University of Bergen
FACULTY OF SOCIAL SCIENCES
Department of Information Science and Media Studies

Master Thesis

Activity-Centric Computing and Context Awareness in Mobile Personal Information Management

Author:
Torstein Thune

Supervised by Victor Kaptelinin

31st May 2013

"Design creates stories, and stories create memorable experiences, and great experiences have this innate ability to change the way in which we view our world."

Christian Saylor

UNIVERSITY OF BERGEN
Master of Information Science

Abstract

Activity-Centric Computing and Context Awareness in Mobile Personal Information Management

by Torstein Thune

Information and tools on modern smartphones are tightly tied to applications. It is hard to move information between applications, and it is often near impossible to collect and compare information from several sources. In this thesis I will explore how we through applying activity-centric computing and context awareness to the current generation of smartphone operating systems can advance the state of mobile personal information management.

The thesis presents two prototypes, Activity Ant 1 and Activity Ant 2, showing how activity-centric computing and context awareness can be implemented from an Human-Computer Interaction perspective. User reactions and suggestions are also presented and discussed.

Acknowledgements

Several people have contributed to the work presented in this thesis, and deserve my thanks and gratitude.

Firstly, I would like to thank my supervisor Victor Kaptelinin for his guidance, supervision, ideas and inspiration. His great knowledge, experience and wealth of ideas were invaluable. His influence on this thesis cannot be understated.

I would also like to thank Per-Øivin Andersen, John Fredrik Asphaug, Bård Bachmann, Siv Berg Snorre Davøen, Eivind Elseth, Lisa Halvorsen, Nils Haukås, Kari Holdcroft, Ståle Nestås and Endre Oma for their invaluable feedback, discussions, ideas and inspiration.

I would like to thank my brother, Gunnar Andreas Thune, that kindly volunteered to read through and correct my grammar.

Lastly I would like to thank everyone else that I have discussed my work with. Their input, ideas and critisisms have helped guide the scope and direction of my work.

Contents

List of Figures

Abbreviations

ABC	**A**ctivity **B**ased **C**omputing
AC	**A**ctivity **C**entric
ACC	**A**ctivity **C**entric **C**omputing
CA	**C**ontext **A**ware
CAC	**C**ontext **A**ware **C**omputing
GUI	**G**raphical **U**ser **I**nterface
HCC	**H**uman-**C**omputer **C**ollaboration
HCI	**H**uman-**C**omputer **I**nteraction
PIC	**P**ersonal **I**nformation **C**ollection
PIM	**P**ersonal **I**nformation **M**anagement
UI	**U**ser **I**nterface
UX	**U**ser E**x**perience
WIMP	**W**indows **I**cons **M**ouse and **P**ointer

Chapter 1

Introduction

Today we live in an age of information. Much of this information lives in our pockets in the form of applications on smartphones. We can access social information through the Facebook and Twitter applications, we can access notes and to-do-lists through the Evernote application, and we can access email through a plethora of specialised applications. The information from these applications are often stored on web servers, and made available to most internet-connected devices. It is also often possible to merge information from several sources, for instance it is possible to import events from Facebook to Google Calendar using the desktop version of these services. On desktops it is also easy to copy and compare data from different services through opening several windows. On mobile devices we do not have this luxury. It is neigh on impossible to collect comparable data in one application, and it is cumbersome to copy or compare data from multiple applications. In essence our phones are full of walled information gardens, which has lead to a situation of almost complete information fragmentation.

Some efforts have been made to dissolve the walls. For instance Android tries to assemble information from multiple services into the context-sensitive Google Now. Samsung has another approach to the problem, and lets users open two applications at a time on their phones. These features help, but I will argue that they are far from enough to fix the state of personal information management on mobile devices. In this thesis we will explore how we through employing activity-centric computing and context awareness can create functions to let users create collections of information and tools.

1.1 Motivation

Work on this thesis is motivated by the observation that information and tools on a smartphone are very strongly tied to applications, and that it is a hassle to operate with several applications at once. For instance an operation as simple as copying a phone number or email address from a text message to an email client can be a labourous and frustrating process. Using several tools from different applications at the same time is also a hassle. For instance sending said phone number as a text message while talking to someone on the phone. The situation is bad enough that we often use physical tools, such as notepads, to act as intermediary information storage instead of using our smartphones directly. The main goal of this project is to suggest a first step on the road to remedying this situation.

1.2 Research Questions

1. **Are there problems or unfulfilled potential with current mobile personal information mangament systems?**
2. **Can activity centric computing be used in order to fulfil potentials or solve problems with current mobile PIM systems?**
3. **How can activity-centric computing be implemented in current mobile operating systems?**
4. **How can the concepts and potentials of activity-centric computing be conveyed to users?**

1.3 Chapter Overview

Chapter 2 presents the research perspective of this project. The chapter begins with presenting relevant theoretical frameworks that have helped the development and frame of thought of this thesis. Next it presents some relevant research on smartphone usage patterns. Thirdly a look at the current state of the art in the smartphone industry is presented. Finally a the chapter contains a brief discussion of problems and potentials.

Chapter 3 presents the design science research methodology that have been followed during the design process of the project. In addition a section on focus groups and a section on prototyping is included, due to the central position these techniques have had during my work.

