starting point is to see end-user development as a learning experience.

Syntactic guidelines are related to how programming elements are put together, the rules of how to do it. A guideline in this area is to make syntactic errors hard, and preferably impossible. The way to do this is to express programming elements as objects and to use drag and drop to assemble these elements.

Semantic guidelines are related to how to complete a program in a meaningful way. This can be accomplished by making domain specific languages, or by introducing meta-domain languages if generality must be met. Annotations that simplify learning the behaviour of and understanding the system are also useful.

Finally there are *pragmatic guidelines* that help make programs personally relevant and the programming process practical. This could be accomplished by supporting development, facilitating decomposable test units, providing multiple views with incremental disclosure, integrating development tools with web services, encouraging syntonicity[8], and scaffolding typical designs (Repenning & Ioannidou, 2006:68 et seq.).

4.1.5 EUD challenges

To support end-users in activities that normally require vast experience and knowledge in computer programming is without a doubt a great challenge. More specifically, the needs to be dealt with are a balance between specificity and generality. The more general the programming language is the more skills are required to master it. A domain-specific language on the other hand deals with objects and actions that are so to speak known in advance. Apart from finding a relevant abstraction level for objects and actions, another challenge is to find appropriate representations for the objects and actions to handle, and to find approaches that support the learning process (Lieberman et al., 2006; Repenning & Ioannidou, 2006).

[8] Papert's concept—borrowed from clinical psychology—that represents a way of learning by identifying with the phenomenon dealt with (Papert, 1980:63).

But the challenges are not only technical and related to a certain way of designing user interfaces. Another acknowledged challenge has to do with how end-users can become confident in their ability to engage in an activity normally left to highly skilled people, and this is described in the next section.

4.2 Technological Self-Efficacy

During the last two or three decades the notion of technological self-efficacy has become an established research area. Self-efficacy is defined as a person's belief in her own abilities to succeed when facing a challenge, ranging from zero to being very confident in one's abilities. The challenge might be exposing oneself to experiences one fears, such as certain phobias, as well as trying out new and challenging tasks. If the challenging or sophisticated task is within technology, we call it technological self-efficacy (McDonald & Siegall, 1992). There are a number of more specialized branches of this area. We find studies in computer self-efficacy (Compeau & Higgins, 1995) as well as in Internet self-efficacy (Torkzadeh & Van Dyke, 2002). There are also several attempts to measure self-efficacy. These measurements need to rely on the individuals' personal estimation; there cannot be a way of measuring this apart from asking individuals how they perceive their own level of self-efficacy, which is also the basis for the procedure followed by Compeau & Higgins (1995).

Many of the studies that measure people's level of self-efficacy are based on the work by Compeau & Higgins, where they suggest a questionnaire with ten questions related to how users estimate their confidence in managing a certain interactive system, see appendix A. Establishing this is however not enough for how to proceed in order to affect people's self-efficacy in a positive way. In the next section self-efficacy theory is introduced with the purpose of describing the sources that affect how people perceive their level of self-efficacy.

4.2.1 Self-efficacy theory

Technological self-efficacy is partly based on self-efficacy theory, a sub-field of psychology. One of the most prominent researchers in this field is Albert Bandura who has studied the relation between self-efficacy and the treatment of phobias (Bandura, 1977). There are, according to Bandura,

four sources that support self-efficacy, and they are performance accomplishments, vicarious experience, verbal persuasion, and emotional arousal.

Performance accomplishments is said to be the most influential source of the four. It has to do with earlier successful experiences. If a person has used a computer-based system in a successful way, where expected outcomes have been fulfilled, conceptions of what will happen in the future are likely to be positive. If instead a person has one or several experiences of not succeeding, that person is likely to expect unsuccessful outcomes in the future.

Vicarious experience could be a colleague or another acquaintance showing a person how to do something, making it obvious that success is possible. This source is considered the second best for reaching a higher level of self-efficacy.

Verbal persuasion has to do with someone verbally encouraging a person to do something. This source is only grounded in someone else's experiences, which makes it the weakest source for an increased level of self-efficacy.

The last one, *emotional arousal*, is a negative source for self-efficacy, thus contributing to a decreased level of self-efficacy. If a person experiences stress, frustration, or anxiety, these emotions are likely to have a negative impact on a person's self-efficacy.

Reconnecting these theoretical findings to the use of computers and computer software provides us with gender-related implications, which are addressed in a number of papers accounted for in the next section.

4.2.2 Gender and self-efficacy

There are quite a few studies showing differences between the genders related to self-efficacy and using computer software. Busch (1995) reports that differences in self-efficacy has been found among college students who were told to estimate their ability to complete complex tasks in spreadsheet software as well as in word processing. No differences between the genders were noticed in *simple* computer tasks in this study. Complex tasks were also covered in the study, and here clear differences between the genders were observed, with the male users rating their level of self-efficacy significantly higher than the female users. An explanation

given to the differences noticed in complex tasks was that the male students in the study had more computer experiences in programming as well as in gaming. The male students were also more encouraged by their parents and friends than the female students. The study was done almost 20 years ago. Still it reflects a skew idea of who is suited to master complex computer tasks.

Much of the research looking at gender differences in self-efficacy and computers settles with stating that there actually *are* differences. But it is also crucial to understand *why* these differences occur. Zeldin and Pajares (2000) and Zeldin et al. (2008) have conducted qualitative studies on self-efficacy beliefs among women and men who pursue careers in mathematics, science and technology. Their case study approach reveals a number of contributing explanations to why females in these areas go against a more traditionally female career, and instead choose a career often considered to be typically male. Factors affecting the women participating in the study were family members and teachers encouraging them to engage in activities that raised their interest, rather than choosing gender stereotypical careers. Also experiencing family members and friends mastering these areas affected them. Even though self-efficacy theory claims that a person's own successful performance accomplishments are the main source for an increased level of self-efficacy, Zeldin's and Pajares' research seems to point to other sources being equally, or even more, important for an increased level of self-efficacy for a certain task or task domain among these women. The sources referred to are vicarious experiences and verbal persuasion. These results only relate to women and their self-efficacy. Men still seem to be more affected by their own successful accomplishments, than by observing other people's experiences and hearing their opinions (Zeldin et al., 2008). Maybe a conclusion to draw from these studies is that women perceive a successful outcome of their doings as receiving positive feedback from people around them, implying that success among women has clear connotations to social relations rather than their own doings.

4.3 Gender Differences in End-User Programming

Most of the literature found on the subject of gender differences in end-user programming comes from the same research group mainly located at

the Oregon State University, and with Professor Margaret Burnett as prominent person, where the first papers were published in the early 2000s. Their research was a continuation of investigating visual programming for managing spreadsheets. The Oregon research group is dominant in this field, but there have been earlier attempts in the same direction, even though they have not been as extensive as those from the Oregon group.

For example Mary Beth Rosson and colleagues have also conducted a number of studies of women and end-user programming (Rosson et al., 2007; Rosson et al., 2010; Harshbarger & Rosson, 2012). Their focus is mainly on how the design of web applications could be planned. In one of the studies the females' feelings of success were greater than those of the males in the study. This result was in contrast to an earlier study, and is explained by the fact that the users in the more recent study are much more familiar with web-technology than the users in the earlier study, that the females in the more recent study were younger, and that the females had expected the tasks to be more difficult than they found them to be (Rosson et al., 2010).

Traditionally gender aspects of technology in general, and IT products in particular, have been studied related to education and the impact of society (Beckwith et al., 2005). The kind of actions taken to overcome such problems have mainly been, as earlier stated, campaigns with the purpose of persuading young women to take a step into the traditionally male realm of technology and technological education. A large number of studies under the supervision of Professor Margaret Burnett have instead put the focus on the product, on software applications, and particularly on software for debugging digital spreadsheets, aiming to investigate whether the design itself causes unforeseen consequences that disfavour women (Beckwith & Burnett, 2004).

Their research builds on a large number of literature studies and theories that are likely to be relevant for studying gender differences in end-user programming. Theories considered are learning styles (Ames, 2003; de Lange & Mavondo, 2004; Heffler, 2001), information processing (Huff, 2002; Meyers-Levy & Maheswaran, 1991), problem solving (Blackwell, 2002; Gorriz & Medina, 2000), decision-making and risk

aversion (Jianakoplos & Bernasek, 1998; Byrnes et al., 1999), information gap theory (Loewenstein, 1994), and self-efficacy theory (Bandura, 1977).

Based on the theories mentioned above—a result from exhaustive literature studies within the Oregon research group—the focus has been on elements that might have an impact on end-users. The researchers have come up with a taxonomy describing potential gender-differences in end-user programming (Beckwith & Burnett, 2004). The taxonomy consists of three main issues, and they are: confidence, support and motivation. Each of these is in turn subdivided into two or three more detailed issues. The main issues are also accounted for in connection to a scientific domain, a summary, and potential impacts on end-user programmers (Beckwith & Burnett, 2004). This taxonomy was then used as the basis for eleven hypotheses derived from the taxonomy mentioned above. The hypotheses were tested in a later study, resulting in an improved design of their test-bed system, and a set of guidelines related to how to design end-user programming environments with a focus on gender. In the sections below I account for their findings divided into the following themes: tinkering, strategies, and self-efficacy. First I account for the circumstances regarding how these studies were conducted.

4.3.1 Design of the Oregon research studies

The studies I have read almost all focus on tasks of debugging spreadsheets with planted bugs, using a spreadsheet application—Forms/3—that was produced as a test-bed with these studies in mind (Burnett et al., 2001). In an early version of the system, there were three features in the system with the purpose of guiding users in how to proceed when correcting bugs. The features were *checkbox*, *x-ing* and *arrows*. The checkbox feature had the purpose of marking that the value in a cell was considered correct. X-ing a cell was a way of marking that the value of the cell probably was incorrect. A third feature was arrows that showed the dataflow in the spreadsheet, an optional feature. In a later study (Beckwith et al., 2006) the system was modified, resulting in two versions that were tried out, a high-support (HS) version, and what was called a low-cost version (LC). The high-support version offered extra support in the form of features for assessing the correctness of values in the spreadsheet. The support consisted of boxes to check, resulting in an

increased number of steps to finish a certain task, which was far from an optimal number of steps. The LC version was designed to minimize the number of steps required for finishing a debugging task. Tool tips were offered, but in as short a version as possible, minimizing reading cost. The idea was that the effort to try something out and then undo it would be minimal, to support an exploring interaction style. The purpose of having two separate versions was to investigate their respective effects on males as well as females regarding self-efficacy, and the tendency for having a tinkering style of interaction. The concept of tinkering is accounted for in the following section.

4.3.2 Tinkering

Originally a tinker referred to a tinsmith who helped people mend their household utensils. "Tinkering" has over time gained the meaning of fiddling with something with the purpose of changing it or mending it, or just playing around with things to see what is possible to accomplish with them. It is an activity with elements of playfulness and experimentation (Blackwell, 2006). As an approach in interaction it is expected to support curiosity and facilitate learning. It is easy to imagine small children playfully exploring or tinkering with almost anything that they come across, and a result of this kind of activity is often some kind of knowledge. Touching a hedgehog probably results in some kind of pain and hopefully the knowledge that one should not pat a hedgehog. Eating dust most of the time results in the child spitting it out and the child might learn that dust is not so tasty. Squeezing wet snow might result in a snowball, or a snow figure, something that the child might find quite pleasant. Putting Lego bricks together results in houses, cars or whatever the child wants to accomplish. Tinkering has led to new knowledge for children testing different materials in different ways. This way of learning and doing is also said to have male connotations (Blackwell, 2006).

Also in engineering tinkering is said to be a fruitful way of learning new courses of action. In such situations tinkering could be described as trying out how components could work together, and trying out in what way the components could be assembled. Here the consequences of such tinkering are also of great importance for the conclusions drawn from the tinkering

activity. Tinkering is thus seen as a fruitful way of learning new skills in engineering as well as in software engineering.

In the Oregon-study described by Beckwith et al. (2006) tinkering was an inspiration for what to observe. In this study there were two versions of the software studied (Forms/3) as described above. The so-called low-cost version of the system was expected to support a tinkering style of interaction. Tinkering was considered to be an efficient way of finding the consequences of trying out a certain feature. For this to be really fruitful, a tinkering style should be characterized as straightforward and uncomplicated. It is also crucial that it is equally easy and straightforward to undo the feature chosen if it does not meet the user's expectations.

In the study described, tinkering was measured in a direct way and simply as the number of times features were chosen.

The findings showed that the male users tinkered more than female users. But the kind of tinkering that the men engaged in seemed to be an unreflective way of tinkering. The sign of this was that men often chose the same feature rather frequently during a use session, without actually remembering the consequences of the feature. The researchers' conclusions were that the male users did not reflect upon what they did, resulting in many actions neither leading to progress in the interaction, nor to insights into how the software worked.

The female users in the study did not tinker as much as the male users. On the other hand the female users seemed to reflect more when they tried out a certain feature, and were more inclined to learn from their tinkering experiences, resulting in positive outcomes for them, to some extent contrasting the results for the males and their tinkering.

4.3.3 Self-efficacy in end-user programming

One of the findings in the Oregon studies was related to how women and men experienced using the system. One particular experience investigated from the start was self-efficacy (Beckwith & Burnett, 2004). The procedure used for investigating this aspect was to let the users fill in the questionnaire developed especially for investigating self-efficacy in computer use accounted for in section 4.2, see appendix A (Compeau & Higgins, 1995). The questionnaire was given to the subjects before and

after the debugging sessions, and then compared, giving a difference in self-efficacy on an individual level.

A somewhat surprising result in the study mentioned above was that the female users' estimated level of self-efficacy dropped significantly after using the high-support version of the system, whereas the male subjects' self-efficacy only dropped a little bit. In the low-cost version the male subjects' self-efficacy increased some, while the female subjects' decreased some (Beckwith et al., 2006).

In an attempt to explain the results that females' self-efficacy dropped significantly for the high-support version, the researchers presented a possible conclusion saying that the female users did not think that tinkering supported their understanding of the debugging features (Beckwith et al., 2006). This is not a result from their research, but rather a hypothesis that they might test in the future.

An account of the SATIN 2 project's work on self-efficacy can be found in Palmquist and Wennberg (2013).

4.3.4 Cumulating self-efficacy

How people estimate their level of self-efficacy seems to have a great impact on future experiences. Hartzel (2003) describes how important it is for users to have a number of positive interaction experiences while using and learning new software, since each experience cumulate into a higher level of self-efficacy. So, designing end-user programming platforms in a way that supports the users in making every session a positive experience for them is indeed desirable, albeit a great challenge.

4.3.5 Strategies and supporting factors

In several of the Oregon studies strategies for debugging spreadsheets are in focus. Strategies are the plans that users follow for accomplishing tasks, how end-users make sure they reach their goals. They can be conscious or subconscious. The kinds of plans or strategies people use differ from person to person, and depend on several factors, such as prior knowledge and experiences. In this section strategies, particularly in end-user development, are accounted for. Also other factors affecting the feasibility in end-user development are accounted for.

The studies by the Oregon group from the first part of the 2000s mainly looked at spreadsheet debugging, and specifically spreadsheet

debugging in their own Forms/3 software. The purpose of investigating strategies in the Oregon studies was first of all to identify strategies relevant to end-user programming activities (Prabhakararao et al., 2003; Subrahmaniyan et al., 2008). A second step was then to come up with ideas for how to support these strategies, support that might be implemented as features in the software investigated (Grigoreanu, et al., 2009).

In one of the more recent studies, the Oregon research group investigated gender differences in strategies for succeeding with debugging tasks in spreadsheets (Subrahmaniyan et al. 2008). From that study they realized that the choice of preferred strategies for dealing with the task of correcting bugs in spreadsheets differed between the genders. A second finding was that strategies that women preferred where not supported in the software to the same degree as the strategies more typically preferred by the men in the study.

The strategies investigated were chosen in relation to the tasks investigated. A consequence of this is that some of the strategies were very specific, and cannot be seen as strategies that are general in end-user development situations. An example of the more specific strategies is "colour following", which is implemented as a feature in the Forms/3 application for signalling how certain it is that a specific value in a spreadsheet is correct (Grigoreanu et al., 2009). A second strategy that is omitted in the following is the fixing formulas strategy, which is obviously very specific for debugging spreadsheets.

There are other things than strategies affecting end-users' successful accomplishments. I refer to them as factors in the following account.

Strategies and factors of a more general kind related to end-user development, found in a number of papers, are the following: testing, code inspection, specification checking, data flow inspection, spatial layout aspects, feedback following, to-do listing, control flow, help, proceed as in prior experiences, a comprehensive view vs. salient features, familiarity, motivation, cost versus benefit, opportunistic usage, pairing, general vs. specific knowledge. These strategies and factors are described in detail in the sections below.

Testing

Using a testing strategy means that the user tests what happens when several different values are used (Subrahmaniyan et al., 2008; Grigoreanu et al., 2009). Features facilitating testing are preferred. Results do not reveal if testing meets females' demands in end-user programming, but there are indications connecting testing strategies and *feedback following*, see below (Grigoreanu et al., 2009).

