RECON

A framework for remote automated field evaluation of mobile systems

Tais Holland Mogensen - Christian Ølholm - Group 1070 - Aalborg University - June 7th 2007

AALBORG UNIVERSITY
DEPARTMENT OF ELECTRONIC SYSTEMS

Fredrik Bajers Vej 7C ▪ DK-9220 Aalborg East

Title

RECON: A framework for remote automated field evaluation of mobile systems

Period

February 2nd 2007 to June 7th 2007

Project group

07gr1070

Group Members

Mogensen, Tais Holland
Ølholm, Christian Rasmus

Supervisors

Jensen, Kasper Løvborg
Larsen, Lars Bo

Number of Copies

5

Number of Pages (report/total)

91/114

Finished

June 7th 2007

Abstract

Mobile devices, such as Smartphones and Personal Digital Assistants, are becoming more and more common, however the devices and applications on them are often evaluated with methods and techniques that are borrowed from traditional laboratory evaluation.

Mobile devices are inherently context dependant, so evaluating these devices in the field is an attractive way to go. However research indicate that the efficiency and effectiveness of field evaluation is poor, as they are timeconsuming and have an unknown added value compared to laboratory evaluations.

The purpose of this study was to investigate if the efficiency of field evaluation could be improved while mainting the same or better effectiveness by automating the evaluation. A working hypothesis was established and a range of research questions derived from it. In order to elucidate these research questions, a framework for automatic field evaluation called RECON was developed.

The RECON framework is capable of capturing usage, context and attitude information. Usage information is captured by inserting hooks in the application being evaluated. Context is captured by querying the State & Notification Broker and via third party API's and attitude is captured by presenting the test subject with a survey on the device at a specific sequence of events. These capabilities where put to the test in a in a range of experiments, done to shed light on the research questions.

The results of the experiments showed that an automated framework for field evaluation could make field evaluation more efficient, however further investigation is required to determine the effectiveness of the framework.

Preface

This master thesis have been completed on the 10th semester of the specialization Intelligent Multi-Media under the Department of Electronic Systems at Aalborg University in the period from the 2nd of February 2007 till the 7th of June 2007.

The report is primarily intended for persons with technical insight corresponding to that of the authors.

Report Structure

The report is comprised of an introduction and an analysis of state-of-the-arts systems, five parts and an accompanying CD. Each part documents a specific area of the project.

Introduction and State-of-the-art systems

First the problem domain will be introduced and the purpose of the study established in the Introduction. Based on the Introduction, a problem statement was established as a working hypothesis. After that an analysis of state-of-the-art systems for automated usability evaluation was done for inspiration and in order to avoid pitfalls.

Methods

The Methods part explore methods and techniques in areas such as context and capturing context information, usability evaluation and how to automate evaluations.

Design and Implementation

In the Design and Implementation part the development of the RECON framework is described, beginning with an overview of the architecture. Also, a more detailed description of the individual parts of the framework is done, describing which methods from the Methods part, that are implemented.

Experiments and Results

The Experiments and Results part document the experiments performed and results gathered in order to elucidate the research questions and the working hypothesis of the study.

Evaluation

The Evaluation part concludes the study with a discussion and a conclusion based on the research questions, the working hypothesis and the results of the experiments.

Appendix

Finally additional documentation that is not necessary for the understanding of the study, but is still relevant, is included in the Appendix.

Reading Instructions

The report is constructed in a successive order; meaning it is assumed that the reader has read the report from page one and forward. Whenever an abbreviation is made, the full word will be written out first followed by the abbreviation, like this: Personal Digital Assistant (PDA). Bibliography references are shown in square brackets, like this: [Nielsen, 1993]. Quotes from other authors or articles are specified in italics followed by a citation, like this: *Usability is...*[Nielsen, 1993].

Mogensen, Tais Holland

Ølholm, Christian Rasmus

Contents

CHAPTER 1

INTRODUCTION

Mobile devices such as Smartphones and Personal Digital Assistants (PDA's) is a rapidly growing business. In 2006 37.4 million Smartphones and 7.4 million PDA's were sold world wide, and these numbers are predicted to increase the next couple of years [Meyer, 2006].

Increased processing power, storage capabilities and bandwidth have made it possible to perform many tasks on mobile devices that were previously limited to desktop computers e.g. handling e-mails, handling appointments, surfing the Internet, leisure, entertainment and informal communication and so forth.

These property makes for challenging design and evaluation considerations for Human Computer Interaction (HCI) theorists and practitioners since mobility imposes significant cognitive and ergonomic constraints affecting device and application usability [Gorlenko and Merrick, 2003].

Most often mobile devices and applications have been evaluated with a range of different methods and techniques borrowed from traditional "desk-bound" usability evaluations [Kjeldskov et al., 2005]. The stereotypical approach is evaluation performed in a controlled laboratory environment with test subjects performing scripted tasks and "thinking out loud" while being observed by test monitors and recorded on video. On Figure 1.2 a picture from a laboratory evaluation. A usability evaluation usually have three activities [Ivory and Hearst, 2001], as can be seen on Figure 1.1. Test data is captured and subsequently analyzed by an HCI researchers, who in turn gives critique on the application or device.

Capture
Analysis
Critique

FIGURE 1.1: The three phases of usability testing.

When evaluating an application, it is important to test it in the situations and environment, or more specifically the context, that it is expected to operate in. The context can either be simulated in a laboratory or a test can be performed on location [Rubin, 1994]. To understand why context can be interesting when conducting usability evaluations, a definition of the word needs to be made. Context is, according to [Dey and Abowd, 2000], *"Any information that can be used to characterize the situation of an entity. An entity being a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*.

Considering that context will change alot while being mobile and this has an impact on the usability of the device, thereby affecting the application tested, this information is as important as the usability data captured from the tested application [Gorlenko and Merrick, 2003]. Despite this fact, only a few methods and techniques addressing this have been proposed [Kjeldskov et al., 2005].

Simulating these environments in a laboratory can be done, however doing this can be costly, depending on how realistic the environment needs to be. An attractive way to evaluate an application in the environments and situations a test subject will encounter in daily use, is evaluating it in the actual operating environment[Kjeldskov et al., 2005]. This is also known as field evaluation.

The stereotypical field evaluation method is done by letting a test subject perform a range of realistic tasks in an application while "thinking out loud", being monitored by test monitors and recorded on video. An evaluation involves up to four persons. A test subject operating the device and interacting with the application being evaluated. A test monitor asking the test subject questions and encouraging him or her to "think out loud". A second test monitor that note down interesting events and finally a cameraman that records the evaluation session [Kjeldskov et al., 2005]. As the name implies, a field evaluation is done by letting the test subjects perform tasks in a location most often chosen by the test monitors, whereas laboratory evaluation is done in a laboratory. In Figure 1.2 a picture from a field evaluation is shown, showing the test subject, the test monitors and the video camera.

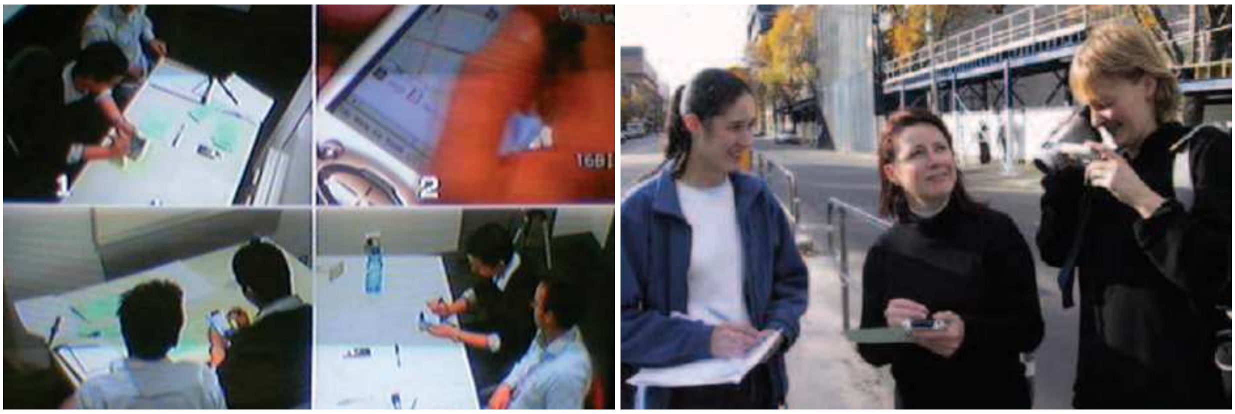


FIGURE 1.2: On the left: A traditional laboratory evaluation [Kjeldskov et al., 2005]. On the right: A traditional field evaluation [Kjeldskov et al., 2005].

When the evaluation is done, the data gathered is subsequently analyzed in order to uncover any usability issues in the application. Finally critique is given based on the issues discovered, as pictured on Figure 1.1.

Two main aspects should be considered, when choosing a specific evaluation method: efficiency and effectiveness. Efficiency defines how much effort is required to use the method. This is typically time required to perform the evaluation, but also the cost. The effectiveness of the test indicate how good the method is at uncovering issues and explaining them.

In 2003 [Kjeldskov and Graham, 2003] reviewed 102 research papers on mobile HCI from top-level conferences and journals between 2000 and 2002 and discovered that of all the papers, 41% involved evaluation of systems, and of these 41% only 19% were done through field evaluations.

In 2004 [Kjeldskov et al., 2004] investigated if there was any added value of performing field evaluation. Their results showed that out of 37 identified usability problems, only 23 were found in field evaluation, whereas 36 were found in a laboratory evaluation, indicating that field evaluation is less effective. The article concludes that field evaluation is not worthwhile considering the cost in terms of time, difficulty of performing the evaluation and usability issues discovered. However, several limitations to the results are also described, such as the fact that the test subjects did not use certain features of the test application during the field evaluation because the test monitor did not force the test subject to use the application. With some parts of the system not covered in the field evaluation, the comparison to laboratory evaluation, which covered the entire system, makes the results questionable. Also the fact that some features were used during laboratory evaluation and not during the more realistic field evaluation could be considered an indication of a usability issue regarding ease of access for the respective features. Finally other publications such as [Kjeldskov et al., 2005] and [Eagle and Pentland, 2006] shows that field evaluations can be as effective as laboratory evaluations with regards to detecting usability issues.

In an effort to decrease the time spent on field evaluation and still produce the same results, several research methods have been developed to automate the capture and analysis of usability data on both desktop computers and mobile devices [Ivory and Hearst, 2001]. The benefits of automating field evaluations is that the cost of capturing data and the amount of equipment required is very low, however the amount of data captured is very high, subsequently resulting in a time-consuming analysis [Castillo et al., 1997], which however can be made easier by automating this part of the evaluation as well. Another benefit of capturing data automatically is that a remote evaluation can be performed, which means that the HCI researchers and test subjects can be seperated in both time and space.

To HCI practitioners, an automated field evaluation framework could make field based approaches more applicable, by improving the efficiency and effectiveness of traditional field evaluations. Using these methods improves the scalability of field evaluation and makes it economically feasible for HCI practitioners to scale their evaluation both in amount of test subjects but also in the duration of the test period, while being able to explain usability issues by considering the context they occur in. An automated field evaluation framework would also allow HCI theorists to perform longitudinal experiments in order to uncover new aspects of HCI on mobile devices.

1.1 Problem statement

Mobile devices are becoming increasingly popular, but no de facto standard of evaluating the usability of applications on these devices has been established [Kjeldskov et al., 2005]. In order to evaluate usability data must be captured and subsequently analyzed before yielding any critique towards the evaluated application. The focus of this study will be on the capture of the usability data, as this is the initial step in an evaluation and it seems that existing methods lack the ability to capture information such as context. However aspects regarding analysis and critique will also be covered, as the captured data must be usable in an analysis, as was also shown on Figure 1.1.

As the use of mobile devices and applications are closely related to the user's context [Kjeldskov et al., 2005], a field-based approach was an attractive way to go. However [Kjeldskov et al., 2004] states that field evaluations are difficult to conduct, time consuming and with unknown added value. Although the results in the article are disputable, as they are based on evaluation of a single application and other publications disagree [Kjeldskov et al., 2005], the time consumption and difficulty of a field evaluation is unquestionable.

Therefore it is interesting to determine if it is possible to automate field evaluation in order to reduce the time spent on briefing and debriefing test subjects, and especially time spent on the evaluation session, by removing some or all of the test monitors. The difficulty mentioned in [Kjeldskov et al., 2004] result from the test monitors not being able to force the test subject to cover parts of the test application. As the evaluation only lasted two days, an extended evaluation might let the test subjects explore the entire application, and could possibly reveal usability aspects that is only apparent after some time, such as learnability or memorability of an application.

The study has both an engineering angle as well as a scientific angle. The purpose of the engineering angle is to analyze methods for extracting usability information, developing a framework for automated field evaluation and verifying that it works. The purpose of the scientific angle was to investigate if an automated field evaluation can be used as a replacement for traditional field evaluation or if it should be considered a supplement or a tool allowing more specialized evaluations. In order to do this, the following working hypothesis is defined:

> *It is possible to produce the same or more elaborate results in an automated field evaluation with the same or less effort as in a traditional field evaluation.*

By working hypothesis is meant a hypothesis that is not necessarily answered, but instead gives inspiration to a range of research questions that can provide a deeper understanding of the hypothesis and an indication as to what the answer could be. By results is meant data which holds evidence to the (lack of) usability of the application and by automated is meant an autonomous framework, that only need manual setup, analysis and critique, and by effort is meant time spent performing the evaluation.

In order to shed light on this working hypothesis, a proof-of-concept framework has been developed. The framework has been named RECON, which is a contraction of REmote and CONtext. The word RECON is also an abbreviated version of Reconnaissance, that means ”a preliminary survey to gain information” [Merriam-Webster, 2007], which also suits the purpose of the framework well. The main goals of the framework is being able to capture usage, context and attitude information and should be time efficient to use.

The results of this study will benefit HCI researchers, by providing a new framework that allows for the capture of information, that is can not be captured with current state-of-the-art systems. Additionally the result will show if it is possible to improve the efficiency and effectiveness of field evaluations by automating them, making them more competetive to laboratory based approaches.

CHAPTER 2

STATE-OF-THE-ART SYSTEMS

This chapter will describe existing state-of-the-art frameworks for automated usability evaluation, for inspiration and to avoid pitfalls when developing the RECON framework. Finally the frameworks introduced in this section will be compared to the RECON framework in the discussion.

Three frameworks have been chosen for further analysis. These frameworks have been created for remote automated evaluation of test subjects. The framework were chosen based on the litterature study, and the fact they are very different in their architecture and way of working, which gives a wider picture of methods to choose when creating a framework. As only three frameworks are investigated, this is not an exhaustive survey.

In Section 2.1 the EDEM framework is described. What makes the EDEM framework interesting, is that it captures user interface (UI) events on the device, and uses an on-device data analysis, in order to achieve a higher abstraction level for events. Doing this allows for easier analysis and storage of the data gathered. The framework can also present a survey for the user to fill out at a specific event, in order to capture the users attitude.

WebQuilt, described in Section 2.2, is a framework that captures usability data in a unobtrusive way by operating between a client and a server. The WebQuilt framework is also platform independent.

Section 2.3 describes the ContextPhone framework, which is not a usability evaluation framework, but can be used to capture a range of interesting context parameters such as location, phone calls and messages, nearby people and many other.

All of the frameworks have been used in an actual evaluation or research project in order to verify their usefulness. In [Hilbert and Redmiles, 1998b] EDEM is used in a proof-of-concept research project, to demonstrate large-scale remote capture of usability data. The study concludes that is possible to perform such a project, and that evaluations using an automated framework scale well and can be used for longitudinal evaluations.

In [Waterson et al., 2002] WebQuilt is used to evaluate an application and compared to a traditional laboratory evaluation. The study showed that click streams and remote evaluation are very useful for evaluating the content of a web user interface, however also notes that by using a proxy approach WebQuilt can not detect issues that does not involve communication through the proxy.

Finally the ContextPhone platform has been used as a research tool for studies of mobility patterns and social network analysis in [Eagle and Pentland, 2006]. The article concludes that the data captured during the study was unprecedented in both magnitude and depth.

In the following sections, the three frameworks EDEM, WebQuilt and ContextPhone, will be described.

2.1 EDEM

The EDEM framework is based on capturing the click stream generated by a user interface. The data is captured by EDEM on the test subjects desktop computer, and after some pre-analysis sent to a central server (Figure 2.1). To lower the bandwidth used by the framework an abstraction is made. The abstraction is basically a combination of one or more UI events to form a single high-level event such as "Opening the print dialog".

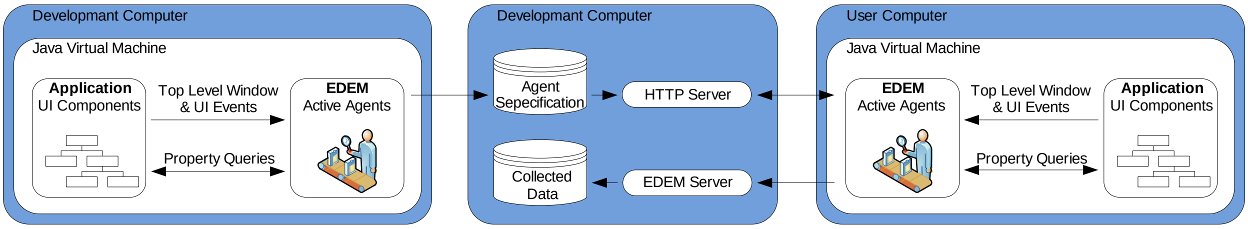
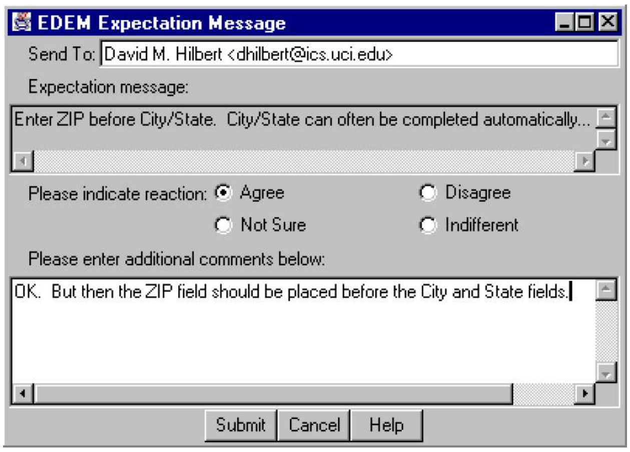


FIGURE 2.1: The basic architecture of EDEM (Modified from [Hilbert and Redmiles, 1998a]).

It is possible to collect usability data with or without user interaction with EDEM. The system can be configured to collect usability data automatically (by simply capturing predefined events) but also to collect user comments by prompting the user when a certain event happens. To decide if an event is expected or not EDEM uses software agents [Hilbert and Redmiles, 1998a]. The user comments are entered in a form, as seen on Figure 2.2, and afterwards sent to the HCI researchers by e-mail.

By collecting user comments, EDEM is able to provide the HCI researchers with information about the test subjects attitude and opinion regarding the evaluated software. This information is normally not available when using an automated evaluation where only the click stream generated is stored [Hilbert and Redmiles, 2000]. However this is also intrusive, in that it can distract the user from his current task.

EDEM is based on a remotely configurable agent based architecture which allows the HCI researchers to reconfigure agents on the fly. The configurations are done by designing a new agent in an agent editor as shown on Figure 2.3 and afterwards deploying it to the device.



Send To: David M. Hilbert <dhilbert@ics.uci.edu>

Expectation message:

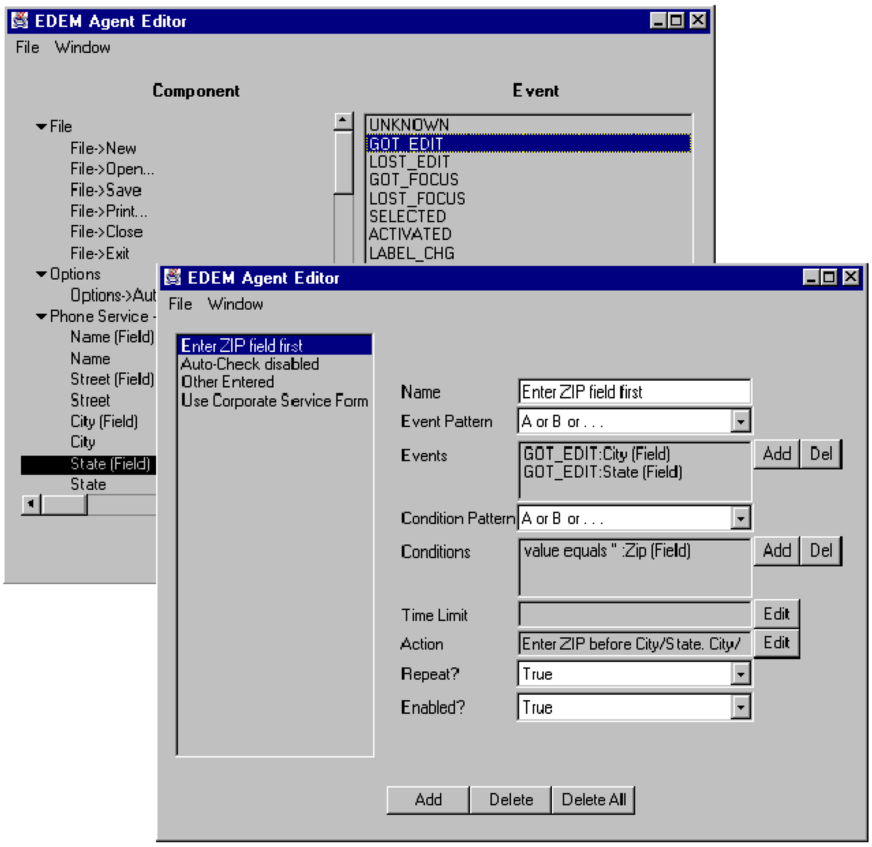
Enter ZIP before City/State. City/State can often be completed automatically...