Chapter 5, 6 and 7 presents the design process and the resulting prototypes. Chapter 5 looks at the design process and presents the CATA-framework, while chapter 6 and 7 presents the two prototypes that were created. Chapter 6 and 7 also presents test results.

Chapter 8 presents findings and discussion.

Chapter 9 is the conclusion of the thesis. The chapter contains a list of general results in addition to some reflections and suggestions for future research.

Chapter 2

Research Perspective

Research on smartphones can be conducted from several different perspectives, each with different motives and goals. For instance a computer scientist can explore algorithms to make smartphones faster, or a media scientist can explore how smartphones have impacted journalism. In this thesis smartphones will be explored from a human-computer interaction (HCI) perspective. I will present how personal information management is done on mobile platforms today. I will also explore how concepts from activity-centric computing and context-awareness can be practically applied to advance the state of mobile personal information management.

This chapter presents the research perspective of the thesis. Firstly the relevant theoretical frameworks that have helped inspire and delineate the research and design process are presented. Next, relevant research is presented. Finally, a section detailing the current state of affairs is presented.

2.1 Theory

This thesis is inextricably tied to several theoretical frameworks. These frameworks have served to refine both the problem space and the solution space through forming a frame of thought in which research was conducted, and possible solutions were developed and reviewed.

2.1.1 User Experience and Usability

The research and results presented within this thesis are firmly grounded within the disciplines of interaction design and HCI. When doing research, and implicitly design, on artefacts within human-computer interaction researchers are interested in some kind of measure of how the artefact is perceived. Two essential terms in that regard are user experience and usability.

User experience (UX) is a concept which encompasses product behaviour and product usage. Both a user's impression of how easy or hard a product is to use and how good or bad it looks like and feels to the touch are intrinsic properties of the UX. It is about "how people feel about the product and their pleasure and satisfaction when using it, looking at it, holding it and opening or closing it" (Sharp, Rogers & Preece, 2007). This intrinsically subjective quality of user experience does however imply that we can only lay the premisses for a wanted experience and not design the actual experience itself.

So, how can user experience be measured? User experience is notoriously vague and undefined. Law, Roto, Hassenzahl, Vermeeren and Kort (2009) talk about UX as "*something* desirable, though what exactly *something* means remains open and debatable". Even the definition given in ISO 9241-210 is vague at best when it defines UX as "a person's perceptions and responses that result from the use or anticipated use of a product, system or service". While disputed it is increasingly recognised that a certain threshold level of usability is necessary for providing a good user experience (Law, 2011). While possible to quantify other UX measures (e.g the level of quality feeling of an iPad on a scale from 1 to 10), it only goes as far as to describing the UX without necessarily revealing what the problem is or where it lies, let alone what potential solutions are. Thus, qualitative data is required to complement and supplement qualitative UX data (Müller, Law & Strohmeier, 2010).

Usability is another key concept within interaction design and HCI. While UX is a very large and vague term, usability is more concrete with a set of metrics which can be used to measure usability. Also, where UX measures can indicate whether there is something wrong, usability measures can often indicate what, or at least where, the cause of a problem is and thus potentially indicate a potential solution. The process of creating good usability is to ensure that the product in question is easy to learn and effective and enjoyable to use. More specifically usability specialists talk about a set of usability goals. Sharp et al. (2007) operates with the following usability goals:

Effectiveness How effective is the product at doing what it is supposed to do?

Efficiency How high level of productivity can a trained user achieve?

Safety How safe is the product to use both in helping the user avoiding actual dangerous situations and helping the user avoid unwanted actions?

Utility Does the artefact provide an appropriate and adequate set of functions that will enable users to carry out their tasks in they way that they want to do them?

Learnability How easy is the product to learn? Is it possible, and how hard is it, to work out functionality through exploring the GUI and performing actions?

Memorability How easy is it to remember how to do certain actions in the product? How does the interface help users remember how to use (especially infrequently used) actions?

In order to design good user experience and usability one often follows a set of design principles. Sharp et al. (2007) presents the following commonly used design principles:

Visibility Generally visible operating functions are easier to figure out and use compared to hidden ones.

Feedback The interface should show that the system has received and used a given input.

Constraints The product should try to avoid facilitating invalid user input and interactions for the given context.

Consistency Similar operations should look and behave similarly.

Affordance When applicable users should be able to use familiarisation from experience in order to naturally understand what an interface element does. For instance a button affords that it is clickable.

These have been used throughout the design process in order to help make informed decisions regarding interface elements. It must however be mentioned that the two prototypes proposed are too far removed from a finished product for UX and usability measures to be accurate for a real-world implementation. I have therefore chosen not to give UX and usability a central position in the discussion of the prototypes.

2.1.2 Personal Information Management

The main element of exploration in this thesis is personal information management (PIM) on smartphones. The field of PIM explores systems which look at or facilitate organisation of information. One of the leading voices in PIM research, William Jones, defines PIM as referring to "both the practice and the study of the activities a person performs in order to acquire or create, store, organize, maintain, retrieve, use, and distribute the information needed to complete tasks (work-related or not) and fulfil various roles and responsibilities (for example, as parent, employee, friend, or community member) (Jones, 2008)". In the context of this thesis PIM refers to how users can organise and access their information on a smartphone.