Code inspection

Code inspection has to do with investigating existing code or the code one has produced oneself. This is a way of learning how the system works, as well as checking out and debugging one's own code. There are studies claiming that females benefit from code inspection as a strategy for learning. This means that looking at existing programming code could actually be very fruitful (Subrahmaniyan et al., 2008; Grigoreanu et al., 2009). For this to actually work well, the syntax of the code must be intelligible.

Specification checking

This strategy was mainly used when looking for errors in formulas. The users compared formulas in the spreadsheets with the descriptions of the formulas to make sure they were in line with them. There were no differences between the genders for this strategy among successful users debugging spreadsheets. Also notable is that this strategy for succeeding does not have much support neither in spreadsheet applications, nor in scripting applications (Subrahmaniyan et al., 2008; Grigoreanu et al., 2009).

Data flow inspection

Data flow has to do with dependencies, and in spreadsheets especially dependencies between formulas. Checking data flow is expected to inform the end-user about these dependencies, and if they are correct according to specifications. The potential support that showing data flow could lead to was not shown in the study investigating debugging in mashup programming (Subrahmaniyan et al., 2008; Grigoreanu et al., 2009). Males in these studies showed signs of having appreciated support for data flow inspection, whereas the results regarding females and the data flow strategy are inconclusive in these studies. A possible explanation is that

there was not much support for this in the systems used. Another explanation is that females seem to prefer forming a comprehensive view of a system, see below (Subrahmaniyan et al., 2008).

Spatial aspects of the layout

A strategy that is probably always present in using software is the spatial layout of the system. There are conventions for this, also related to culture. Generally, following how the code is spatially located hopefully informs users about meaning and functionality of the code (Subrahmaniyan et al., 2008; Grigoreanu et al., 2009). A spatial aspect of the layout is not possible to skip, even if no such conscious decisions are made, a spatial layout will always be there.

Feedback following

Feedback following is a generalization of the colour following used in the Forms/3 spreadsheet application. The idea is that users utilize built-in features that give feedback, helping them to assess the state of the system or potential "bugs" appearing while using the system (Grigoreanu et al., 2009). This was a strategy used by a successful female in that study.

To-do listing

In spreadsheet debugging as well as in more general end-user programming, to-do listing is a strategy for keeping track of suspected problems in the code, and a way to remember following up on these problems (Grigoreanu et al., 2009). In the more specific context of debugging spreadsheets, the Oregon research group implemented a feature helping users to keep track of what they had done, in the shape of checkmarks for indicating what parts had been checked, and what parts had not (Subrahmaniyan et al., 2008).

The difference between to-do listing and code inspection is simply that to-do listing also has an element of noting how far one has come in the activity of code inspection.

Control flow

In traditional programming control flow is basically how programs are constructed. It has to do with the order of instructions in programs. This strategy is not always in focus; in spreadsheet activities and debugging we have a data flow orientation rather than control flow (Grigoreanu et al.,

2009). Support for this strategy is often present in programming and end-user programming software, probably because it is fundamental in traditional programming.

Help

Help is simply the strategy of asking people for help or using certain resources in the software that help the user understand what to do. Also other sources of information, such as searching the Internet for documentation, are part of this strategy (Grigoreanu et al., 2009).

Proceed as in prior experiences

Earlier experiences could be an important factor contributing to successful use in end-user programming. If the user recognizes the course of action from other occasions, this might have a positive impact on the outcome in end-user development situations (Grigoreanu et al., 2009). It is not only a source for success, if the similarities from prior experiences do not match the current situation, conclusions made of how to proceed might be misleading.

Comprehensive view vs. salient features

Women seem to prefer a comprehensive view of a system, where they try to grasp the meaning of the entire system if possible. This information processing style could be an explanation of why women sometimes claim that they do not understand or master a certain system, even though they seem to use it without hesitation and successfully. Men on the other hand seem to use more of a serial information procession style, and also go for the most salient feature in the interface, and try it out. Supporting a comprehensive view might imply that the number of features should be quite limited, and that it should be possible to quickly understand the entire system (Cao et al., 2010).

Familiarity

Women show signs of preferring systems with features that are familiar. Especially women with low self-efficacy seem to rely on using features that are familiar. Even though the familiar features do not directly support what they wish to accomplish, the familiar features are still preferred (Cao et al., 2010, Ko et al., 2011). This strategy is related to a person's willingness to try new features. If the interface gives a feeling of

familiarity, where the users recognize elements and their functionality, it is more likely that all users dare try out all features since it gives a feeling of confidence.

Motivation

One of the factors affecting users returning to certain software is motivation, to have a clear motive for using the software (Beckwith & Burnett, 2004; Wiedenbeck & Engebertson, 2004). There are several elements here that affect users' motivation such as ease of use and usefulness. An interesting observation is that what motivates males differ from what motivates females. Women are more motivated by the possibility to help other people through technology, whereas men are more often motivated by the technology in itself (Beckwith & Burnett, 2004). This factor is related to *cost versus benefit* accounted for below.

Cost versus benefit

Studies show that women in general are more likely to weigh benefits of an activity against cost and particularly risk. This means that it is important that end-user development software is self-explanatory and quick and easy to learn so that females are more likely to feel that it is worth the effort to use it (Subrahmaniyan et al., 2007, 2008).

Opportunistic usage

Using features that emerge most clearly and adapting to them has proven to be a strategy that is more common among males than females (Ko et al., 2011). This strategy is often considered in interaction design, and is believed to be an efficient way to support learning new systems and their features. There is even an approach to programming called opportunistic programming that is said to be rapid and efficient in terms of reusing code (Brandt et al., 2009).

Pairing

A strategy that seems to benefit women in particular is to form pairs while taking on a new challenge. This strategy is also known as pair programming. When female students enter computer science programs this strategy is advocated, and has been found to benefit female students pursuing their computer science studies, giving them the confidence for completing their studies (Werner et al., 2004).

General vs. specific knowledge

General knowledge is more often not trusted among females as a source to rely on in new situations. This indicates that knowledge about the specific benefits females more, and that the interface should give a feeling of familiarity. The design should "deceive users" into believing that what is present in the user interface, is common knowledge (Hartzel, 2003).

4.4 How Making Is to Be Made in Design

Most of the research accounted for in this chapter has a common factor, which is guidelines, strategies, and theory that *support the designers of products for end-user development and design*. The Oregon group has focused primarily on how to design with the purpose of supporting self-efficacy when trying to manage end-user programming software. They have also looked at the effects of tinkering as well as a large number of strategies that a designer of such systems might consider in order to support design activities.

In the present research, finding ways to support designers of end-user development systems has been in focus. The basis that this work has built on is mostly studies by the Oregon group and self-efficacy theory accounted for in this chapter.

5

SATIN — An End-User Development Case

In this chapter I use the SATIN 2 project to illustrate the phenomenon studied, which is end-user programming, and the challenges associated with adjusting end-user development environments to female users in particular. The framework used for describing the case is based on theory on gender and technology in general, and more specifically female strategies in end-user programming—as described in section 4.3. I have also used self-efficacy theory, and more specifically technological self-efficacy theory, described in section 4.2.

This particular case is described from mainly three perspectives. First there is an account of a software development project with a clear focus on gender and diversity in design in section 5.1.

The second perspective is how a design team uses insights about gender-related issues in end-user programming aiming at developing a high-quality design, in section 5.2.

Thirdly the focus is on what can be said about reactions and improvements to a fully working end-user programming prototype with the specific purpose to build smartphone applications, or apps, described in sections 5.3-5.6.

The account of the case is given from my own perspective and experiences, with the consequence that the account is partial, and does not cover all activities, challenges and outcomes of the project. Still I am convinced that these experiences contribute to at least a slightly improved

understanding of what it means to design for end-user development from a feminist perspective.

5.1 The SATIN 2 Project

The SATIN project started in 2008 as a pilot study and continued in SATIN 2 in the end of 2009. The main purpose of the project has been to design and implement a web-based environment where users with no programming experiences or programming skills still should be able to assemble and build their own smartphone apps, by choosing suitable components from a large set of components or building blocks.

The idea behind the project is not only to provide people in general with the possibility to realize their own ideas of services and functionality in their own smartphones, but also to encourage and motivate people to incorporate their apps as part of a business plan. This ambition is accompanied with quite a lot of challenges, where we have to ask:

- How do we design in order to support motivation?
- How do we design in order to support creativity?
- How do we design in order to support someone with a business idea?

These questions have been central when carrying out the project in the course of the project period.

5.1.1 Goals and aim according to project application

A clear goal of the SATIN project has been to create a platform where smartphone applications can be built, where these applications can be spread to interested users, and even with functionality supporting the app builder in charging for their apps. Other business related features have also been discussed. For the purpose of making the demands mentioned in the project application clear, I used an evaluation technique that is part of Semiotic Engineering (de Souza & Leitão, 2009), a fairly new method suitable for specifying requirements for graphical user interfaces. Semiotic engineering consists of two parts, the *semiotic inspection method*, and the *communicability evaluation method*. I decided to review the project application using the semiotic inspection method. My motive was mainly to try the method out, and to become familiar with using it. The result of the analysis was well received within the project group, and the method seems to be a useful method for clarifying demands and requirements stated in

documents related to the design of a computer system, and especially with a focus on interaction design aspects.

In the semiotic inspection analysis of the project application, four questions made up the basis for a deeper understanding of the goals and the framework for the system. Normally, there are five questions to consider, but the last one did not result in any answers in this case. The questions are:

- Who are the users?
- What do the users want or need?
- How do the users want to do things?
- Why do they want to do what they do?
- Why do they want to do it in a certain way? (no answers were found)

In the following sections I account for the results of the analysis, based on the questions. The results are also found in Appendix E in tables.

Who the users are

There was a rather short answer found in the project description, about who the users are, and the answer is "women", and particularly young women. Another perspective taken on users was considering diversity, such as age, ethnicity, and obviously gender.

Users' needs

The needs identified in the inspection were divided into five categories. The categories were: needs of the individual user (Appendix E, table E.1), design-related needs (Appendix E, table E.2), business-related needs (Appendix E, table E.3), specific tools (Appendix E, table E.4), and societal needs (Appendix E, table E.5). The categories were chosen after identifying the needs mentioned in the project application, and are thus chosen and decided by me with the purpose of having a more manageable representation of the needs.

In summary users are assumed to have an *individual need* to express their personality and in so doing, develop mobile services to be displayed in a user-driven marketplace.

Design-related issues (Appendix E, table E.2) that would support the needs above are possibilities to have a platform with features that support developing a design ability, where unique experiences could be created while combining and further refining services from existing services.

These activities should be integrated into a single portal, where the users are also part of an innovative development process.

A third category of needs is *business-related*. A rather large number of such needs can be found in this category (Appendix E, table E.3). The needs should thus support the entire process from an initial business idea based on a certain app, to developing the app, having market services, finding partners and other services necessary for managing this kind of challenge.

Three specified *tools* for succeeding could also be found in the project application (Appendix E, table E.4), tools for developing user-driven services, for communicating and refining ideas, and also support for finding tools related to individual needs.

Finally two needs here labelled as *societal needs* were identified (Appendix E, table E.5). These needs are to initiate a shift in actions and attitudes related to equality and diversity, and to influence women in the IT sector.

How to do it

The answers to how users want to act in order to fulfil their needs are divided into five categories, and they are: technological considerations, concrete design, certain usability criteria, methodological aspects, and finally structure demands. Again, the categories identified are mine. Other categories are probably possible as well, with other groupings as a consequence.

Within technological considerations we find clear directions for designing a web-based product (Appendix E, table E.6).

The design suggestions (Appendix E, table E.7) cover using a visual programming style, with building blocks or components. Then these assembled components are interpreted, and they automatically result in a service. A search feature should also be available. The visual concepts should reflect the underlying program design.

Just a few criteria supporting usability were found (Appendix E, table E.8), and they state that the application should be well defined, extremely easy to use, and be accessible for users without IT expertise.

Methodological considerations—as I understand the project application—were also mentioned (Appendix E, table E.9). In this context this should be interpreted as the kind of support we want to provide

novice users with when they first approach the product. This is said to be through early establishment and through habituation, the users should have the possibility to slowly get used to the product, without loosing interest.

The final category identified that deals with how users want to act is structure demands (Appendix E, table E.10). Here we see a demand saying that search features should be based on service usage. In this category I have also chosen to include a demand already mentioned in the category of design suggestions, i.e. a structure fulfilling the demand of having a well-defined relationship between visual concepts and underlying program designs / constructions. This choice makes clear that categories are not evident and unique, but often quite subjective.

Why users want to do what they do

The final question that I have found answers to in the project application is concerned with why users want to do what they do. The answers are covered in two categories, individual wishes and market related wishes.

In the first category a desire among users to be part of designing products and services is mentioned, also expressing their personality, and to be the individuals they are (Appendix E, table E.11).

The second category—market related wishes—include wanting to affect services in the smartphone, and to reach the entire market (Appendix E, table E.12).

However, I found no direct or obvious answers explaining the previous question concerning how the users want to do things.

Summing up, the findings of the analysis of the project applications is as follows: SATIN 2 aims at supporting female users in particular, providing a tool for coming up with business propositions in a part of Sweden where job opportunities do not grow on trees, so to speak. It is also aimed at finding aspects of the actual system design that support female strategies in end-user programming systems.

5.1.2 Preconditions — organisation and methods

As earlier stated, the SATIN 2 project has been quite a large-scale project. The project was set to last three years, with participants from universities as well as from SMEs. The participants of the project were to cover

competences in technical aspects, in design aspects, in business aspects, and last but not least in gender and diversity aspects.

The individual participants varied to some degree throughout the project, and especially representatives from the SMEs varied due to employments finishing and starting, and depending on other responsibilities within the SMEs.

The participants from the universities remained almost the same throughout the project, even though some of the participants did not stay until the very end of the project due to employment conditions.

The extent of my own contribution to the project was working half time during two years.

Apart from active participants there were also two companion researchers providing guidance and advice during the course of the project. They based their advice on documents, such as notes from meetings, that project members published on a common web-based portal. Their feedback was presented during quarterly project meetings where most of the project members participated.

Group responsibilities

The project members were divided into three main groups. The so-called CoreTech group had the responsibility to develop and implement the platform, using relevant and advanced technological solutions, and a thoughtful and flexible technological structure.

The design group was mainly responsible for design decisions regarding interaction aspects.

Finally the business group was assigned the responsibility to develop business models and find partnerships that might emerge from the project.

The project management team consisted of representatives from all groups. And finally a steering group with representatives from trade and industry, as well as financiers, was responsible for following the progress of the project and for giving directions for further progress within the project.

Gender and diversity aspects were mainly addressed by the gender and diversity group, with representatives from the other groups, including myself.

Design group

The main responsibility for the design group was to propose design suggestions regarding how the users could put together their own mobile services, supported by a web-based platform and editor. The design group discussed concepts for how to support this with an emphasis on interaction aspects and how to connect components.

The design group held teleconferences every other week to start with, and towards the end of the project every week. In the meetings each of us accounted for what we had accomplished since the last meeting. We then discussed current issues, and made plans for what to do the next week or weeks.

CoreTech group

Apart from making sure that the technological aspects of the system were of high quality, the group also implemented design suggestions from the design group. The communication between the design group and the CoreTech group was crucial for succeeding with this. To ensure that the communication worked, at least one project member participated in meetings for both groups, reporting design suggestions to the CoreTech group, and implementation progress to the design group.

When the SATIN editor was stable enough to start using among project members, some of the CoreTech members were assigned to build components that would be the building blocks to assemble into new smartphone apps. All project members could suggest components to be added to the editor.

Gender and diversity group

As mentioned above, the members of the gender and diversity (G&D) group were also members of one of the other project groups. The purpose here was to bring knowledge and inspiration from the G&D group into the groups that members normally worked within.

One of the specific aims for this group was to identify a working process where G&D issues could be incorporated into the development process in a straightforward way. So the members of the group were encouraged to reflect upon these issues and account for their experiences.

It seemed quite obvious for the representatives from the design group and business group that G&D issues were important for their work. Even

though the representatives from the CoreTech group (one member left the project and was replaced by a second member) did not see an obvious effect of G&D issues in their everyday work, they were still loyal to the activities and meetings, and were open to taking in new perspectives.

At the end of the project, a web site referred to as the Gender and Diversity toolbox, was launched. The content of the toolbox is based on seminars, workshops and literature studies within the SATIN 2 project, and other collaborating projects. One example of the toolbox content is the persona approach for describing an imaginary user or participant. Research on female strategies and self-efficacy in end-user programming is also described and published in the toolbox[9].

Quarterly project meetings

During the SATIN 2 project, lasting three years, there have been project meetings four times a year, to which all project members have been invited. Each group has had the opportunity to present the work done since the last meeting, and present plans for the next quarter. The project groups have also collected this information in documents presented to the persons continuously evaluating the project. One purpose of these presentations has been to coordinate the ongoing work between the groups. The gender and diversity group has also had the opportunity to account for their work and findings in every project meeting. This part of the meetings has been as self-evident as those of the other groups, which has benefitted the ambitions found in the project description and application.