Please indicate reaction: ● Agree ○ Disagree ○ Not Sure ○ Indifferent

Please enter additional comments below:

OK. But then the ZIP field should be placed before the City and State fields.

FIGURE 2.2: Screenshot of the user comment dialog from EDEM [Hilbert and Redmiles, 1998b].



Name: Enter ZIP field first

Event Pattern: A or B or . . .

Events: GOT_EDIT:City (Field)
GOT_EDIT:State (Field)

Condition Pattern: A or B or . . .

Conditions: value equals " :Zip (Field)

Time Limit:

Action: Enter ZIP before City/State. City/

Repeat?: True

Enabled?: True

FIGURE 2.3: Screenshot of the EDEM Agent Editor [Hilbert and Redmiles, 1998a]

2.2 WebQuilt

WebQuilt [Hong et al., 2001] was originally build for automatic web usability evaluations on desktop computers but has also been used for usability evaluations on mobile devices [Waterson et al., 2002]. Because of the architecture in WebQuilt it is relatively simple to integrate with almost any system. WebQuilt is developed to function as a proxy server with the capability of logging all communication (Figure 2.4).

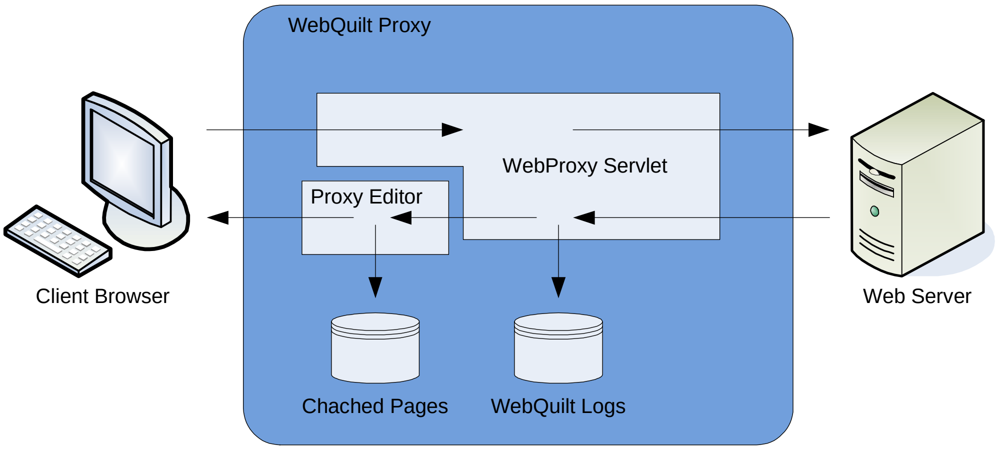


FIGURE 2.4: The basic architecture of WebQuilt Proxy (Modified from [Hong et al., 2001]).

Normally a proxy server is expected to function transparently from the client's point of view, but WebQuilt is in fact the only thing the client sees. When a client requests a web page he actually connects to WebQuilt and in the HTTP request sends the desired URL as a post or get parameter. WebQuilt afterwards fetch the web page and modifies all links to point at it self e.g.

```
http://www.yahoo.com
```

would become

```
http://webquilt-address/webproxy?replace=http://www.yahoo.com
```

(when WebQuilts location is: `http://webquilt-address/`) [Hong et al., 2001] before returning it.

The modification of links takes place in the component denoted ”Proxy Editor” on Figure 2.4 and is performed in order to ensure that all requests pass through WebQuilt. The translation of links is furthermore done in order to add tracking information which is necessary in the logging component denoted ”WebProxy Servlet” on Figure 2.4. The tracking information is necessary for the ”WebProxy Servlet” to be capable of producing a sequence of linked events, as the HTTP protocol used is state-less.

2.3 ContextPhone

ContextPhone is a platform that can be used when developing context-aware applications for mobile devices. As such the platform is not developed specifically for usability evaluation, however with the logging capability that is part of the platform, it proves to be a useful tool for studying mobility patterns [Raento et al., 2005]. Compared to EDEM and WebQuilt, ContextPhone captures information about the users context instead of focusing on usage information. The philosophy of ContextPhone is to supply context as a resource, meaning that the context information should be easily understandable for humans. It should incorporate existing mobile devices without the need to add additional hardware. It should offer fast interaction and unobtrusiveness, meaning that it should not interfere with the users interaction with the application. It should ensure robustness, by incorporating watchdogs that will recover the framework in case of a crash. It should let users control seams, which is gaps in interaction. The framework should emphasize timeliness, meaning that the response latency of the framework must should be low. And finally the ContextPhone framework should enable rapid development, by being able to easily add new data sources and by being able to easily create new applications.

The ContextPhone is built as a set of C++ libraries that work on mobile devices using the Symbian operating system and the Nokia Series 60 Smartphone platform [Raento et al., 2005].

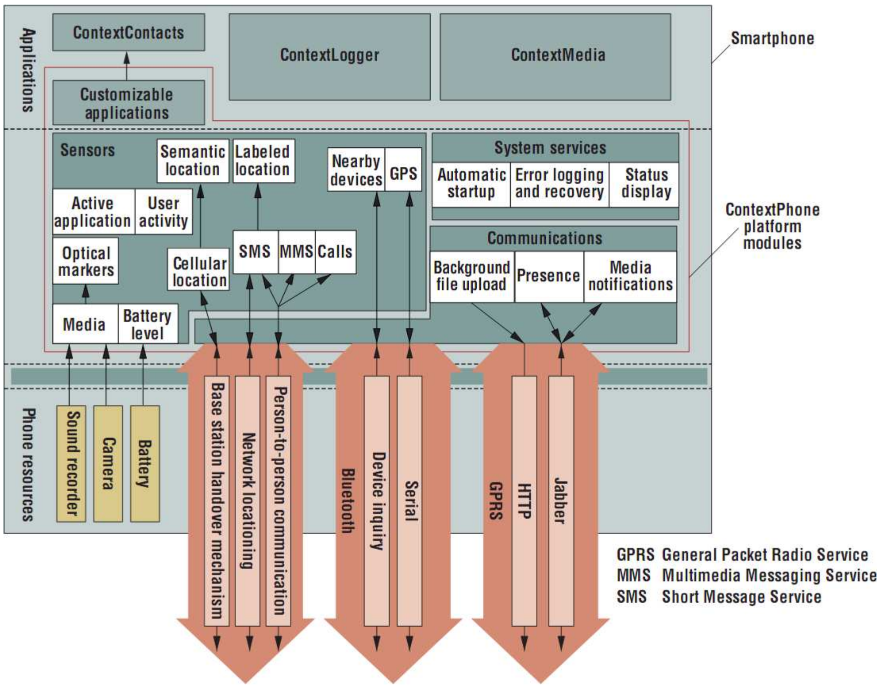


FIGURE 2.5: The ContextPhone platform architecture [Raento et al., 2005].

ContextPhone platform consist of four modules, which can be seen on Figure 2.5. The modules are:

Sensors that acquire context information.

Communications which is used for exchange of information.

Customizable applications are applications included with the platform that can replace build-in applications on the phone.

System services which handle background services, error logging, recovery and status.

The platform supports four sensor types: location, user interaction, communication behaviour and physical environment. The location can be based on the Global System for Mobile Communications (GSM) cell information and Global Positioning System (GPS) via a Bluetooth GPS receiver. User interactions is done by capturing information about the current application, idle/active status, phone profile, battery level and media devices. Communication behaviour is measured by capturing information about phone calls and sent and received messages. Finally attributes of the physical environment is done by capturing information about nearby Bluetooth devices and using optical marker recognition using the built-in camera.

2.4 Summary

Several frameworks already exist for automated usability evaluation of applications and they use very different methods of extracting their data. In this short survey, the focus was on three frameworks called EDEM, WebQuilt and ContextPhone.

All frameworks have proven to be viable for usability evaluation, by being able to capture information about test subjects use of an application, and they allow longitudinal evaluation and scalability. However results from [Waterson et al., 2002] showed that proxy-based frameworks such as WebQuilt is only useful for capturing application related usability data, and that in order to capture more device specific data such as context, the framework must be present on the mobile device.

EDEM allows a HCI researcher to present a survey to the user at specific events, and the framework can also abstract low-level events to form high-level events. This means that the researcher can query the users about their attitude or opinion towards certain elements of the application. That way some of the attitude information, that usually can be extracted in traditional field evaluations by asking a user to ”think out loud”, can also be extracted in an automated framework.

Finally one of the shortcomings of EDEM and WebQuilt is the fact that they do not gather any information about the users context. This is however possible with the ContextPhone platform.

Combining the best of all approaches, allows the capture of usage information such as use patterns, capture attitude as can be done with EDEM and capture context as can be done with ContextPhone.

In the following part of the report, the methods used in these frameworks and similar methods will be further analyzed, in order to select specific methods for implementation in the RECON framework.

Part I

Methods

This part explores methods and techniques in areas such as context and capturing context information, usability evaluation and how to automate evaluations.

Chapter 3

Context

In the Introduction, context and context awareness were briefly discussed. ContextPhone was also described in Section 2.3, which is a framework for making applications context-aware. The purpose of this chapter is to further define what context is, how it can be captured and why it is interesting in a usability evaluation.

Definition

In general context is defined as *"Any information that can be used to characterize the situation of an entity. An entity being a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"* [Dey and Abowd, 2000]. Depending on academic areas, the word "context" can have different meanings, however in this study the term refers to task context. Task context means everything related to the situation in which a task is performed. These context parameters could be location, network quality of service or environment and so forth.

In traditional evaluations, either in the laboratory or the field, test subjects are often encouraged to "think out loud", which makes it easy to for HCI researchers to explain a usability issue. "Think out loud" makes the evaluation very artificial to the test subject, as they are forced to do something they would not usually do. Instead of this method, usability issues could possibly be explained by using context information. For example a wrong selection in an application on a mobile device, could be explained, if it was known that the current season was winter, the test subject was located outside, and the temperature outside was below freezing. In such a case, one could conclude that either the buttons on the device were to small, or that the user were wearing gloves. In any case, the context information makes it possible to narrow down possible causes for a specific usability issue.

3.1 Context awareness

Context-aware computing is characterized as systems being able to adapt their behavior to an environment it has little or no control over. Context awareness is a term often mentioned together with mobile computing because of the changing contexts occurring when being mobile [Chen and Kotz, 2000].

The purpose of making systems context-aware, mobile as well as stationary, is to be more effective and adaptive to users information needs without consuming too much of a users attention [Chen and Kotz, 2000]. Examples of such applications include mobile tourist guides, such as [Kjeldskov et al., 2005] and [Cheverst et al., 2000], presenting the user with information depending on location, and adaptive applications on mobile devices that change font size and color depending on user activity and light level.

To make an application context-aware it needs to be able to sense context-relevant information from the environment and situation. This can be done by either querying the user or capturing the information automatically. As the purpose of context awareness in most cases is to not consume user attention, the focus will be on automated capture. Which context parameters that needs to be captured depends on the type of application i.e. the aforementioned mobile tourist guide might not necessarily need to capture the air pressure, but more likely the position.

In the following section the paramters that are most interesting and relevant to usability evaluation are described.

Location

Location is considered as one of the most important context parameters [Gorlenko and Merrick, 2003]. Many of the tasks people perform are by nature location dependant, so this information can be used to reason about the users current task and environment.

An obvious choice for automated location is GPS, however this technology does not work satisfactory indoor because of poor penetration and radio wave reflection, and would in most cases require an external unit as few mobile devices have a builtin GPS today. An alternative solution could be positioning by cellular data such as GSM cell information, using either template matching or triangulation [Laasonen et al., 2004].

Indoor several methods are available such as badge location, infrared location and location by triangulating (Figure 3.1) with either 802.11 (WiFi) or Bluetooth access points [Chen and Kotz, 2000]. Choosing a specific method generally depends on the granularity required and whether the positioning needs to be performed inside, outside or both. Combining methods could improve the accuracy of the positioning, such as using GPS outdoors and WiFi triangulation indoors.

Another concern when choosing a method is privacy, however this concern is not considered in this study, as the purpose of the RECON framework is to gather usage, context and attitude information.

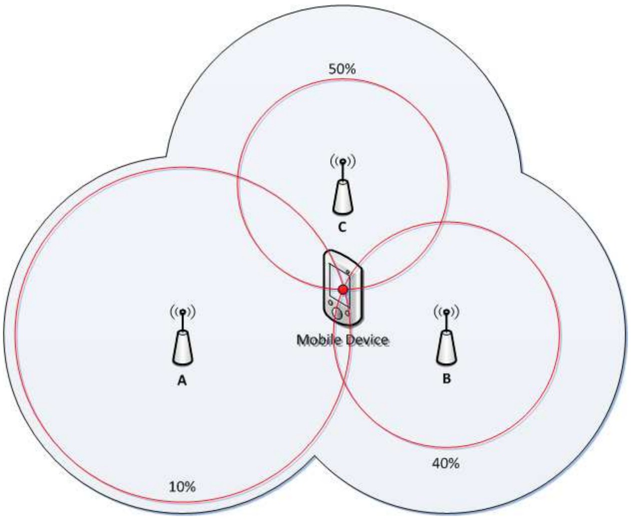


FIGURE 3.1: Triangulation based on three network access points. Based on the signal-strength and the known location of the three access points, the device can calculate a location.

Time

As most people follow a regular schedule most weekdays and a slightly altered one in the weekends, time can tell about a test subjects context. Time information can be several things other than time of day and date, such as season and time zone. Time can be obtained from the built-in clock on the mobile device.

Nearby objects or persons

This can tell about location or social context. One example could be that the test subjects are in close vicinity to their boss as well as the meeting room printer, a reasonable assumption could be that they are in a meeting. Detection of nearby people and objects could be performed with Bluetooth, by searching for other nearby Bluetooth devices. This method has a weakness though, as Bluetooth is notorious for its power consumption[Linsky, 1995], some people leave their Bluetooth off to conserve battery life. Another more complex method is using the microphone and/or camera on the mobile device to reason about nearby people or objects.

Network Quality of Service

Network Quality of Service can affect how a test subject is using an application using the network. Quality of service include network bandwidth, delay, jitter and errors. For example low bandwidth could cause a person to close his or her media player while streaming a video, because of the sudden drop in quality or disconnection. The quality of service can usually be monitored via the mobile device's API.

Computing

This area covers parameters such as load on the central processing unit (CPU), available memory and running applications on the mobile device. This is very similar to Network Quality of Service, as it affect the use of an application, however this affects all applications, network enabled or not. If an application makes heavy use of the CPU, and other running applications are doing so as well, this could explain a high busy-time for the application, which ultimately could mean that the test subject get tired of waiting and ends the application.

Environment

This is the area surrounding a test subject. Information about the environment such as sounds, light level, temperature, humidity, air pressure and such could allow an application to adapt accordingly or in an evaluation could explain an issue. For example, measuring the sound level could explain why the test subject was distraction.

Social

Social contexts such as current activity and motives of a test subject can be hard to capture [Chen and Kotz, 2000]. One possible method is to access the subject's calendar to figure out what the test subject is supposed to do at a certain time and what the his or her plans or motives are. In [Eagle and Pentland, 2006] a framework for sensing complex social systems was developed, using ContextPhone [Raento et al., 2005] framework and user modelling methods.

External resources

A weather webservice could allow an application to query the current temperature and weather given a specific location or a traffic webservice could show if a user might be caught in a trafficjam. To get information like this, the webservice needs to be available as well as an Internet connection and a robust location system.

Several other parameters could be captured such as vibration, tilt, radiation and so forth, and these parameters can in some cases be combined to form new context information.

During a field evaluation all these different parameters can either affect the user's directly or through the evaluated application, which is the reason why it is so important to also gather context information in an evaluation.

3.2 Summary

In this chapter context was defined and context awareness in general was described. Afterwards a list of seven common types of context information were described in detail. For each of the types, a method for capturing the information was described and related to how they could be useful in determining a user's general context. The user's context can tell why a certain usability issue occurs, such as stalling in an application because they are distracted.

CHAPTER 4

USABILITY EVALUATION

The purpose of this chapter is to explain why design and evaluation of applications with regards to the user is important and how User Centered Design (UCD) can be applied to achieve this. The chapter will also explain important terms associated with usability evaluation and what remote and automated evaluation means.

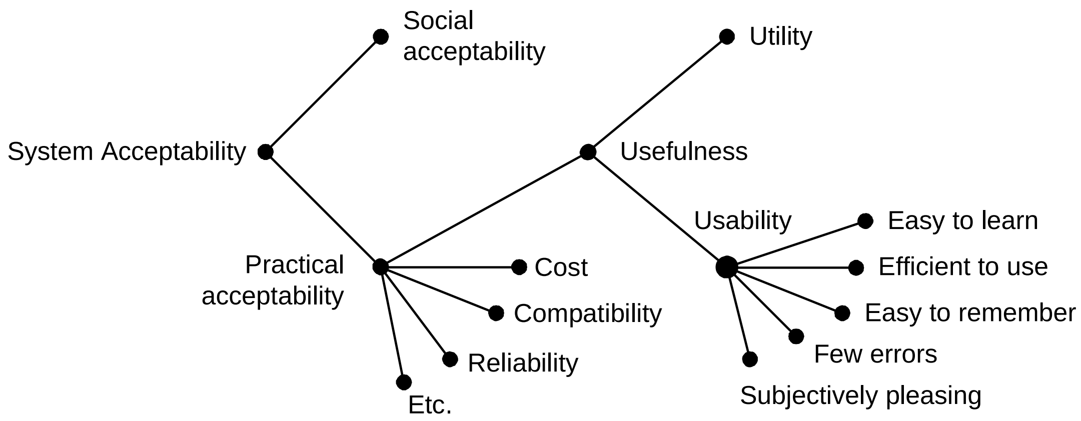


FIGURE 4.1: System acceptability according to Jakob Nielsen (Recreated from [Nielsen, 1993])

Whether or not a system will be accepted by its end users is determined by many factors. On Figure 4.1 the overall acceptability of a system is divided into social acceptability and practical acceptability. Social acceptability refers to ethics, for example one operating system might not be socially accepted by fanatics of another operating system, even though it might have a higher practical acceptability. Practical acceptability is a combination of acceptability categories such as cost, compatibility, reliability and usefulness. According to [Nielsen, 1993], usefulness is the issue of whether the system can be used to achieve some desired goal. Usefulness can be further divided into utility and usability. Utility defines whether the system is capable of doing what is needed, whereas usability defines how well users can use that functionality [Nielsen, 1993].

Usability is per definition a combination of different factors. [Nielsen, 1993] and [Rubin, 1994] disagree in naming these, however seem to agree on what is important, which is shown in Table 4.1.

Learnability & memorability	Indicate how easy a system is to learn and remember.
Effectiveness, efficiency & errors	Indicate how efficient the system is, meaning that it allows the user to have a high level of productivity and a low error rate at the expenditure of minimum resources.
Attitude & satisfaction	Indicate the user's opinion and feelings about the system.

TABLE 4.1: Combined usability factors from [Nielsen, 1993] and [Rubin, 1994].

All these factors apply to all aspects of a system with which a user might interact [Nielsen, 1993]. One way of achieving high system usefulness is to focus on the user throughout the development process, which is what the UCD approach proposes.

4.1 User centered design

The philosophy of UCD is, as the name indicates, to place the user in the center of the design process, which means designing everything around and for the user, instead of making the user fit the design. The design process entails that a products goals, objectives, context, and environment as well as all task-related aspects, are derived from the users viewpoint.

There are three principles in UCD, which are:

- An early focus on users and tasks.
- Empirical measurement of product usage.
- Iterative design whereby a product is design, modified, and tested repeatedly.

