Interaction Design of User-Supplied Data in Health Care Systems

Katarina Lejonlid

Institutionen för informationsteknologi
Department of Information Technology
Abstract

Interaction Design of User-Supplied Data in Health Care Systems

Katarina Lejonlid

Health care staff contribute to quality registers in order to improve health care and to provide data for research. The largest registry center is Uppsala Clinical Research (UCR) which supports over twenty national registers, for instance SWEDHEART and SENIOR Alert. The purpose of this thesis is to create a design hypothesis for a highly usable interface to the forms developed by UCR. The first step was understanding the potential for improvement through analyzing the system and observing users. This showed that the design should increase the learnability of the system and minimize errors. Two prototypes of an overview side panel were created that were presented to the users. An overview panel with process indication was the preferred design. The design was implemented in the evaluation phase using a simple form with grouped variables. The design hypothesis of an overview panel serves as a highly usable interface to the forms developed by UCR. The results showed that there are many aspects to consider, because the users have to gather information from many systems and sometimes paper forms. The implemented design could be presented to users and be further evaluated in an iterative design manner according to human-centered design.
## Contents

1 Introduction ........................................... 3  
   1.1 Background ........................................ 3  
   1.2 Purpose ............................................. 4  
    1.2.1 Limitations .................................... 4  
   1.3 Related Work ...................................... 4  

2 Theory .................................................. 7  
   2.1 Interaction Design .................................. 7  
    2.1.1 Norman Doors and Human-Centered Design ..... 7  
    2.1.2 Keystroke-Level Model ......................... 8  
   2.2 Design Thinking .................................. 9  
    2.2.1 Designing for Errors ......................... 10  

3 Method .................................................. 12  
   3.1 Understanding the Potential for Improvement ....... 12  
    3.1.1 Analysis of the Existing System ............... 12  
    3.1.2 User Observations ............................. 12  
   3.2 Creating a New Design ............................. 13  
   3.3 Evaluating the Design ............................. 13  

4 Results ................................................ 15  
   4.1 Analysis of the Existing System .................. 15  
   4.2 User Observations ................................. 17  
   4.3 Potential for improvement ....................... 22  
   4.4 Creating the Design ................................ 22  
    4.4.1 Prototypes .................................... 22  
   4.5 Evaluating the Design ............................. 27  
    4.5.1 User Input .................................... 27  
   4.6 Implementing the Design ......................... 27  

5 Conclusions ........................................... 31  

6 Future Work ........................................... 32  

7 Discussion ........................................... 33
1 Introduction

In order to improve health care nursing staff in Sweden is contributing to the quality registers on a daily basis. The data contained in the national quality registers is used by researchers after approval and ethical review. Uppsala Clinical Research (UCR) is the largest Registry Center in Sweden on behalf of the Swedish Association of Local Authorities and Regions. [18]

1.1 Background

The UCR Registry Center supports over twenty national registers in various fields. [18] There are several registers for a number of heart diseases [9]; register for health and care [6]; and a register for preoperative quality follow-up after surgery [4];

The Swedish Web-system for Enhancement and Development of Evidence-based care in Heart disease Evaluated According to Recommended Therapies (SWEDEHEART) is a national register of all patients hospitalized for acute coronary syndrome (ACS) or undergoing coronary or valvular intervention for any indication. [9] The register is the result of merging five existing registries with the goal to keep together the whole process of care. [17] Data are entered in one web-based form by different units and hospitals following the patient’s transfers. [9]

According to Jernberg [9] the register recorded in 2010 approximately 80000 cases each year and there are close to 2700 users, who mostly were physicians and nurses. The register is web based and available online, built on the technical platform OpenQreg. The data are filled out directly by the caregiver and relocated in an encrypted format to a central server. A personal identification number, that Swedish citizen have, name, address and hospital identity, are added to the register.

The main purpose of SWEDEHEART is “to support the improvement of care by providing information on care needs, given therapies and outcome.” In a long-term to reduce the occurrence of mortality and morbidity as well as to improve cost efficiency in coronary care. The users, who can be physicians, nurses or local managers, can obtain regular reports from the system helping them follow the processes of care and outcomes and compare with other caregivers over time. The system can provide them with reports that can be used in the documentation of the patients with information of treatments, interventions and a synopsis of the total care. [9]

Another national register of UCR, Internet-based SENIOR Alert, launched in April 2008, aims to provide a more systematic approach to improve the preventive care process for the elderly. To be included in the register the
elderly person must have a care contact. The most common group of users are nurses, thereafter nursing assistants and they register patients in nursing homes, hospitals wards and home care. The users can either file at the time of the patient encounter or use a paper form for later entry in the system.