5.1.3 The SATIN editor

The product, which is the concrete outcome of the SATIN 2 project, is a web-based platform with a number of features supporting smartphone application programming, with support features, business features etc. Today the product consists mainly of what we call the SATIN editor, and features closely related to the editor. This is the environment where components are connected and assembled into fully working smartphone applications that are possible to download to one's own smartphone from the web-based editor.

[9] http://www.gdtoolbox.eu/

During the course of the project there have been a number of versions of the editor—based on different design concepts—ranging from mock-ups (on paper and computer-based) to fully functional prototypes.

The versions covered in this thesis are mainly three (there might have been some differences between versions as well, since the development of the editor has been a constantly ongoing process). The first computer-based version tested was a mock-up where the interaction was fully functional, although the components were only dummies with no functionality implemented, see figure 5.1. The purpose of the mock-up was to test the interaction, and thus to find weaknesses and alternative ideas for how to design the interaction.

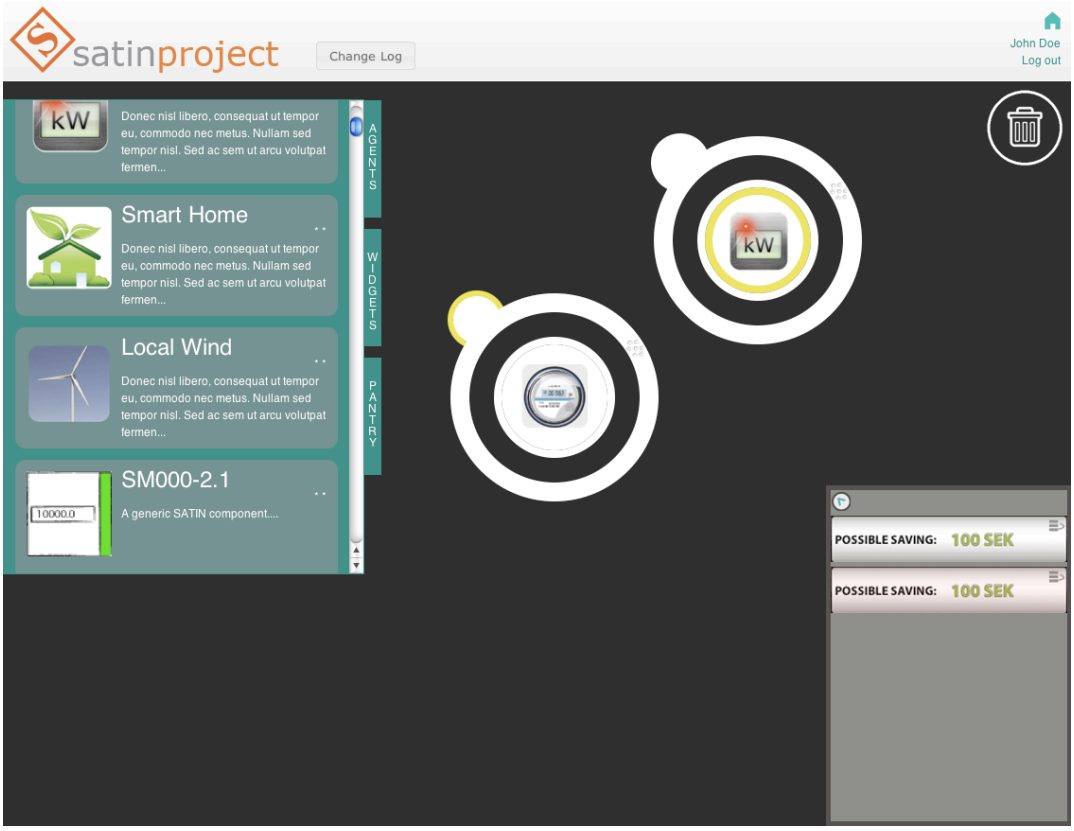

Figure 5.1 The SATIN editor mock-up from November 2011. See Appendix F for a larger version.

In this version the editor consisted of a canvas (the black area in figure 5.1) where the components were connected. The components were accessed from a list to the left in the editor. These components were then chosen, by dragging them into the canvas. The way to connect

components in the mock-up version was based on an agent-inspired interaction style. The agent-inspired style resulted in components being connected by clicking on the middle part (representing input) of one component, and then clicking the "bump" placed on the outer part (representing output) of another component supplying the data. The result of this procedure was components being connected.

Based on the tests of the mock-up (see section 5.2) a new version was developed, where the shape of the components was altered. The "bump" was removed, since its purpose was unclear. Instead the whole outer part of the component served as a clickable area. Apart from that change, nothing else was altered in the design concept.

Another feature present in both versions was that a connection was represented by a dashed pattern in the connected components, where this indication of connectedness corresponded in colour as well as in kind of pattern, see figure 5.2.

In this first prototype the components were implemented with working functionality and, when appropriate, real web-based data.

In the most recent version of the SATIN editor, some major changes concerning the appearance have been made. The components now have a rectangular shape. Another major change is the way components are connected. In this version there is a connector or plug representing an output, to be connected to a socket (an input part) in another component, see figure 5.3. These connections are also colour-coded.

Another major change that is implemented in the most recent version is that there is a preview of the application depicted as a smartphone, showing the user interface of the application to be built. We also changed the representation of settings for the components. Examples of settings are distances to be set between two locations in a component where these locations are compared, text messages in a component alerting a message based on a trigger being set. In the earlier version these settings were reached by clicking on an icon placed on the outer part of each component. In several of the test sessions the icon was mistaken for some kind of handle for moving a component. Another problem occurring was that the test subjects seldom understood that settings existed, and were even less aware that these settings are often crucial for the functionality of

the final application. Some test subjects understood that the icon represented settings for the component, but had problems finding the exact spot to click, since the clickable area was rather small.

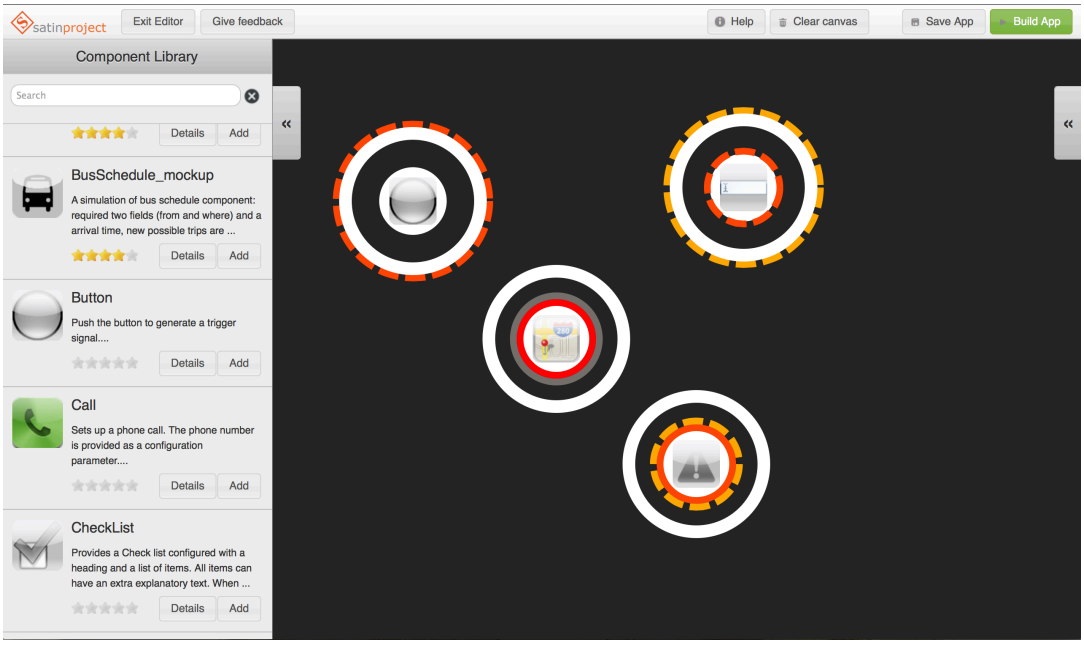

Figure 5.2. An enhanced version of the SATIN editor, a fully functional prototype. Connections between components are represented by colour and dashed patterns. See Appendix F for a larger version.

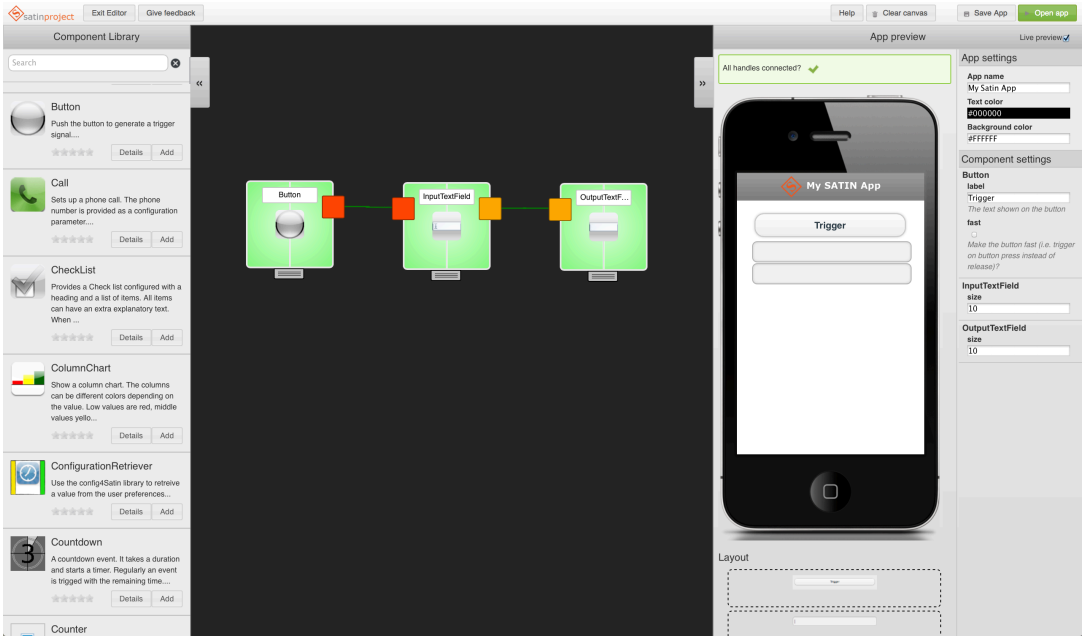

Figure 5.3. The most recent version of the SATIN editor, showing the square shape of components, the connections, the preview, and the settings on the far right. See Appendix F for a larger version.

These test experiences resulted in the settings being displayed continuously on the far right in the editor, see figure 5.3.

5.2 Interaction Design Experiences

Early in the design process, the idea of having an agent-based interaction style was discussed, and sketches to illustrate the concept were developed. One of the intentions with an agent-based style was to avoid a cluttered, difficult to read representation of the "code" describing the application (see figure 5.4 for an example of a wire-based version, where the code is somewhat cluttered and difficult to follow). The agent-based version is further described in section 5.2.1.

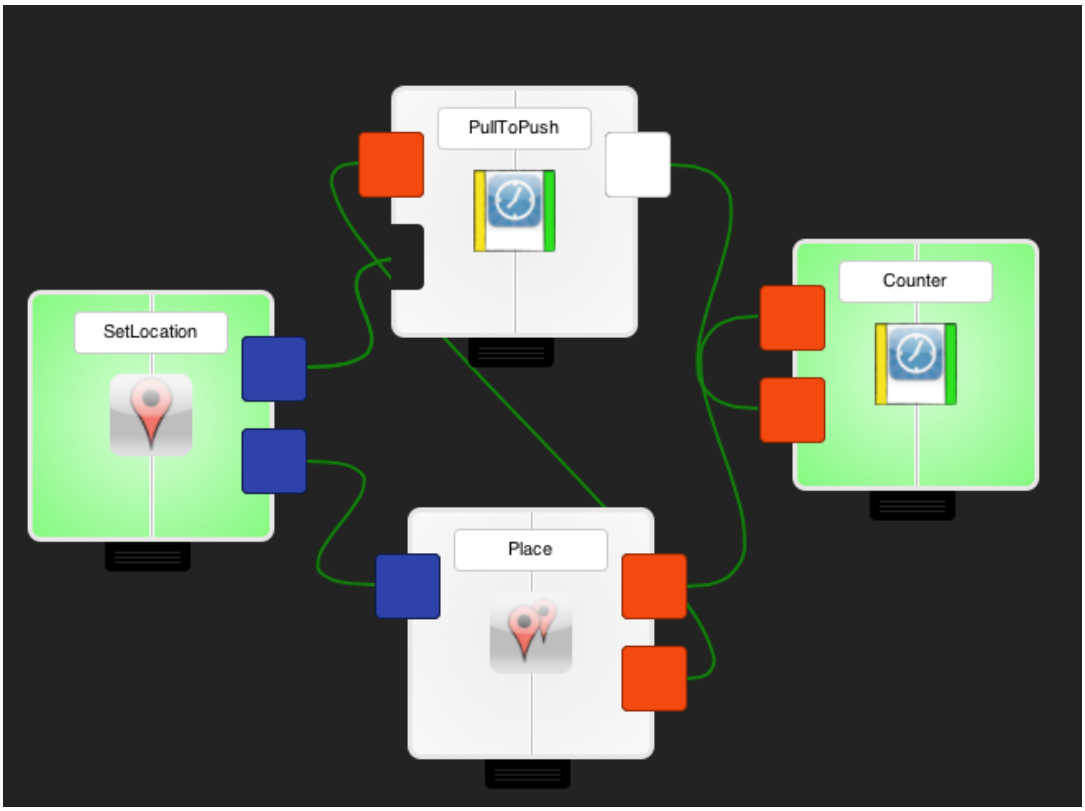

Figure 5.4. The "code" of this app is rather difficult to analyse due to connections crossing, and thus making them hard to follow. See Appendix F for a larger version.

5.2.1 Connections

In the agent-based version, connections between components were represented *within* each component, as described in section 5.1.3, rather than *between* components, thus resulting in a minimally cluttered view. This

was also how the components were to be connected in the first two versions, illustrated in figures 5.1 and 5.2.

Based on two test sessions where the first two versions were tested respectively, the agent-based approach was considered too difficult for beginners to understand. This result was also obvious from the test sessions performed by students as a part of their examination project in a course in human-computer interaction (HCI). So the outcome of these test sessions was to have a different approach to how to connect components, a more traditional way of connecting objects or components where the actual connection is represented by something directly showing the connection, such as a wire.

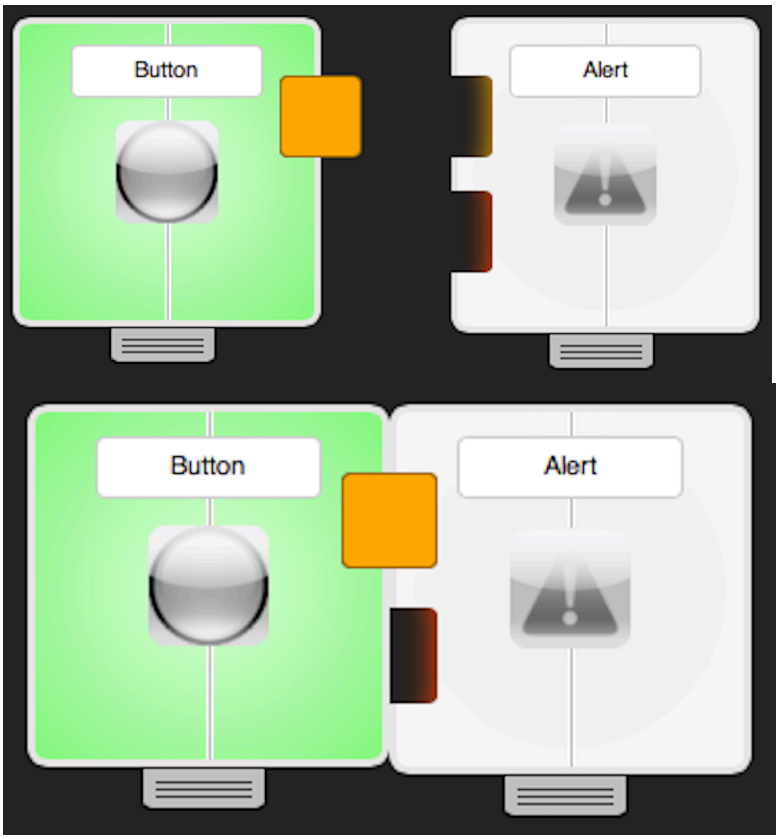

Figure 5.5. The illustration at the bottom shows how some users thought connections were made, i.e. by placing the connector into the "socket" in the rightmost component.

The wire-based version was tested in two courses in HCI in 2013, where groups of students evaluated the editor with subjects representative of the target group. Even though the tests showed that it seemed easier

for the users/subjects to understand the new version, the students still presented a number of suggestions for how to improve the understanding even further. Problems encountered were for example that the matching colours, indicating that a connection between components is possible, were not clearly shown. The colour-code of the input socket is a bit dimmed, which makes it difficult to decide whether the colours really match. Another criticism against this solution is that colour-coding is not recommended as the only way to give information, since as much as 7-8 per cent of all males have some kind of colour deficiency. Females are much less represented in this group though. Another problem occurring in the students' evaluations where that the part to be dragged from one component to another or the plug (in figure 5.5 it is the orange part on the component to the left, pointing to the right) was not understood as something to grab, instead many users tried to place the whole component close to the other in order to make a connection, which did not work, see figure 5.5.