UCD is not usability evaluation, however usability evaluation is a method for achieving a good UCD [Rubin, 1994]. Most often usability evaluation gathers information about the users use of the product, which will be referred to as use patterns. Use patterns can be defined as a sequence of events that show what parts of the software the user are using, how they use it and when they use it. Additionally changes over time in these three areas can provide further information, such as learnability and memorability. The events that can be captured differ alot in abstraction level, as can be seen on Figure 4.2. Each of the events on the figure a plotted according to their expected duration. Events such as mouse clicks are categorized as high frequency events, as their duration is short and therefore have a tendency to happen at a high frequency, whereas events such as projects have a long duration and therefore happen at a low frequency. An example of an abstraction of HCI events, could be a sequence of mouse clicks and UI events that define the task of printing. A project could then be comprised of several tasks, such as the printing task and so forth. This kind of abstraction is also part of the EDEM framework, which was described in Section 2.1.

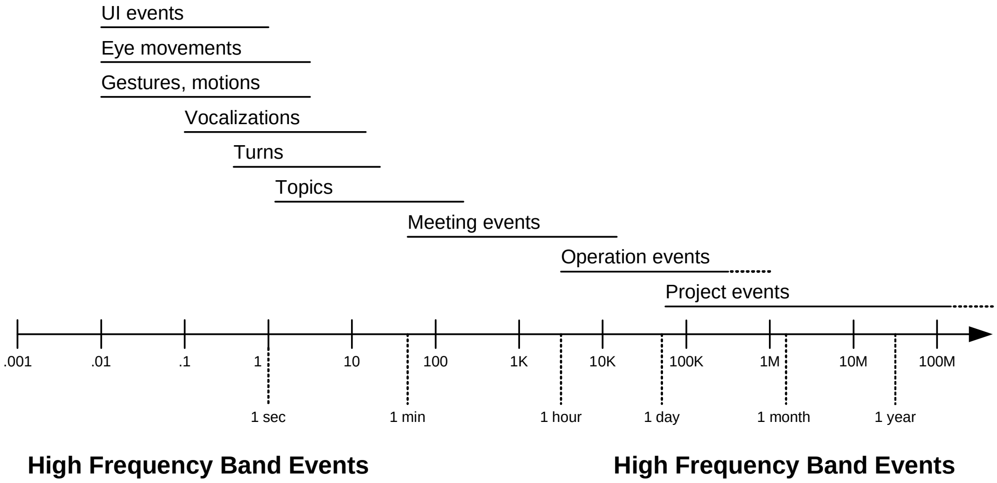


FIGURE 4.2: A spectrum of HCI events. Adapted from [Sanderson and Fisher, 1994].

According to [Gorlenko and Merrick, 2003], the methods and techniques of UCD will need to undergo certain changes to be an effective and efficient design method for mobile computing. One of the challenges mentioned is that in order to understand how a user interacts with mobile applications, it is not only required to examine use patterns but also the users context.

For this reason it can be troublesome to choose a specific technique or method. In an attempt to shed light on some of the different methods available, [Kjeldskov et al., 2005] evaluates a mobile application using four different approaches - Field evaluation, Laboratory evaluation, Heuristic walkthrough and Rapid reflection - and compare the results. 22 distinct usability issues were detected and categorized according to Molich's taxonomy [Molich, 2007], from cosmetic problems, serious problem to critical problems.

	Field evaluation	Lab evaluation	Heuristic walkthrough	Rapid reflection	Total
Critical	4	4	4	4	5
Serious	7	6	6	5	11
Cosmetic	2	3	3	4	6
Total	13	13	13	13	22

TABLE 4.2: Summary of results from [Kjeldskov et al., 2005].

The results of [Kjeldskov et al., 2005] is summarized in Table 4.2. Arguably the amount of issues found is relatively low, and the results are only based on one specific application. However the results seem to indicate that field evaluation is a valid alternative to a laboratory evaluation based on the assumption that problems categorized as serious are more important than issues categorized as cosmetic. If the time spent is taken into account the conclusion can be slightly different, as field evaluation is notorious for being very time consuming compared to laboratory evaluation. In [Kjeldskov et al., 2005], the field evaluation took 64% longer then the laboratory evaluation (82 versus 50 hours).

Since the time spent on the field evaluation (compared to the laboratory evaluation) is not negligible a more efficient way of performing the evaluation must be found, before it is a real alternative. One way to achieve this is to use remote evaluation, to automate it or do both. In Section 4.2 remote evaluation will be further described and automation of the evaluation will be described in Section 4.3.

4.2 Remote evaluation

The concept of remote evaluation is to separate the test subject and the HCI research team in space and/or time. Separation in space means that the test monitor can stay at one location and the test subjects can be geographically independent of the test monitor. Separation in time means that the test monitor does not have to be working during the actual evaluation process, but can analyze the usability data gathered whenever he or she wishes to [Thompson et al., 2004][Hartson et al., 1996].

When performing usability evaluation it can be difficult and expensive to find representative test subjects nearby. To overcome those problems remote evaluation can be a possibility since the HCI research team and the test subject can be separated in both space and time.

In [Hartson et al., 1996] nine types of remote evaluation are described and two case studies performed. One of the methods described is Semi-Instrumented Remote Evaluation.

In the semi-instrumented evaluation the test subjects was instructed how to spot usability problems to be capable of reporting usability problems only. The advantages of this approach is that many test subjects can be instructed at the same time and it is possible to get detailed description of attitude and opinion towards the software being evaluated. This method will result in differences in the data being reported meaning it will be difficult to compare results among different test subjects. Another problem is that instruction of test subjects is relatively time consuming which must be dealt with.

Instead of instructing the test subject in spotting usability problems finding them automatically could make the data collection more uniform and more qualified for research.

4.3 Automation

Automation means removing the cameraman and test monitor that is usually present in a field evaluation, as well as the manual processing of the data captured. Instead relevant usability data is captured by the use of software and hardware.

In Chapter 2, EDEM, WebQuilt and ContextPhone were presented as automatic evaluation frameworks. An automatic evaluation framework captures system and UI events, but can also capture more indirect information such as context and attitude.

By the use of extensive data capturing, a history can be built and by visualizing the data captured, the analysis is made more straightforward.

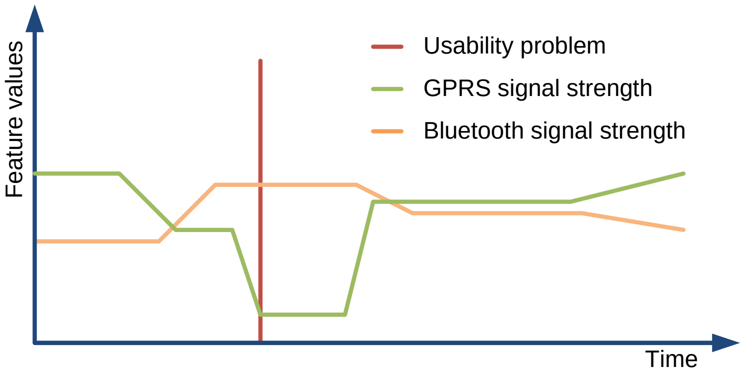


FIGURE 4.3: Conceptual view of some desired features over time.

For example on Figure 4.3 some fictive data is plotted. The idea is that by plotting context information over time the reasoning for some usability issues could be identified. In this example the application could be dependant on a network connection, which for some reason suddenly disconnects (the situation outlined at Figure 4.3).

One of the benefits of automatic evaluation is that the test monitor recording the evaluation with a camera is obsolete, the HCI evaluation team is only required to setup the framework and instruct the test subjects. After that, the system gathers usability data, and stores it for later automatic or manual processing by either a piece of software or a HCI researcher.

Automated evaluation will be further explained in Chapter 5.

4.4 Summary

In this chapter system acceptability was described as being a collection of many parameters such as utility and usability. In order to achieve a high system acceptability, it is important to have high utility and usability. In this study the focus is on usability, and to achieve high usability the method UCD was presented. This method makes the user the focus in all phases of a products development life cycle.

A comparison of four of the methods in UCD showed that field evaluation is good way of identifying critical issues, however it is also time consuming. One way of conducting more time-efficient field evaluations is to do remote evaluation, letting the HCI researchers analyze the results of the evaluation at another time and place. A further improvement to remote evaluation is to automate the process of gathering usability data and in some cases analyze it, allowing the HCI researcher to focus on the critique of the application.

Automation is however quite complicated, and the following chapter will explain the concept of automated evaluation more thoroughly.

CHAPTER 5

AUTOMATED EVALUATION

The purpose of this chapter is to explain the possible benefits of an automating usability evaluation and explore some of the methods that can be used to achieve this. Of the methods described, one or more will be chosen for implementation in the RECON framework.

Automation can be done in any of the three phases - capture, analysis and critique - as was mentioned in the Introduction, however frameworks for capturing usability data is more common than the latter two.

Collecting usability data from several mobile devices and storing them at a central location is not trivial. Choices such as where to store the data and at what abstraction level is affected by factors such as storage capacity, storage reliability, network bandwidth and so forth.

On mobile devices, which is the focus of this study, a limited amount of resources usually leads to an architecture where the data is gathered on the device, temporally stored and later reported to a server for storage and further analysis.

When capturing usability data in a system, the architecture of the system determines which data capture methods are viable. Some systems, such as an instant messenger application, needs to communicate with other systems through i.e. the Internet. Other systems work by themselves, such as a calculator application.

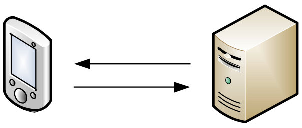


FIGURE 5.1: An example of a software system using a client-server architecture.

On Figure 5.1 a client-server based system is shown. In this case two primary methods can be chosen - one functioning as a proxy and the other as a stand-alone system. The two different types of capture techniques are shown on Figure 5.2 and 5.3. Other methods exist (e.g. capturing data to a memory card in the mobile device and manually collecting the storage cards) but they are not considered feasible alternatives, since they can be time consuming or expensive to use (because the HCI research team will have to manually retrieve the memory cards). Another drawback is that it does not support real time data access. When a system does not have a client-server relationship, the proxy method is obviously not usable, so in this case only the stand-alone approach is viable.

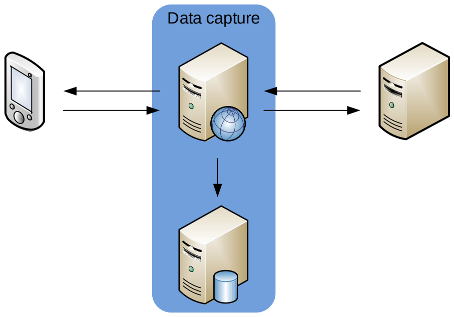


FIGURE 5.2: Deployment of a proxy based logging system in a client-server based system (Like WebQuilt).

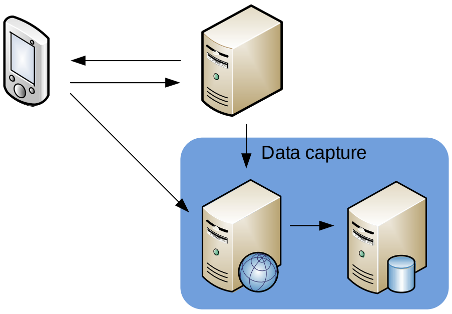


FIGURE 5.3: An example of a stand-alone logging system (Like EDEM).

Which capture method to choose depends on the level of abstraction and type of information desired e.g. if the communication between the client and server is the interesting part, the proxy based system would be desirable. In this study a system like the one shown on Figure 5.3 is selected because the data capture is desired to function as a standalone system where information from various sources is combined and stored together. The information to combine could be information from the mobile device about radio signal strength and UI events from the application. Finally the framework should not be limited to evaluation of client-server architectures, so the proxy method is not applicable.

As mentioned earlier, usability evaluation is often divided into three parts; capture, analysis and critique, but in practice capture and analysis often melt together or overlap e.g. like in EDEM (described in Section 2.1). In EDEM the two parts are separated but there is still some overlap since the capturing mechanism is designed and implemented in relation to the information necessary in the analysis. Furthermore an abstraction is made in the capturing process to form high level events where the abstraction can be categorized as a pre-analysis. In Section 5.1 and 5.2 some of the practical aspects of the data capture and analysis will be described.

5.1 Data capture

Depending on the hardware and software platform of the mobile device used in the evaluation, different types of information and methods for capturing this information is available. Some information is more important than other depending on the goal of the evaluation. The goals of the RECON framework was to capture the following:

Usage information tells how the test subject is using the application i.e. how he or she navigates the UI and how long it takes.

Context information tells about the current context of the test subject, such as location, time, plans, nearby objects and so forth.

Attitude information tells about the test subjects state of mind or his or her opinion towards the application.

In the following sections different methods for extracting information will be presented.

5.1.1 Usage information

The purpose of the following section is to explore different possibilities for extracting information about the use of a specific application such as use patterns. This is done to uncover possible usability issues in an application. Several different techniques are available, and these will shortly be examined.

Hooking

Hooking is a programming technique used to perform a chain of methods in an application. An example could be when a user clicks a button; The button will probably have some method associated that is executed when the button is pressed i.e. printing a page, which will be referred to as the original method. When a hook is inserted the button will still execute the original method, but will also execute whatever method the hook points to. The method that the hook points too can be either performed before, after or during the execution of the original method depending on where the hook is inserted.

To be able to insert hooks the source code is required as well as somewhere to put the method that the hook points to. This can either be incorporated as additional source code in the project or as a dynamic linked library.

One of the benefits of the hooking technique is that it allows the software developer to specify when the hooked method is executed, and what data is available to the hooked method. The drawbacks are that the source code is required and additional code or libraries needs to be added to an existing system. HCI researchers who uses the framework are also required to have some programming insight, which might not be the case.

Aspect Oriented Programming

A technique that is very similar to hooking is Aspect Oriented Programming (AOP). AOP is a programming paradigm primarily concerned with separation concerns in programming. Separation concerns entails splitting a system up in modules with as little overlap in functionality as possible to avoid what is known as *cross-cutting* [Tart and Moldovan, 2006]. Examples of modules with these *cross-cutting* concerns, could be security modules or database module because these modules often contain code that needs to be implemented in or accessed by many different modules. AOP is very similar to hooking in that it intercepts certain events and the source code needs to be available, however AOP only modifies the source code at compile time, so the AOP parts can be easily left out i.e. on release of a product.

Windows Messages

Windows-based applications are event-driven. Unlike MS-DOS-based applications that make explicit function calls to obtain input, a windows-based application wait for the system to pass input to them. The information is passed to an application in the form of a windows message. Therefore all windows must have a procedure that can process these messages known as a Window Procedure.

Windows Messages originate from both the Microsoft Windows operating system as well as other applications. For example a message is generated when the user types, moves the mouse, or clicks a control in an application.

All Windows systems maintains a global system message queue and a thread-specific message queue for all GUI-threads. The Windows operating system will automatically sort the messages in the thread-specific queue, so that only windows messages intended for the specific thread will be received. However, it is possible to listen for messages in the global message queue.

By listening to windows messages for an applications, it is possible to know when a user clicks a specific button, opens a certain window and so forth. The benefit of this approach is that it makes it possible to record usage information of an application in any other process without having access to the source code. Handling the windows messages and relating them to the controls in the application can be difficult though, especially if no source code is available, as names of controls not necessarily dictate their position or action.

5.1.2 Context information

With the release of the Microsoft .NET Compact Framework 2.0, one of the new additions to the framework was a so-called State & Notification Broker. This software module provides access to system states and properties that previously required invocation of methods in native API's.

The State & Notification Broker allows event handlers to register for changes in system properties such as battery level, signal strength, unread messages and so forth. Many of the available properties can indicate current context.

Additionally to identify nearby object and to determine a location (See Section 3 for more information), access to wireless networks on the mobile device is a possible solution. Most mobile devices are also Bluetooth-enabled and more rarely capable of connecting to WiFi networks. How these technologies work and why they can be useful in this project is described and considered in the following subsections.

GSM

In Denmark GSM covers most of the country (See the GSM coverage map on Figure 5.4), and in most cases an area is covered by more than one cell tower. Using Cell ID's and signal strength it is possible to calculate a location provided that the locations of the cell towers are known.

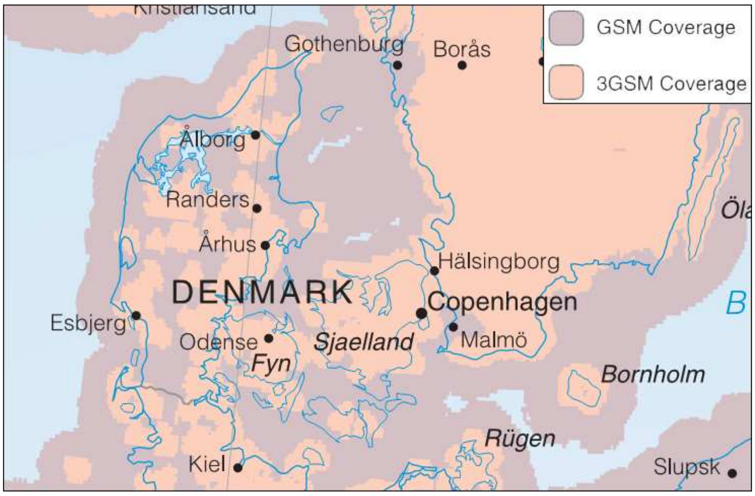


FIGURE 5.4: GSM Coverage map [Association, 2007]

WiFi

WiFi networks are becoming increasingly popular, and as can be seen on `http://www.openwifi.dk`, most major cities in Denmark are covered with access points. The range of WiFi varies, but knowing nearby access points and cross-referencing that information with information from a website like `http://www.openwifi.dk` would give a rough estimation of a location.

No managed API for WiFi networks exist in the Microsoft .NET Compact Framework 2.0, however it is possible to access a native API that has this functionality. By using the OpenNETCF Smart Device Framework 2.0 (SDF) [ope, 2007], WiFi methods are accessible from managed code without having to deal with a native API. The SDF exposes a method that returns all nearby access points. Each access point contains information such as SSID, MAC-address and Signal to Noise Ratio (SNR).

Bluetooth

Bluetooth is a short range radio network technology created for personal area networks. Most mobile devices today are Bluetooth-enabled, meaning that they can connect to and communicate with other Bluetooth devices such as handsfree headsets. Bluetooth is not actively searching for other devices, but needs to perform an inquiry to detect them. If any devices are detected it is possible to query them about what device they are, their MAC-address and a name.

Performing inquiry on a regular interval will tell an application about nearby Bluetooth devices. Nearby devices can provide information about context, such as whether or not a user is alone or with his or her friends, if the user is in a meeting or on the bus. Some Bluetooth devices such as printers or scanners are usually stationary and can therefor be used for location like WiFi access points.

Like GSM and WiFi, Bluetooth cannot be accessed via managed code in the Microsoft .NET Compact Framework 2.0 API's, however by using 32feet.NET [Hand, 2007] it is possible to access the Bluetooth interface on the mobile devices.

5.1.3 Attitude information

To collect data about the test subjects opinion and attitude a smart prompt based on a digital survey can be used, as was done in the EDEM framework [Hilbert and Redmiles, 1998a]. To decide when to prompt the test subject some kind of identification of possible usability issues is necessary which in fact is part of the data analysis but still has a strong relation to the data collection. The detection of possible usability issues can be done more or less intelligent but based on the assumption that the RECON framework will be used to analyze small parts (of an application) at a time methods based on expert knowledge is preferable.

An expert system to detect possible usability issues could be based on finite state machines (FSM). The HCI researcher or software developer can define a FSM for each activity to evaluate and define which states the test subject can go through to receive a prompt. An example of a FSM for a print activity is shown a Figure 5.5. At the figure the test subject should enter through one of the green states and leave through the blue state. In case the user leaves through the red states a questionnaire will appear and the user's opinion collected.

Comparing the result which can be obtained by the use of a survey (as described above) e.g. with results

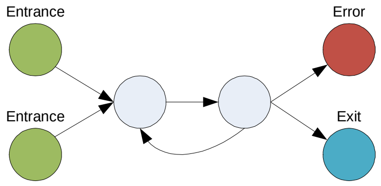


FIGURE 5.5: An example of a FSM for detecting unexpected behavior.

from a ”thinking out loud” evaluation would definitely be a disadvantage for the survey. It is expected that it is possible to obtain a better a result by the use of a ”think out loud” evaluation since this method can give a continuous description of the test subjects opinion and attitude. The survey is only capable of sampling the user’s opinion and attitude by ”asking” predefined questions or showing a text box in which the user has the opportunity of write whatever he or she would like. On the other hand the data collected by the survey would be easier to compare between different test subjects and also perform statistics on.

5.2 Data analysis

To perform analysis on the data collected during an automated usability evaluation several different techniques are available. In [Ivory and Hearst, 2001] the following four methods are reviewed:

- Metric-Based analysis of data.
- Pattern-Matching analysis of data.
- Task-Based analysis of data.
- Inferential analysis of data.