Within the field of PIM one often talks of a hierarchy of information collections. The outermost node of this hierarchy is the personal information space (PIS). The PIS is the set of all information a person has direct control over. Elements included in PISes are locally stored references to documents as well as applications, tools and constructs such as folders that support the acquisition, storage, retrieval and use of said information. However, information that a user does not have direct control over, like for instance a website administered by another party, is not a part of a person's PIS. On the other hand, a locally stored copy of the same website is a part of the person's PIS.

Underneath the PIS information is organised in personal information collections (PICs). PICs are essentially subsets or "islands" within an individual's PIS in which a conscious effort has been made to control the information that goes in and the way in which it is organised. (Jones, 2008). When talking about PICs in a digital context it is important to note that the members of a PIC typically share a technological format and are accessible through a particular application (Boardman, 2004). There have however been efforts to widen the scope of digital PICs to include tools and information spread over several applications. One such effort is activity-centric computing, which is presented in section 2.1.4.

In an ideal implementation of PIM, we would have the right information in the right place in the right form in sufficient quality and completeness to meet our current needs (Jones, 2008). Today we have a lot of tools and technologies which help us with our information organisational needs. For instance we have tools, such as Evernote and OneNote, which provide useful features for creating and organising notes. However, such tools have led to information fragmentation through them employing their own organisational schemes. While OneNote and Evernote keeps the same kind of information the way in which the two applications stores information is incompatible, therefore making making one application unable to access and use information from the other. Essentially, this phenomenon forces a user to keep many similar, but essentially incompatible organisational schemes for digital assets such as text documents, emails and references. In addition to this, the problem increases in complexity when a person has several devices, with differing organisational capabilities, such as laptops and smartphones.

Lately PIM research has become increasingly popular due to the problems introduced by the plethora of new applications and tools. Advances reached through this research may have large and varied forms of pay-off, both on a personal and a societal level. Jones (2008) names some of these. For instance better PIM may lead to a better quality of life through allowing us to make better use of our time, money, energy and attention. Within organisation advances to PIM can improve productivity and lead to better teamwork. On a societal level better PIM may translate into better support for the ageing population, which in turn increases the chances that our mental lifespans matches our physical lifespans.

2.1.3 Activity Theory

Activity Theory is one of the main branches of thinking within HCI, and is on of the main sources of inspiration for activity-centric computing. Emerging from early twentieth century Russian psychology, Activity Theory forms a broad theoretical framework for understanding humans through analysing the creation, structure and process of activities (Kaptelinin & Nardi, 2006). Though for many years contained to the Russian psychology tradition, activity theory was introduced to the rest of the world in the late 1970s and 1980s through the translation of Leontiev's "Activity, Consciousness and Personality" (1978), and Wertsch's collection of activity theory papers in 1981. Presently Activity theory is one of three dominant post-cognitivistic theories of human-computer interaction, the other two being Phenomenology and Distributed Cognition.

At the centre of activity theory is the notion of activities. Within activity theory activities are defined as "a purposeful interaction of the subject with the world, a process which through mutual transformations between the poles of subject-object are accomplished" (Leontiev (1978) as referenced in Kaptelinin and Nardi (2006, p. 31)). Activities lie in the interaction between the subject that wants to perform some action and the object that the subject want to perform that action with or on. This object, or objective, can be physical or social in nature. The subject on the other hand can only be a living agent. This principle of directedness is called object-orientedness.

$$Subject \quad \longleftrightarrow \quad Object$$

FIGURE 2.1: An activity (Kaptelinin & Nardi, 2006)

The base case of activity is one in which a subject interacts with an object. As seen in section 2.1.3 there is a reciprocal influence. On one hand the object is manipulated by the subject's actions. On the other hand the subject goes through a developmental process through interacting with the object. In the context of user interfaces this can be a learning process in which the subject learns how to manipulate a graphical user interface (GUI) in order to retrieve desired information.

An activity can be divided into activities, actions and operations. Actions are conscious goal-directed processes undertaken to fulfil the object (Kaptelinin & Nardi, 2006). Different actions can be performed in order to reach the same goal. Goals can be divided into a hierarchy of sub-goals. Where actions are defined as conscious operations are defined as automatic. For instance hitting the keys on a keyboard can be considered an operation for an experienced typist, while the thinking required to formulate the words can be considered actions. This can change for instance if a key is stuck and the typist is forced to look for a solution, at which point the operation would turn into an action due to consciousness being involved. Thus activity theory holds that the constituents of activity are not fixed (Kaptelinin & Nardi, 2006)

Another principle is the notion of internalisation and externalisation. Activity theory proposes the idea that the human mind is not separated from culture and society, and that it is related through internalisation and externalisation. Internalisation is the process through one can for instance automate operations through creating an internal representation of the action in question, for instance through learning the layout of a keyboard in such a way that one is able to touch-type. Externalisation is the opposite, transforming internal activities into external ones. [1]

While not explicitly used in this thesis, activity theory constitutes a frame of thought that implicitly informed the design process. Activity theory is also one of the main influences of activity-centric computing, which is of paramount importance to this thesis.