The data are stored in a database at UCR with a direct admission to the Swedish population records. Nationally, the reports provided by SENIOR Alert are means for comparison between regions, counties, and municipalities. Between 2009 and 2014, the number of risk assessments were more than 1 000 000 totally. In 2014, 287 municipalities out of 290 used the register SENIOR Alert. The register supplies researchers with data that can be used in future studies for risk assessment and prevention to be able to offer better health care for the elderly.[6]

The Swedish Perioperative Register (SPOR), founded during 2011, is based on the Swedish Anaesthesia Register. It is sponsored by the Swedish Society of Anaesthesiology and Intensive Care Medicine. On a general level the goals of the register are to be an evaluation instrument of perioperative outcomes, to aid in the follow-up, to be a base for research and quality improvement. The register has been implemented in nearly all of the operative units in Sweden and it covers the whole perioperative process. It was anticipated that by the end of 2016, 95 percent of surgical procedures would be included.[4]

1.2 Purpose

The purpose is to create a design hypothesis for a highly usable interface to national quality register forms.

1.2.1 Limitations

There are other national registers that are developed by others than UCR, and they are not included in the thesis. The techniques behind the registers can differ and therefore the focus is the registers of UCR.

1.3 Related Work

Studies in the field of interaction design, designing for usability and related to medicine or user-supplied data were retrieved using an online scholar search engine, Google Scholar. The search keywords were "interaction design", "interaction design" and "health care" as well as "input methods".

There are several studies that design or re-design a health care application with a user-centered design involving the users in the development
process. The methods used were review documentation[11], interviews[11], focus groups discussions[11], questionnaire[1], usability tests [14] and user groups [15].

Khammarnia[11] states that more studies are needed to evaluate the application that was developed while the rest concludes that the user involvement has been beneficial for the study.[1, 14, 15] Niës[15] says that the solutions require the integration of Human Factors(HF) expertise. During the study the HF experts helped re-formulate the users’ needs building a bridge to the development team to fully understand the users’ needs.

Nelson[14] case report of software prototyping used an iterative development that through surveys received better than average usability and gained positive feedback from the users. To facilitate a user-centered design they used low and high-fidelity design prototypes. For the initial design process they used low-fidelity prototypes that were quickly sketched drawings on paper and then moved on to usability testing using high-fidelity interactive prototypes. The advantage of the rapid prototyping process is that it can identify major usability or interface problems quickly and cheaply, without the need to rewrite the code. [14]

A research case within health care information systems development in medical care experimented with participatory methods in a hospital. Martikainen [12] used user interface drawings for prototyping, and according to the feedback the drawings enabled them to describe their requirements and create common understanding with the system developers. Viitanen[19] showed that health care professionals in Finland are dissatisfied with current tools such as patient record systems and other software products. A national questionnaire study on clinical information and communication technology(ICT) environment, which health care professionals use daily, showed that there were several usability problems of clinical ICT.[19] There is a great deal of available health information management systems but still nurses and doctors are using traditional paper forms for planning and following up medical care. In order to collect user feedback the study used observations, interviews and simulations. The article also states that there is a huge amount of data in health records; “the problem is not data input but the access to the relevant information in the right situation at the right format”. [12]

As Johnson[10] says, there are many health care systems that are designed without concern towards user-centered design guidelines. A case study is presented as an example of a successfully applied framework for guiding the redesign process. Software developers in health care tend to ignore issues such as the users’ characteristics, tasks, preferences and usability overall. This results in systems declining productivity or with low usability. The cost of adjusting issues during development is ten times more that making the
adjustments in an early stage of design.

When designing it’s important to learn about the people using the applications. Novice users ask for continual feedback providing information while the expert users require fast response time and shortcut availability.

The article mentions that one of the best ways to gather large amounts of data concerning the users’ tasks is field observation, since it obtains detailed information about people’s tasks in their natural environment. Three recommendations are proposed for redesign process, collaboration among different parts of the redesign team, education of the matters of user-centered design and from the user’s point of view not to tolerate poorly designed systems. [10]
2 Theory

The starting point for creating a design hypothesis on a highly usable interface is the theory field of interaction design. The Keystroke-Level Model (KLM) is used to highlight the time aspect of performance of the design hypothesis. The design process presented in design thinking is used to give structure.

2.1 Interaction Design

The focus of interaction design is upon how people interact with technology. The goal is to enhance people’s understanding of what can be done, what is happening and what has just occurred. Interaction design extracts principles of psychology, design, art, and emotion. [2]

2.1.1 Norman Doors and Human-Centered Design

In our everyday life we are surrounded by design, good and bad. When the design works as promised we are satisfied but bad design can lead to frustration. An everyday object such as a door should not need an instruction or sign to be opened, still they can be very confusing. Norman [16], the author of “Design of Everyday Things”, has the phenomenon of poorly designed doors named after him as “Norman doors” since his problems opening doors have become widely known.