User interfaces on mobile devices mostly are very limited in the number of different screens and UI events. Because of the limited number of screens and events a Task-Based analysis of data is considered as being appropriate. A Task-Based analysis technique is earlier used in automated analysis such as QUIP [Helfrich and Landay, 1999] where a graph, representing the users path through the application, is generated based on data captured (see Figure 5.6), in this study however the analysis will be performed manually.

Each circle on Figure 5.6 represents an application state and the arrows width the number of transactions. The last parameter on Figure 5.6 is the time spent on switching state (visualized by color coding).

To use the Task-Based analysis method, the data must be well formatted, describing exactly what happened at a given time e.g. a data entry for an application could look something like the one shown in Table 5.1.

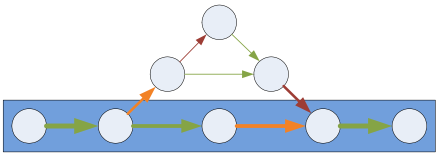


FIGURE 5.6: A sketch of a graph representing user navigation in a fictive application (Modified from [Helfrich and Landay, 1999])

Date	Time	Event ID	Extra information
2007-06-06	20:00:00	1	PRINTING:report.pdf

TABLE 5.1: An example of a well formatted data entry.

An expert path (how an expert would do) is defined through the different states of the application, according to the designers usage expectations. By comparing the expert path with the different paths through the generated graph it is possible to draw a wide range of different conclusions about the application, in the following two of the primary are described.

Intuitiveness and understandability If the generated graph looks something like the one shown on Figure 5.7 a usability issues is located. The graph indicates that the users do not understand the application or that it is not intuitive to use the application. This conclusion is based on the fact that eight states outside the expert path was visited by several users.

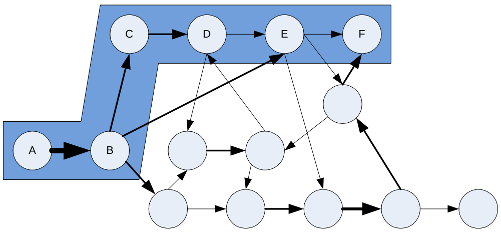


FIGURE 5.7: A sketch of showing an application defect where test subjects does not understand the application structure (Modified from [Helfrich and Landay, 1999]).

Omission of states is possible to detect by comparing the graph with the expert path. An example of how an omission would look like is shown on Figure 5.8 where the correct path through would be A-B-C but some of the users managed to omit state B which e.g. could be a confirmation dialog.

Drawing graphs as the one shown on Figure 5.7 and 5.8 all the paths, for all the test subjects, through the tested application must be analyzed and two identically paths collapsed e.g. if the sequence A-B is part of three test subjects path they must be joined together. To determine equal paths the Longest Common

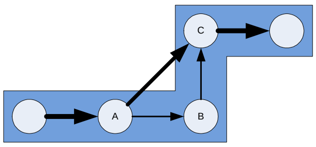


FIGURE 5.8: A sketch representing an application defect where some of the test subjects omit an important action (Modified from [Helfrich and Landay, 1999]).

Substring (LCS) algorithm [Cormen et al., 2001] can be used (as done in [Helfrich and Landay, 1999]). Given two or more strings the LCS will find the longest string which is a substring of the given strings e.g the longest substring of the two strings ”ABAB” and ”AABA” would be ”ABA”.

5.3 Summary

In this chapter possible methods and technologies used in an automated evaluation were described. Afterwards possible ways of capturing usage, context and atittude information were described. A standalone based system was chosen, where data from numerous sources could be stored together and afterwards combined.

The data capture was further divided in three categories. Possible ways of obtaining date regarding each category was afterwards described. The first category was usage information which describe possible methods of obtaining information about a user’s interaction with an application. The second category was context information which covers information regarding the context a potential user would be in. The last category was attitude information which describes how to capture a test subjects attitude.

The data analysis describes one method for analyzing data captured by an automated evaluation framework. The described method is a task-based analysis where the user’s navigation is represented as a state graph, which will be used when analysing the results obtained from the RECON framework.

Part II

Design and Implementation

In this part the development of the RECON framework is described, beginning with an overview of the architecture. In the following sections a more detailed description of the individual parts of the framework is done.

CHAPTER 6

RECON FRAMEWORK

This chapter will present the development process of the RECON framework. First an overview of the system will be introduced. Afterward more detailed descriptions of the individual parts of the framework will be described, and any considerations elaborated.

6.1 Overview

The main objective of the RECON framework is to capture information about usage, context and attitude described as described in the Introduction. In this section some of the primary and general design concerns regarding an information capturing framework will be described, after which more specific requirements will be elaborated.

It is expected that a large amount of data will be gathered, so one essential decision to make, is where to store the collected data. The two primary possibilities are centralized or decentralized e.g. on the device or at a central server. The centralized date storage approach allows the research team to have data access from day one. This can be an advantage, because test problems can be identified early and corrected, avoiding useless data. However in order to be able to transmit data to a server a stable network connection is required in order not to loose any data. The decentralized does not suffer from this limitation, but does not have the benefits either. In order to gain all benefits, a combination of both is chosen. Data gathered will be temporarily stored on the mobile device, and reported to the server when a network connection is available.

This choice means that the RECON framework will be comprised of both a mobile part and a server part. The mobile part will do the data collection, temporary storage and report to the server which will handle final storage and allows the HCI researcher to extract the information.

On Figure 6.1 the system concept is shown. As can be seen, the captured data is being transferred from the mobile devices to the server. After this the HCI researcher can access the data in the database from his own computer via database administration software such as PHPMyAdmin. Configurations can be generated by the HCI researcher and sent to the mobile devices via the server. The configurations tells

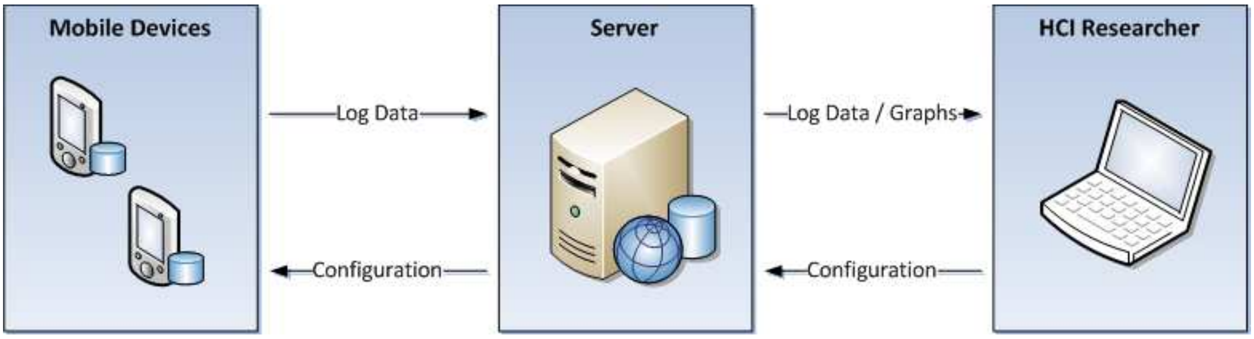


FIGURE 6.1: System concept

the mobile devices what to capture and how often to do it.

Based on the goals of the RECON framework defined in the Introduction, the requirements to the are, that it is capable of extracting the information described in Table 6.1.

Usage	describes how the test subject is using the application being tested. Usge and use patterns was described in Chapter 4.
Context	describes location, nearby objects and persons, environment and more. Context was described in Chapter 3.
Attitude	describes the users feelings towards the application. Attitude was described in Chapter 5.

TABLE 6.1: Framework requirements.

Use patterns and attitude information are related to the application being tested, however context information is more general. Context could also be interesting to capture even when the user is not currently using the application, whereas the other types of information are less interesting.

In order to be able to capture context information when the tested application is not running, it has been decided to split the mobile part of the RECON framework up into two parts. One part that monitors the application and one part that monitors the context information. That way the part that monitors the application only needs to be active while the application is running. This also means that an application, named RECON log, needs to be running on the device at all times during an evaluation. As the mobile device could run out of power, or be turned off by the test subject, it is important, that the RECON log starts itself, as soon as the device is powered on again, which is achieved by placing a shortcut in the startup folder, present in a standard Windows Mobile installation, on the mobile device.

On Figure 6.2 the modularization of the RECON framework can be seen.

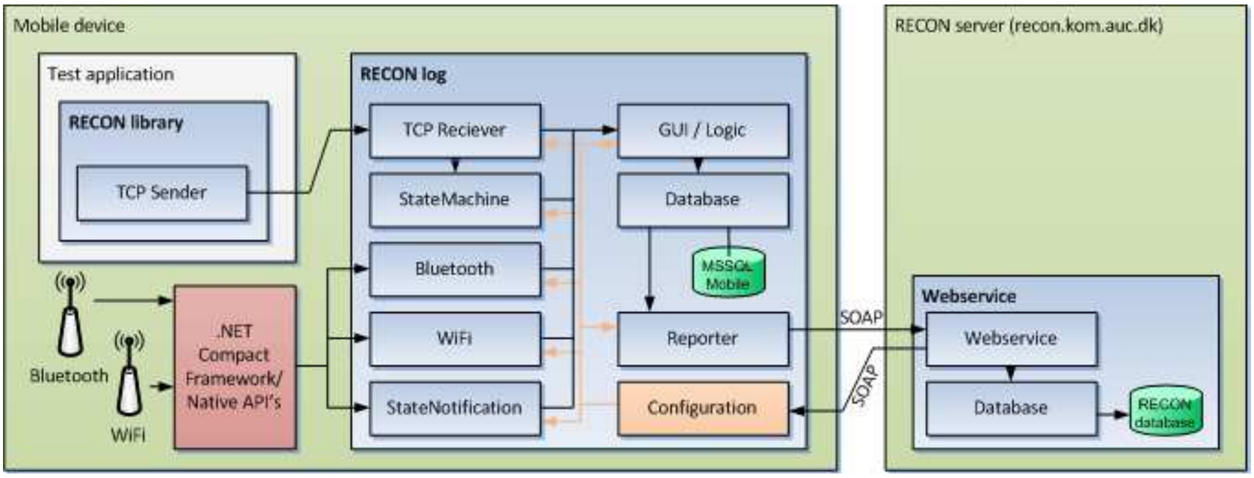


FIGURE 6.2: System overview

As can be seen on the figure, part of the RECON framework is located in the application being tested, and part of it in its own dedicated application as mentioned above. The reason for this software split up, is because of the technique used for capturing events, which will be explained later in the section Usage information. If another method such as Windows Messages has been used for achieving usage information the software split up has not been necessary.

Hardware and software platform

In this study the hardware platform is a Qtek 8300 and the software platform is Microsoft Windows Mobile 5.0. This platform was chosen because it allows for easy development and deployment because of the Microsoft .NET Compact Framework 2.0 and also makes some of the data capture trivial by using the Microsoft Windows State and Notification Broker, which provides access to a wide range of context information. In order to temporarily store captured data on the mobile device the software platform also include Microsoft SQL Mobile server, which was chosen as it integrates easily in Microsoft .NET applications for the Microsoft Windows Mobile platform. In addition a range of third party API's are used in order to access information from other sources, which will be further described in their respective sections in the following.

In the following section the choice of communication method is made, based on the hardware and software used in the study. In the sections following this, the mobile part of the RECON framework will be explained and afterward the server part described. In these sections each specific module of the RECON framework will be described in detail and further design considerations documented. All modules are continually tested during the development process and a final pilot test is done in order to verify that all modules are working correctly and that the RECON framework is capable of capturing data as specified in this chapter.

6.2 Communication

The purpose of the following section is to describe the interface between the client and the server including the choice of communication method. The Microsoft .NET Compact Framework 2.0, which is used by the RECON framework on the mobile device, supports fewer methods than the Microsoft .NET Framework used on the server-side of the RECON framework, however several different ways of communicating are still available:

- Socket
- HTTP POST/GET
- Webservice

Sockets allows for a very customized communication, as the communication protocol needs to be defined by the developer. This can, depending on the developer, lead to a very efficient exchange of data. However sockets requires some effort to implement, as communication with sockets is very low-level and issues with connection and error handling needs to be addressed by the developer in his own code.

HTTP POST/GET is what is used on most websites today to exchange information between a client and a server via a web browser. Using POST/GET does not require as much effort to use as sockets, but as results are returned in one stream, some post-processing needs to be performed in order to extract information from the stream.

Webservice is very popular nowadays, since it requires little effort to implement and manages the post-processing mentioned under HTTP POST/GET by itself. Webservices use a protocol Simple Object Access Protocol (SOAP) which is standardized like HTTP, meaning that it is not limited to one platform e.g. Microsoft .NET or Sun Java. Since SOAP is based on XML, one of the disadvantages is the amount of meta-data in a message, which results in a significant overhead with small messages.

As this is a proof-of-concept implementation, the efficiency is not prioritized and as the development and evaluation of the RECON framework will be performed in an environment with good WiFi coverage, the overhead of using a method like webservice is insignificant. Therefore the server of the RECON framework will be implemented as a webservice, that the client can communicate with.

As mentioned, small messages sent via SOAP have a large overhead. One way of lowering the general overhead of the RECON framework, is to either gather a set amount of data or transmitting it at set intervals.

6.3 Mobile device

As mentioned earlier, the mobile part of the RECON framework consist of a library and a standalone application. The RECON library is trivial, as it all it does is connect to the RECON log, and send a string whenever a hook is triggered. The RECON log is more complex, and the remainder of this section will describe how this application is designed.

The main module of the RECON log is responsible for configuration and creation of all other modules. The module will capture all events received from the modules and store them in a dedicated database, but will also show them in a graphical user interface (GUI) for debugging purposes.

Whenever the RECON log is started on the mobile device, the following steps are performed by the main module:

- A connection to the database is established and a ”system start” indication inserted.
- A configuration is loaded.
- The Usage Information module is started (TCP Receiver).
- The State & Notification Broker module is started.
- The Bluetooth module is started.
- The WiFi module is started.
- The Reporter module is started.

For the modules that needs activation at certain intervals, dedicated timers are created after the module has be started.

In the following sections, the most interesting of the modules in the application are more thoroughly explained and design considerations described.

6.3.1 Usage information

All the methods for capturing usage information, described in the previous chapter, would work in the RECON framework that is to be developed in this project, however the methods vary in the amount of effort required to implement as well as use them in an evaluation.

Window Messages allows monitoring without tampering with the evaluated application, however they are difficulties such as finding out what windows messages to process.

The hooking and AOP approaches require access to the source code of the application, with hooking requiring manual addition to the source code, whereas AOP automatically adds to the source code at compile time. AOP is a new programming paradigm, and using it would require the developer to know about the technique.

As the source code of the application used for testing in this project is available, hooking and AOP are valid choices, however if the goal of the RECON framework would be source code independence the windows message approach would be the way to go. Also as the mobile part of the RECON framework is split up in two parts, one that is only active while the application is running (the RECON library), and one that is always active (the RECON log), an obvious choice would be to implement the application-active part as a dynamic linked library in the application.

With these pros and cons in mind, the choice falls on the hooking approach, as it allows easy use of the RECON framework without requiring the developer to know about AOP. The hooking approach will be implemented as a dynamic linked library that can be included in the application being tested to insert the hooks. This of course entails communicating with the always active part of the RECON framework, which will be done with sockets as no alternative is available in the Microsoft .NET Compact Framework 2.0 that achieves asynchronous communication. This means that part of the module will exist in the tested application as a linked library and part of it will be in the RECON log.

6.3.2 Attitude information

As described in earlier chapters it is necessary to collect the test subjects opinion and attitude. It is chosen to collect the user opinion and attitude by the use of a smart prompt based on surveys and FSM's. The choice to use FSM compared to other methods, such as grammar used in [Hilbert and Redmiles, 1998a], is based on the idea that a FSM relates well to the navigation in most applications and would be easy to comprehend for HCI researchers, who does not necessarily have a background in software development. For example, on Figure 6.3, a small application with different screens can be seen. A straight forward way to represent the navigation in this application, is to consider each screen as a state.

The prompting system is task based meaning that each task, a test subject is supposed to perform, is

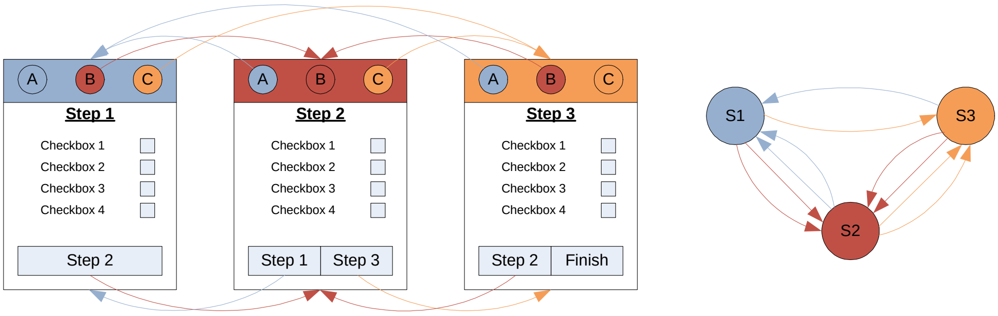
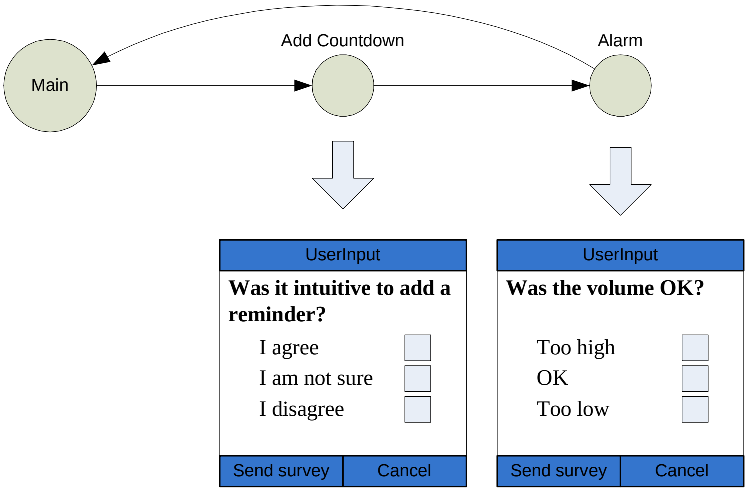


FIGURE 6.3: An example of how transitions between three simple GUI screens can be expressed as a simple FSM

represented by a FSM. One FSM can contain numerous states in which it is possible to show a survey. The surveys is configured and deployed at the mobile devices through the RECON server database.

On Figure 6.4 is an example of a simple state machine, with two surveys, shown.



Was it intuitive to add a reminder?

I agree □
I am not sure □
I disagree □

Was the volume OK?

Too high □
OK □
Too low □

FIGURE 6.4: FSM with two surveys. A FSM can have one survey for each state and a survey can have unlimited number of questions each with unlimited number of options.

In the same way as the configuration, the state machines are downloaded from the server when the mobile device is booted, on the condition that the state machines on the server are newer than the ones already present on the device.

6.3.3 Context information

As mentioned in the previous chapter, the State & Notification Broker provides access to a wide range of properties, however only a select range of these are of interest in this project. The properties that have been chosen are mainly related to Network and Power information and Social Context as explained in Section 3.

Network and power information

In Table 6.2 the chosen properties related to Network and Power information are shown.

Using these properties will not indicate specific network problems, such as latency, jitter or delay, nevertheless poor connection and disconnects can be explained.

Connections	Four different network technologies are monitored: Cellular, Modem, Bluetooth and WiFi. The State & Notification Broker signals whenever a connection is made and returns the amount of established connections.
GPRS coverage	This indicates whether the mobile device is able to connect to the Internet via GPRS or not. In combination with the amount of Bluetooth or Network connections, this can tell whether or not the mobile device is able to connect to the Internet.
No phone service	This indicates if the mobile device can receive phone calls or messages. When there is no service available, the user will not be distracted by incoming phone calls or messages.
Signal strength	Indicates the GSM signal strength to the operator. A low signal strength might explain why a phone call is abruptly ended or why an application streaming via GPRS is stopped.
Battery level	Denotes remaining battery power, expressed as a percentage of fully charged. A user might decide to turn off Bluetooth on his or her mobile device when battery is low in order to extend the life of the mobile device on the current charge.

TABLE 6.2: Properties in the State & Notification Broker related to Network and Power information.

Social Context

As described in Section 3 social contexts could be the current activity of the user, the motives or plans for the day. In Table 6.3 the properties related to context are shown.

Active application	The active application on the mobile device gives an indication of the users current activity or context i.e. if the current application is a game, the user might be in a boring context such as on the bus.
Phone calls	Incoming, outgoing and missed phone calls can be used to explain sudden distractions as well as indicate whether or not the user is available.
Unread messages	Just like phone calls, the amount of unread SMS, MMS and E-mail messages can indicate whether the user is busy or not (or if the user forgot the mobile device).
Device profile	By capturing the current device profiles status it is possible to collect information (defined manually by the user) about the context. The devices profile changes settings such as ringtone, ringvolume and vibrator. Typical settings could be: ”Normal”, ”Meeting” and ”Outdoors”.
Calendar and events	The calendar is also a source for context information specified manually by the user. Appointments in Pocket Outlook (the default calendar and event application on a Microsoft Windows Mobile enabled mobile device) describe parameters such as subject, time, location, expected duration, reoccurence and more.