2.1.4 Activity-Centric Computing

Research shows that most people spend very little continuous time on single activities, and that most of us experience frequent interruptions to our work flow (González & Mark, 2004). This is likely an even larger problem on mobile devices that are designed to be used in the public sphere, where distractions are abundant. One problem with the current computing paradigms, document-centric computing and application-centric computing, is that they provide little support for aggregating PICs, thus not facilitating the frequent interruptions and multi-tasking realities of modern computer use.

Activity-centric computing (ACC), sometimes referred to as activity-based computing, is a computing paradigm in which human activities, as opposed to applications or documents, take the centre spot. In general one can say that the goal of ACC is to devise digital artefacts that facilitate human activities in everyday contexts. One definition of activities as proposed by Balakrishnan, Matthews and Moran (2010) is "a set of interrelated actions and events around a common goal, involving a particular group of people, set of resources, and time framework". In ACC one often separates between computational activities and human activities. Computational activities are in this case something that "...collects in a coherent set a range of services needed to support a user carrying out a certain human activity (J. E. Bardram, 2007)". In other words one can separate activities into

[1] Taken from my home exam, http://thunemedia.no/2012/11/home-exam-hci/361-home-exam/

two levels: the abstract human activity level and the computational level referring to the services/applications that are used in the human activity (Christensen & J. Bardram, 2002).

One of the earliest ACC systems was the ROOMS system, in which a digital workspace was divided into several virtual desktops, called "rooms", each with its own specific purpose (Card & Henderson, 1987). One can say that each room constituted a digital representation of a PIC. Concepts from ROOMS have later been implemented by several X window managers, such as the window managers used in the Gnome and KDE Linux desktop flavours.

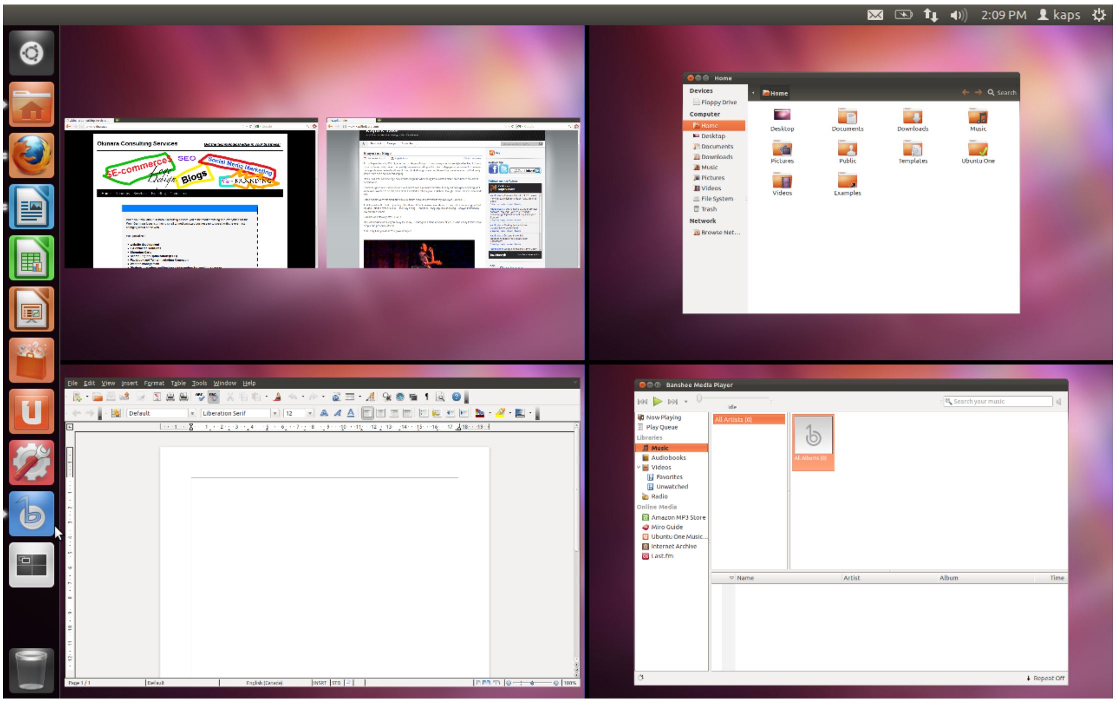

FIGURE 2.2: Unity, the default desktop environment on Ubuntu Linux supports virtual desktops.

There have been several attempts at implementing ACC through creating novel GUI solutions. One such system is GroupBar (Robertson et al., 2007)[104-114]. GroupBar was designed to work as a taskbar replacement in Windows XP. In this new taskbar users were able to create groups of applications. These groups could in turn be manipulated in a similar fashion to how normal windows are manipulated. That is, users could maximise and minimise groups. This let users create activities in the form of open and positioned application windows. When later tested on users it was found that the participants were generally positive. Some even wanted to continue to use the GroupBar in spite of it lacking features, such as the system tray, from the normal Windows XP taskbar. Other similar interface-based ACC implementations include Scalable Fabric (Robertson et al., 2007)[115-122] and Task Gallery (Robertson et al., 2007)[122-132].

ACC can also be implemented through changes at the information level. In the Haystack project Karger (2007) aimed at creating a system where there was no information segregation based on applications. In order to do this data is stored in a structured manner, for instance as RDF triples. This information is then displayed in specialised views designed to display and provide tools to edit an object. These views can in turn be mixed freely, through the user adding them to for instance his/her email inbox view.