Machines follow a set of rules predefined by the engineers who designed it. When we try to follow these rules and fail we are, as operators of the machines, blamed for not understanding the machine and its rules. The field of human-machine interaction tries to reverse this situation; it’s the machine and its design that are the fault, not the people operating. The reason for the deficiencies in human interaction design are numerous but mainly it is the complete lack of understanding the design principles necessary for effective human-machine interaction.

Design of things is overall getting better and better as this field evolves, although the rapid rate of technology change outpaces the advances in design. The solution is human-centered design (HCD) an approach that puts human needs, capabilities and behavior first and then designs to accommodate those needs. [16]

A human-centered design is all about thinking about what people want to do and not what technology can do, designing new ways to connect with people, involving people in the design process and designing for diversity. [2]
2.1.2 Keystroke-Level Model

The Keystroke-Level Model (KLM) is a simple model expressing the performance outcome, meaning the time it will take an expert user to perform a specific task on a specific given system. Keystrokes, low-level operations, the user’s mental preparations as well as the system’s response will all together compose the model.

The model addresses only a single aspect of performance, that is time. There are more aspects to the performance of a user-computer system such as errors, learning, functionality, concentration, fatigue and recall. There is also not a single kind of user. They can differ in many dimensions such as the extent of knowledge of tasks, knowledge of other systems and experience with the system. Regarding the experience of the system they can be divided into three groups; novice users, who know little of the system; casual users with a moderate experience using it irregularly; expert users who know the system well and use it frequently.

The central idea behind the model is that the time to fulfill an assignment on an interactive system is determined by the time it takes to do the keystrokes. The keystroke-level model has several restrictions since it measures only one aspect, time, and it considers only expert users. It also considers only routine unit tasks and error free keystrokes.

The keystroke-level model consists of six operators, $K$ (keystroke or button press), $P$ (pointing to a target on a display with a mouse), $H$ (homing the hand(s) on the keyboard or other device), $D$ (drawing manually), $M$ (mentally preparing for executing physical actions) and $R$ (response time of the system).

**K (keystroke or button press)** This is the most common operator and it stands for keys, not characters. The motor skills of the user determine this operation which skills are decided taking one-minute typing tests. The total typing-test time will then be divided by the total number of keystrokes without errors.

**P (pointing to a target on a display with a mouse)** The time for this operator is constant and does not contain the mouse click itself. The mouse click will count as a separate operation ($K$).

**H (homing the hand on the keyboard or other device)** The operator consists of movement that appears between two devices of any kind and the final positioning of the hand.
D (drawing manually) This is a very specialized operator, restricted to the mouse.

M (mentally preparing for executing physical actions) It contents the time it will take the user to think or make decisions. The total amount of this operator will depend on how skilled the user is and the knowledge level.

R (response time of the system) The operator is relevant if the user have to wait for the system to respond and will depend on the system, the command and the context of it.

As previously mentioned the operator K depend on the skills of the typer where the best typer will have 0.08, an average skilled typist 0.2 and the worst typist and unfamiliar with the keyboard will have 1.20. The operator for pointing the mouse P will have 1.1.[3]

2.2 Design Thinking

In design the secret to success is to understand what the problem is. A thumb rule is never to solve the problem given to you because you first must be sure that this is the real fundamental root problem. As a good designer you have to try and understand what the real issues are. Two of the most powerful tools of design thinking are human-centered design and the double-diamond-diverge-converge model of design.

The double-diamond model of design, seen in figure 1, uses a double diverge converge pattern that was first introduced in 2005. This repeated divergence and convergence is important in properly determining the right problem to be solved and then the best way to solve it. The pattern is expanding the scope of the problem, diverging it to examine the fundamental issues that underlie it and then converge upon a single problem statement. During the solution phase they expand the space of possible solutions and then finally converge upon a proposed solution. [16]

The human centered design process takes place within the double-diamond-diverge-converge process. There are four different activities within the design process; observation, idea generation, prototyping and testing and they are repeated over and over until the desired solution. The observation phase is the initial research to understand the nature of the problem. The research is to find the potential customers in their natural environment and follow them in order to understand the real situations they encounter. The group of people being observed is not always of high importance, since what matters most is the activities to be performed. In same cases when a product is intended
for a subculture the exact population must be studied. The next step is the idea generation for potential solutions and there are several methods for example brainstorming, generating numerous ideas and to be creative without constraints, meaning not to criticize ideas in an early stage. Another rule to add is to question everything even what at first seems obvious because even the obvious can be subject of clarification.