TABLE 6.3: Properties in the State & Notification Broker related to Context Information.

In the RECON log, a specific module for accessing the State & Notification broker is created, which deals with configuring and registering event handlers and relaying events to a the mainform that saves the event information in the database on the device.

Wireless Networks

Most newer mobile devices have access to network technologies such as GSM, Bluetooth or WiFi, however some are easier to access than other when using the Microsoft .NET Compact Framework. Power consumption of these technologies also needs to be considered as some network technologies can drain battery without the proper settings.

When performing field evaluations, the location is important, and as mentioned earlier, a lot of methods exist to extract this information, but some require more effort to realize than other.

A GPS module is not available on the device, and no external hardware module was available. GSM was not accessible through the Microsoft .NET Compact Framework 2.0 and no third party API for this exist at the time of writing.

As a result of these difficulties, the use of WiFi and Bluetooth are chosen, as both of these are available and easily accessible on the device. It is expected, that a rough estimate of a location can be made by looking at nearby WiFi access points or Bluetooth devices.

6.3.4 Reporter

The reporter is the component in the RECON log which is responsible for the communication between the mobile device and the server. Since it is not possible for the server to send data to the mobile devices without a request they must regularly connect and request data. The process of triggering a hook till it ends up in the database on the server is illustrated on Figure 6.5.

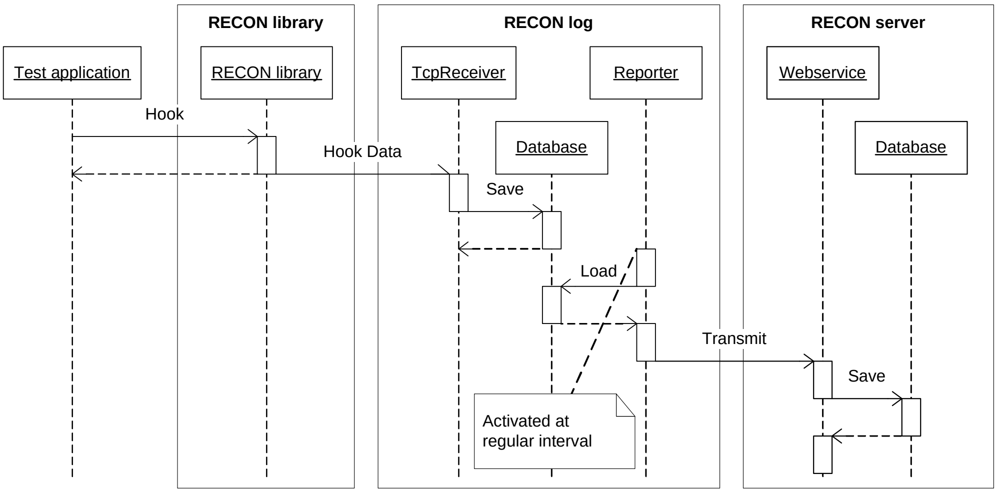


FIGURE 6.5: A sequence diagram showing the communication process.

In the RECON framework there exist two types of information which must be sent between the server and the mobile devices. The information which needs to be sent from the server to the mobile devices is configurations-files and FSM's and in the opposite direction captured data.

Figure 6.5 shows a hook triggered in the test application, which activates the RECON library. The RECON library transfers the hook data via a local TCP connection to the TcpReceiver in the RECON log. The TcpReceiver saves the hook data in the database. At a specific interval, the Reporter will attempt to transfer the contents of the database on the mobile device to the server. First the Reporter loads the data, and afterwards transmits it to the server via the webservice, that subsequently saves it in the database located on the server. The server part is explained in Section 6.4.

How often the RECON log must connect and transmit or receive data is defined in the configuration-file. Furthermore the preferred connection type is also defined in the configuration-file. If a mobile device fails to obtain a configuration-file form the server a boot time a default configuration will be used.

6.4 Server

The server part of the RECON framework will be referred to as the RECON server. The RECON server has two primary tasks to perform; it will regularly receive and store captured data from one to many mobile devices at once and it should send configurations to the same devices when they request it.

As was mentioned in Section 6.2, the RECON server will be implemented as a webservice. The server will expose a range of methods that the clients can use to either report data or retrieve configuration files. A graphical representation of the server architecture is shown on Figure 6.6.

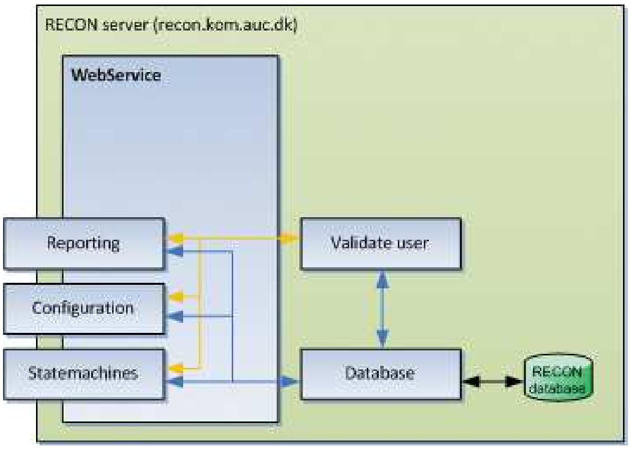


FIGURE 6.6: The architecture of the RECON server.

Other than three methods, the webservice also contains a user validation module and access a database via a database module. The webservice methods and modules will be further described in the following sections.

6.4.1 Webservice methods

The webservice is implemented on an Internet Information Server 5.1 and exposes the following three methods:

Reporting In order for the client to be able to report data captured, the server exposes a webservice method that saves entries of captured data to a database. To identify which client is reporting, the user name is saved along with all data entries.

Configuration Whenever the RECON log is started on a mobile device, some of the modules in the client needs a configuration. A default configuration file is shipped with the client, allowing it to work even though no network connection is available. The configuration-file is simply a xml file in which it is possible to defines walues for all the essential functionalities in the RECON log. During the startup, if a network connection is available, the client queries the server about the date of the configuration on the server. If the configuration residing on the server is newer than the one currently on the mobile device, a new configuration file is retrieved. This way a new configuration is only transmitted to a client whenever it is needed.

State machines As mentioned earlier, the client can manage a list of FSM's than can trigger surveys when a certain path is chosen in an application. The FSM is requested from the RECON Server whenever the client starts. Information about what to ask for in a survey, and what options should be available are sent with the FSM from the server.

6.4.2 Validation

The module ”Validate user” is used by all the methods in the webservice (”Reporting”, ”Configuration” and ”Statemachines”) for determine whether a username and password is correct or not and to distinguish between captured data from different test subjects. In this proof-of-concept study, the username and password is always the phone number of the mobile device on which the evaluation is performed.

6.4.3 Database

The database module is the central module in the RECON server and handles all communication with the RECON database (Figure 6.2). Besides from handling all database communication this module also function as an abstraction layer over the database. The abstraction is important since it in a final implementation properly would be desireble to replace the MySQL database. The choice of database is based on availability and the fact that it is free.

The RECON database

The RECON database (Figure 6.7) is a MySQL version 5 database which consists of nine different tables. Based on functionality it is possible to divide the nine tables in three times three tables. The three groups of tables and their function is described in the following.

User management The three tables related to the user management in RECON is: ”group”, ”groupMember” and ”phone”. The table ”phone” holds all user data. In the table ”group” is all the different user groups defined. A user group could e.g. be a research team evaluating a specific application. Assignment of users to groups is done in the last table ”groupMember”.

Survey functionality The tables which is related to the survey functionality is: ”questionnaire”, ”questionnaireQuistion” and ”questionnaireOption”. A survey is created in the table ”questionnaire” and questions added to the survey in ”questionnaireQuistion”. The options for each question is added in the table named ”questionnaireOption”.

State machines The tables related to the FSM's in RECON database is: ”task”, ”states” and ”transition”. Each FSM is in the RECON Freamework supposed to model one task (a task could e.g. be printing a file) and is defined in the table named ”task”. Each FSM consist of a number of states and transitions defined in: ”states” and ”transition”.

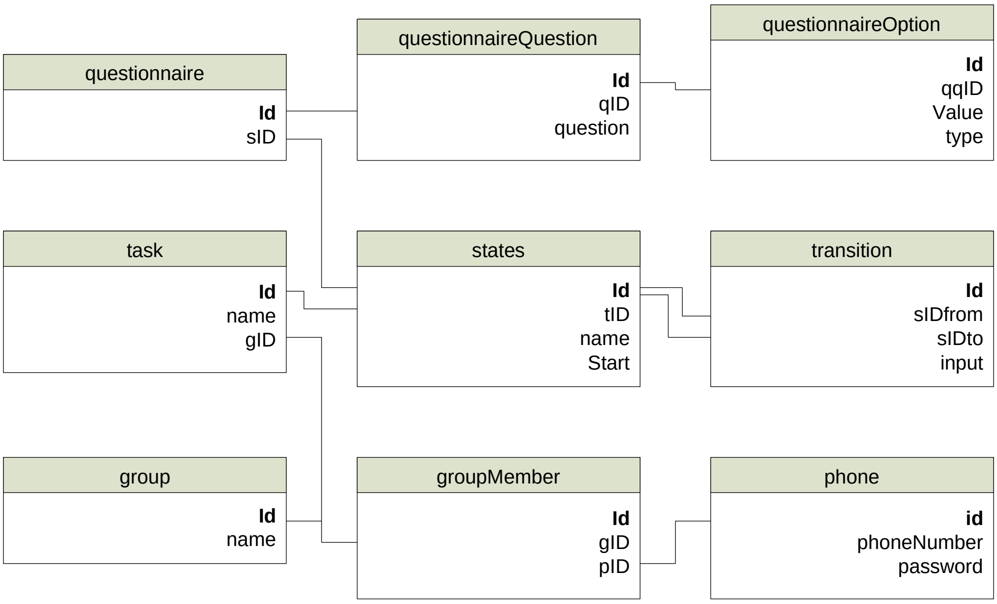


FIGURE 6.7: The RECON database.

6.5 Summary

This chapter is divided in the following four sections: Overview, Communication, Mobile and Server.

First the overall design of the RECON framework and the split in a mobile and server part was described. Each part of the RECON framework was named according to their functionality; the dynamically linked library will be referred to as the RECON library, the mobile standalone application as the RECON log and finally the server part as the RECON server. Additionally several methods for storing log data were presented, and based on pros and cons it was chosen to temporarily store captured data on the mobile devices and transmit any captured data when the desired network connection is available.

In the Communication section, different communication protocols were presented. Based on pros and cons it was decided that the interface between the mobile devices and server would be based on SOAP and webservices.

After defining the overall architecture and the communication in the RECON framework, the RECON library and RECON log was described. Four subsections each described a functionality of the RECON frameworks mobile part. The subsections were; Usage information, Attitude information, Context information and Reporter. The first three subsections describe how information regarding each category is obtained. In the section Reporter, the communication module of the RECON log was described.

The RECON server was described with a short overview of the webservice. Afterwards the methods that the RECON service exposes were described in detail, followed by a description of the validation module. Finally the database on the server side, where all the captured from the mobile devices will be stored, was described.

Part III

Experiments and Results

This part documents the experiments performed and results gathered in order to conclude on the research questions created to elucidate the working hypothesis of the study.

CHAPTER 7

EXPERIMENT DESIGN

This chapter is split in two sections, one establishing a list of research questions (each with the purpose of uncovering different aspects of the working hypothesis) and the other section defining experiments to capture data in order to shed light on the research questions.

How the research questions are established and relate to the working hypothesis is covered in Section 7.1. The establishment of the desired experiments and their relevance for answering the research questions is described in Section 7.2. Results obtained from the experiments will be presented and concluded upon in Chapter 8.

7.1 Research questions

As explained in the Introduction (Chapter 1) the purpose of the working hypothesis was to form the basis for the study. Since the working hypothesis is broad it is decided to elucidate it through a range of research questions, which will be presented in this section. For clarification, the hypothesis is restated below:

> *It is possible to produce the same or more elaborate results in an automated field evaluation with the same or less effort as in a traditional field evaluation.*

As was written in the Introduction, this study contains both an engineering aspect as well as a scientific. The scientific aspect is achieved by establishing a working hypothesis and research questions and design and perform experiments in order to shed light on these. In order to do these experiments, the RECON framework was developed, which provides the engineering aspect of the study. By being able to capture data to evaluate the working hypothesis and research questions, the engineering aspect of the study is also evaluated.

Based on the working hypothesis, the following research questions have been derived, and was elaborated on, through the use of data captured with the RECON framework. The research questions was divided in two groups; questions related to effectiveness and questions related to efficiency. Research question RQ1 to RQ5 all relates to evaluating the RECON fremeworks effectiveness and RQ6 the efficiency.

RQ1 Is it possible to detect usability issues?
- What type of usability issues occurred according to Molich's taxonomy [Molich, 2007]?
- Why the usability issues occurred e.g. based on usage, context or attitude information?

RQ2 Is it possible to extract use patterns?
- How about changes over time?
- Can an extracted use pattern be linked to the context they occurred in?

RQ3 Is it possible to determine the users attitude?
- At any point i.e. continuous?
- For a specific part of the application?

RQ4 Is it possible to detect a distraction of the test subject?
- By persons nearby?
- How does the detection of persons nearby affect power consumption?
- Events on the mobile device (incoming calls and sms, appointments and alarms)?

RQ5 Is it possible to determine the test subjects position?
- If so by which accuracy?
- How does the positioning affect the power consumption?

RQ6 How efficient is the RECON framework?
- Must the source code be available for alteration or recompilation?
- What is the time consumption of the evaluation for 1, 2, 10 and 100 test subjects?

In the following, each of these research questions will be elaborated as to why they are worth investigating.

RQ1 - Detection of usability issues

When developing a framework for usability evaluation the goal will typically vary depending on who the users are. For HCI practitioners the goal could be to detect usability issues whereas for a HCI theorists the goal could be to perform longitudinal experiments in order to uncover new aspects of HCI on mobile devices. However detecting usability issues is not enough, if it not possible to explain why issues occurred or why a test subjects uses an application the way he or she does. The usability issues detected can, based on [Molich, 2007], be rated as either cosmetic, serious or critical, indicating to which degree they affect the overall usability.

RQ2 - Use patterns

Detecting and explaining usability issues can be done by looking at a test subject's use pattern in the application. Change in use patterns over time, such as a faster execution of certain tasks, can give an idea about the memorability and learnability (described in Section 3) of an evaluated application. If a particular use pattern can be linked to a certain context, the application evaluated can be designed to adapt to this specific context, thereby making the application context-aware.

RQ3 - Attitude

When performing a traditional field evaluation, the test subject is often encouraged to ”think out loud” in order for the HCI researchers to determine the users opinion and attitude towards the application. If the attitude is directed at a specific element of the application, the developer can attempt to improve it based on the opinions of the test subjects i.e. if they do not like the color of a specific button or find that an icon is misrepresentative for the functionality it performs.

RQ4 - Distraction of the test subject

When a test subject suddenly pauses in his or her interaction with an application, this could be considered a usability issue such as being confused as to where to go next or the result of a specific action, however it could also mean that the test subject is paying attention to someone or something else. Therefore detecting these distractions can help explain sudden pauses.

The only way to detect persons nearby in the RECON framework, is by the use of Bluetooth inquiry scan, which indeed is known to be power consuming. Due to this, it is necessary to evaluate the power consumption further.

RQ5 - Position

The position of a test subject is a valuable information, when determining the subjects current context. An example where the the position can tell a lot about the context is when the test subject e.g. is located at home or at work. Since the only way to determine position is via either WiFi og Bluetooth, which both require power and might not necessarily be activated, the same power considerations as in RQ3 applies.

RQ6 - Efficiency

If one evaluation method requires less effort and time to use than another, and both yield the same results, it is easy to choose which method to use. Therefore the efficiency of evaluation method has been investigated. Some of the parameters that can affect efficiency is the amount of extra code or time required to use the RECON framework.

In the next section the experiments performed in order to answer the research questions will be defined.

7.2 Experiments

The research questions set forth in the previous section will be elucidated by analyzing data from experiments captured with the RECON framework. The purpose of this section is to explain the setup and execution of these experiments, performed to gather these results.

Instead of performing one large experiment, four smaller experiments and a questionnaire have been performed.

The four experiments is numbered from E1 to E4 and described in Section 7.2.1 to 7.2.4. Table 7.1 shows how each experiment and the questionnaire relates to one or more of the research questions (RQ1 to RQ6) described in Section 7.1. As can be seen E2 is the main experiment.

Experiment	**RQ1**	**RQ2**	**RQ3**	**RQ4**	**RQ5**	**RQ6**
E1 - Proof of Concept		✓				✓
E2 - User Test	✓	✓	✓	✓	✓	✓
E3 - Power Consumption						
E4 - WiFi Evaluation					✓	
Questionnaire						✓

TABLE 7.1: Matrix describing the relations between the performed experiments and the research question.

7.2.1 E1 - Proof of concept

The purpose of this experiment is to verify that the RECON framework can capture usage information, and that it is possible to derive use patterns from the information in order to answer research question RQ2. The application chosen is called DIAL2 (developed in the summer 2006 by Kenneth Holm Andersen and Tais Holland Mogensen). Finally the effort and time required to use the RECON framework in this experiment, gives an indication of its usefulness, thereby supporting research question RQ6.

Setup

In order to setup DIAL2 for the evaluation, the guide in Appendix B was followed.

The RECON library was included in the DIAL2 C#-project and the desired hooks inserted; everything was compiled and afterward deployed at the mobile device. Before DIAL2 was started the RECON log was deployed and started on the device.

To collect information describing the users navigation in the DIAL2 application a total of 18 hooks was necessary; one hook in each GUI-screen, four hooks in the data transmission and an extra hook in each of the GUI-screens which handles user input (Figure 7.1, the placement of the hooks is marked with read bullets).

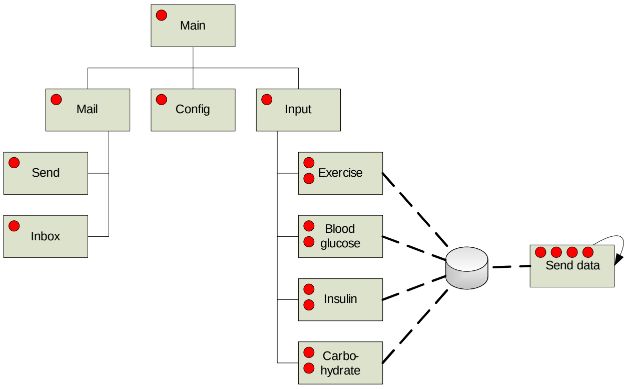


FIGURE 7.1: Graphical representation of the DIAL2 structure with read bullets indicating the RECON hooks.

Since DIAL2 originally was developed for a PDA without Microsoft Windows Mobile 5.0, it was necessary to port it to the new platform. The port was a simple replacement of non-supported GUI components e.g. buttons and radiobuttons.

Conceptual model DIAL2 is an application which can be used in the treatment of diabetes. DIAL2 was originally intended to replace the patients' old paper based log book and by the replacement making the patients life easier.

As shown on Figure 7.2 DIAL2 is based on a client-server architecture. The client is a simple user interface which allows the user to enter measurements (blood glucose, carbohydrates and level of exercise) manually. Besides from user input the client also supports automatic data collection by the use of Bluetooth.

The server part of DIAL2 is supporting the client with two main functionalities which is logging and smart suggestion of input values based on domain knowledge and user history. The smart suggestion of input

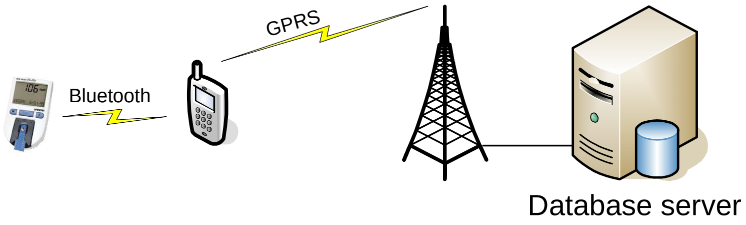


FIGURE 7.2: Overall architecture of DIAL2.

values was one of the new concepts in DIAL2 compared with DIAL (an earlier version developed a year and a half earlier at AAU). The concept behind the smart suggestion of input values is that by using log data and domain knowledge it is possible to develop a more user friendly UI. The UI proposes, depending on the time and previous measurements, values and types e.g. a blood glucose level of 4.2 mmol/l at 11AM. Unfortunately the smart suggestion of input values part in DIAL2 was only implemented as a code stub.