J. E. Bardram (2007) has conducted much research on how ACC can be used to facilitate multi-user and distributed computing scenarios. Through conducting research on Danish hospitals and clinics, Bardram has devised a novel pervasive computing framework called activity-based computing (ABC). In this framework the activities of the user are considered the basic computational unit. In ABC computational activities can be initiated, suspended, stored and resumed on any computing device in any infrastructure at any time. The activities can also be shared or handed over to others. Another aspect of the activities in ABC is that they adapt to the usage contexts of the users, thus bringing in a level of context-awareness. ABC has also been implemented, with user interfaces for both stationary desktop PCs, tablet PCs and wall-based computers. All of this makes ABC well suited for situations with frequent interruptions and distributed computer terminals and differing information needs.

ACC has largely remained within the realm of academic research, and has not seen wide adoption in consumer devices. Research on ACC has also been largely limited to desktop operating systems, and not mobile operating systems. There is however one example of a consumer-oriented ACC system for tablets. KDE Plasma Active (KDE, 2012) is a system that was intended to run on Linux powered tablets, development has however seemingly stagnated. Plasma Active is briefly presented in section 2.3.5.

2.1.5 Context-Aware Computing

Context-Aware Computing is a computing paradigm in which applications can discover and take advantage of contextual information (Chen & Kotz, 2000). Examples which are readily available to modern smart phones are for instance the time of day, the location of the device, the direction the device is pointed in, nearby networks and user activity. This has been used for instance to provide turn-by-turn navigation in Google Maps. In addition to purely sensory information more complex contexts can be calculated through monitoring and comparing sensory output over time. For instance a system that has recorded data of its user moving in circles in a previously unvisited city would be able to infer that its user is potentially lost and might need help.

In the context of ACC and mobile computing context-awareness could be used in order to manipulate activities, or to act as another tool to help facilitate human activities in that it can remind or enforce routines.

Several research projects have been conducted within the field of activity-centric computing. Some early research projects are presented below. Context-awareness has also begun to appear in consumer devices. One notable example of this is Google Now, which is presented in section 2.3.2.1.

2.1.5.1 Cyberguide: A mobile context-aware tour guide

In the mid 1990s a group at the Georgia Institute of Technology researched context-awareness as a possibly intrinsic part of future computing environments. The result of this research was a series of prototypes for a mobile, hand-held tour guide system named Cyberguide. The ultimate goal was a system that ”knows where the tourist is, what she is looking at, can predict and answer questions she might pose, and provide the ability to interact with other people and the environment" (Abowd et al., 1997). They were however confined by technology to creating an application for hand held computers which only took into account the location of the user.

2.1.5.2 CybreMinder: A Context-Aware System for Supporting Reminders

Dey and Abowd (2000) describe a system that based on context, including time and place, determines when an user needs a reminder. The paper describes how the tool was used and how it was developed.

2.1.5.3 Web 2.0 integration in a context-aware mobile PIM application

Grønli and Ghinea (2009) shortly outlines the implementation of web 2.0 technologies in a mobile PIM system in order to automatically display data and news from the user's location. The system used GPD, Wi-Fi and communication mast triangulation in order to retrieve information about the user's location. The paper shows that similar systems have been implemented and have undergone some field testing, thus implying that these kinds of systems are at least feasible to implement.

2.2 Relevant Research

While many of my assumptions are based on my own personal relationship with smartphones it is important to look at actual data for how the general population interact with their phones, since it may give important clues as to which problems exist and what phones are not used for, which in turn might give clues as to what potentials there are. In order to somewhat alleviate the impact of cultural and social differences I will present data from three studies originating from different countries. It must however be noted that the data presented is limited and that they are not necessarily indicative of behaviour in non-western cultures.

2.2.1 Usage Patterns and Feature Requests in Norway

In her master thesis Øvrebø (n.d.) looked at what different groups of Norwegians were using their phones for, and what features and improvements they wanted.

According to the study, which admittedly was conducted on a limited sample, the average installed application count was 28,4. For males the average was 26,5 while for females the average was 30,9. This number was however coloured by a few significant statistical outliers. The median for the whole sample was 20 installed applications, while the numbers were 20 for males and 16 for females. From these numbers we can hypothesise that the average Norwegian smartphone user has approximately 20 installed applications.

One very interesting question in the survey was: "How often do you use your phone to organise your day?" To this one answered never, 13 answered sometimes and the vast majority, 47, answered every day.

Applications used for personal organisation: [2]

1. Calendar: 51
2. Alarms: 23 (includes both usage as an alarm clock and for reminders)
3. Mail: 21
4. Note taking: 17
5. Social media: 13
6. To-do lists: 4

It is clear that people use their phones for reminders. 51 use the calendar application, which in most cases will give you reminders. In addition many use alarm features. Communication is also important, with many using mail and social media applications in order to organise their days.

Feature requests

[2] Based on work in progress data.