A quick prototype or mock-up for each design idea is a way to test an idea. Prototyping during the problem specification phase is done mainly to understand the problem. In the testing it’s applicable to gather a small group of persons similar to the target population. Testing is done in the problem specification phase to ensure that the problem is well understood, then done again in the problem solution phase to ensure that the new design meets the needs and abilities of those who will use it. Iteration has a central role in the human-centered design because it enables continual refinement and enhancement. [16]

2.2.1 Designing for Errors

Occurrence of errors can depend on many reasons such as the nature of tasks and procedures requiring people to behave unnaturally. Multitasking and doing several things at once can be such an unnatural behavior. Multitasking can appear very efficient but in fact, all the evidence prove that on one hand, quality and efficiency decrease and on the other hand, errors increase. Interruptions are also a common reason for errors. There are two types of errors; slips and mistakes. Slips occur when the user intends one action but ends up doing something different. A slip can be holding the coffee in one hand and the milk in the other and putting the coffee in the refrigerator. Slips will occur most frequently when the mind is distracted. A mistake is
when the wrong goal is established. An example of a mistake is to compute in the wrong unity using pounds instead of kilograms.

Designing for errors is complicated since it is easier to design for the case where everything goes as planned. There are several things that can be done; understand the causes of error; design to minimize them; make it easier for people to discover the errors that do occur and make them easier to correct.[16]
3 Method

Three activities are needed to fulfill the purpose; understanding the potential for improvement; creating a new design; evaluating the design.

3.1 Understanding the Potential for Improvement

Understanding the potential for improvement is divided into two phases, analyzing the existing system and observing users.

3.1.1 Analysis of the Existing System

The first step in understanding the potential for improvement is to analyze the existing system. The platform used is QReg5, developed by UCR, in which a registration form was created as an example to elaborate in. The main programming language is Java, one of the most important languages for general-purpose programming as well as for computer science instruction. The user interface is developed in Wicket, a popular open source web application framework for the Java platform. The classes and inheritance hierarchy were thoroughly studied and a UML diagram over a class of variables was created. In order to get acquainted with Wicket and Java a Wicket tutorial was carried out as well as a preliminary application was developed testing some initial ideas of design. The Wicket theoretical model was studied through literature. Two registers, SWEEDEHEART and SENIOR Alert, were studied in a test environment to understand the interface.

3.1.2 User Observations

Observations of users were an important choice of method to understand the potential of improvement. As Benyon [2] says it is difficult to describe in words all the details and that the actual procedure may differ from the "official" one.

The users were recruited with the help of project managers at UCR. The requirement was to find users with different levels of expertise. The users themselves were helpful in finding new observations. The approach is similar to the so called snowball sampling meaning recruiting subjects through acquaintances. The first contact was a section at the Academic hospital of Uppsala. During the initial discussion it appeared that it was difficult to do as many observations as desired at this section. The majority of nurses only filled out forms irregularly during hours when the work load was low, which was impossible to predict ahead. On the other hand there
was an expert user with an extra responsibility for filling out the quality
register and a longer observation was conducted with the user during week
13. Next set of observations was, during week 15, at a section where severals
members of the team would fill out forms during surgical procedures with
patients.

The observed users were medically trained and employed as medical staff
in the Swedish health care system. They had different levels of experience
filling out quality registers forms. In total five users were observed, three
nurses and two doctors.

The users were initially presented with the thesis worker, the study pur-
pose and how the findings were going to be used. They also gave their consent
for documentation. The documentation was through fields notes and photos.
The users observed were sent the notes of the observations regarding the facts
and comments they gave. Photos are published with the users* consent.

3.2 Creating a New Design

According to Benyon [2] prototypes may be used to demonstrate a concept in
early design and are concrete but partial representations or implementations
of a system design. Therefore use of prototypes was chosen to gain input
from the users in an early stage of design. The representations are partial
because in an early stage it was interesting to present a concept with focus
on input and overview. The prototyping started after the first observation to
present a group of users with two alternative prototypes that could serve as
a starting point for discussion. When creating the design whichever register
form could have been used in the design. Since the users were familiar with
one register this particular register served as an example. The prototypes
were designed using a free online mock-up tool called Mockingbot [13] with
drag and drop elements in order to design the desired appearance of a web
application.