The intended use of DIAL2 is that the patient uses the client application every time he eat, exercise, measured blood glucose level or took insulin. Over time DIAL2 would adopt time, amount and type of user input and automatically suggest values.

Execution

The DIAL2 application was evaluated with the RECON framework, with the author of this study who was not familiar with DIAL2, as test subject. The test subject was instructed in what the application could do and what limitations there were. A longitudinal evaluation with a truly neutral test subject would be ideal, however a short pilot evaluation has been chosen because of the study time limit. The pilot evaluation took 15 minutes and all functionality in the DIAL2 application was activated in order to verify that all hooks were inserted properly and were working.

7.2.2 E2 - User test

The goal of this experiment was to capture realistic usage data by the use of the RECON framework combined with traditional video recordings. The experiment supported many of the research questions as the evaluation was done with four test subjects and the subjects were instructed to follow a range of tasks at various locations.

Setup

An experimental application was developed for the test, and hooks inserted. The development of this application is described in the following.

Conceptual model The experimental application to use in the evaluation is a time management and scheduler application named ”Time and Notes”. The application allows a test subject to add reminders, countdowns and notes to the application. Whenever a reminder is due or a countdown expires, the user will be warned with a messagebox and a sound.

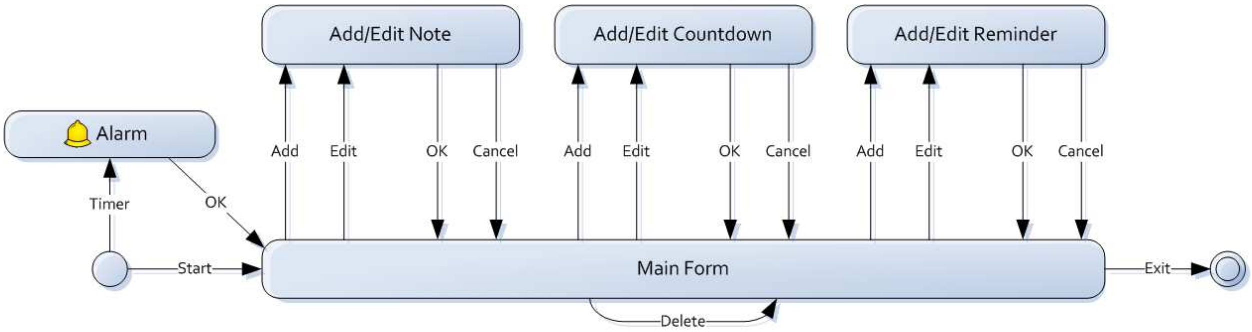


FIGURE 7.3: A state-diagram of ”Time and Notes”.

On Figure 7.3 a state-diagram of ”Time and Notes” can be seen. The application was implemented in C#.NET and the RECON library included in the application in order to insert the hooks.

The actual screens of the finished application can be seen on Figure 7.4. The application is included on the CD at ”Source\TimeAndNotes\”.

Execution

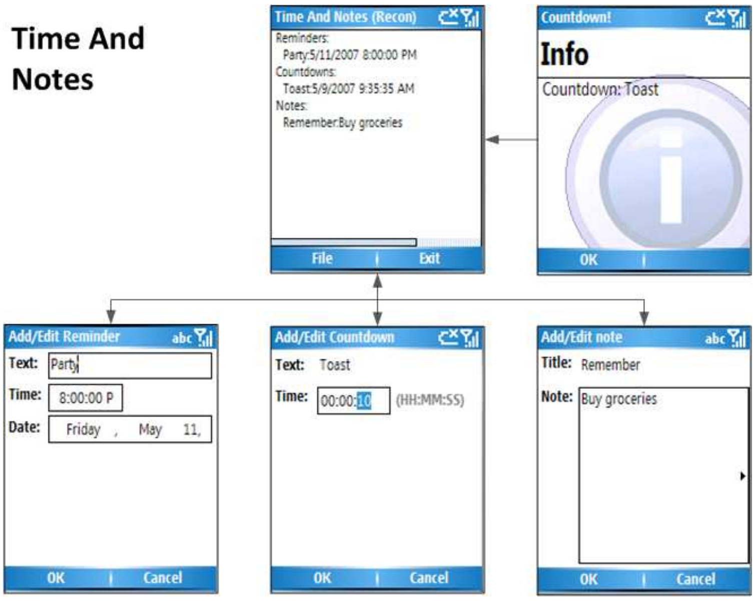
The experiment consist of four test cases which each of the test subjects performed at the premises of Aalborg University, Fredrik Bajers Vej 7, more specifically in A4-205, in the cantina and at the bus stop. The test cases performed is shortly described in the following and an extended description of the test cases can be found in Appendix A.1.

Test case 1 In A4-205: Add a countdown.

Test case 2 In the cantina: Create a note with the daily menu.

Test case 3 At the busstop: Add a reminder which must trigger at 20.00 with the message beer at pollyt’s.

Test case 4 In A4-205: Change profile on the mobile device to ”meeting” or ”silent”.



Reminders:
Party:5/11/2007 8:00:00 PM
Countdowns:
Toast:5/9/2007 9:35:35 AM
Notes:
Remember:Buy groceries

Info
Countdown: Toast

Text: Party
Time: 8:00:00 P
Date: Friday , May 11,

Text: Toast
Time: 00:00:10 (HH:MM:SS)

Title: Remember
Note: Buy groceries

FIGURE 7.4: Screen of the finished ”Time and Notes” application.

All test subjects were asked to sign a declaration of consent (Appendix A.3), in order to use the data captured in the study. After signing the declaration of consent the test subject performed the four test cases described above. To gather more qualitative information from the test subjects who participated in the experiment they were asked to fill out a short questionnaire and if possible to elaborate on the questions. The questions on the questionnaire was selected in order to support one or more of the research questions. The questionnaire used can be seen in Appendix A.2.

7.2.3 E3 - Power Consumption

As the RECON framework makes intensive use of WiFi and Bluetooth inquiry on the device, the power consumption of different configurations of the framework was investigated. The RECON framework works by sampling the nearby WiFi access points Bluetooth devices at a regular interval.

Setup

In order to measure the power consumption, a simple test application has been developed. The applications purpose is to write down the time of startup in a file and then write down the current time every minute. When the mobile device runs out of battery it shuts down, and the program is closed. When the mobile device is recharged, the start time and end time of the application can be read from the text file.

Test	WiFi inquiry interval	Bluetooth inquiry interval
Test 1	Off	Off
Test 2	5 mins	Off
Test 3	5 mins	7 mins
Test 4	5 mins	10 mins

TABLE 7.2: Power consumption for different inquiry setups.

This test application is run alongside the RECON framework, which use different WiFi and Bluetooth configurations.

Execution

The experiment was divided in four sub-tests one for each of the WiFi and Bluetooth configuration listed in Table 7.2.

The experiment is performed without any user interaction on the device. The experiment is done in an uncontrolled environment, meaning that nearby devices and access points could change, which can affect the results of the experiment.

Each of the four sub-tests was performed as described below.

- When the battery in the mobile device was fully charged the test begun.
- RECON is configured in accordance with the actual test number e.g. Test 1: WiFi and Bluetooth inquiry disabled (See Table 7.2).
- WiFi and Bluetooth were enabled on the device. Bluetooth was enabled so that it was not visible to other devices.
- The power supply was disconnected.
- The developed test application and RECON was booted.
- When the mobile device was switched off due to power loss the power supply was plugged back in and the device booted.
- The battery standby time was written down (read from the text-file).

7.2.4 E4 - WiFi Experiment

The purpose of the WiFi experiment is to visit the three primary locations of the user test (Experiment E2), in order to detect available WiFi access points and afterwards creating templates describing there SNR values. This was done to be able to infer the test subjects location in the user test, based on the available access point at the location and their specific SNR values.

Setup

The RECON framework was configured to sample the WiFi access points every 15th second and started on the device.

Execution

The device was moved between each of the desired locations. At each location the device was held still for minimum five minutes which produced 18 samples due to the sample rate used.

CHAPTER 8

RESULTS

In this chapter the research questions introduced in Section 7.1 will be discussed based on the data obtained through the four experiments E1 to E4, and the questionnaire.

8.1 RQ1 - Detection of usability problems

As mentioned in Section 7.1, a framework for evaluating the usability of an application must be able to detect usability issues. Research question RQ1 asks if it is possible to detect usability problems, and if so, if it is possible to explain them. Since experiment E1 and E2 was the only experiments able to discover usability issues they are of interest for answering RQ1, however as E1 was performed primarily to verify application independence focus will be on experiment E2, the user test.

The user test that was performed allowed the RECON framework to capture data for four test subjects while performing four test cases as described in Appendix A.1. Despite the simplicity of the application and the short evaluation, usability issues could be derived from the gathered data.

Date	Time	Log	Description
04-05-07	12:32:21	1;Time And Notes (Recon):Add/Edit note	Open the dialog: Add/Edit note
04-05-07	12:32:21	7:Add Note	Add Note selected
04-05-07	12:32:26	9:AddEditNote Cancelled	Dialog cancelled
04-05-07	12:32:26	25:MainForm Focus	MainForm got focus
04-05-07	12:32:26	1;Add/Edit note:Time And Notes (Recon)	Switch dialog
04-05-07	12:32:29	1;Time And Notes (Recon):Add Reminder	Open the dialog: Add Reminder
04-05-07	12:32:29	1:Add Reminder	Add Reminder selected
04-05-07	12:33:13	2:Reminder saved (04/05/2007 20:00:00 - pollys)	Reminder saved
04-05-07	12:33:14	25:MainForm Focus	MainForm got focus
04-05-07	12:33:14	1;Add Reminder:Time And Notes (Recon)	Switch dialog

TABLE 8.1: A part of the captured data indicating a usability issue for test subject 4.

Table 8.1 shows an annotated version of the captured data containing events revealing a possible usability issue. The purpose of the test case was to add a reminder, but as can be seen from the captured data, the test subject select Add Note instead of Add Reminder, and five seconds pass before he goes back to the main form of the application.

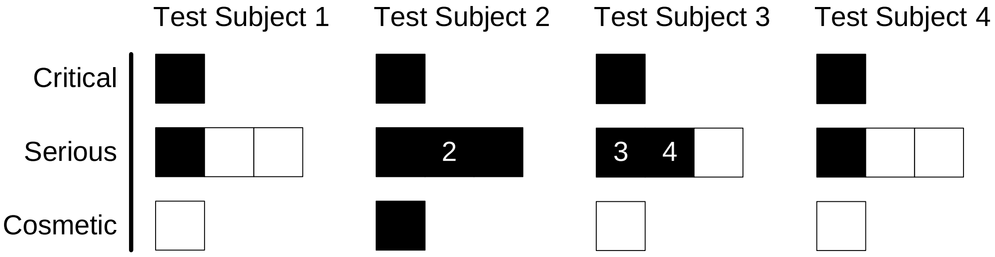


FIGURE 8.1: Overview of issues detected by the RECON framework. A number in the box specifies occurrences, if the issue happened more than once.

In Figure 8.1 an overview of the five unique issues detected and their severity according to Molich's taxonomy [Molich, 2007] which categorizes issues according to the following:

Critical issues.

- Recurred across all subjects.
- Stopped subjects completing tasks.

Serious issues.

- Recurred frequently across subjects.
- Inhibited/slowed down users completing tasks.
- Subjects could (eventually) complete tasks.

Cosmetic issues.

- Did not recur frequently across subjects.
- Did not inhibit subjects severely.
- Subjects could complete tasks.

The single critical issue was discovered across all subjects, and was caused by the mobile device returning to the Microsoft Windows Mobile Home screen when ending a phone call. This issue was not fatal, but inhibited the test subjects who were unexperienced with the mobile device used. The issue can be explained, as the issue occurred as soon as an incoming call ended.

The serious issues that were discovered, was mainly caused by cancellations in dialogs and accidental button pushes, which either switched to Internet Explorer or Media Player. The cancellations was due to usability issues in the test application and the button pushes due to a pure design of the mobile device.

The single cosmetic issue experienced by subject 2 happened when he decided to create a note, where a reminder was supposed to be created according to the test briefing (See Appendix A.1). It was not possible to explain the cause by looking at the captured data and the video recording of the same issue did not provide additional information, as the subject did not provide any explanation for the choice.

Partial conclusion

The results of the user test experiment E2 resulted in 5 unique usability issues (18 occurrences) detected by the RECON framework, however some of them were impossible to explain based on the data captured by the framework.

One of the reasons for not being able to explain these issues was that they were not due to the test application (Time and Notes), but the design of the mobile device or the operating system. Also the cosmetic issue was explainable only due to the users following a scripted test, and as such it was known what the user was supposed to do.

The results captured, indicate that the RECON framework can actually detect usability issues, however explaining them can be troublesome based on context with a non-context-aware application. It was possible to rate the usability issues according to Molich's taxonomy, however this was made easier because of the controlled nature of the evaluation.

One way of explaining cancellations such as the one described could be to query the test subjects about their actions just as one might query them about their attitude, which is described in research question RQ3.

8.2 RQ2 - Use patterns

By inspection of the captured data from experiment E2 it was possible to see exactly what parts of the application the subjects visited, when they did it and the amount of time between each UI interaction. The data shown in Table 8.2 is part of the data collected for test subject 1 during test case 1.

On Figure 8.2 the same data has been manually visualized as a state diagram using the method described in Section 5.2, in order to give an overview of the use pattern.

The use patterns show how the user perform certain tasks in the application, and capturing this information for several repetitions of the task can give an indication as to whether the test subjects are improving

Date	Time	Data
04-05-07	12:09:49	Start
04-05-07	12:09:51	25:MainForm Focus
04-05-07	12:09:51	1;:Time And Notes (Recon)
04-05-07	12:10:01	4:Add Countdown
04-05-07	12:10:01	1;Time And Notes (Recon):Add Countdown
04-05-07	12:10:08	25:MainForm Focus
04-05-07	12:10:08	6:AddEditCountdown Cancelled
04-05-07	12:10:08	1;Add Countdown:Time And Notes (Recon)
04-05-07	12:10:22	4:Add Countdown
04-05-07	12:10:22	1;Time And Notes (Recon):Add Countdown
04-05-07	12:10:30	25:MainForm Focus
04-05-07	12:10:30	5:Countdown save (04/05/2007 12:10:39 - done)
04-05-07	12:10:30	1;Add Countdown:Time And Notes (Recon)
04-05-07	12:10:40	18:CountdownAlarm
04-05-07	12:10:43	24:CountdownOK
04-05-07	12:10:43	25:MainForm Focus
04-05-07	12:10:44	1;Time And Notes (Recon):UserInput
04-05-07	12:10:52	25:MainForm Focus
04-05-07	12:10:52	1;UserInput:Time And Notes (Recon)
04-05-07	12:10:52	1:1:3:true
04-05-07	12:10:52	SURVEY_SEND

TABLE 8.2: Captured data concerning one of the test subjects use pattern form test case one.

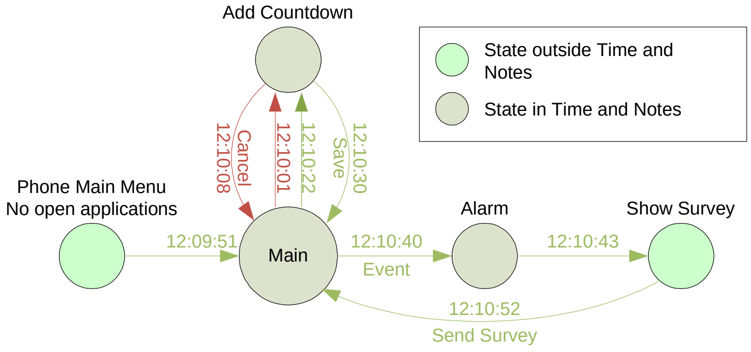


FIGURE 8.2: Graph showing one of the test subjects use pattern from test case one.

their efficiency or not. It is not possible to observe this in the data obtained during experiment E2 since each test subject only performed each task once. It is indeed still possible to verify indirectly by comparing the obtained results across different test subjects (pretending that all data obtained during a single

Test subject	Date	Time	RECON log
1	04-05-07	11:04:40	4:Add Countdown
	04-05-07	11:04:57	5:Countdown save (04/05/2007 11:05:06 - test)
	Total:	*17s*	
2	04-05-07	11:26:54	4:Add Countdown
	04-05-07	11:27:11	5:Countdown save (04/05/2007 11:27:21 - hej)
	Total:	*17s*	
3	04-05-07	12:10:22	4:Add Countdown
	04-05-07	12:10:30	5:Countdown save (04/05/2007 12:10:39 - done)
	Total:	*8s*	
4	04-05-07	12:26:15	4:Add Countdown
	04-05-07	12:26:33	5:Countdown save (04/05/2007 12:26:43 - test)
	Total:	*18s*	

TABLE 8.3: Captured data obtained during experiment E2 - Task: ”Add countdown”.

test case is from the same test subject). In Table 8.3 a comparison across different test subjects is done for test case 1 of the user test (Experiment E2). The Add Countdown event signals that the test case has been started, and when the countdown is saved the test case is complete. The results show that test subject 3 is faster than the other test subjects, however this is not considered remarkable, as the application is new to all the subjects. In order to get significant results, each test subject should perform the task several times.

Partial conclusion

The RECON framework is capable of capturing all user actions (that is hooked) in the application. As each event is timestamped, use patterns can be derived from this event and visualized e.g as shown on Figure 8.2. If it is possible to determine, what task a test subject is performing, it would be possible to detect improvements (over time) in the way they use the application. When comparing the results from the RECON framework with the video recorded during the user test, extracting the use patterns is also significantly easier with the RECON framework as every event is timestamped.

8.3 RQ3 - Attitude

Without being able to record audio or video, a framework will have a hard time capturing continuous attitude information from the test subjects even if the ”think out loud” method is being used. However by using surveys at certain events in the system, it was possible to sample the users attitude towards a specific part of the application.

In the user test (Experiment E2), all test subjects were presented with a survey after an alarm in the ”Time and Notes” application. The survey asked if the volume of the alarm was ”Too high”, ”Too low” or ”OK”. The results of the experiment are shown in Table 8.4.

	Too low	**OK**	**Too high**
Test subject 1	✓		
Test subject 2		✓	
Test subject 3	✓		
Test subject 4		✓	

TABLE 8.4: Captured data concerning one of the test subjects use pattern form test case one.

In this case, the survey functionality of the framework was used to ask the test subject about his opinion towards one specific feature of the application. However it can also be used to ask the user about his current location or, as mentioned in the partial conclusion for RQ1, allow the test subject elaborate on his actions at a certain event.

Partial conclusion

It was possible to capture attitude information towards a single aspect of the application, however the RECON framework was not capable of capturing continuous attitude information.

This way of capturing attitude information will remind the test subject that he is part of an evaluation and will seem intrusive. In the questionnaire, all four test subjects disagreed that the survey functionality was disturbing, however two commented that the reason was, that it was only a single survey. Therefore a more thorough experiment needs to be done in order to compare the intrusiveness versus the benefit of the results from this way of sampling the attitude in order to give a final answer on this question.

8.4 RQ4 - Distraction of the test subject

As was mentioned under RQ1, it is difficult to explain the cause of some of the usability issues detected. During test case two in the user text (Experiment E2), one of the test monitors phoned the test subject. This call resulted in a critical usability issue in the application.

In Table 8.5, data from this test case can be seen. It can be verified that the RECON framework is capable of collecting information about events on the mobile device (the events of interest is marked with green in Table 8.5), which can be a distraction of the test subject. The table shows the events in the data that indicate the incoming phone call and a phone of a nearby friend.

Date	Time	Log	Description
04-05-07	12:28:32	7:Add Note	
04-05-07	12:28:33	1;Time And Notes (Recon):Add/Edit note	
04-05-07	12:28:42	6;0	
04-05-07	12:28:45	1;Add/Edit note:	
04-05-07	12:28:47	1;:Phone - Incoming...	Answer - Incoming call
04-05-07	12:28:47	12;1	One active call
04-05-07	12:28:51	12;0	No active calls
04-05-07	12:28:52	1;Phone - Incoming...:	End - Phone call
⋮	⋮	⋮	
04-05-07	12:32:52	Qtek 8310;0012377CAEA2	Buddy's phone
04-05-07	12:33:09	6;1	
04-05-07	12:33:13	2:Reminder saved (04/05/2007 20:00:00 - pollys)	

TABLE 8.5: Captured data concerning one of the test subjects use pattern from test case two.

Despite being able to detect distractions such as incoming phone calls or nearby persons, the RECON framework is not capable of detecting other distractions from the environment such as people who do not own a mobile device or have Bluetooth enabled or available on their mobile device.

Detecting nearby people by Bluetooth is power consuming, as they an inquiry must be performed every time the information is needed. In order to measure the power consumption of the Bluetooth inquiry scan, experiment E3 was done. In the experiment the RECON framework was started with four different configurations and the uptime measure. The results of the experiment can be seen in Table 8.6.