So based on these most recent test results, there is obviously still room for much improvement of the user interface, especially regarding how to connect components.

5.2.2 Component settings

As mentioned in section 5.1.3, each component has a number of settings. These settings are almost always essential to set in order to give the components information needed for a certain app behaviour and to make the app work as expected. Examples of what settings could be are distances between for example the user's location, and another specified location. It might also be a message to be displayed as a result of using the app. Descriptions of these settings were not always clear to the subjects in the two tests, and sometimes they were too difficult for the them to figure out. In some cases the subjects also forgot how to access the settings, which was done through the small settings icon located on the outer part at the upper right of the components, explained in section 5.1.3, see figure 5.2.

In the first mock-up version of the editor, these settings were accessed through an icon (a couple of grouped dots) located on the outer part of the component. This icon was not recognised as a settings icon by the test

subjects. In fact, they did not even recognise it as an icon to be clicked on, but rather as some kind of handle for moving the components, which was touched upon in section 5.1. A first change here was implemented in the prototype version with actual components, where the icon was changed from the dots to a more traditional settings icon, in the shape of a gear-wheel. Even with this expected improvement, the test subjects' reactions in the second test round did not convince us that this was a good enough design idea.

In order to overcome these difficulties it was decided to make the settings more comprehensible and more visible. They were thus placed on the canvas for permanent visibility. A consequence was that the canvas (where the components are placed and assembled into apps) was reduced in area corresponding to that of the settings. This design choice has a clear trade-off. On the one hand it is important that users easily become aware of the settings, on the other hand it is also important that there is a large enough canvas where the applications are assembled, supporting a comprehensive view. A possible solution is to be able to temporarily hide the settings while connecting components.

When the students tested the latest version—where the settings part is on display all the time—with inexperienced users, some of the students suggested a design solution where the settings should be hidden and accessed through a tab for reaching a drop-down settings box. In the existing version, such a tab is used for displaying information about each component. The students motivated their design with the claim that people are used to this kind of solution, and thus would have no problem finding it. Their design suggestions have not yet been evaluated, but it is worth noting that they suggest a design solution that has some characteristics in common with a design suggestion that was abandoned earlier in the development process. We could ask ourselves whether we were too eager to abandon a settings design where the settings are not visible all the time. Was the initial problem the lack of visibility, the icon chosen, or maybe the location where the icon was placed?

5.2.3 Component scope

A rather wide range of components was implemented and incorporated into the editor by the CoreTech team. In order to test the SATIN editor

thoroughly—in the way it is intended to be used—a large enough set of components was needed. Ideas for components came from members of the project team; anyone with an idea for a component or an app could suggest that their idea be implemented as a component or a set of components. A designated group within the CoreTech team took care of implementing the components. The directions given to the CoreTech group was mainly related to the functionality of the components. Sometimes only a description of what the final application would do was given. A consequence was that the components' scope were not actively designed, but rather decided without much thought and probably in a hurry, since no directives regarding this aspect were given to those implementing the components.

According to one of the design suggestions identified in the semiotic inspection, there should be a clear relationship between visual concepts, such as the elements that we call components, and underlying program designs. This ambition could be interpreted as having components that users experience as logical not only in how they appear, but also in what they do, and what they do not do.

One example of a component that was not well understood beforehand by the test subjects was the CheckList component that displays a checklist defined in the component settings. The component needed to be connected to another component that displays a message. The purpose with the message was that the app would tell when all items in the list were checked. This is not necessarily something that people in general associate with a checklist, even though it is not totally illogical.

The scope is thus an aspect of the component design that should be deeply considered as opposed to just left to chance. Unfortunately, we never got as far as evaluating the components in the SATIN 2 project. The topic was discussed though, in design meetings and in a quarterly project meeting at the end of the project period. We mainly addressed identifying a set of quality criteria that all components should meet, although, these criteria were never formally documented. One of the few criteria that we addressed was the language used in the descriptions of the components. The descriptions quite clearly revealed that they were

formulated by representatives from the CoreTech group, focusing more on technological aspects than on aspects that end-users would relate to.

Component scope also involves the settings of each component. What should be handled by having settings in a component, and what should be handled by connecting the component to another component with the corresponding functionality to the settings considered? These questions are related to identifying quality criteria for the components.

5.3 Gender Issues in SATIN

From the very start of the SATIN project, there has been a clear focus on gender, even though resources for this have not been more than five per cent of the total budget. A conceivable consequence of this approach has been that, as far as I have seen, there has practically never been anyone questioning this focus during project meetings, at least not publically. This approach has given at least me a confidence to report on my findings related to gender in the course of the project. It is also worth noting that my own experiences from lifting gender issues in other situations are connected to something that I fear, and to lacking the confidence to do. Situations that come to my mind is when students express almost a rage in course evaluations to be "forced to" listen to gender aspects in a course, claiming that it has nothing to do with their future professional role. Also, among colleagues there has been a criticism towards incorporating gender perspectives in education, as well as taking measures aimed at meeting the expectations of our female students of our study programmes. There has also been somewhat of a backlash in society in general among ordinary people (such as people working as programmers or designers or business developers) regarding incorporating a gender perspective in their work. Something that probably has contributed to this backlash in Sweden is a television documentary in two episodes by journalist Evin Rubar broadcasted in May 2005, with the title "Könskriget" (The gender war[10]). The documentary depicted one of Sweden's organisations for women's shelters, and its chairperson, as representing extreme opinions and with extreme methods. Also a former Swedish minister for gender equality was interviewed, and questioned. A third focus was on professor Eva Lundgren, accused of seeing irrational signs of ritual murder of small

[10] http://en.wikipedia.org/wiki/The_Gender_War

children among members of secret religious organisations. The documentary depicted feminism in general and certain people and organisations in particular as far from trustworthy, and even as quite crazy and ridiculous.

Ironically the second episode of the documentary was awarded one of the most important Swedish prizes in journalism, "Guldspaden" (The Gold Shovel), and at the same time the Swedish Broadcasting Commission convicted the first episode of bias.

A consequence of this show was that among people in general and especially among anti-feminists, the documentary was taken as a proof of feminism being totally wrong, and that feminists were at the best misguided, but more likely out of their minds.

Not only did this TV documentary affect the feminist debate in Sweden in general where feminism was accused of being sectarian, but it also resulted in actions taken towards feminist researchers, and the support from the state given to women's shelters was questioned in debates in the Swedish parliament. All these factors have affected me as a feminist, and as a feminist researcher into fearing lifting issues related to feminism. I have felt that such issues have been labelled ridiculous, and as an academic and a researcher, being considered ridiculous is definitely not desirable, for several reasons. All of these fears never felt justified in the SATIN project.

In the SATIN project, my own activities related to gender were that I came across some papers reporting on gender and end-user programming, see section 4.3. I found out that there were significant differences between men and women in a certain end-user setting, where users were asked to correct bugs in spreadsheets. The differences were not so much a matter of interaction styles, but rather a matter of perceived self-efficacy (see section 4.2). Some of the papers also mentioned specific strategies that women seem to prefer when dealing with end-user programming, specifically when debugging spreadsheets.

When I presented the findings in these papers to the project members, I felt that the project members were almost relieved to see something rather concrete to deal with related to our actual design. We did not have to discuss aspects of power and other theoretical feminist issues, in order

to fulfil the demands mentioned in the project application related to gender.

Inspired by the results presented in the Oregon papers, and other sources, the design group took on to try to put the female strategies into practice through our design suggestions. A consequence of the insights we gained was to attempt to test our product with the strategies in mind, which was the approach chosen in my final evaluations within the SATIN project. I also used these strategies as the main aspect to evaluate when my students took on to suggest alternative design solutions to the SATIN editor. Each project group (all in all 10 groups) was given a specific and unique strategy to focus on in the evaluation phase of their design projects. Their results were not always what I wished for, regarding methodology and performance, but several of the groups made a satisfactory contribution to our understanding of how to implement these strategies into the SATIN environment.

5.4 Self-Efficacy and Strategies Evaluated

The results of the second observation study are accounted for in this section. The idea was to support strategies that would contribute to a high level of self-efficacy among test subjects. First, showing the test subjects how to assemble a smartphone application was a way of mimicking a feature that could be present in the system giving a *vicarious experience*. Such a feature might be footage showing the end-user the same thing as I did during the test session. Another strategy that this might have supported is *code inspection*. I also went through all of the functionality in the SATIN editor with the purpose of supporting a *comprehensive view* of the system. Having seen all features could also have contributed to the test subjects feeling familiar with the editor.

All test subjects managed to finish building the apps they were asked to build, and almost all of them expressed satisfaction with their accomplishments. A few also showed signs of being pleasantly surprised at their accomplishments, hopefully a sign of successful *performance accomplishment*.

Signs of creativity were clearly expressed by several of the subjects. One of the subjects had an inaccurate perception of what a certain component could do. She then asked why there was no such component in the editor.

She expressed that she wanted an app in her smartphone based on the kind of behaviour she expected from the existing component. Some of the other subjects described rather specific ideas of apps to build, if suitable components had been available.

The results from the questionnaires (Appendix C) indicated that most subjects (eight of eleven) were more positive of their abilities to successfully build apps after testing the SATIN editor than they were before they tried it out, which is shown in table 5.1, the Survey-column, where the first figure is showing the mean of the ten questions before and the second after using the SATIN editor.

Table 5.1. Information about test subjects, and their assessment results.

Subject#	Age	Gender	Progr. exp.	smartphone	Download	Survey	Fun
1	23	F	minor	No	no	6.9/9.2	7-8
2	25	M	extensive	Yes	moderate	8.2/9.7	7.5
3	20	F	none	Yes	moderate	2.6/6.7	5
4	23	M	minor	Yes	moderate	6.1/4.6	7
5	24	M	moderate	Yes	some-moderate	6.2/9.4	8
6	27	M	<minor	Yes	some	3.5/4.7	7-8
7	28	M	moderate	No	moderate	2.8/5.1	8
8	25	F	none	Yes	<moderate	7.4/5.8	7-8
9	23	M	none	Yes	moderate-often	5.9/6.0	6-7
10	36	F	minor	No	no	8.8/7.5	8-9
11	56	F	none	Yes	no	4.1/5.4	6

The last question that the subjects were asked in the interview was how much they enjoyed using the editor and building apps. The person showing least enthusiasm gave the experience a five on a scale from one to ten. Most assessments were between seven and nine, shown in table 5.1, the Fun-column. Some of the assessments were hypothetical, provided that the quality of the components had been better (subjects 2 and 10).

In the final questionnaire, where the subjects assessed help features, the subjects' answers were not unanimous, see table 5.2. Still there were tendencies toward them appreciating being shown the functionality and features in the editor, with an average of 6.9, being shown an example of an app built from scratch with an average of 8.1, and having the possibility to ask questions throughout the session with an average of 8.4, the scale ranged from one to ten. The design of the interface was also considered

rather helpful with an average of 6.4. Only a few of the subjects used the online help and thus only five of the subjects gave an assessment of that feature ranging from four to eight, where the highest value was given by a subject who consciously decided to try to find information in the on-line help instead of asking me, se section 5.5.2.

Table 5.2. The subjects' assessments of help features.

Subject#	Features shown	Example	On-line help	Interface	Asking
1	7	7	5	8	6
2	8	8	4	5	9
3	7	10	-	8	10
4	8	10	4	8	9
5	6	9	-	7	10
6	8	7	-	7	10
7	5	8	-	7	6
8	6	8	-	6	9
9	8	9	6	4	7
10	7	7	-	5	8
11	6	6	8	5	8

5.5 A Closer Look at Two Subjects

In this section I describe two of the observation studies in more detail. I hope to give a richer account of what happened, and thereby strengthen the case. I have chosen two quite different observations. The first one is the youngest subject who was the fastest to assemble the two apps. The second subject was actually the oldest one, and in the following two sections I account for how they approached the task of making an app.

5.5.1 "The young and successful subject"

Test subject number three is a 20-year old female student with no programming experience, and no interest in computers other than using them the way young people in general do.

The first app she built, an app with two components, displaying real estate for sale, took 2.5 minutes for her to build. She immediately chose the correct components, connected them in the correct way, and then asked what to do in order to display the results. I then informed her that displaying the results was embedded in the component. After that she checked the settings without hesitating, and noticed that the green button

indicating a working app had appeared. She asked if she should press the green button, which she did, and when she saw the results she was clearly pleasantly surprised, saying something like "wow, it actually works".

The second app she was asked to build was a checklist app. Again she managed to find the appropriate main component almost immediately, but was probably somewhat confused by the fact that there were several versions of the component. The reason for having several checklist components in the editor was that they worked in slightly different ways, and were evaluated by project members during the project. There were also other components with several versions, which was clearly something that confused the test subjects, and often much more than this one.

A complication with the Checklist component is that the actual list is entered through the settings, and in what could be called a programming style. This could definitely be a complication, making subjects unsure of how to proceed. Still this subject only asked how to separate the items in the list, and then finished the app with a triumphant "yeah".

The overall impression from this observation was that the subject clearly grasped the interaction of the editor, and also the concept of using components to make an app. Her success is probably better explained by her paying attention when the editor was introduced to her, than through an intuitive user interface.

In the interview she also expressed that she did much better than she had expected. She appreciated the introduction and believed it would have been difficult to manage without it. The introduction provided her with a comprehensive view of the system, she said.

A difficulty in the editor that the subject expressed was how to find suitable components. At the same time she managed very well with this in the test, which might have had to do with the example shown in the introduction, and the fact that she immediately recognized the icon of the correct component from prior experiences of a corresponding web site.

Even though this subject had no programming experience, and had the lowest assessment of the building experience (five), she was the most successful subject in terms of time taken, and reaching a working solution without getting lost in the interaction in the editor.

The strategies she used were looking for familiar icons symbolizing the components, and also reading the descriptions of the components.

In spite of her successful accomplishments, she did not seem very queen on using this kind of software in the future.

5.5.2 ”The middle-aged and persistent subject”

The last observation session was with a woman of 56 years with no prior programming experience and no experiences of downloading smartphone apps. As with the other subjects, I first showed her all the features and how to assemble an app, and then I informed her that she could ask me anything anytime during the test session.

When I gave her the instruction to try to build a checklist app she asked if that was all the information she was going to get at this point. She felt that she had no idea of how to proceed, but concluded that she had to start searching for something to use. She then found a component in the component list called Checklist. But then she was not sure how to ”get hold of it” and start using it. She first guessed she should double-click it, and then asked how to find information about the component. Double-clicking the icon of the component in the component list is actually how to display information, but there was a complication here that contributed to the subject's confusion. This complication was using a laptop without a mouse in this test, which was the only test where a mouse was not used. The subject was not used to this particular kind of laptop, and did not immediately succeed in double-clicking the touchpad.

This subject also found out that there were several versions of the Checklist component, which was obviously a bit confusing. We concluded that it probably did not matter which one she chose. Next she wondered what to do next, and suggested she should drag the component to the canvas, which is the correct way to do it.

The subject's next reflection was to understand how to add the list itself, where it was to be found. I then told her that I had done something similar when I introduced the system to her. She then decided to consult the help feature. Together we concluded that what she needed to do was to look at the component's settings. She found the settings, read the instructions, and started entering items to the list, even though she felt quite unsure of what she did. She wondered if there was any limit to what

she could add to the list, which there is not. Next she asked how she could know what she had done. I interpreted this comment as asking for a preview of the app, which would have made her more certain about what the app would actually do. An interesting comment from her was that she realized that such an app could actually be useful for her in her work situation, some kind of app for planning her activities at work. She thus showed signs of creativity and had ideas for an app for her own needs.

An interesting comment from her at this stage was that she said that there was a list ”somewhere”, but she did not know how to access it. This reflection was to me a clear sign of the need for a preview feature in the editor.

Next the subject wondered what to do next, and I informed her that she needed a second component for the app to work. Her reflection was that she thought she either needed a component for displaying a message or a component that would take care of the actual checking in the checklist. Being informed that the Checklist component also made sure the checkboxes in the list were displayed, the subject immediately dragged the alert component to the canvas, but still stating that she did not really understand what she was doing. Without hesitation she then chose the settings of the alert component and wrote ”Done” in the text box and commented that this message would be the result from checking all the items in the list. Then followed a moment of silence. I asked the subject what her thoughts were, and she wondered if a start button component was needed. I informed her that the Checklist component did not need a trigger starting it.

Something was still missing she concluded, since the button indicating that the app could be finished had not yet turned green. I informed her that she had actually not yet connected the components, whereby she realized that she did not remember how to do that. I offered to tell her that, but she wanted to figure it out herself, and after consulting the help function she managed to connect the components, with a ”look, hah!”.

When she looked at the app, it became clear that she had expected an interactive app, where she could have changed the items in the list as it was running, rather than when the app was created. This aspect of having

interactive apps was not even addressed in the project prior to this observation session, so this raised an important aspect to consider.