During the process KLM and design thinking was taken under consider-
ation. The results of the expert user observation helped formulate the real
issue regarding the design which is fundamental according to Norman in de-
sign thinking.[16] KLM was important in the aspect that the design would
preserve the time performance of the expert user.[3]

3.3 Evaluating the Design

A meeting with a group of users took place after the first observation was
made to evaluate the design during week 14. The meeting was at their
work place, a care unit, in the beginning of their evening shift. A small
conference room with a window was booked for this purpose. They were
three participants, working as nurses, and they were casual or novice users of
the register system. The meeting started off with a presentation of the thesis
worker, the purpose and approval of documenting with audio recording and
notes. A set of general questions were then asked regarding the time spent for
filling out forms, the situation in which they filled out forms and advantages
and disadvantages of the current form. After this initial discussion they
were presented with one design at a time, where the overview with process
indication was presented first, seen in figure 8. The prototypes were low-
fi d elity [2] since they were printed on large paper and there were four screens
available for each prototype to visualize the flow. After each presentation
the group had a discussion. They had the opportunity to directly comment
on the design. At the end they were asked which design they preferred.
4 Results

The results come from understanding the potential of the system, creating the design and evaluating it. The main components in understanding the potential for improvement of the current system were to analyze the system and making user observations.

4.1 Analysis of the Existing System

The system is build with Java and Wicket, where Java is used for the logic and Wicket for the user interface. The first part of the analysis was to learn the code platform in interaction with the Wicket framework.

A registration form is created using the proprietary registration API. The registration contains a tree structure of variables with rule handling and variable types, the regforms dynamic form library, and the registration process.

The VariableTree is an evaluation tree structure of variables. When evaluated it can generate user feedback and state, changing consequences, for instance to make a variable required, based on rules placed upon the same tree. The VariableTree is an abstract serializable Java class. Each variable is an instance of the Variable class, a subclass to VariableTree. The Variable class has several subclasses such as IntVar, TextVar and ChoiceVar. The UML diagram for VariableTree is shown in figure 2.

The variables are also affected by the rules that can be applied and their consequences. Rules are different types of evaluations such as comparing two values and checking the validity of a value. Consequences can make a VariableTree not applicable, applicable or required. For example a variable can be dependent on a previous variable if the variable is relevant or not. A variable can also be required by the user. These set of rules and consequences make the register forms dynamic and not static.

The RegForm is a component that contains the actual form and the field containers that corresponds to the variables that are to be handled. The constructor of the regForm will take a String id of the component, an IModel of the type RegFormModel and LayoutInformation. The RegFormModel aggregates other models commonly used by RegForm and other components that for example handles registrations, variables and input components. The LayoutInformation is an object containing the layout of the registration form.

The RegForm is created in a few steps, firstly by initializing the model needed for registration state and evaluation. The next step is to create a model to keep track of form feedback. A main RegForm model is then created with the state model and feedback model. Lastly the form instance
can be created with the RegFormModel in the constructor. The RegForm model binds together the state and feedback of the form that is created.

The registration process library aids in managing the registration process in a register. It is based on stateless4j, a lightweight Java state machine. The key-classes of the registration process are Step, Process and StateManager. Step is a step in the registration process and the StateManager governs the state of the process and its steps. A process can have the following steps; complete, incomplete or deleted.

The registration process manages the life cycle of a registration while the RegForm contains the actual form with its associated variables.

As described by Dashorst [5] Wicket is a part of the apache software foundation and aims to bridge the gap between object-oriented programming and HTTP, which is a stateless protocol. It provides a stateful programming model. In Wicket, following the object oriented design approach the concepts of application, session and request are translated into classes.

The central concepts in Wicket are components, markup, models and

Figure 2: VariableTree and its subclasses
behaviors. Components, like objects, are reusable software modules. The difference between objects and components is mainly that components encapsulate processes and can be perceived as end-user functionality. The component works together with the model and the markup. The markup is metadata that describes text, here HTML, and contains the bulk of what the user sees. A wicket id attribute is defined in a tag and attaches Java components to it. The model provide components with an interface to data and the concepts comes from the Model View Controller (MVC) pattern. [5] MVC is an architectural design pattern for interactive applications. MVC organizes an interactive application into three separate modules: one for the application model with its data representation and business logic, the second for views that provide data presentation and user input, and the third for a controller to dispatch requests and control flow. [20] The model represents the domain model which includes objects, the view renders UI elements and displays it while the controller uses the user input to update the model. The model in Wicket is more than the data the component is interested in, but a locator to the actual Model object.

Behaviors are the last of the central concepts in Wicket and are mainly used for modifying attributes of HTML tags and responding to events or calls to the components they are bound to. Behaviors much be attached to components and each component can have several behaviors. [5]

4.2 User Observations

Most of the staff observed would fill out about two forms a day while an expert could use two workdays a month to fill out forms. In order to fill out the forms the users used mainly the hospital information system called Cosmic and a web browser to open the quality register form. Sometimes other systems were needed such as Kovis, a information system in which the Emergency Room (ER) and the ambulance personnel scanned paper forms. Kovis only provided an image view. Sometimes when they needed to define which category a certain medication belonged to they searched an online medication list, called Fass, using a web browser.