1. Context sensitive information
2. Better notes
3. Better reminders
4. Payment solution (NFC)
5. Ticket handling
6. More possibilities to change how things look like on the phone
7. Time settings for the length of soundless mode
8. Do things faster and more reliably
9. Integrated note taking
10. To-do-list integrated in iCal
11. Notifiers/blinking/lights if you have unread messages/unanswered calls
12. Whistling function in order to find the phone

In general the survey seems to indicate that the average Norwegian smartphone user has a large amount of applications installed, but that they predominantly use a small subset of these applications with any regularity. The dominant uses within the sample are communication and reminders. While largely inconclusive, the set of features explicitly requested by the participants seems to indicate that there might be a potential for a larger degree of context-awareness, as shown by the requests for "Context sensitive information" and "Better reminders".

2.2.2 Usage Contexts and Usage Patterns

In another study Karikoski and Soikkeli (2011) tried to identify how the context of use affects the usage of smartphone communication services. The study was done through employing a computing algorithm on 140 Finnish smartphone users through 2009 and 2010.

The study identified five place-related usage contexts: (i) home, (ii) office, (iii) other meaningful (a place where a considerable amount of time is spent, but that is not home or office, e.g parents home), (iv) elsewhere (other than meaningful, for instance on the move, or infrequently visited places) and (v) abroad.

When analysing the contexts Karikoski and Soikkeli (2011) found that there was, as expected, patterns to the times at which a phone was in different contexts. There was a peak in the home context during early morning hours, and a dip during the early afternoon. The office context was active during normal working hours. The elsewhere context peaked during early evening, but did not show any clear peak during the morning commute time. The other meaningful context accounted for less than 10% of the time, and peaked during the late afternoon.

The study found that the average time spent interacting with the phone was 74 minutes per day. It also found that shares of interaction times per contexts were: home 53%, office 12%, other meaningful 8% and elsewhere 24%.

The study also found that usage patterns varied by context. The intensity of usage, measured on user level as interaction time in minutes per hour spent in context, was lowest in the home context (even when taking into account time spent sleeping), while the other contexts showed similar levels of usage intensity. A difference in application usage was also detected. Voice calls were longer in the home context than they were in the office context. Emails and SMS were also used more in the office context than in the home context.

The study clearly shows that there are some more or less clearly defined contexts that we go through during a day. While the contexts found in the study were not detailed they still prove the point that it is possible to find meaningful contexts for smartphone users, and that these contexts have differing usage patterns.

2.2.3 Usage Patterns

In a study performed by Falaki et al. (2010) on 255 smartphone users, it was shown that smartphone usage varies to a large degree. The researchers installed tracking software on 33 Android phones and 222 Windows Mobile phones. The software gathered data about which applications that were used. In addition to applications the software tracked interaction length, the number of interactions, time of day and data usage. An interaction in this context refers to a usage session. So, for instance checking the time would count as an interaction and having a phone conversation would count as another interaction. The users studied were computer science researchers and high school interns.

The study found that both the time spent on an interaction session and the number of interaction sessions during a day varied immensely between users. The average number of interaction for a user was for instance between 10 and 200 times per day, and the mean time spent on these sessions varied from 10 to 250 seconds. They also found that for any given user most interactions are short, but that some interactions are very lengthy.

Another finding was that application popularity varies for different times of day. Through analysing their logs they found that for instance one of the messaging applications for Windows Phone was more popular during the day than during the night. The researchers also identified eight categories of applications, and their relative popularity.

1. **Communication** (email, text messaging and instant messaging)
2. **Browsing** (e.g web browser, search and social media applications)
3. **Maps**
4. **Media** (pictures, music and videos)
5. **Productivity** (e.g calendars, alarms and applications for handling text-documents)
6. **System applications** (e.g settings and file explorer)
7. **Games**
8. **Other applications**

Of these categories communications was by a large margin the most used application category. Browsing was another category with significant use. Maps, media and games were found to have lower levels of use.

The study also showed that as much as 90% of interaction sessions were with only one application, implying that users tend to interact with their phone for single tasks, and that these tasks tend to involve the use of only one application.

The findings of the study are very interesting in that they show that while users use their phones a lot, most of the usage is quick interactions with single applications. This indicates that the most common usage scenario is quick tasks limited to the scope of one application. Examples of such tasks can for instance be to check the time or calendar appointments, or to read an email. The study did however also show that as much as 10% of interaction sessions involve the use of more than one application. Another finding was that some applications are vastly more used than others, with the general communications category of applications topping the usage list.

2.2.4 Summary

Users seems to have a large quantity of applications installed. However, as shown by Falaki et al. (2010), the level of usage varies vastly, with some applications being used exponentially more than others. Communication and browsing applications were found to have high levels of usage, while maps, media applications and games were found to have lower levels of usage.

Research also indicates that there are some common usage contexts, such as the five identified in Karikoski and Soikkeli (2011). Research also indicates varying usage patterns at different times of day (Falaki et al., 2010; Karikoski & Soikkeli, 2011).

The length of interaction sessions vary immensely, but most are short in length. In addition, approximately 90% of all interaction sessions are limited to one application (Falaki et al., 2010).

2.3 The state of the art

In this section we will look at some of the dominant actors in the smartphone operating system market in order to identify how information and tools are handled by the different vendors. This section will generally outline the dominant smartphone operating systems, and some of the new features that have recently been introduced on these platforms. The operating systems presented are Apple iOS, Google Android and Microsoft Windows Phone 8. Additionally two lesser actors, Ubuntu Touch and KDE Plasma, are presented due to their novelty.