Table 8.6 shows that WiFi inquiry does not have a significant effect on the power consumption, however Bluetooth does.

Partial conclusion

The RECON framework is capable of capturing some distraction events, as long as they originate from the mobile device. In order to detect more distractions, the framework should implement a way of using the microphone on the device and possibly the camera to e.g. measure the sound intensity and light level etcetera.

The Bluetooth inquiry scan will definitely affect the uptime of the mobile device on which it is installed, but a configuration can be made to at least keep an uptime of more than 24 hours. For a longer uptime, a tradeoff between battery life and Bluetooth inquiry interval must be done.

Test	WiFi inquiry interval	Bluetooth inquiry interval	Uptime
Test 1	Off	Off	24+ hours
Test 2	5 minutes	Off	24+ hours
Test 3	5 minutes	10 minutes	10 hours 15 minutes
Test 4	2 minutes	5 minutes	7 hours

TABLE 8.6: Power consumption for different inquiry setups. The resulting uptime is rounded off to nearest 15 minutes.

8.5 RQ5 - Position

No direct position information is capturable by the RECON framework, however as mentioned in the Context chapter (Chapter 3) several methods exist to obtain a position based on network information. The only network information available in the RECON framework that can be used to relate to a position is WiFi, because of the stationary access points. One way of inferring a position is by using triangulation, however this requires alot of information about access points positions and radio wave propagation, which unfortunately is not available.

Instead the location was inferred by templates based on the SNR values of the seven most common access points for all test subjects during the user test (Experiment E2). The purpose of experiment E4 was to sample the three primary locations used in the user test, in order to create a range of templates, so that positions could be inferred from the data captured during the user test.

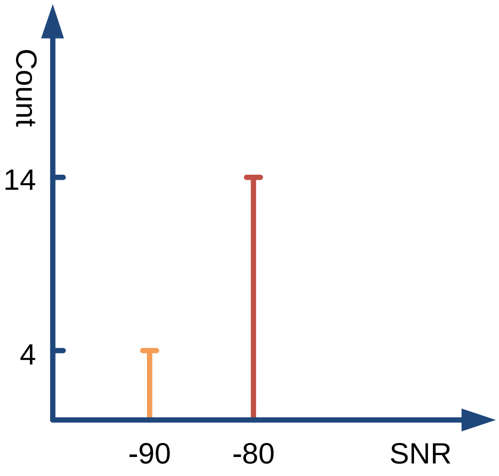


FIGURE 8.3: Summation of the data from Table 8.7.

After performing experiment E4 the RECON database contained 18 entries for each of the seven access points of interest. Table 8.7 is the part of the data captured concerning the access point with MAC address: 00-0E-84-C2-C2-A0 (AP4). By inspection of the table or Figure 8.3 it is possible to see that the SNR values differ even if the position is fixed e.g. the lowest SNR value measured for AP4 is −90 and the highest −80. In such a case, the most prevalent value is chosen, which is −80.

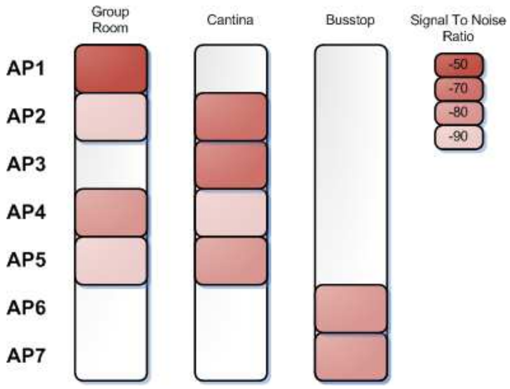


FIGURE 8.4: WiFi templates

Based on data from the seven most common access points, similar to what is shown in Table 8.7, is it possible to infer the three WiFi template shown on Figure 8.4.

Date	Time	MAC address	SNR
15-05-07	10:43:51	00-0E-84-C2-C2-A0	-80
15-05-07	10:44:11	00-0E-84-C2-C2-A0	-80
15-05-07	10:44:31	00-0E-84-C2-C2-A0	-80
15-05-07	10:44:51	00-0E-84-C2-C2-A0	-90
15-05-07	10:45:11	00-0E-84-C2-C2-A0	-80
15-05-07	10:45:31	00-0E-84-C2-C2-A0	-90
15-05-07	10:45:51	00-0E-84-C2-C2-A0	-90
15-05-07	10:46:11	00-0E-84-C2-C2-A0	-80
15-05-07	10:46:31	00-0E-84-C2-C2-A0	-80
15-05-07	10:46:51	00-0E-84-C2-C2-A0	-80
15-05-07	10:47:11	00-0E-84-C2-C2-A0	-80
15-05-07	10:47:31	00-0E-84-C2-C2-A0	-80
15-05-07	10:47:51	00-0E-84-C2-C2-A0	-80
15-05-07	10:48:11	00-0E-84-C2-C2-A0	-90
15-05-07	10:48:31	00-0E-84-C2-C2-A0	-80
15-05-07	10:48:51	00-0E-84-C2-C2-A0	-80
15-05-07	10:49:11	00-0E-84-C2-C2-A0	-80
15-05-07	10:49:31	00-0E-84-C2-C2-A0	-80

TABLE 8.7: WiFi data for a single access point at a specific location for five minutes from experiment E4.

After obtaining the three template shown on Figure 8.4 it is possible to decide if the test subject is at one of the template locations. On Figure 8.5, a diagram is drawn based on WiFi information collected during the four test cases of experiment E2 for a single test subject. The figure shows WiFi access points nearby and their SNR.

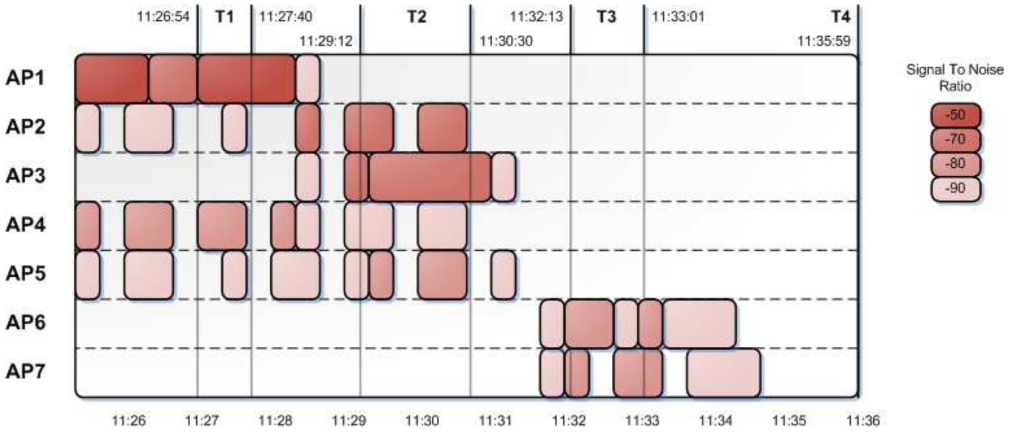


FIGURE 8.5: WiFi access points nearby and their SNR.

It is clear, that the templates match the users location at the three first test cases, which are all performed at the locations that were sampled in E4. The data captured for the remaining three subjects were also inspected and compared to the SNR values of the templates shown on Figure 8.4, yielding similar results.

The accuracy of using template matching is not particularly high, mainly because the granularity of the SNR values (steps of 10) is quite low due to the API provided by OpenNETCF [ope, 2007]. Also it is unknown if more than one location could match the templates created.

The WiFi based position did not seem to have any influence on the power consumption which was shown in Table 8.6 below RQ4.

Partial conclusion

When comparing the data captured by the RECON framework with the video recordings it is easier to identify the geographical position by the use of the video recordings, than with the template matching. However it is believed that in a large scale evaluation, the use of RECON would improve the efficiency, as video annotation typically takes a long time compared with the time spent on generating graphs as the one shown on Figure 8.5 and location specific templates.

The position accuracy is in the video is local, meaning that it is only possible to determine a position compared to the room the test subject is located in. This is the same for the RECON framework, as it is only possible to determine the position compared to the access points. The accuracy of the RECON framework is not very high either, as the template matching is unstable, as was also seen from the data in table 8.7.

One way of improving the positioning in the RECON framework would be to automate the template generation and matching. The ability to query the user about his current location could also be used to make the template matching better.

If the template generation could be automated the conclusion would be slightly different meaning that it is an advantage using RECON over video recordings. Finally another positioning method such as GSM or GPS would likely yield better results.

8.6 RQ6 - Efficiency

In Section 7.1 the research question RQ6 asked about the usefulness of the device. The question was made more specific by the two sub-questions relating to the source code and the time efficiency of the framework.

Using the RECON framework does not require access to the source code, unless usage and attitude information needs to be captured. The RECON framework can be run on the device without any other applications running, and only gather context information and report it to the server.

To infer use patterns and to query the test subject at specific events, the source code must be available, and the RECON library integrated in the application. The framework will require one line of code for each hook that needs to be inserted, however in some cases can require more than a single line, in order to catch events that are not usually used in the application i.e. focus in a specific control.

The time efficiency of using the RECON framework compared to a traditional field evaluation is impossible to determine by the results of experiment E2 as they were carried out simultaneously. Instead the time taken to prepare experiment E2, that is related directly to a specific evaluation method will be listed for that method, and any task that was relevant for both methods will count for both. The measurements is rounded and divided in the following test phases: Code preparation, test case design, briefing, execution and debriefing. Afterward the measurements is organized in a table (Table 8.8), in the table it is taken into account if it is possible to brief more than one test subject at a time or perform several tests simultaneously by multiplying the time with either $\frac{n}{d}$ or n (where n is the number of test subjects and d the number of available mobile devices).

In order to determine, if and when the framework is more time efficient, a comparison to traditional field evaluation for different amounts of test subjects is shown in Table 8.9.

The performed experiment E2 is not expressive for all evaluations, but based on the above it is believed that the RECON framework is a faster alternative to a traditional field evaluation if the number of test

	RECON [minutes]	**Traditional [minutes]**
Code preparation	80	0
Test case design	0	60
Briefing	$10 \cdot \frac{n}{d}$	$10n$
Execution	$5 \cdot \frac{n}{d}$	$5 \cdot 2n$
Debriefing	$10 \cdot \frac{n}{d}$	$10n$
Total	$80 + \frac{25n}{d}, \text{for} n \geq d$	$60 + 30n$

TABLE 8.8: Schematic representation of the time spent on testing, with n indicating the number of test subjects and d the number of mobile devices $(n \geq d)$.

n	**RECON [minutes]**	**Traditional [minutes]**	**RECON fastest**	**Traditional fastest**
1	$80 + \frac{25}{d}$	90		$d \geq 1$
2	$80 + \frac{50}{d}$	120	$d > 1$	$d = 1$
10	$80 + \frac{250}{d}$	360	$d \geq 1$	
100	$80 + \frac{2500}{d}$	3060	$d \geq 1$	

TABLE 8.9: Schematic representation of the time spent on testing, with different numbers of test subjects (n). d indicates the number of available mobile devices $(n \geq d)$.

subjects is higher than one and there is more than one mobile device available.

Partial conclusion

The RECON framework requires access to the source code because hooks needs to be inserted to use the survey functionality of the framework. Despite this, only a single line of code is needed for each hook. The RECON framework becomes more time efficient than traditional field evaluation when the amount of test subjects increase beyond one. The amount of time spent in preparation of an evaluation using the RECON framework varies depending on the complexity of the navigation in the application and the amount of FSM's that needs to be created.

Despite this, the results indicate that the RECON framework is more time efficient than traditional field evaluation.

Part IV

Evaluation

This part will conclude the study with a discussion and a conclusion based on the research questions, the working hypothesis and the results of the experiments.

CHAPTER 9

Discussion

The purpose of this chapter is to further discuss the results of the experiments and the methods used to obtain them. As mentioned in the problem statement (Section 1.1) the projects main focus was on the data capturing which is why the results and the discussion also primarily focused on the data capturing. In Section 9.1 the main results of the study will be emphasized and reflected upon. In Section 9.2 the weak points and strong points of the methods chosen for implementation in the RECON framework will be discussed.

9.1 Experiments and results

Although it is not possible to either accept or reject the working hypothesis, the results obtained through the research questions (RQ1 to RQ6 from Section 7.2) indicate that it indeed is possible to automate field evaluations.

One significant discovery, was that a critical and serious usability issue could be explained by using the context information collected by the RECON framework. This indicates the importance of capturing context information on mobile devices when evaluating application usability using an automated framework. Despite this, some detected usability issues were still unexplainable, but could possibly be explained by inserting surveys at these events. With the ability to update the configuration and statemachine on the device at any time, such an addition could be easily done. This was not done in this study, as the data gathered was analyzed after all test subjects had performed the evaluation. In addition, a more extensive amount of context information, such as audio and video, might improve the results further, as HCI researchers would have an even better basis for explaining an issue.

In the user test (See Experiment E2 in Section 7.2) only a single, simple application was used. 5 unique usability issues were discovered, however it is likely that one or several more complex applications would show more issues, thereby making way for a better comparison between the traditional field evaluation and automated field evaluation.

It was not possible to detect a link between use patterns and a specific context, however it was possible to capture the actual use patterns and at the same time the context. Had the user evaluation lasted for several days instead of half an hour, a connection might be discovered. A longitudinal evaluation could also show a change in use patterns over time, which will happen, according to the results of the questionnaire, where all the test subjects either agreed or strongly agreed to changing their use of an application over time (See Appendix A.4).

Another compelling result, was that it was possible to ”sample” the test subjects attitude, by presenting them with a small survey. This has also been done earlier by [Hilbert and Redmiles, 1998b], although using a different method and platform than the one applied in the RECON framework. A possible drawback of this, is the increased intrusiveness and the fact that the test subjects will be reminded that he is participating in an evaluation. This effect is not investigated in this study, but is important to take note of nonetheless.

[Kjeldskov et al., 2004] states that the expenditure of time in a traditional field evaluation is considerably high which the results of this study agrees with and in addition has poor scalability with regard to number of test subjects and length of the study. However the results of this study also show that an automation could make field evaluations more time efficient with as little as two test subjects, and significantly more when the amount of test subjects increases. The time consumption results are based on theoretical assumptions, so in order to verify them, a comparison of the traditional field evaluation versus an evaluation with the RECON framework would be necessary. This could be done by evaluating the same application, in the same context, with both methods, and afterwards let HCI researchers analyze the gathered data and compare the results. Such an evaluation would be able to compare the efficiency and effectiveness of each evaluation method, and give a more precise answer to the working hypothesis.

The power consumption of the RECON framework with different configurations was very unstable, which could be caused by the fact that the experiment was performed in an uncontrolled environment. A result of the experiment was that changes in Bluetooth inquiry interval had the greatest impact on the uptime of the mobile device. Most people tend to turn off their Bluetooth, which was also shown in the questionnaire (See Appendix A.4), so the benefit of querying Bluetooth is questionable and needs further investigation.

9.2 Methods

One of the main goals of the RECON framework was to capture context information. The software platform chosen in this study was Microsoft Windows Mobile 5.0 or later based primarily on the availability of the State & Notification broker, which allowed for easy access to a range of context information. The choice of software platform unfortunately limits the number of available hardware platforms. Choosing this platform meant that access to some features such as WiFi and Bluetooth would have to be done through thirdparty API's or by invoking native API's. This resulted in some context information not being captured, such as audio and video. The accuracy of some of the information captured, such as the location information, which is based on WiFi templates, is questionable. However no other hardware for estimating a location with a higher better accuracy, such as GPS, was available on the device, and technologies such as GSM could not be accessed. These were considered the only solutions other than WiFi triangulation, which is a study in itself. Also using Bluetooth to detect nearby objects or in particular people, proved to be troublesome, as many people tend to turn off their Bluetooth to save power. This assumption is supported the fact that all of the test subjects in the user test usually leave their Bluetooth off (See Appendix A.4). However the ability have proven to be useful in other experiments such as the one described in [Eagle and Pentland, 2006], where all test subjects are working in the same building, and can be requested to leave their Bluetooth on.

A traditional field evaluation is platform and application independent, as long as it is performed without requiring any software present on the device. The RECON framework is not completely hard- and/or software independent, as it requires Microsoft Windows Mobile 5.0 or later. The experiment E1 and E2 performed in Section 7.2 showed that the RECON framework worked as expected, even in an application not developed specifically for the framework. This means that the RECON framework can be used in any application, as long as it is deployed on a mobile device with the required software as described in Appendix B, and if needed, WiFi and Bluetooth. Since the RECON framework is application independent, it would be interesting to perform an experiment where independent HCI researchers were asked to do an evaluation by the use of the RECON framework, thereby evaluating its efficiency. By efficiency is meant the amount of extra code that needs to be added to the evaluated application and time spent preparing and performin an evaluation with the RECON framework.

In order to capture test subjects attitude, it was chosen to present them with a survey on the mobile device at the occurrence of events. To decide when to ask the subjects, a list of FSM's were used to model the sequence of events, which could trigger a survey. The choice to use FSM compared to other methods, such as grammar used in [Hilbert and Redmiles, 1998a], is based on the idea that a FSM relates well to the navigation in most applications and would be easy to comprehend for HCI researchers, who does not necessarily have a background in software development. A straight forward way to represent the navigation in this application, is to consider each screen as a state. A transition in the application would then result in a transition in the FSM's in the RECON framework. Each of these transitions can be inserted in the application by a developer with a single line of code, making even large FSM's trivial to implement. However in order to know where to insert the hooks, would require technical insight of programming. The hooking approach was chosen in preference to windows messages and AOP, as these required more effort from the developer. This allows a HCI researcher without programming experience, to create a FSM for his application, and then let a developer insert the required hooks. In order to verify this benefit, an usability evaluation of the RECON framework would be interesting, focusing on areas such as the usability of the hooking approach.

As mentioned in Chapter 2, several other frameworks exist for usability evaluation, however all of them cover one area really well but seem to neglect others. In Table 9.1, each of these frameworks are compared to the RECON framework.

Type of information	RECON	EDEM	WebQuilt	ContextPhone
Usage	✓	✓	✓	✗
Context	✓	✗	✗	✓
Attitude	✓	✓	✗	✗

TABLE 9.1: The type of information that the RECON framework can capture, compared to the frameworks analyzed in Chapter 2.

EDEM aswell as WebQuilt is well-suited for capturing usage and information, however in WebQuilt's case only when the application evaluated is based on a client-server architecture. In the case of EDEM it is also possible to capture attitude information. Common to both frameworks, is the fact that neither EDEM or WebQuilt captures context information and is not developed specifically for evaluation of mobile systems. Contrary to EDEM, ContextPhone captures context information, but can not capture attitude information and only limited usage information. These frameworks perform better in their specific area compared to the RECON framework, however the RECON framework covers all areas by being able to capture usage, attitude and context information resulting in a combination, that can provide results that are not attainable by the use of the existing frameworks, such as being able to explain usability issues.

Chapter 10

Conclusion

It is commonly agreed that traditional field evaluation is too difficult, time-consuming and it is questionable whether it adds any value. To counter these drawbacks of field evaluation a new framework was proposed for performing automated field evaluation with the goal of requiring less effort while still yielding the same or better results.

This aim resulted in the working hypothesis of the study, as stated below:

> *It is possible to produce the same or more elaborate results in an automated field evaluation with the same or less effort as in a traditional field evaluation.*

A traditional field evaluation can be carried out in many ways, and a single experiment could not answer the working hypothesis. Instead a list of research questions was created in order to give an indication as to what the answer of the working hypothesis could be.

This study contains both an engineering and a scientific angle. The purpose of the scientific angle, was to answer the research questions derived from the aforementioned working hypothesis, whereas the purpose of the engineering angle was to develop a framework for automated field evaluation.

The development of the RECON framework was done by analyzing state-of-the-art frameworks for usability evaluation. Analysis of methods for automatically capturing usage, context and attitude information on mobile devices was done, in order to uncover candidates for implementation in the RECON framework. The resulting RECON framework is a scalable and flexible framework which is application independent and can be used on various mobile devices as long the required software platform is available. Furthermore the framework allows HCI researchers to access data gathered, while an evaluation is being executed and make adjustments to the settings in framework, such as inquiry interval on the network devices.

The RECON framework was used in a range of experiments to capture information with the purpose of elucidating the research questions, that are based on the working hypothesis. The experiments involved capturing usage information by inserting hooks in an application. Also capturing context by communicating with the State & Notification broker in the Microsoft .NET Compact Framework 2.0 as well as WiFi and Bluetooth API's. Finally attitude information was captured by presenting the test subject with a survey at points in the application based on FSM's symbolizing a subjects navigation in the application.