One of the users was an expert with a particular responsibility for the quality register of the section and part of the employment was dedicated for this purpose. The user had one screen at hand when filling out forms. The expert user met with all the new nurses in their section to introduce them to the quality register. The nurses in the section always filled out a paper form at first because they didn’t have access to a computer when working at the care unit with patients. The use of paper forms in health care was also seen in Martikainen [12]. The paper form is a double sided form, filled
out by nurses and doctors, and contains the most important fields for the particular quality register. When working a night shift they had easier access to a computer to fill out forms. Sometimes the nurses and doctors at the care unit had already filled out the paper or online form. Even if there was no paper form filled out from the care unit beforehand, an empty paper form was used to take notes from Cosmic or Kovis before filling out the register.

The users at the surgery used two screens for filling out forms and there were also several screens for the examination and procedure. One nurse at a time was responsible for all the documentation during the operation including the quality register form. The nurse would fill out all the information and leave some fields blank for the operating doctor to fill out. Using two screens the user would have Cosmic at one screen and the quality register at the other and fill out the form while looking at the screen with Cosmic. In Cosmic they shifted between different parts of the journal, for example medication and test results. No notes were taken but there was a note with a patient’s height and weight during the observation. According to one nurse they sometimes took a note when they asked a patient something, otherwise they normally didn’t take any notes. The nurse started off with reading the patient’s journal.

When it was the doctor’s turn to fill out the form, they filled out the parts left by the nurse. None of them looked at the hospital journal. One doctor looked at a small sticker with patient information and the other at one of the screens showing the results of an examination.

When filling out the form most of the users used mouse and keyboard when filling out a text field or numbers, such as medicine dosages. The expert user used mostly the tab key and arrows when filling out the form (see figure 3).

Some different cases of filling out forms are presented visually to exemplify the observations findings. In case one (see figure 4) a nurse fills out the form for a new patient brought in for an examination in the operation room and has two screens. In case two a nurse fills out a quality register form and has one screen. The paper form has not been filled out so the nurse takes an empty paper form. See figure 5. Case 2 is a good example on how gathering data can be time consuming more than the data input itself, an issue pointed out by Martikainen[12]. The expert user had to first collect the information from several sources and transport them manually on a empty paper form in order to finally fill out the form. Giving the user the opportunity to use two screens could be a simple way to increase efficiency.
The expert user holds a pen in the left hand because of previously taking notes on an empty form. The left hand is placed on the tab key which jumps to the next field on press. The right hand is placed on the arrows, using them the answers from drop down menus will shift.

The user will sometimes have to move the right hand to the numeric keys when filling out test results or medication.
Figure 4: Case 1, nurse fills out form
Figure 5: Case 2, nurse fills out form
4.3 Potential for improvement

The main findings are:

- the QReg5 platform generates dynamic forms and supports navigation with tabs
- there are different levels of expertise among the users
- the conditions in which they fill out forms vary

The main potential for improvement of the interface would not be improving the efficiency since the platform already supports tabs. When creating a design idea one should consider all users, novice, casual and experts, and their different needs. The expert user will mainly use the keyboard when filling out forms and should still be allowed to. There are also many new users that need to learn the system. The novice user will, according to Johnson [10], require frequent informative feedback as opposed to expert users who will require rapid response time and the availability of shortcuts. The findings show that the potential for improvement is to increase understanding and avoid errors for novice or casual users.

4.4 Creating the Design

4.4.1 Prototypes

The design hypothesis was created using the double-diamond model of design, according to Norman[16]. A preliminary application was developed testing an initial idea of a flexible design with multiple views. It would allow the user to choose freely which variable to fill in. The form consists of three panels, two of them in the same row and a third in the next row (seen in figure 6). The left top panel would include all initial and empty variables, the right top the variables with a value and the bottom panel would be where the value is filled in. When clicking on an empty variable the variable would move to the editing panel and when a value was entered it moved to the right top panel.
The simple task of submitting a variable and moving to the next field was compared using KLM[3] in a regular form and in the flexible design with multiple views. As seen in table 1 and table 2 the task was completed in seven steps using multiple views and three steps in a regular form. The total time is 3.1 in option 2 for the multiple views and only 0.6 for the regular form.

### Flexible design with multiple views

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Point on top panel(P)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Click on top panel(K)</td>
<td>2P + 5K = 2<em>1.1 + 5</em>0.2 = 3.1</td>
</tr>
<tr>
<td>3</td>
<td>Press arrow to choose alternative(K)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Point on submit button(P)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Click on submit button(K) or tab and enter (K) (K)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Point on top panel for next(P)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Click on top panel for next(K)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: KLM encoding for Multiple views.
Table 2: KLM encoding for a regular form.