2.3.1 Apple iOS

The iOS operating system was created, and is maintained, by Apple Computers. The system is used on Apple's own mobile computing devices, including the iPad and iPhone. According to a report by market research firm IDC iOS devices 18.8% of total shipped smart phone devices in 2012 (IDC, 2013). Through the integrated application store iOS devices have access to buy and download an abundance of third party applications. These applications continuously extend the functionality of iOS devices, thus making them ever more capable of performing ever more complex tasks. The iOS operating system is based around the application-centric paradigm, but with a new twist in that no file explorer is available [3]. This leads to information being tightly connected to the applications it originated from.

When booting up an iOS powered smart phone a user is presented with a home screen consisting of a grid of installed applications. By clicking an application the user is taken to the user interface of the application in question. Due to a strict application review process, specialised toolsets, and design guidelines Apple has managed to shape the user interfaces of these applications into a mostly coherent and consistent experience where the same gestures and interface elements are tied to the same behaviour.

[3]It is however possible to attain a file explorer through a process called Jailbreaking.

FIGURE 2.3: Screenshot showing the iOS home screen on an iPad 2.

Applications within the iOS ecosystem are mostly very specialised to a certain domain of information, providing tools for all allowed tasks within said domain. They are at the same time more or less completely segregated from other applications. For instance the Facebook application provides all the tools necessary to browse content from Facebook.com, including a general purpose web browser used to display external links. The same is true for the application Alien Blue, which is tailored to browsing content from the social sharing platform Reddit.

There are some exceptions to the segregated nature of iOS applications. For instance there is often a "share"-button, which allows a user to send content to a predefined set of compatible applications. One example of this would be to send a URL to the Safari web browser application, or to send a link along with meta information to the Facebook application so that the user can share the link on his or her social stream. The applications in these share-menus are however determined by the application developer, and often limited to a small subset of applications.

iOS has some features that enable users to quickly switch between applications, thus facilitating multi-application usage. On iPhone double clicking the physical button below the screen brings up a row displaying the icons of previously used applications. Clicking these icons will take you to the corresponding application. On iPad users can switch between applications using a four-finger gesture. Using this four finger gesture will switch to the last used application.

2.3.2 Google Android

The Android operating system is based on the open source Linux kernel, and is used to run a vast number of different devices ranging from smart phones to media centres. Android is currently the largest smart phone operating system with a share of 68.8% of shipped units in 2012 (IDC, 2013).

When booting an Android device a user is taken to a screen containing application icons and widgets (fig. 2.4). Widgets are small information containers connected to an application, which can be used in order to display a snippet of information or to control certain functionality. For instance a weather application can display temperature through a widget, and a music application can display playback controls in a widget. The home screen is divided into several screens, each containing icons and widgets. The user is able to move, remove or add new icons and widgets.

FIGURE 2.4: The Android home screen, as displayed on a Sony Xperia Z

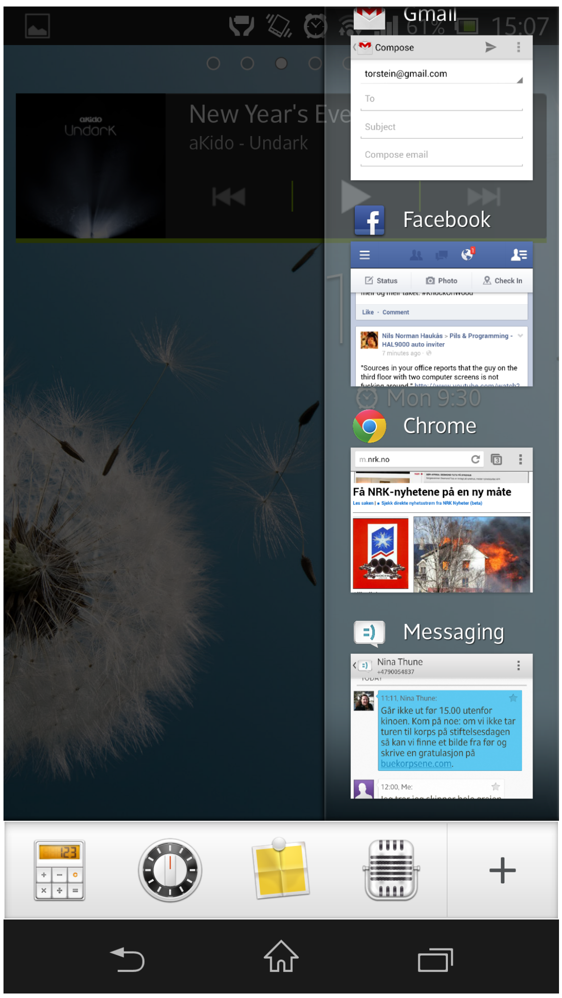

FIGURE 2.5: The Android application switcher, as displayed on a Sony Xperia Z with Android 4.1.