It was clear from early in this session that the subject had decided to have a totally different approach than most subjects. When she hesitated and did not remember exactly how to continue, she used the help feature and thoroughly looked it through, and then continued with her task.

Summing up her experiences the subject thought she did better than she had expected. Understanding the concept of components to connect, and information to control was rather easy, she thought. The difficult part was to know exactly how to do it, indicating that the version of the editor that was tested did not come across as very intuitive. She was also clear about her expectations of what she thought the components did according to the information given were not met. In her own words she said: ”I went from understanding nothing to believing that I understood, but then maybe I didn't understand, then in the end I probably understood a great deal anyway”[11].

5.6 Summary of the SATIN Case

The questions that were introduced in section 5.1 addressed design issues for the SATIN project. Finding precise answers to the questions is not within the scope of this work, and is probably not within the scope of any single project exploring design solutions for computer-based products in general, and end-user development environments specifically.

What happens in a project like this is that steps are taken towards identifying answers rather than giving precise answers to questions concerning design solutions. Sometimes the result is even more a matter of identifying relevant questions, rather than finding answers.

The SATIN project has presented me with some steps towards identifying answers related to the design of end-user development. The subjects' comments and reactions indicated that some of them felt quite motivated as well as creative during the test sessions. The question of how to support someone with a business idea was not investigated in this work.

In a broader perspective, the experience of looking at the SATIN project has taught me a lot regarding so many aspects of how to succeed with an end-user development project from a gender and diversity

[11] Translated from Swedish by the author.

perspective. It has taught me about the challenges to be dealt with in the development process as well as how to incorporate gender and diversity aspects, and also gender aspects directly related to specific design decisions.

These experiences are further addressed in chapter 6, where my observations are analysed.

6

What It Tells Us

Being a part of the SATIN 2 project has given me personally new knowledge and insights. These insights are related to a number of aspects relevant to a project such as SATIN 2, which is accounted for in the following sections. Even though some of the insights are not directly connected to the research questions, they are still important for reaching a successful product supporting the needs and wishes that end-user developers expect from a system such as SATIN.

I account for insights related to project management challenges, gender and diversity integration, end-user programming design, and finally insights not yet found.

6.1 Project Management Challenges

First, the progress of developing the SATIN editor has had "logistic" challenges. Implementation work depends to some extent on design suggestions, and at the same time, developing interaction design is an iterative process that largely depends on having a system or a prototype of the system to be used while evaluating with test subjects. Even though paper mock-ups were used early in the project, there are limitations to what kind of results such tests give. Paper mock-ups demand test subjects that are quite dedicated to the system being developed, and since there were no existing "natural" users that we could ask, these early versions of the system were mainly evaluated by the project members, since they had a clear interest in developing the SATIN editor.

The CoreTech group started to work on implementation tasks at the same time as the design group started to take on the challenging task of designing for end-user programming. The experiences of this situation on my behalf was that the design group became more certain regarding the explicit challenges and potential solutions to the challenges at the very end of this rather short project. When the members of the CoreTech group were wrapping up their contributions to the project, the design group were left with a long wish list with items that they would have wanted to see implemented. The reasons for this situation is probably not evident, maybe the project should have been planned in a different way, or maybe the design group could have been more efficient than it was. These explanations are just speculation, but should be reflected upon in future projects of a similar kind.

A clear challenge for the design group when advocating design propositions to the entire project team—and the CoreTech group in particular—was to be clear about exactly what we wanted implemented. Communicating design ideas is thus also a challenge in a project such as SATIN 2. A consequence of the unsynchronized pace between the design group and the CoreTech group was that when the design group finally suggested features to be implemented, only features that were not too complicated to implement were dealt with. At the same time, it was crucial to have a computer-based version of the system when testing these features. Otherwise the results would become rather hypothetical, for reasons accounted for above.

Altogether the SATIN 2 project has made it very clear to me that a project of this extent depends on a large amount of resources, in terms of plenty of time to spend on the project and many different competencies needed, including a well-informed project management.

6.2 Gender and Diversity Project Integration

One of the insights within the SATIN 2 project has to do with gender and diversity aspects. Since these aspects were on the agenda from the very start of the project, inscribed in the project application, it felt safe to raise questions regarding especially gender during our project meetings and design group meetings. It is significant that I use an expression such as "felt safe". Gender aspects are often looked upon as special interests only

relevant for a small group of women, or as a way for women who do not have a successful career to put the blame on something. In the SATIN 2 project we were actually obligated to follow up on these issues, since one of the aims of the project was to consider women's needs in particular.

Also related to gender aspects, I had the impression that when I presented the research results from the Oregon group in late 2011, almost all project members felt a relief. Now that rather obscure and difficult part to fulfil in the project had found "a way out". We could then "tick off" that part of the project. There were other gender-related goals as well, but they were mostly handled by the G&D group. Ticking off the gender aspects as focusing on women's self-efficacy meant that the more troublesome and maybe threatening aspect of gender and power could be left aside. When I describe it in this way I assume that some of the members of the SATIN 2 project would otherwise have reacted in a less positive way. There is no evidence of this that I have seen or that I can claim, apart from some comments from a couple of people on a few occasions. These comments still convinced at least me that it might have been a lot more difficult if gender had not been mentioned in the project application, or if our focus had been more power-centred, since those few comments were related to such issues, and questioned a gender perspective to some degree.

Despite all, I would describe the project members as most willing to embrace the gender and diversity goal of the project to a high degree. A possible explanation of this is the rather concrete design suggestions related to gender presented to the project members. Such an approach might be recommendable to get project members engaged in gender issues.

6.3 End-User Programming Design—the SATIN Editor

The experiences directly related to end-user programming were obviously part of the activities in the design group. Repenning's and Ioannidou's guidelines (see section 4.1.4) were not explicitly taken into account during the development process, but we still discussed how to assemble elements, or as we called them, components, which is one of their *syntactic guidelines*. Also in at least one of our workshop meetings, we discussed conceptual and technical aspects of making syntactic errors hard to make, and

preferably impossible (also among the *syntactic guidelines*). The *semantic guidelines* were also touched upon when we discussed characteristics of a suitable design language. Finally the *pragmatic guidelines* were related to our discussions concerning different views, such as a rather simple preview of the app, and other more spectacular views in three dimensions illustrating the entire data flow in the app. The purpose of these views was to—as clearly as possible—visualize the ideas that the user might be trying to achieve. With such a view the users can then assess whether they are on the right track or not.

6.3.1 Experiences

Focusing on experiences is likely to benefit designing for diversity in general, and gender in particular. The subjects in my tests thought it was rather fun, they seemed to think it was possible to successfully build apps, and they showed in their comments that the system brought them creative ideas, even though they did not express it in that way themselves. Using the system also seemed to raise a feeling of self-efficacy within the subjects with no programming experience, rather than reducing it, based on the questionnaire. The test sessions thus indicate that the idea of a rather general end-user development platform is possible, if the challenges encountered are considered and met.

Having used self-efficacy theory and female end-user development strategies as a framework for the observations has contributed to insights that should be considered in further development of the SATIN editor.

6.3.2 Learning and tinkering

More exploring studies need to be done in order to find better ways of supporting tinkering in our system. The studies showed that test subjects hesitated to test building an app, even if they were aware that it was possible. The most recent design with a clear preview might support the users to adopt a tinkering interaction style. The preview provides a kind of information that the test subjects in my observations seemed to lack, something giving them a more apparent representation of the app, but this remains to be tested.

6.3.3 Stages in the design process

The way the work in the project progressed indicates that not only potential users, but also people involved in the project as designers as well as implementers, understood more and more about the system's potential as the project went along. Each step, and each test brought on more insights regarding challenges that have to be met, and how to meet these challenges.

In the tests that I have been involved in, the potential for non-programmers to use and understand how to build apps have been confirmed. Not all subjects understood the mock-up version, and at the same time some subjects accounted for a clear understanding of what the system, and a particular app, was all about. In the second test phase, all subjects understood what they had done, and were rather confident that they could build apps with the system. But in parallel with this finding, the test clearly showed the challenges that remained to be met.

To my perception it has been quite obvious that there are no shortcuts in the design process, it is necessary to iterate design, testing, and implementation several times. With each iteration the challenges have become clearer, and have thus helped decide how to continue and focus in further development and testing.

6.3.4 Designing for design

I mentioned in the first chapter a number of end-user programming environments. A feature that the SATIN editor is supposed to support is the possibility to create or require new components to be added to the system. This feature is not present in most end-user programming environments. The SATIN editor as a platform for combining components that make an app is thus not all that the SATIN system should support. It should also support the process of adding new components to the system. The stages of this process at least consist of features in the system signalling that users' ideas could probably be converted into new components, functionality for requiring someone to build the component in case one does not know how to do it oneself, and finally some kind of support for communicating ideas to the person who undertakes to build the suggested component.

6.3.5 Design of supporting features

The observations showed that the editor could be a tool for building apps, even among users with no programming experiences under the circumstances given. There were also clear signs of motivation, creativity and self-efficacy among female as well as male subjects. At the same time there were also signs of frustration and confusion. It is possible that these experiences originate from the rather poor quality of the existing components in the SATIN editor. Also related to the components is the need to study and try to find an appropriate level of scope or detail for the components.

Since the observations were conducted, an improved version of the SATIN editor has been implemented. The main differences between the tested version and the new version are an added preview of the app interface, the settings are given a more salient appearance, and finally how to connect components is now more obvious, see figure 6.1.

A clear difference between the studies made within the Oregon research group and our studies, is that the Oregon group had test subjects that were familiar with the kind of task tested, and were thus probably more motivated and had higher expectations of their abilities than the subjects in the present study.

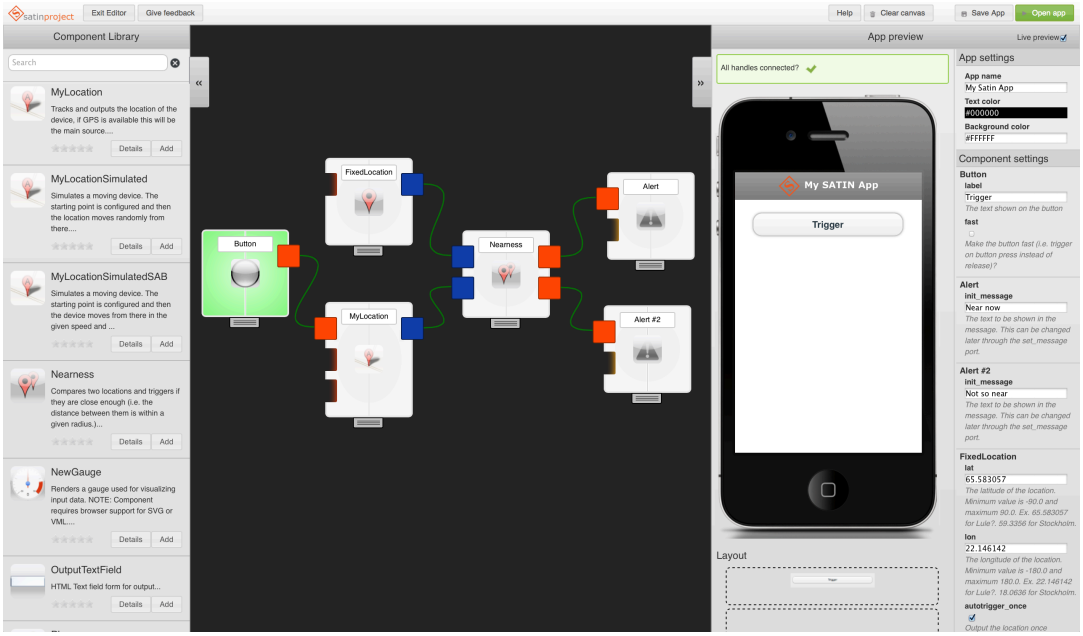

Figure 6.1. The most recent version of the SATIN editor. See Appendix F for a larger version.

Even though the subjects expressed that using the SATIN editor for building smartphone apps was almost a surprisingly positive experience, we do not know if the supporting features are possible to implement, and if they are, whether or not they will result in the same positive outcome as here observed. In the present test the users received direct and unlimited support from me as the observer.

In a future study it would be crucial to test a system with implemented features, and with a higher quality of the available components. It is also desirable to have enough components to support users' own ideas of apps to build. Not until then could we be more certain about the potential of the SATIN environment, and especially the potential to support self-efficacy and female end-user strategies.

6.3.6 A gender perspective of the observations

For me, it is not surprising that a subject managing to use the SATIN editor seemingly effortlessly, still is surprised that she succeeds, and before using the editor predicts her results to be much worse than how they actually turned out, in the way the subject described in section 5.5.1 did. A totally successful use of a "technological" product does not seem to be enough to give this subject the feeling of mastering the product or system. Theoretical explanations of this observation could be found in Judy Wajcman's account of "technology as masculine culture" (Wajcman, 1991). The perception of technology as masculine still seems to be so inscribed into the minds of even young women, that they do not perceive themselves as masters of technology even when they obviously do master technology.

Another theory explaining the reactions of this subject is related to self-efficacy, and especially how young women seem to rather be affected by the encouragement and success of acquaintances when it comes to how they perceive their self-efficacy (Zeldin et al., 2008). In the test situation, it was only the subject's own successful accomplishments that affected her self-efficacy, which she graded rather high (an average of 2.6 before and 6.7 after). Still, her way of reacting to her accomplishments indicated that they were highly unexpected. Even though her average after using the editor had raised from 2.6 to 6.7 which was the largest difference of all

subjects, she actually had an estimation of her future abilities that was somewhere in the middle of all subjects. Her relatively low estimation of her abilities strengthens the interpretation that this is related to the idea of technology as a male phenomenon.

The second observation described in section 5.5.2 also had elements indicating a lack of self-esteem of the subject in the situation she was, but in this case it is clearly expressed in what she said, for example that she said that she had no idea of what to do or how to proceed on a few occasions. Even though she did not manage to finish building her app as fast as the first subject, there were still elements of success in her session. Some of the stages she took towards finishing the app seemed self-evident to her. And in fact, some of her comments were valuable design suggestions that are now incorporated in the most recent version of the editor. Her perceived level of self-efficacy for future use of similar systems, according to the survey, was as low as 5.4 (the forth lowest) in spite of her succeeding in finishing the app she was asked to build, and in spite of her having many innovative ideas and interpretations of how the editor should work. These accomplishments did not seem to be indications to her of her technological skills. These outcomes are again in line with the findings of Zeldin et al. (2008) saying that personal accomplishments might not be the most important factor for a significantly raised self-efficacy, especially among women.

6.3.7 An end-user development design apparatus

Inspired by the theoretical framework of Barad, accounted for in section 3.4, an outline of the apparatus that end-user development design is composed of is presented here.

Clearly gender and technology stereotypes affect how successful end-user development systems turn out to be among women in general. Also the strategies that women prefer in such situations should definitely be considered, as well as self-efficacy theory. In accordance with the approach of the Oregon studies, looking at concrete design suggestions is of uttermost value when designing for end-user programmers from a gender perspective. In the present study especially the design—or rather lack of design—of the *components* proved to be critical for how the system

was perceived by the test subjects. So, component quality is definitely part of what constitutes the apparatus of end-user development design.

This account of the apparatus of end-user development design is far from complete, and should be further explored and expanded in future studies.

6.3.8 EUD design conclusions

In conclusion a platform such as SATIN is most likely possible to design, with features that support the kind of strategies that women as a group prefer in such computer applications. As mentioned in section 5.5, some of the subjects immediately grasped the concept for building apps in the editor without having any kind of experiences of similar situations. The features supporting the strategies are yet to be implemented and tested, but the present work implies that this way forward is definitely worth following.

Maybe the most important results from the SATIN 2 project is not primarily the SATIN editor but rather the knowledge about what is not yet designed and implemented in the SATIN platform.

6.4 A Feminist Approach

What makes this research *feminist* research? According to the criteria mentioned in section 3.1 feminist research is characterized by a wish to make a difference, to meet the needs, demands and conditions that women request to the same degree as those of men. This research focuses on designing end-user development software, and specifically investigating what happens when female strategies are considered in the design process. It is thus concerned with achieving a transformation of women's conditions in the specific context of end-user development of digital products and services.

Looking more closely at another design challenge, we have the quality of components. The relation between component quality and feminist research is explained by the same course of reasoning as given above. If components are too difficult to figure out, there are not only the traditional obstacles of an area of life considered to be male, there are also obstacles related to severe usability violations. The likelihood that women considering IT products as clearly male, will definitely not be convinced

that they suit them if they are impossible to figure out. It is probably also likely that they attribute the difficulties to their being female, consciously or not. Therefor focusing on an improved component design benefits women using the SATIN editor, or other similar products or services.

6.5 . . . And What It Doesn't Tell Us

The studies and experiences gained from the SATIN project have not disproved the strategies and guidelines covered in the literature study in chapter 4. Neither are the concrete realizations of the design suggestions tested in this research disproved. However, claiming that our design and design ideas do provide women with technological self-efficacy is far beyond what the present research can guarantee. Even though designing in support of self-efficacy is a commendable aspiration, evidently a well-designed computer-based system does not necessarily change the self-efficacy that a woman experiences while using a computer program. The contribution a certain design can make is rather to bridge a lack of self-efficacy and offer features covering the strategies that women seem to appreciate.