<table>
<thead>
<tr>
<th>Regular form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Press tab (K)</td>
</tr>
<tr>
<td>Press arrow (K)</td>
</tr>
<tr>
<td>Press tab (K)</td>
</tr>
<tr>
<td><strong>Total time</strong></td>
</tr>
<tr>
<td>$3K = 3 \times 0.2 = 0.6$</td>
</tr>
</tbody>
</table>

The result from this calculation was to reject the initial idea of a flexible design with multiple views. As Norman points out [16] the problem given to you is not always the real fundamental problem of the user, something that became clear during prototyping. Instead the design was reformulated into an overview presented as a side panel to the regular form. The overview can help minimize errors since multitasking and interruptions are sources of errors, as Norman [16] points out, and the overview can be an aid in finding the main theme again. The two prototypes are two variations of a side overview panel and both contain a search field on top of the form.
Overview with filled in, non-filled fields  The side panel includes the headlines of the form and a plus icon on the right side indicating the possibility to expand each headline to its containing fields. On the top of the panel there is a label called "Non-filled fields". The panel is separated in two sections where the bottom section has the label "Filled out fields". When the fields of a headline will be filled out the headline will move to the bottom section. This way the user will see which parts are already complete. The prototype is seen in figure 7.

Overview with process indication  The side panel includes the headlines of the form with a plus icon on the left in order to expand the headline. On the right side of each headline there is a badge with a red number, indicating the number of non filled out mandatory fields of the form. When the user will fill out the form the number in the badge will be updated until all the mandatory fields of a headline are filled out, then the badge will be replaced with a green check icon. If there are fields that are empty but non mandatory a badge with a gray number will appear. The prototype is seen in figure 8.
Figure 8: Prototype of overview with process indication

**Search field**  A search field is placed on top of the form to search through the form with auto-complete when the user types.

**Expandable headlines in the form**  Instead of showing the form all at once all sections but the current is shown. When the user completes a section the next will automatically expand. The user can also expand the sections by clicking at the plus icon on the right side of the headline.
4.5 Evaluating the Design

4.5.1 User Input

All of the participants were nurses and they filled out forms on an irregular basis but agreed upon consuming about one hour per week for the purpose. The group agreed upon that the current form for the particular register has a good layout. Two out of three users use mostly the computer mouse and sometimes the tab key when registering the form while one mostly use tab and arrow keys. In order to switch between the register in the web browser and the hospital information system the latter used alt key together with tab key.

The group of users was after the initial discussion presented with two prototypes, as previously seen in figure 7 and 8. The overall idea of an overview was considered good and also the search function on top of the form. They wished that the overview would scroll to clicked category or field. The search function was not very relevant for them as they didn’t need to move between different parts of the form. They expressed the need to search within the help information given by the form when they need to decide the type of medication given to the patience. They only need to fill out the category in which the medication belongs to. The expandable headlines in the form could be a good thing when used at the care unit but not so good when used during operation. This function maybe would not suit all the sections and forms.

For the prototype with process indication the group agreed on that it was helpful to see the number of non filled out fields. This prototype was considered to be the best of the two presented. The main reason was that the user could see the number of fields that were not filled out. This would be useful when the user has neglected one mandatory field and gets an error message when trying to submit. Currently it’s not easy to find a blank field.

4.6 Implementing the Design

The overview panel with process indication is placed on the side of a register form. To implement the overview panel there has to exist a form from which the panel could get information. Therefore a simple-fictional form was created in the main panel by creating a variable key, a variable and then adding the variable to the form. The form was created using the registration API without any changes to it.

In order to implement the overview panel with process indication two Wicket panels were created, GroupPanel and ListPanel. Each Wicket Panel consist of a Java class and a markup file. The GroupPanel contains instances
of ListPanel. The GroupPanel is placed on the right of the registration form and retrieves information from the form.

The next step in the implementation was to formulate several subtasks. The first task was to list all variables from the registration form in a panel. In the ListPanel the Wicket Component method onInitialize() is overridden and a ListView is created where each list item is populated. Each list item is a variable and the variable name has to be translated with the class TranslationService, which is a service for translation of both feedback messages and variables.

The next task is to show which variable in the list is filled out, meaning that the variable has a value separated from null. This information exists in the registration form which was created in the main panel of the system. In order to update the information in the overview panel the form needs to update the component in which the overview panel is in. This is achieved with an OnChangeAjaxBehavior(). A Wicket label with a check icon was added which is shown only when a variable has a value separated from null. The variables shown in the list should only be the ones that are required in the form, so the panel filters non required variables. Therefore the variable is only shown when it is applicable which also goes for the label with the check box.