Based on the results of the study, it can be concluded that it is indeed possible to develop a framework for automated usability evaluation. The results indicate that an automated field evaluation framework can lower the time spent on testing, however the framework developed in this study can not match the effectiveness of a traditional field evaluation, as more varied data can be gathered with a video camera. As such the framework will not be a replacement for traditional field evaluation, but instead be a supplement or a method for more specialized investigations that can be used when certain requirements needs to be satisfied e.g. longitudinal evaluations or evaluations in environments that does not allow traditional field evaluations. Examples of environments which would not allow a traditional field evaluation could be cramped or dangerous places e.g. at a roof which is the typical workplace of chimneysweepers.

Although the working hypothesis can not be answered, the discussion shows that there are many interesting challenges and considerations that needs to be done, in order to develop a complete and thouroughly tested framework for automated usability evaluation, and this study have given an indication as to which studies to perform, and in some cases what the outcome could possibly be.

10.1 Future work and perspectives

As was concluded, the RECON framework is not a replacement for traditional field evaluation in its current state. As the time limitation of the project only allowed a set amount of time for each subject in order to cover all interesting aspects of the development and evaluation of such a framework, some areas of this study could be further investigated, the specific areas is divided in two categories (namely functionality and experiments) and described below. Further work in these specific areas could make the framework a more valid alternative.

10.1.1 Functionality

The context information gathering of the RECON framework could have been improved in several areas. One of the most obvious is the location capture, which could have been done by using either GPS, GSM cell information or WiFi triangulation. Another potential improvement in regard to context information is adding support for capturing audio and video from the mobile devices.

On the administrative side, an easier way to generate FSM's and configuration-files e.g. via a website would improve the usability of the actual framework. For the analysis of the captured data, some kind of visualization or pre-analysis would improve the use of the framework by making it easier to use for HCI researcher, and could make it easier to explain usability issues in large amounts of data.

10.1.2 Experiments

The evaluations done in this project were limited to proof-of-concept evaluations and a more thorough field evaluation using both video and the RECON framework. These evaluations were assessed to be enough to obtain results to shed light on the working hypothesis of the study, however further evaluations such as an actual longitudinal evaluation combined with efficiency evaluation of the RECON framework could prove that the framework is even more benficial. To perform such evaluations a context aware application and two independent usability evaluation teams would be necessary. The test could be performed as illustrated on Figure 10.1. A test application is developed and the two independent HCI research teams evaluates the application, while timing each task they perform. After the two usability evaluations are performed the results are grouped in accordance with Molich's Taxonomy [Molich, 2007]. This experiment would produce qualitative measurements, comparable to those used in previous research such as [Kjeldskov et al., 2005] and [Kjeldskov et al., 2004]. The time measurements will show exactly how long time each HCI research team spent on testing and further give an indication of the efficiency of the RECON framework.

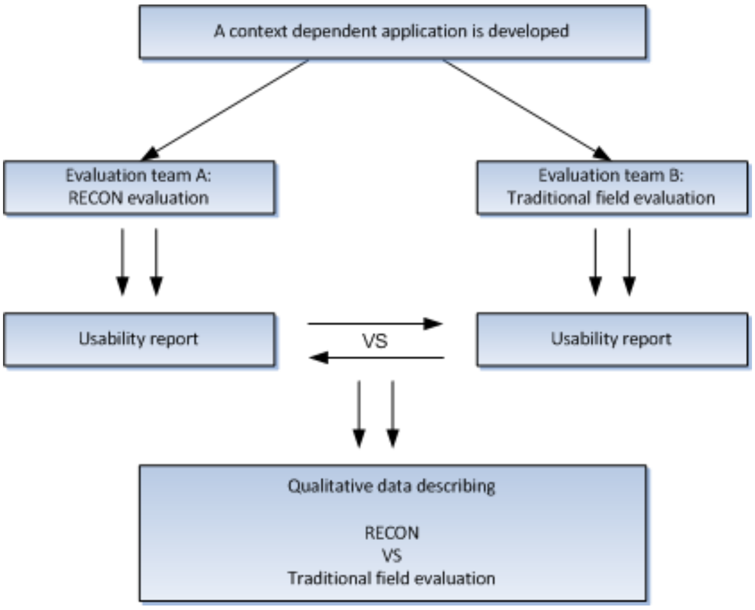

A context dependent application is developed
Evaluation team A:
RECON evaluation
Evaluation team B:
Traditional field evaluation
Usability report
VS
Usability report
Qualitative data describing
RECON
VS
Traditional field evaluation

FIGURE 10.1: Graphical representation of the desired usability evaluation with two evaluation teams.

Bibliography

[ope, 2007] (2007). Opennetcf smart device framework 2.0. http://www.opennetcf.org.

[Association, 2007] Association, G. (2007). Gsm european poster. http://www.coveragemaps.com.

[Castillo et al., 1997] Castillo, J. C., Hartson, H. R., and Hix, D. (1997). Remote usability evaluation at a glance.

[Chen and Kotz, 2000] Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. Technical report, Dartmouth College.

[Cheverst et al., 2000] Cheverst, K., Davies, N., Mitchell, K., Friday, A., and Efstratiou, C. (2000). Developing a context-aware electronic tourist guide: some issues and experiences. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–24, New York, NY, USA. ACM Press.

[Cormen et al., 2001] Cormen, T. H., Rivest, R. L., Stein, C., and Leiserson, C. E. (2001). *Introduction to Algorithms*. MIT Press. ISBN: 0262032937.

[Dey and Abowd, 2000] Dey, A. K. and Abowd, G. D. (2000). Towards a better understanding of context and context-awareness. Technical report, Georgia Institute of Technology.

[Eagle and Pentland, 2006] Eagle, N. and Pentland, A. (2006). Reality mining: sensing complex social systems. *Personal Ubiquitious Computing*, 10:255–268.

[Gorlenko and Merrick, 2003] Gorlenko, L. and Merrick, R. (2003). No wires attached: Usability challenges in the connected mobile world. *IBM Systems Journal*, 42(4):639–651.

[Hand, 2007] Hand, I. T. (2007). 32feet.net. http://InTheHand.com.

[Hartson et al., 1996] Hartson, H. R., Castillo, J. C., Kelso, J., and Neale, W. C. (1996). Remote evaluation: The network as an extension of the usability laboratory. In *Conference on Human Factors in Computing Systems*, pages 228–235. ACM Press. ISBN: 0-89791-777-4.

[Helfrich and Landay, 1999] Helfrich, B. and Landay, J. (1999). Quip: Quantitative user interface profiling. http://citeseer.ist.psu.edu/helfrich99quip.html.

[Hilbert and Redmiles, 1998a] Hilbert, D. M. and Redmiles, D. F. (1998a). Agents for collecting application usage data over the internet. pages 149–156. University of California, ACM Press. ISBN: 0-89791-983-1.

[Hilbert and Redmiles, 1998b] Hilbert, D. M. and Redmiles, D. F. (1998b). Why let perfectly good usability data go to waste.

[Hong et al., 2001] Hong, J. I., Heer, J., Waterson, S., and Landay, J. A. (2001). Webquilt: A proxy-based approach to remote web usability testing. *ACM Transactions on Information Systems*, 19(3):263–285. ISSN:1046-8188.

[Ivory and Hearst, 2001] Ivory, M. Y. and Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Compuing Surveys*, 33(4):470–516.

[Kjeldskov and Graham, 2003] Kjeldskov, J. and Graham, C. (2003). A review of mobile hci research methods. In *Proceedings of the 5th International Mobile HCI 2003 conference*, pages 317–335. Springer-Verlag.

[Kjeldskov et al., 2005] Kjeldskov, J., Graham, C., Pedell, S., Vetere, F., Howard, S., Balbo, S., and Davies, J. (2005). Evaluating the usability of a mobile guide: the influence of location, participants and resources. *Behaviour & Information Technology*, 24(1):51–65.

[Kjeldskov et al., 2004] Kjeldskov, J., Skov, M. B., Als, B. S., and Høegh, R. T. (2004). Is it worth the hassle? exploring the added value of evaluating the usability of context-aware mobile systems in the field. In *Proceedings of the 6th International Mobile HCI 2004 conference*, pages 61–73. Springer-Verlag.

[Laasonen et al., 2004] Laasonen, K., Raento, M., and Toivonen, H. (2004). Adaptive on-device location recognition.

[Linsky, 1995] Linsky, J. (1995). Bluetooth and power consumption: issues and answers. *RF Wireless Connectivity*, pages 74–95.

[Merriam-Webster, 2007] Merriam-Webster (2007). Merriam-webster online dictionary.

[Meyer, 2006] Meyer, D. (2006). Smart device sales soar. Website. http://news.zdnet.co.uk/communications/0,1000000085,39283931,00.htm.

[Molich, 2007] Molich, R. (2007). *Usable Web Design*. Nyt Teknisk Forlag. 978-87-571-2526-9.

[Nielsen, 1993] Nielsen, J. (1993). *Usability Engineering*. Academic Press. 0-12-518406-9.

[Raento et al., 2005] Raento, M., Oulasvirta, A., Petit, R., and Toivonen, H. (2005). Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 04(2):51–59.

[Rubin, 1994] Rubin, J. (1994). *Handbook of Usability Testing*. John Wiley & Sons. 0-471-59403-2.

[Sanderson and Fisher, 1994] Sanderson, P. M. and Fisher, C. (1994). Exploratory sequential data analysis: Foundations. *Human-Computer Interaction*, 9(3-4):251–317.

[Tart and Moldovan, 2006] Tart, A. M. and Moldovan, G. S. (2006). Automatic usability evaluation using aop. *Automation, Quality and Testing, Robotics, 2006 IEEE International Conference*, 2(4):84–89. ISBN: 1-4244-0361-8.

[Thompson et al., 2004] Thompson, K. E., Rozanski, E. P., and Haake, A. R. (2004). Here, there, anywhere: remote usability testing that works. In *CITC5 '04: Proceedings of the 5th conference on Information technology education*, pages 132–137, New York, NY, USA. ACM Press.

[Waterson et al., 2002] Waterson, S., Landay, J. A., and Matthews, T. (2002). In the lab and out in the wild: Remote web usability testing for mobile devices. pages 796–797. U.C. Berkeley and Seattle University, ACM Press. ISBN: 1-58113-454-1.

Part V

Appendix

In this part additional documentation that is not necessary for the understanding of the study is included. Also available in this part are the documents used in the experiments involving test subjects.

APPENDIX A

Test Documents

A.1 Test briefing

The purpose of this test is to evaluate the RECON Framework compared to traditional field test. A small application named ”Time And Notes” will be used in this test. You will be given an introduction to this software after which three test cases will be performed at different locations at Aalborg University. You are expected to use the phone as if it is your own, meaning that should the phone ring, you should answer it. Also for all the test cases, you are supposed to use the ”Time And Notes” application.

Your actions during the test will be recorded on video, monitored by a test monitor and logged by the RECON framework. After the test, a debriefing will be performed, where you will be asked to fill out a questionnaire and a short openended interview will be done, after which you will be able to enjoy a small refreshment.

Thank you for your participation.

- Tais & Christian, Group 1070

Test case 1

You are sitting in your group room and are about to test a piece of software you wrote. It is supposed to complete a task within 10 seconds.

Test case 2

Your eating your breakfast in the cantina. Your friend asked you yesterday to find out what the menu for today is. Your friend will join you at the University in a couple of hours, so in order to remember the menu, you better write it down somewhere.

Test case 3

You and your friend are waiting for the bus at the busstop. You plan to meet tonight at the pub Polly's at 20.00 for a beer. In order to not forget the appointment, you decide to use your phone to remind you.

Test case 4

During one of the previous three scenarios you received a warning from the phones builtin calendar about a meeting you are to attend. This meeting is very important to you, so you do not wish to be disturbed while attending it.

A.2 Test debriefing

The purpose of this questionnaire is to gain further insigt into the experience of using the RECON framework and the evaluation in general.

Please answer the questionnaire by selecting one of the five boxes symbolising statements spanning from ”Strongly agree” to ”Strongly disagree”. When possible feel free to elaborate on why you agree or disagree with the statement in the questionnaire.

Name:

Questions	**Strongly agree**	**Agree**	**Neither Agree nor Disagree**	**Disagree**	**Strongly disagree**	**Comments**
1. The surveys on the phone were annoying/disturbing	□	□	□	□	□	
2. The camera man and/or test monitor was a distraction/disturbance	□	□	□	□	□	
3. Performing the test cases felt natural	□	□	□	□	□	
4. The test felt natural despite the presence of the test monitor and camera man	□	□	□	□	□	
5. The way I use an application can change over time, i.e. when I learn the application	□	□	□	□	□	

Name:

Questions	**Strongly agree**	**Agree**	**Neither Agree nor Disagree**	**Disagree**	**Strongly disagree**	**Comments**
6. I would be concerned about privacy issues, if I were to participate in an automatic evaluation logging context information such as location, incoming calls, SMS etc.	□	□	□	□	□	
7. I act differently when I am aware of being monitored	□	□	□	□	□	
8. I would prefer an automatic evaluation without test monitor and cameraman compared to a traditional field evaluation with these	□	□	□	□	□	
9. I usually have bluetooth enabled on my mobile device/phone (Please elaborate in the Comments field)	□	□	□	□	□	

A.3 Declaration of consent

I, ______________________________ , have heard and understood the information regarding this study. All questions regarding the study have been answered to my satisfaction.

I agree to participate in this evaluation and understand that I can withdraw from the study at any time, should I wish to do so.

I give consent to the recording of the evaluation, in order to maintain raw data, provided that it is not made available to the public.

I give consent to the publication of the evaluation data, on the condition that anonymity is maintained, and I am not recognizable.

Name of participant: ______________________________

Date: __________ Signature: ______________________________

Name of test monitor: ______________________________

Date: __________ Signature: ______________________________

A.4 Questionnaire results

The results gathered from the evaluation of the framework. Four testpersons participated in the evaluation and filled out the questionnaire.

Questions	**Strongly agree**	**Agree**	**Neither Agree nor Disagree**	**Disagree**	**Strongly disagree**
1. The surveys on the phone were annoying/disturbing	-	-	-	4	-
2. The camera man and/or test monitor was a distraction/disturbance	1	1	-	1	1
3. Performing the test cases felt natural	-	-	2	2	-
4. The test felt natural despite the presence of the test monitor and camera man	-	1	1	2	-
5. The way I use an application can change over time, i.e. when I learn the application	2	2	-	-	-
6. I would be concerned about privacy issues, if I were to participate in an automatic evaluation logging context information such as location, incoming calls, SMS etc.	-	2	1	1	-
7. I act differently when I am aware of being monitored	1	3	-	-	-
8. I would prefer an automatic evaluation without test monitor and cameraman compared to a traditional field evaluation with these	3	1	-	-	-
9. I usually have bluetooth enabled on my mobile device/phone (Please elaborate in the Comments field)	-	-	-	1	3

The following are the comments from the test subjects. When more than one subject wrote the same comment, it is noted in brackets after the comment.

Question 1: Only one survey(2).

Question 6: If the test subjects would be using their own phone they agree(2). If the test personnel are serious and trustworthy.

Question 9: Because of the power consumption(4).

Appendix B

How to use RECON

The RECON framework is a framework designed for automated field evaluation of mobile systems. The framework can capture usage information based on hooks in an application, context information such as nearby wifi accesspoints or bluetooth devices and can query the user at specific events.

If all software and hardware requirements of the RECON framework is met, the following steps needs to be taken in order to perform an evaluation.

- Setup the Webservice and database on the server
- Insert hooks in the application and recompile it
- If surveys are needed, generate a state machine and survey for it
- Start the evaluation

Software and Hardware Requirements

The primary requirement, is a mobile device running Windows Mobile 5.0 or newer. In order to gain full benefit of the RECON framework the following additional requirements need to be met. The hardware requirements should soft, meaning that if they are not available, the framework should still capture information from the other modules of the framework, however this has not been tested thoroughly. The software requirements are hard requirements, and the indivindual libraries or packages needs to be installed in order to use the framework.

Hardware:

- Bluetooth Enabled
- Wifi Enabled

Software:

- .Net Compact Framework 2.0
- Microsoft SQL Server Mobile 3.0
- OpenNETCF's Smart Device Framework 2.0
- 32Feet.Net 2.1

Server Setup

The server setup is in most cases limited to creating a user account for each test subject and adding the test subjects to a predefined group. The creation of user accounts and adding the accounts to one group is done in the database named ”recon” (shown on Figure B.1). The database is accessible by the tool named phpMyAdmin (`http://kom.aau.dk/group/07gr1070/phpmyadmin`) with the username ”root” and password ”creative”.

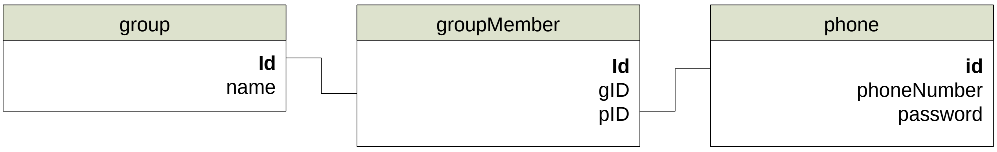


FIGURE B.1: Database diagram showing the tables involved when creating user accounts and groups.

User accounts is created by adding a row in table ”phone” after a new account is added the user must be added to a group which is done by adding a row in table ”groupMember”. If it is necessary to add a new group is this done by adding a row in table ”group”.

If a special configuration, of the mobile devices, is desired it is also possible to configure. The configuration of the mobile devices is done in the xml-file (`c:\temp\config.xml`) located on the recon-sever (recon.kom.auc.dk).

Mobile Setup

To evaluate an application access to the source code is required. This is because the RECON framework library needs to be linked in the actual application and hooks inserted wherever an event needs to be logged.

When the RECON library is included in the project, the library needs to be initialized when the application is started. This is done in the Main method, the following way:

Recon.Init();

When the library is initialized, an event can be reported to the framework at any point in the execution of the code by inserting the following line of code:

Recon.ReportEvent(1);

The ReportEvent method takes an integer as argument, identifying a specific event(this integer is also saved in the database). Additionally the method has an overload that takes two argument, the first is the event-identifying integer and the second a string that can hold additionally data. An example is shown below:

Recon.ReportEvent(2, "Time is now: " + DateTime.Now.ToString());

Whenever all hooks are inserted, the application can be recompiled and deployed to the mobile device. In order to use the hooks in the application to trigger a survey of the user, a state machine needs to be generated, which is described in the following section.

State Machine Generation

The configuration of a state machine with survey functionality is normally done in two steps. In the first step the state machine is configured and in the second is the survey created.

Step 1: State Machine Configuration

Setting up a new state machine is done by adding a row in table ”task” describing the task which the state machines models e.g. ”Printing a document”. After a task is defined all the states in the task is created by inserting one row for each state in the table ”states”. The last step in creating a state machine is defining the transitions between the defined states which is done by adding rows in table ” transition”.

The database tables involved in the creation of new state machines is shown on Figure B.2.

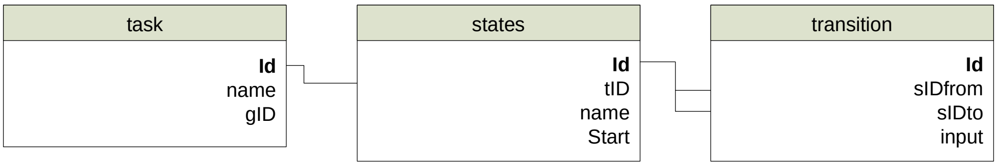


FIGURE B.2: Database diagram showing the tables involved when creating a state machine.

Step 2: Survey Configuration

When the desired state machines it created it is possible to crate one or more surveys for each machine.

Setting up a new survey is done by inserting a row in table ”questionnaire” where the field ”sID” indicates the state in which the survey should be shown. Questions is added to a survey by adding rows in table ”questionnaire” and options is added to a question in table ”questionnaireOption”.

The database tables involved in the creation of surveys is shown on Figure B.3.

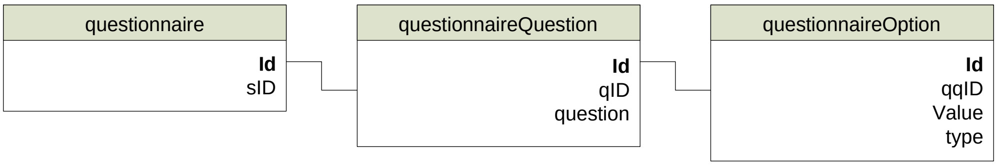


FIGURE B.3: Database diagram showing the tables involved when creating a survey.

Evaluating with the RECON framework

When the server is setup, a configuration file created, hooks inserted in the application and a state machine and survey configuration generated, the evaluation is ready to commence.

It is important to start the RECON application and the evaluated application in the right order. In order to not miss out on any of the events in the evaluated application, the RECON application must be started first. When the RECON application is up and running, it immediately starts capturing context information. At this point it is safe to starte the application that needs to be evaluated.

To verify that the RECON framework is working correctly, switch to the ReconLog window in the task list and make sure that both context information and usage information for the application is received properly.