The way to overcome the lack of self-efficacy among women—as a group—in relation to computer-based products is thus a task that is far too complex to be solved in the present work.

The features that I tried to mimic with my presence and by answering questions in the second study seemed to be important for how successful the users were in building applications. These features still need to be implemented and tested again. There is no guarantee that the features in their implemented digital form will be experienced as supportive as a person being present and ready to answer questions or explain features. Neither is there an apparent design of these features. A feature for asking peers is not only a design decision, but also requires a large enough group being present as users or members of the SATIN environment. Tutorials and instruction videos also require testing in order to find the right way of addressing such issues. We do not know the difficulty of this task at this moment. It is not just a matter of showing the features of the SATIN platform. We also need to consider who shows a SATIN session in a tutorial, since this person represents the kind of person considered to be knowledgeable in the area of end-user development. The severity of the

task shown, and many more factors that affect such a feature also need to be considered.

7

A Look at the Future

For more than three years I have been part of and have reflected upon the SATIN 2 project and its platform and editor, how to design them, and what features to add to them. Sadly, the journey for me ends with this thesis. Still I have lots of ideas for how to continue to work with the development process.

In accordance with the case study research approach, many aspects work together in a design project such as SATIN 2, and all are required when testing and drawing conclusions. There is however a line of work that I would recommend for future development of the SATIN portal, and in particular the SATIN editor, which I account for in the following sections.

7.1 Components — Quality and Scope

As mentioned in chapter 5, the components in the editor were not designed with the involvement of the design group, and design decisions probably relied more on technological aspects, than usability and interaction aspects. This lack of informed design decisions turned out to be an obstacle while testing the comprehension of how to build apps, and probably also how end-users graded their level of self-efficacy. As a consequence, the way forward would thus be to redesign the components from a number of aspects.

In the SATIN 2 project we discussed quality criteria for the components at the end of the project, but we did not end up with any

specific recommendations. Therefore a recommendation for the future is to investigate the quality of the components. Based on the observations I have made, aspects to consider in such a study are:

- the quality of descriptions of functionality and data-passing for each component
- reviewing settings for each component, are they the right kind of settings and are the settings described in a relevant way?
- do the present components have a relevant scope, are they building blocks where each component comprise functionality and features that are considered logical for an end-user? One way of doing this is to look at every component, and let a number of independent people give their opinions on the scope of the component.

When the investigations of component quality are finished, the recommended changes should be implemented to facilitate new test sessions. Implementing these improved components is in itself a problem, since the old versions of the components must be kept in order for all existing apps to work the way their creators meant them to work. Still, the old versions of the components should not be exposed to new users in order to avoid confusion.

A second outcome of the present research is the importance of finding suitable scopes for the components, the building blocks of the SATIN editor as described in section 5.2.3. This aspect of the components concerns how much each component should cover for users to quickly grasp what they do, and how they fit in with other components. Investigating how end-user developers apprehend a suitable scope for components is highly recommended as a research project, since the reliability of other investigations depends on this.

Component descriptions as well as component scope are part of what component developers should consider when they design and implement new components for the SATIN editor. Clear instructions for these aspects are desirable as a result of such research.

A final component-related challenge that appeared in one of the observation studies—as an idea to explore—is the possibility to have interactive apps. The components that exist today are all designed in a manner where all information needed is specified using their settings,

which is done while the app is assembled or built. There is however a need to have apps where certain settings are made when they are used, such as the content of a checklist. Such a solution is definitely more flexible than what the present design offers. Challenges for adding interactive features are related to technical solutions as well as design demands.

Some important steps forward, not covered in the present research, have actually already been taken. Two students in cognitive science at Umeå University have in their bachelor's thesis examined the kind of criteria that should be considered for component design (Larsson & Lindström, 2014). Their thesis work covers specific criteria for the components, as well as interaction design aspects of how components are assembled and handled in the SATIN editor.

7.2 General Design Aspects

During the SATIN 2 project, and also later on, students have been investigating the SATIN editor, and have come up with alternative design suggestions. Among these investigations are the projects of three courses in Interaction Design that I have taught in the last two years. The results of these projects should be analysed and then result in a number of changes in the user interface of the SATIN editor. There is thus a vast amount of results and suggestions that definitely should be made use of. Especially, they have provided us with a diverse range of design suggestions for almost every part of the SATIN editor. Following the recommendations of interaction design methodology, these design suggestions could all be tested in observation studies with potential users, and compared in order to find a design as good as possible, based on existing materials.

7.3 Features Supporting Self-Efficacy

The features that I had the intention to mimic in the observation studies should be implemented to be useful. Until this is done, we cannot know that such features really make a difference serving as a support for self-efficacy.

Do the features mimicked in the tests give similar results if implemented? Do they in that case support the strategies that female users

prefer to a higher degree than male users? Or are these features yet another set of "training wheels" that will lower the self-efficacy of female users in a more reliable test situation—in the same way as in some of the Oregon studies?

Is it possible to come up with a set of guidelines that makes it easier for designers and implementers to more easily support the "female strategies?"

7.4 SATIN Editor and Innovation

Innovation aspects are also relevant to the SATIN 2 project. A study where preferably young women (the target group for the project) with innovation ambitions test the SATIN portal would be appropriate. It would require finding a group of people with a particular interest for innovation related to building apps. Collaborating with teachers of university courses with innovation themes might be fruitful.

7.5 End-User Development in the Future

The area of end-user development is still in its infancy. Even though a vast amount of studies has been produced in this area, there still needs to be a good enough system to continue to develop and study. Also a large enough group of enthusiastic users, preferably women, that contributes to ideas as well as implementations of components is crucial. Only then will it be possible to truly understand the factors, or the apparatus, of end-user development design, and thus take the design of such systems one step or preferably several steps forward.

The SATIN editor could serve as an experimental platform for further development of end-user programming design for the purpose of learning more about this certain area of interaction design.

References

Abrahamsson, Lena (2002), Just när det blev viktigt blev det manligt (When it became important, it became male). In Swedish. *Kvinnovetenskaplig tidskrift*, Vol. 23, No 1, pp. 37-52.

Ames, Pat C. (2003), Gender and learning style interactions in student's computer attitudes. *Journal of Educational Computing Research*, Vol. 28, No. 3, pp. 231-244.

Bandura, Albert (1977), Self-efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological Review*, Vol. 84, No. 2, pp. 191-215.

Barad, Karen (2003), Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter. *Signs: Journal of Women in Culture and Society*, Vol. 28, No. 3, pp. 801-831.

Barad, Karen (2010), Quantum Entanglements and Hauntological Relations of Inheritance: Dis/continuities, SpaceTime Enfoldings, and Justice-to-Come. *Derrida Today*, Vol. 3, No. 2, pp. 240-268.

Bardzell, Shaowen & Bardzell, Jeffrey (2011), Towards a Feminist HCI Methodology: Social Science, Feminism, and HCI. In *Proceedings of CHI 2011,* ACM Press, pp. 675-684.

Beckwith, Laura, Burnett, Margaret (2004), Gender: An important factor in end-user programming environments? In *Proceedings of 2004 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 107-114.

Beckwith, L. Burnett, M. Wiedenbeck, S. Cook, C. Sorte, S. Hasting, M. (2005), Effectiveness of End-User Debugging Software Features: Are There Gender Issues? In *Proceedings of CHI 2005,* ACM Press, pp. 869-878.

Beckwith, Laura, Burnett, Margaret, Grigoreanu, Valentina, Wiedenbeck, Susan (2006), Gender HCI: What about the software? *Computer*, Nov. 2006, pp.97-101.

Bergvall-Kåreborn, Birgitta, Broberg, Anders, Lassinantti, Josefin, Davoli, Lorenzo, Kuenen, Stoffel, Palmquist, Lena, Parnes, Peter, Ståhlbröst, Anna, Synnes, Kåre, Wennberg, Paula (2012), User Toolkits for Citizen-centric Mobile Service Innovation. In *eChallenges e-2012 Conference Proceedings.*

Björkman, Christina (2005), *Crossing Boundaries, Focusing Foundations, Trying Translations: Feminist Technoscience Strategies in Computer Science*. Dissertation, Blekinge Institute of Technology.

Blackwell, Alan (2002), First steps in programming: a rationale for Attention Investment models. In *Proceedings of the IEEE 2002 Symposia on Human-Centric Computing Languages and Environments*, pp. 2-10.

Blackwell, Alan (2006), Gender in Domestic Programming: from Bricolage to Séances d'Essayage. *CHI 2006 Workshop.*

Brandt, Joel, Guo, P.J., Lewenstein, J., Dontcheva, M., Klemmer S.R. (2009), Two Studies of Opportunistic Programming: Interleaving Web Foraging, Learning, and Writing Code. In *Proceedings of CHI 2009,* ACM Press, pp. 1589-1598.

Burnett, Margaret (2009), What is End-User Software Engineering and Why Does it Matter? In *Proceedings of the 2nd International Symposium on End-User Development*, Springer, pp. 15-28.

Burnett, Margaret M., Atwood, John, Djang, Rebecca Walpole, Reichwein, James, Gottfried, Herkimer, Yang, Sherry (2001), Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm. *Journal of Functional Programming*, Vol. 11, No. 2, pp. 155-206.

Burnett, Margaret M., Beckwith, Laura, Wiedenbeck, Susan, Fleming, Scott D., Cao, Jill, Park, Thomas H., Grigoreanu, Valentina, Rector, Kyle (2011), Gender pluralism in problem-solving software. *Interacting with Computers*, Vol. 23, No. 5, pp. 450-460.

Busch, Tor (1995), Gender differences in self-efficacy and attitudes toward computers. *Journal of Educational Computing Research*, Vol. 12, No. 2, pp. 147-158.

Byrnes, J., Miller, D., and Schafer, W. (1999), Gender differences in risk taking: a meta-analysis. *Psychological Bulletin*, Vol. 125, No. 3, pp. 367-383.

Cao, Jill, Rector, Kyle, Park, Thomas H., Fleming, Scott D., Burnett, Margaret, Wiedenbeck, Susan (2010), A Debugging Perspective on End-User Mashup Programming. In *Proceedings of 2010 IEEE Symposium on Visual Languages and Human-Centric Computing*.

Compeau, D.R. & C.A. Higgins (1995), Computer Self-Efficacy: Development of a Measure and Initial Test. *MIS Quarterly*, Vol. 19, No. 2, pp. 189-211.

Croon, Anna, Palmquist, Lena (2001), Virtualisering och den sociala konstruktionen av kön och teknik. In Swedish (Virtualization and the social construction of gender and technology) in *Från siffror till surfning* edited by Lena Kallin Westin & Lena Palmquist. Lund: Studentlitteratur.

Dahlström, Margareta (1996), Young Women in a Male Periphery — Experiences from the Scandinavian North. *Journal of Rural Studies*, Vol. 12, No. 3, pp. 259-271.

de Souza, C. S. and Leitão, C. F. (2009), *Semiotic engineering methods for scientific research in HCI*. Princeton, NJ: Morgan & Claypool Publishers.

de Lange, Paul, Mavondo, Felix (2004), Gender and motivational differences in approaches to learning by a cohort of open learning students. *Accounting Education*, Vol. 13, No. 4, pp. 431-448.

Evans, Judith (1995), *Feminist Theory Today*. London: Sage Publications.

Fausto-Sterling, Anne (1992), *Myths of Gender*, second edition. New York: Basic Books.

Feak, Christine B., Swales, John M. (2009), *Telling a Research Story: Writing a LIterature Review*. The University of Michigan Press.

Gorriz, C., Medina, C. (2000), Engaging girls with computers through software games. *Communications of the ACM*, Vol. 43, No. 1, pp. 42-49.

Grigoreanu, V., Brundage, J., Bahna, E., Burnett, M., ElRif, P., Snover, J. (2009), Males' and females' script debugging strategies. In *Proceedings of the 2nd International Symposium on End-User Development*. Springer, pp. 205–224.

Haraway, Donna (1988), Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective. *Feminist Studies*, Vol. 14, No. 3, pp. 575-599.

Harding, Sandra (1986), *The Science Question in Feminism*. Milton Keynes: Open University Press.

Harding, Sandra, ed. (1987), *Feminism & Methodology*. Bloomington: Indiana University Press.

Harding, Sandra (1991), *Whose Science? Whose Knowledge?* Ithaca, NY: Cornell University Press.

Harding, Sandra (1993), Rethinking Standpoint Epistemology: What Is "Strong Objectivity"? In *Feminist Epistemologies*, edited by Linda Alcoff and Elizabeth Potter. London: Routledge, pp. 49-82.

Harding, Sandra (2004), A Socially Relevant Philosophy of Science? Resources from Standpoint Theory's Controversiality. *Hypatia* Vol. 19, No.1, pp. 25-47.

Harding, Sandra, Norberg, Kathryn (2005), New Feminist Approaches to Social Science Methodologies: An Introduction. *Signs: Journal of Women in Culture and Society,* Vol. 30, No. 4, pp. 2009-2015.

Harshbarger, N.L., Rosson, M.B. (2012), wProjects: Data-centric Web Development for Female Nonprogrammers. *Proceedings of 2012 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 67-70.

Hartzel, Kathleen (2003), How Self-Efficacy and Gender Issues Affect Software Adoption and Use. *Communications of the ACM*, Vol. 46, No. 9, pp.167-171.

Heffler, Bo (2001), Individual learning style and the learning style inventory. *Educational Studies*, Vol. 27, No 3, pp. 307-316.

Henwood, Flis (2000), From the Woman Question in Technology to the Technology Question in Feminism: Rethinking Gender Equality in IT Education. *European Journal of Women's Studies*, Vol. 7, No. 2. pp. 209-227.

Hirdman, Yvonne (1988), Genussystemet — reflexioner kring kvinnors sociala underordning. (The gender system — reflections on women's social subordination) In Swedish, *Kvinnovetenskaplig Tidskrift*, Vol. 9, No. 3, pp. 49-63.

Hirdman, Yvonne (2001), *Genus – om det stabilas föränderliga former*. (Gender – regarding the changing shapes of what is stable) In Swedish. Malmö: Liber.

Huff, Chuck (2002), Gender, software design, and occupational equity. *SIGCSE Bulletin*, Vol. 34, No. 2, pp. 112-115.

Jianakoplos, N., Bernasek, A. (1998), Are women more risk averse? *Economic Inquiry*, Vol. 36, No. 4, pp. 620-630.

Ko, A.J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M.B., Rothermel, G., Shaw, M., Wiedenbeck, S. (2011), The State of the Art in End-User Software Engineering. *ACM Computing Surveys*, Vol. 43, No. 3, article 21.

Larsson, A., Lindström, E (2014), *Quality Attributes of Components in a System for User Driven Service Innovation*. Bachelor's thesis in Cognitive Science, Umeå University.

Lieberman, Henry (2001), *Your Wish is My Command: Programming by Example*. San Francisco: Morgan Kaufmann.

Lieberman, H., Paterno, F., Klann, M., Wulf, V. (2006), End-User Development: An Emerging Paradigm. In *End-User Development: Empowering people to flexibly employ advanced information and communication technology,* edited by Lieberman, H., Paterno, F., Wulf, V. Dordrecht: Kluwer.

Loewenstein, George (1994), The Psychology of Curiosity: A Review and Reinterpretation. *Psychological Bulletin*, Vol. 116, No. 1, pp. 75-98.

McDonald, T., Siegall, M. (1992), The effects of technological self-efficacy and job focus on job performance, attitudes, and withdrawal behaviors. *The Journal of Psychology*, Vol. 126, No. 5, pp. 465-475.

Mernik, Marjan, Heering, Jan, Sloane, Anthony M. (2005), When and How to Develop Domain-Specific Languages. *ACM Computing Surveys*, Vol. 37, No. 4, pp. 316-344.

Merriam, Sharan B. A. (1994), *Fallstudien som forskningsmetod*. Malmö: Studentlitteratur.

Meyers-Levy, Joan, Maheswaran, Durairaj (1991), Exploring Differences in Males' and Females' Processing Strategies. *Journal of Consumer Research*, Vol. 18, No. 1, pp. 63-70.

Mörtberg, Christina (1999), Technoscientific challenges in feminism. *NORA - Nordic Journal on Gender Research*, Vol. 7, No.1, pp. 47-62.

Nardi, Bonnie A. (1993), *A Small Matter of Programming: Perspectives on End User Computing*. Cambridge, MA: MIT Press.

Palmquist, Lena, Wennberg, Paula (2013), SATIN 2 — How to Support Self-Efficacy and Diversity in End-User Development. Unpublished

paper presented at the *Equality, Growth and Innovation conference* in Luleå, Sweden, October 10th, 2013.

Papert, Seymour (1980), *Mindstorms: Children, Computers, and Powerful Ideas*. Brighton: Harvester Press.

Prabhakararao, S., Cook, C., Ruthruff, J., Creswick, E., Main, M., Durham, M., Bumett, M. (2003), Strategies and Behaviors of End-User Programmers with Interactive Fault Localization. In *Proceedings of the* 2003 *in Symposium on Human-Centric Computing Languages and Environments*, pp. 15–22.