Since the forms include many fields there’s a possibility to group them using LayoutInformation creating a headline for each group. The layout of the form is created in the main panel of the project where each variable is added to a particular group. The information about the layout of the form has to be retrieved by the GroupPanel. The overview panel shows the groups headlines. A list of headline labels is extracted from the layout and used in a Wicket Model. For every group in the GroupPanel a ListPanel is created showing the variables in the particular group. When creating a ListPanel a model of variable list is passed to the constructor of the ListPanel. The list of variables for each group is created in GroupPanel by extracting it from the LayoutInformation. Each group has a list of variables and each variable has a key. The key is then matched with the list of all variables in the form and those who match are added to the desired list.

When the variables are displayed as groups in the overview panel there is a need to sum the process indication on a group level. If the fields of a group are all filled out the group is complete and a check icon can appear next to the group headline. A private method in GroupPanel checks if all variables have a value separated from null and in that cases return a Boolean value groupIsComplete true. Another private method in GroupPanel will count the number of variables that have the value null and are applicable. If all the variables in a group are filled in and the group is complete a Wicket Label
with a check box is visible next to the headline. If the group is not complete another Wicket Label will be visible with a badge and a number, from the number of variables that still are empty.

If the user finds that there are blank fields the overview should provide a link between the panel and the form. This was achieved using an AjaxLink in the ListPanel with a Wicket Label that includes the variable name. The AjaxLink overrides the onClick behavior, what happens when the link is clicked on. A protected abstract class onVariableClick with a Ajax target and a variable was created in ListPanel and GroupPanel and implemented in the main panel of the system where the registration form is created. This procedure to create a protected class was chosen so that the ListPanel will not have to contain information on the form which exists in a higher level of the project. The variable clicked on will be be matched with the variable key and the Ajax target will set the focus of the form to the specific target which automatically will scroll the page to the variable in focus.

The overview should also at first show the headlines with the possibility to expand each group for its content. So when the ListPanel is created inside the GroupPanel it is not visible but when clicking on a plus icon the panel is visible and a minus icon replaces the plus. When the page renders a new instance of ListPanel is created every time. Therefore a map of Boolean holding the value of visibility as true or false is created in the GroupPanel. When the map is empty the value is false. A Wicket Label is created for expand and one for minimize and each of those contains the appropriate icon. The implementation result is shown in figure 9.
Figure 9: The overview panel implementation
5 Conclusions

The design hypothesis of the overview panel:

- increases understanding of the form for the novice and casual user
- decreases the occurrence of errors in the forms
- allows the expert user to navigate through the form as usual

The overview with process indication is an aid for the novice and casual user. The functionality prevents the users in making errors by making it easier to find blank fields in the form, by clicking on the overview panel. The placement of the overview panel on the right side of the form enables expert users navigating as usual in the form. Through fulfilling the above criteria the design hypothesis serves as a highly usable interface to the national register forms developed by UCR.
6 Future Work

There are several parts of the design hypothesis that could be further developed. There were some aspects from the created design that were not implemented, such as a search field for searching in the form and the implementation of expandable headlines in the form making the form easier to overview. A functionality in the overview panel was to also indicate non filled-out fields that were not mandatory using a gray colored number in a badge icon. The overview panel has a fixed position making it follow the form during scrolling, an option that could be on or off.

The implementation of the overview panel would allow users to elaborate with the form and see how the panel reacts when they start filling it out. This gives an opportunity to further evaluate the design with usability tests and other techniques used in the field of interaction design such as the think aloud method. [2]

The potential of improving the system in the aspects of making it easier to learn and prevent errors in the form could be further discussed and evaluated by users. Could the process indication for example be implemented directly in the form so the check box and number of empty fields are seen directly next to the forms headline? Or can the overview panel be further developed through user analysis and evaluation to better meet the user needs?
7 Discussion

This work highlights the need of user studies when creating new technology. The users’ daily lives must be the guidance in developing for people. The user input played an important part in finalizing a design hypothesis for a highly usable interface.

The users recruited for the user input represented all expertise, different professions as well as different situations in which the forms are filled out and therefore the results can be reproduced.

The intention was to make observations of users in different quality register forms but unfortunately some contacts didn’t lead to actual observations. The results could be applied on more registers but the best solution would be to complement the results with observations on other registers.

The implemented design was intended to be evaluated by users but they couldn’t find the time to participate. The health care personnel’s focus is, with righteousness, the patient and its needs and not serving the researchers and their million questions. Evaluating the implemented design to further improve it, would be according to the design process which is repeated over and over until the desired solution.[16]
References