Repenning, A, Ioannidou, A., Zola, J. (2000), AgentSheets: End-user programmable simulations. *Journal of Artificial Societies and Social Simulation,* Vol. 3, No. 3.

Repenning, A., Ioannidou, A. (2006), What Makes End-User Development Tick? 13 Design Guidelines. In *End-User Development: Empowering people to flexibly employ advanced information and communication technology*, edited by Lieberman, H., Paterno, F., Wulf, V. Dordrecht: Kluwer.

Rosson, M.B., Sinha, H., Bhattacharya, M., Zhao, D. (2007), Design Planning in End-User Web Development. In *Proceedings of 2007 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 189-196.

Rosson, M.B., Sinha, H., Edor, T. (2010), Design Planning in End-User Web Development: Gender, Feature Exploration and Feelings of Success. In *Proceedings of 2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 141-148.

SCB (2012), *Women and Men in Sweden. Facts and figures 2012*. http://www.scb.se/statistik/_publikationer/LE0201_2012A01_BR_X10BR1201ENG.pdf accessed 2014-04-02.

Schirmer, C., Hecht, M., Maass S. (2011), Inspiring Innovative Practice: Gender and Diversity as Key Factors in Software Migration Processes. Paper presented at *Gender in ICT 2011* in Umeå.

Spinellis (2001), Notable design patterns for domain-specific languages. *The Journal of Systems and Software,* Vol. 56, No. 1, pp. 91-99.

Subrahmaniyan, N., Kissinger, C., Rector, K., Inman, D., Kaplan, J., Beckwith, L., Burnett, M.M. (2007), Explaining debugging strategies to

end-user programmers. In *Proceedings of the 2007 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 127–134.

Subrahmaniyan N., Beckwith, L., Grigoreanu, V., Burnett, M., Wiedenbeck, S., Narayanan, V., Bucht K., Drummond, R., Fern, X. (2008), Testing vs. Code Inspection vs. ... What Else? Male and Female End Users' Debugging Strategies. *Proceedings of CHI 2008*, ACM Press, pp. 617-626.

Torkzadeh, G., Van Dyke, T.P. (2002), Effects of training in Internet self-efficacy and computer user attitudes. *Computers in Human Behavior,* Vol. 18, No. 5, pp. 479-494.

Wajcman, Judy (1991), *Feminism confronts technology*. Cambridge: Polity Press.

Wajcman, Judy (2010), Feminist theories of technology. *Cambridge Journal of Economics*, Vol. 34, No. 1, pp. 143-152.

Webb, Christine (1993), Feminist research: definitions, methodology, methods and evaluation. *Journal of Advanced Nursing*, Vol. 18, No. 3, pp. 416-423.

Werner, L.L., Hanks, B., McDowell, C. (2004), Pair-Programming Helps Female Computer Science Students. *ACM Journal of Educational Resources in Computing*, Vol. 4, No. 1, Article 3.

Wiedenbeck, Susan, Engebretson, Alec (2004), Comprehension Strategies of End-User Programmers in an Event-Driven Application. In *Proceedings of the 2004 IEEE Symposium on Visual Languages and Human Centric Computing*, pp. 207—214.

Wylie, Alison (2004), Why Standpoint Matters. In *The Feminist Standpoint Theory Reader: Intellectual and Political Controversies*, edited by Sandra Harding. New York: Routledge, pp. 339–351.

Yin, Robert K. (2009) *Case Study Research: Design and Methods*, 4th edition. Thousand Oaks, CA: Sage publications.

Zeldin, A.L., Pajares, F. (2000), Against the Odds: Self-Efficacy Beliefs of Women in Mathematical, Scientific, and Technological Careers. *American Educational Research Journal*, Vol. 37, No. 1, pp. 215-246.

Zeldin, A.L., Britner, S.L., Pajares, F. (2008), A Comparative Study of the Self-Efficacy Beliefs of Successful Men and Women in Mathematics, Science, and Technology Careers. *Journal of Research in Science Teaching*, Vol. 45, No. 9, pp. 1036-1058.

Appendix A
Design Group Agenda, Weekly Meetings

1) Someone takes the minutes for the meeting, who?
2) The agenda and other questions to discuss.
3) Weekly scrum, round the table presentation of last weeks work related to SATIN. 3 minutes presentation 2 minutes for comments
    - I have worked with…last week!
    - I will work with … next week!
    - Do you see any impediments or problems for the work next week?
4) Discuss the "Quality criteria project"
    - The status and what to do next
5) Discuss the "Verification project"
    - The status and what to do next
6) Discuss the "Design language project"
    -The status and what to do next
7) Discuss the "Toolbox project"
    -The status and what to do next
8) Discuss the "Big-picture project"
    - The status and what to do next
9) Discuss the "Eco-system project"
    -The status and what to do next
10) Discuss the "Improvements on existing prototype project"
    -The status and what to do next
11) Other items to discuss

Appendix B
Test Protocol, Initial Test

SATIN Mock-up
http://www8.cs.umu.se/~bopspe/SATIN2/mock-up/mockup/

Goal

Have a large number (>50) of short impressions from various people on the agent-based mock-up paradigm.

Basic set-up

The test-taker and the test-subject are not in the same location. Each are seated in front of a computer with a fast internet connection.

The test subject runs a screen-sharing software and shares the screen to the test-taker (e.g. VNC or skype.

The test-taker runs a screen-capture software (e.g. Quicktime (MAC), or … (Windows)).

> Quicktime on the MAC:
> choose file>New Screen Recording;
> in the window that opens, click on the white triangle on the right, choose 'Internal-microphone' as audio-input.
> You will have to be in a quiet room, and NOT use headphones, so that you can record the audio from the test-subject as well as your own voice. (Test this out!)

The test-taker will record the whole session of approx. 5 minutes, including the introduction of the SATIN mock-up editor to the test-subject.

Before the actual recording sessions, do a test run to establish:

- all software runs fine
- a connection can be set up between the test-taker and test-subject
- the connection is fast enough
- the screen layout shows the relevant windows, test subject understands how to 'behave' during the session.
- the recording captures properly

Session
Once the screen-sharing is set up and the screen-recording is ready to go.
The test-taker starts the recording.
The first part of the introduction is to explain what is required of the test-subject (see below 'Introduction text').
The second part of the introduction is to explain what is SATIN (see below).
Continuously encourage the test subject to think aloud about the expectations, predictions, actions taken and what is seen on the screen. If a subject gets completely stuck, help a little to get things going again.
At the end of the session, make sure to ask and note down the following:
- age
- gender
- profession
- type of mobile-phone (and OS)
- owns a tablet-like device?
- works with computers?
- has any programming/scripting or similar experience

Introduction text
You are probably familiar with mobile devices such as smartphones and tablets (e.g. iPad). You probably also know that you can choose, download and install apps (applications) with particular functionality that you want on such a device. (Are you indeed, or not at all?)
In the SATIN project we are working to take this one step further and allow you to build your own (mobile) applications yourself, with functionality that you desire. We have therefore developed the SATIN editor, with which you can assemble your application. The editor that you will be working with is not yet capable of generating a working app, but it does show the basic functionality of how you could assemble an application.
In this session we will ask you to think aloud while you explore the SATIN editor to create your own app yourself. There is no good or bad, we are interested in what you think while you do things, why you do things, what you expect that things will do, etc. Please keep on thinking aloud at all times. Try to find solutions yourself if you get stuck. I can only help you a little bit.
The session will be short, approximately 5 mins.
Afterwards I'll ask you a few general questions about you.

Appendix C
Computer Self-Efficacy Questionnaire

I COULD COMPLETE THE JOB USING THE SOFTWARE PACKAGE...

		NOT AT ALL CONFIDENT				MODERATELY CONFIDENT				TOTALLY CONFIDENT	
		I-I				I-I					I-I
Q-1. ...if there was no one around to tell me what to do as I go.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-2. ...if I had never used a package like it before.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-3. ... if I had only the software manuals for reference.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-4. ...if I had seen someone else using it before trying it myself.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-5. ...if I could call someone for help if I got stuck.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-6. ...if someone else had helped me get started.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-7. ...if I had a lot of time to complete the job for which the software was provided.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-8. ...if I had just the built-in help facility for assistance.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-9. ...if someone showed me how to do it first.	YES.....1 2 3 4 5 6 7 8 9 10 NO										
Q-10. if I had used similar packages before this one to do the same job.	YES.....1 2 3 4 5 6 7 8 9 10 NO										

Free Comments:

Appendix D
Help-Experience Questionnaire

Questionnaire concerning how you experienced the kinds of help offered during this session

When I built the app, the following helped me according to the grading below

	NOT MUCH				MODERATELY					TOTALLY	DON'T KNOW
	I				I					I	
Q-1. ... being shown the functionality and features in the editor. SPECIFICALLY:	YES 1	2	3	4	5	6	7	8	9	10	☐
	NO										
Q-2. ... being shown an example of an app built from scratch. SPECIFICALLY:	YES 1	2	3	4	5	6	7	8	9	10	☐
	NO										
Q-3. ... the on-line help.	YES 1	2	3	4	5	6	7	8	9	10	☐
	NO										
SPECIFICALLY:											
Q-4. ...the design of the interface.	YES 1	2	3	4	5	6	7	8	9	10	☐
	NO										
SPECIFICALLY:											
Q-5. ...the possibility to ask questions throughout the session. SPECIFICALLY:	YES 1	2	3	4	5	6	7	8	9	10	☐
	NO										

Free comments:

Appendix E
Results of a semiotic inspection

Table E.1, Needs of the individual users found in the project application

Users' needs
- develop and affect mobile services
- express one's personality
- become motivated to develop apps
- to have a user-driven market place

Table E.2, Design-related needs found in the project application

Design-related needs
- developing a design ability
- being part of innovative development
- combining and coordinating innovations
- combining and building services from existing services
- one integrated portal
- creating unique experiences
- further refining existing services

Table E.3, Business-related needs found in the project application

Business-related needs
- offer and sell mobile services - facilitate marketing - design products and services - service generator that will be available in mobile phone - produce / consume services - support processes, transaction processes - be able to receive payment, or donate profits - contact producers - invest in services or components - dictate conditions - allocate revenues under agreements - marketing - seek partners - user-driven marketplace - executable services - searching for services - cohesive portal

Table E.4, Needs for specific tools found in the project application

Specific tools
- tools for developing user-driven services - finding tools supporting individual needs - tools for communicating ideas, refining ideas

Table E.5, Societal needs found in the project application

Societal needs
- a shift in actions and attitudes related to equality and diversity - the influence of women in IT

Table E.6, Technological considerations in the project application

Technological considerations
- with technological aids - via communication media such as the Internet - in the shape of a web site

Table E.7, Concrete design suggestions found in the project application

Design suggestions
- using visual programming, component-based - storyboarding - with a service generator that interprets the merged picture symbols and automatically creates the corresponding service - through searchable building blocks - through a well-defined relationship between visual concepts and underlying program designs

Table E.8, Usability criteria found in the project application

Usability criteria
- well-defined - without IT expertise - extremely easy

Table E.9, Methodological considerations in the project application

Methodological considerations
- through early establishment and habituation

Table E.10, Structure demands found in the project application

Structure demands
- search based on the service usage - through a well-defined relationship between visual concepts and underlying program designs / constructions

Table E.11, The individuals wishes found in the project application

The individuals wishes
- a desire among consumers and users to assist in the design of products and services - to express one's personality - individuals know best what they need

Table E.12, Market related wishes found in the project application

Market related wishes
- Affect services in one's mobile phone - Reach the entire market

Appendix F
Larger versions of the figures

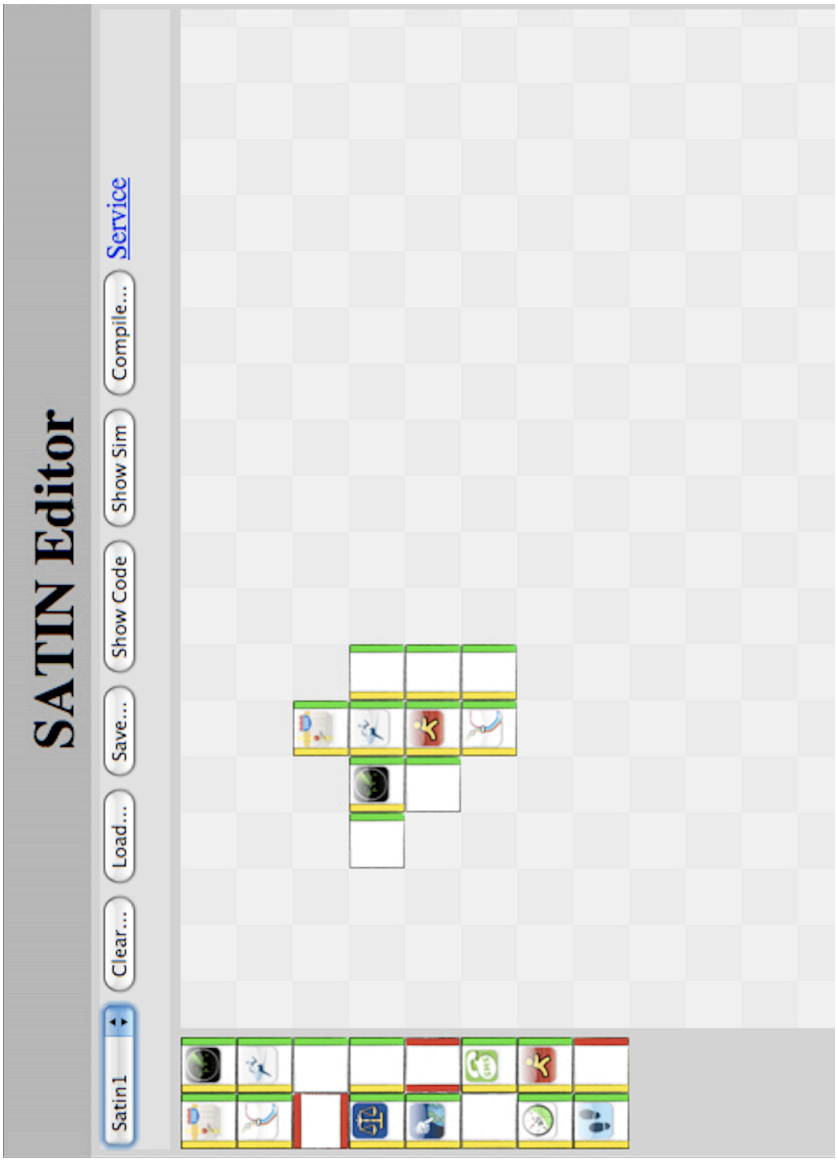

Figure 1.1: The first prototype of the SATIN editor—an important result of the SATIN 1 project—designed and implemented by Anders Broberg, Department of Computing Science, Umeå University.

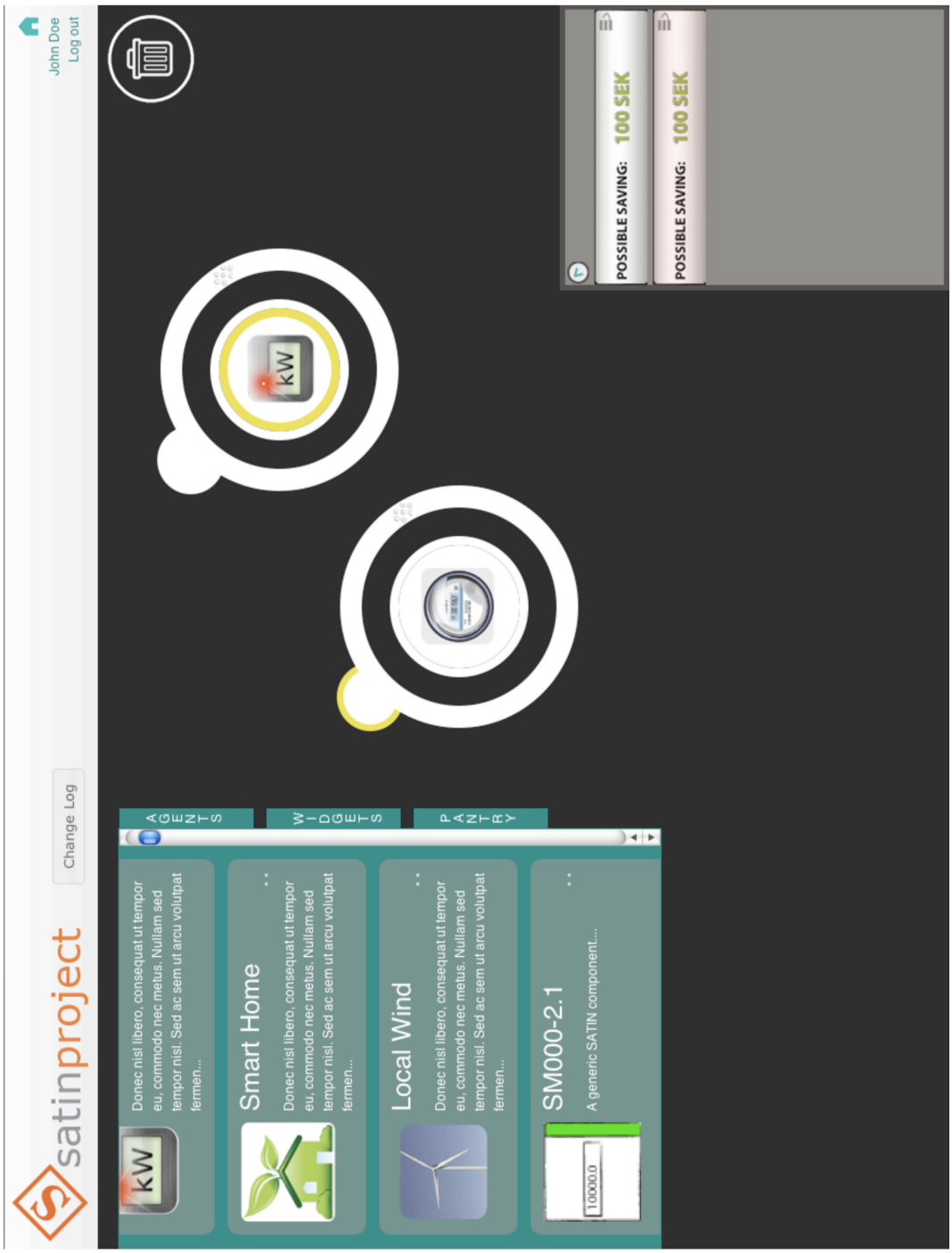

Figure 5.1 The SATIN editor mock-up from November 2011

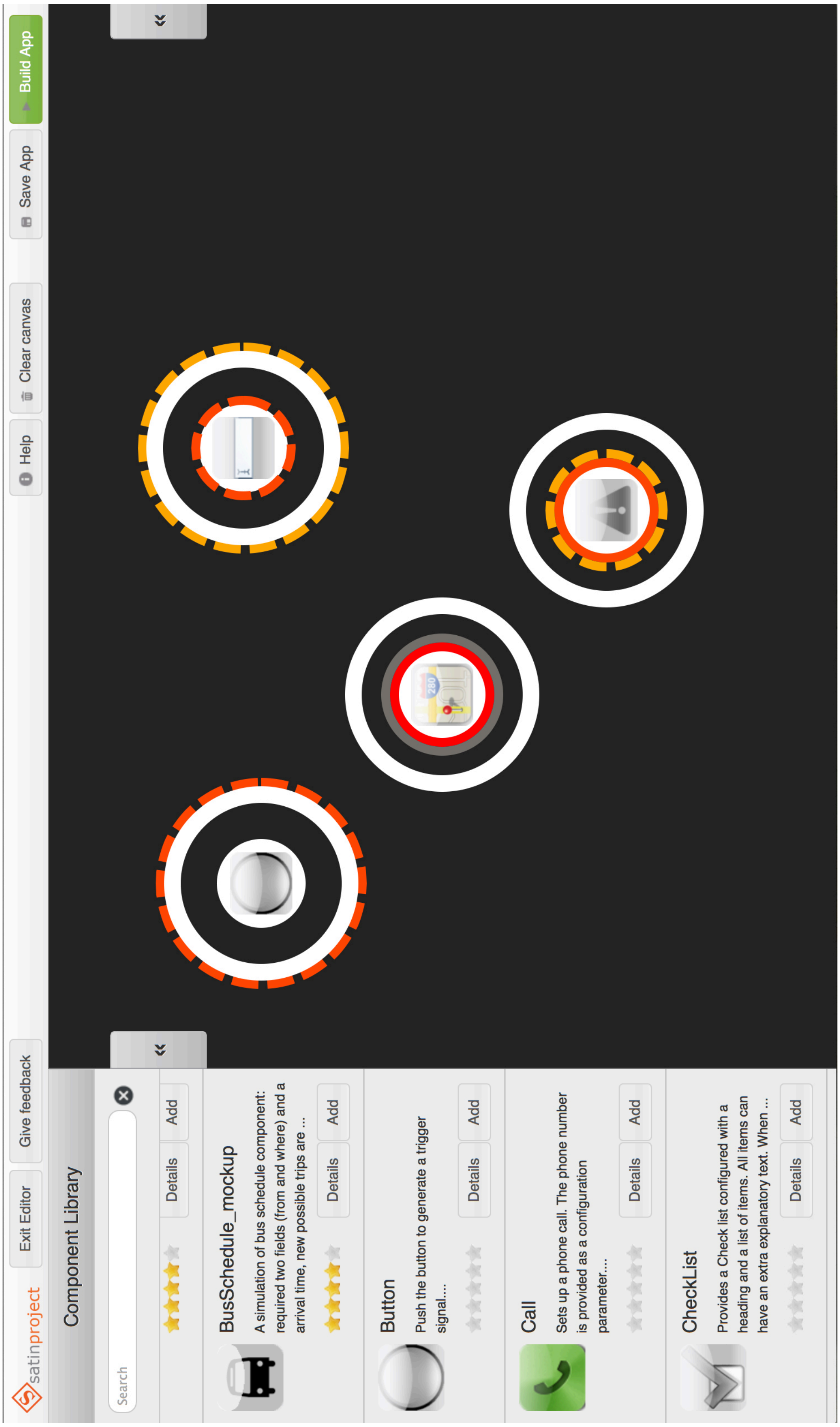

Figure 5.2. An enhanced version of the SATIN editor, a fully functional prototype. Connections between components are represented by colour and dashed patterns.

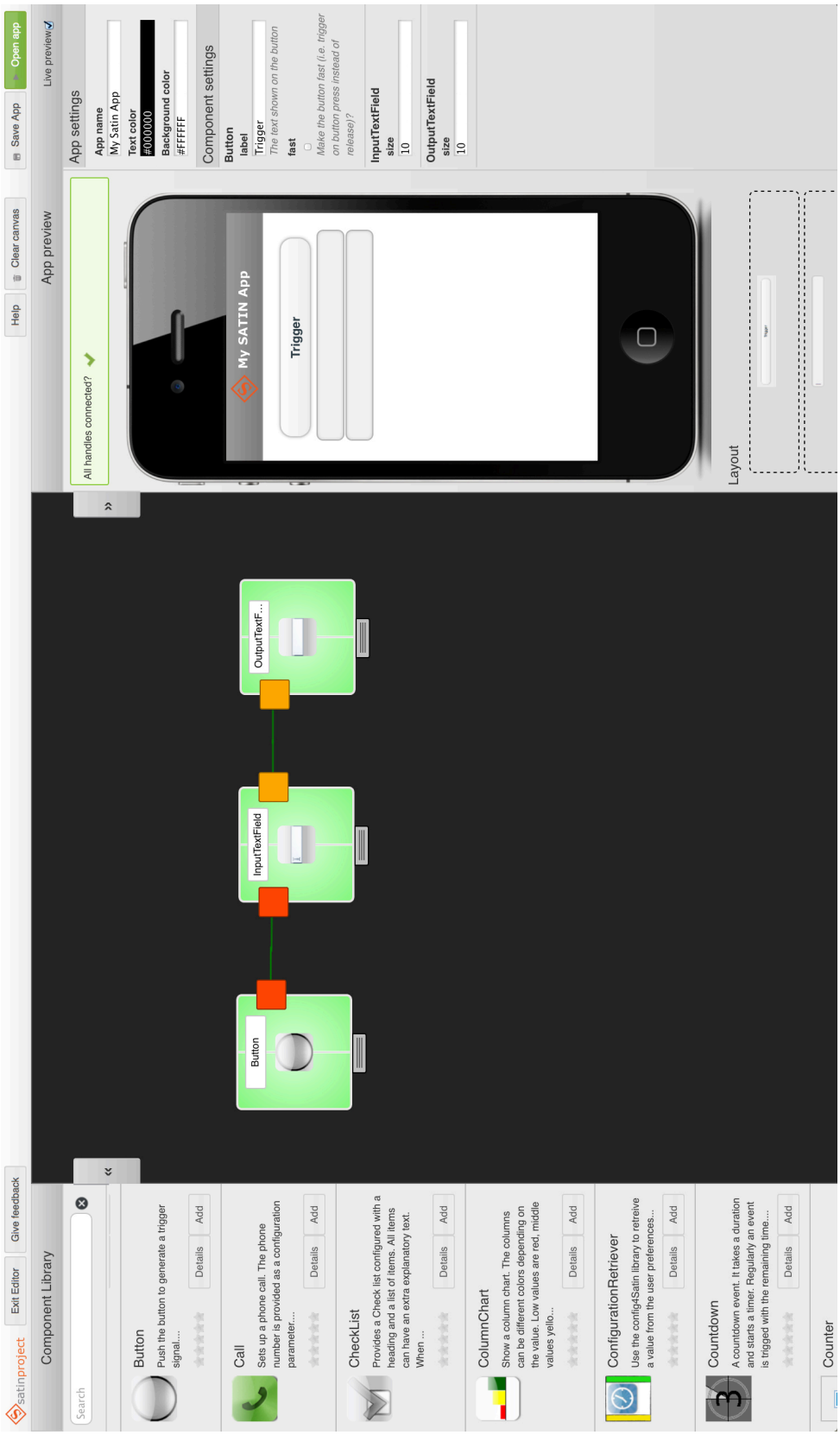

Figure 5.3. The most recent version of the SATIN editor, showing the square shape of components, the connections, the preview, and the settings on the far right.

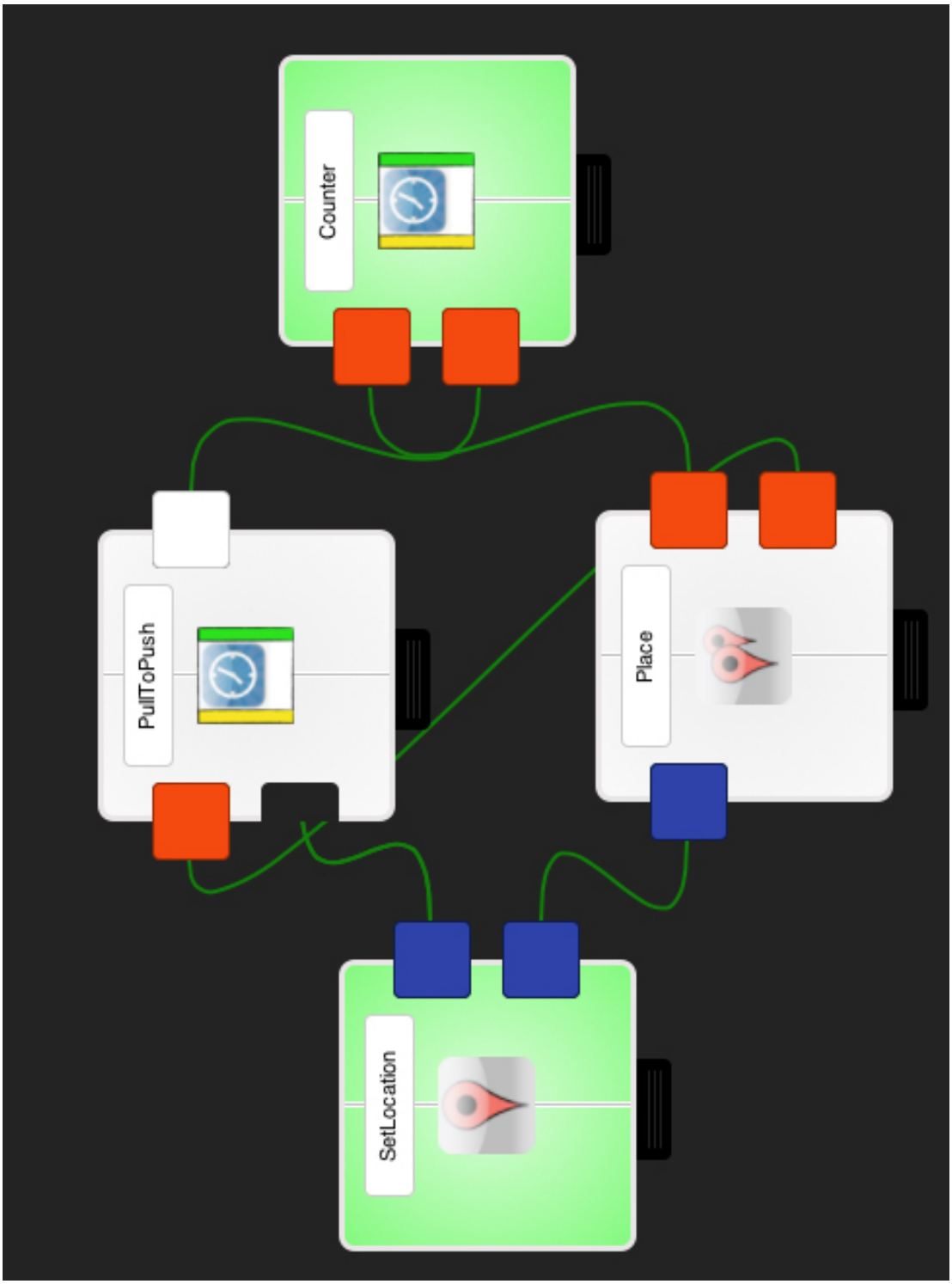

Figure 5.4. The "code" of this app is rather difficult to analyse due to connections crossing, and thus making them hard to follow.

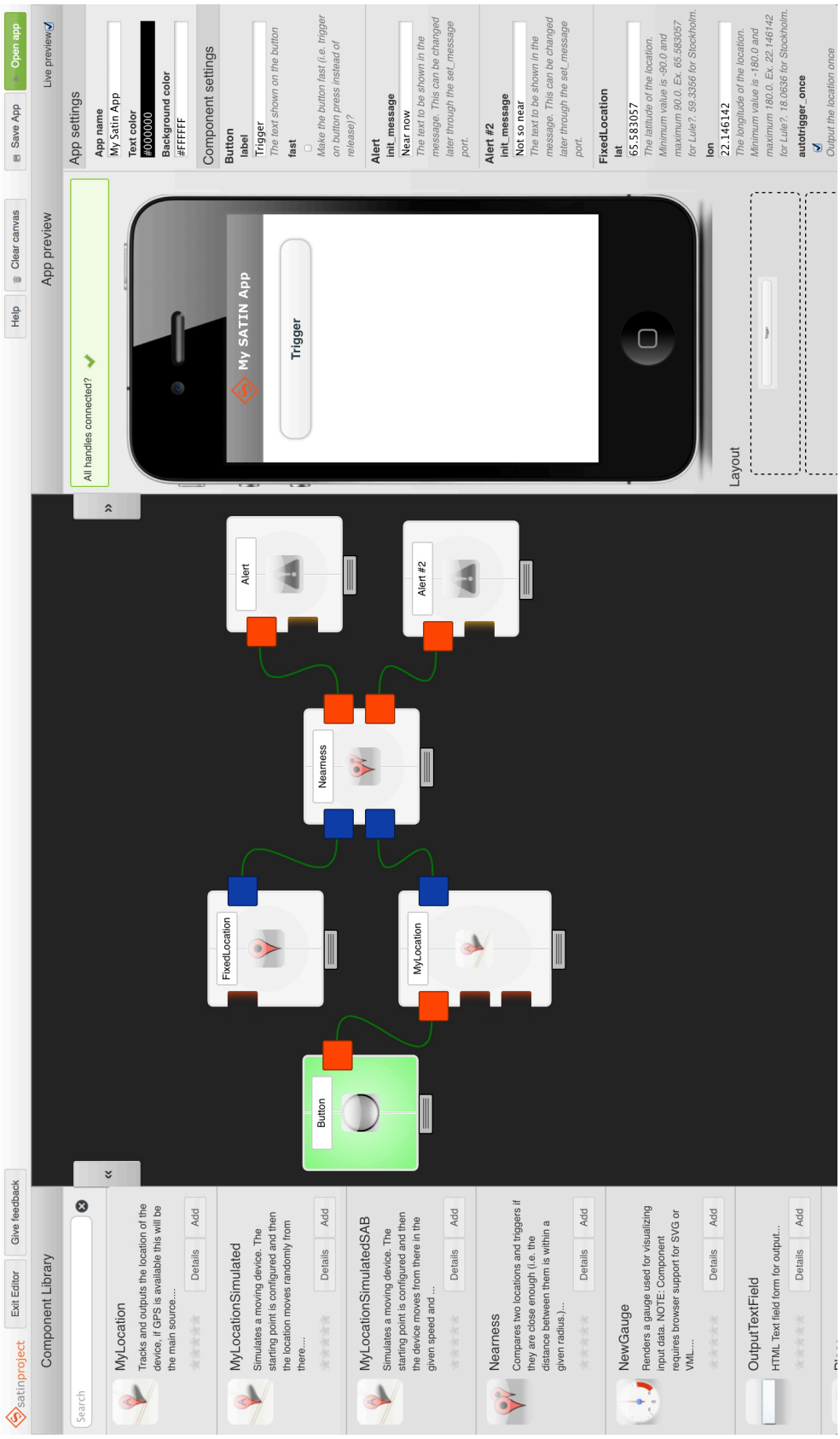

Figure 6.1. The most recent version of the SATIN editor.