While applications in iOS are largely independent entities, this is not always true in the Android-universe. Applications in Android are often intent-driven. This means that an application can "sign up" to open and receive certain information if something is done in another application. A real world scenario would be to click a link to a Youtube video within a social media application. The operating system would detect that a Youtube video has been clicked and would give you a choice of applications that have told the operating system that they are capable of receiving such links. In the Youtube example such applications could be the web browser and a dedicated Youtube application. This differs quite radically from the "share" or "send-to" functionality in iOS, since it is an implicit action instead of the explicit action on iOS. It is also different since it lets the operating system handle intents, thus avoiding the situation in which an application developer has to explicitly choose a set of applications, thus letting less ubiquitous applications appear as a choice to receive information.

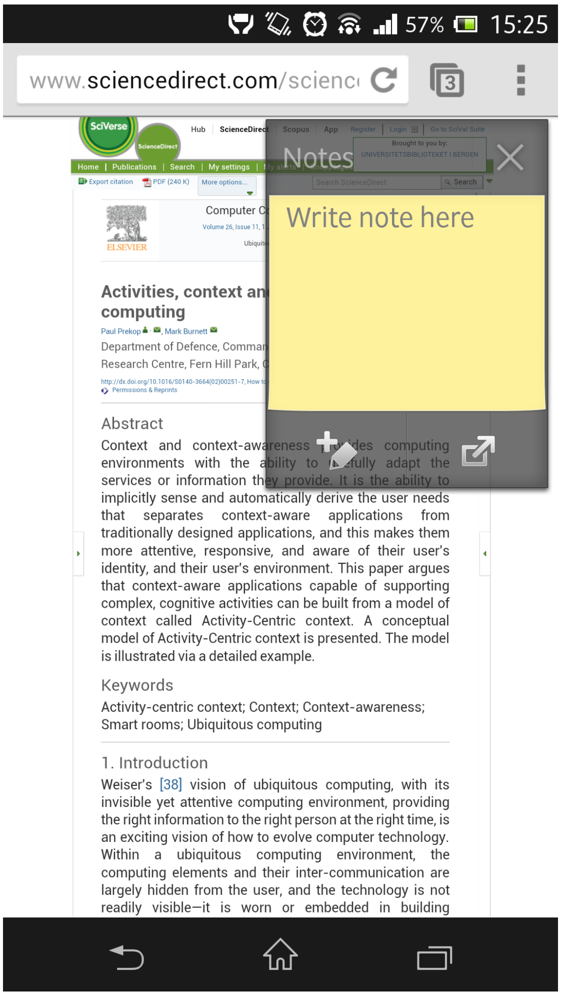

FIGURE 2.6: "Small apps", in the form of a notebook application, as displayed on a Sony Xperia Z

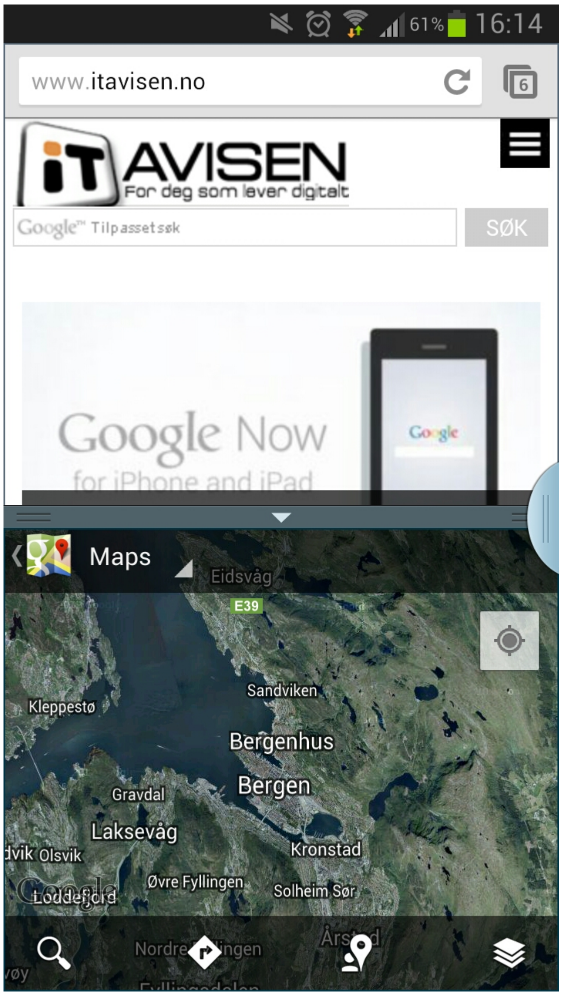

FIGURE 2.7: Multi-view, here displaying a browser and Google maps, on a Samsung Galaxy Note 2.

Android is a fragmented landscape with different vendors providing different functionality. For instance in recent Samsung devices, such as the Galaxy S4, Samsung-flavoured Android lets the user use a featured named "multi-view" (fig. 2.7). Multi-view is basically a function that lets a user start and display two applications simultaneously. Sony, on the other hand, provides something called "small apps" (depicted in fig. 2.6), which are small utility applications that will stay active and on top of other applications until you close them.

While most Android versions have some way of quickly switching between recently used applications Android 4.1 and 4.2 provide a more standardised and unified experience. Through clicking a button that is almost always visible the user is presented with a screen displaying active applications (as seen in fig. 2.5). Clicking a screenshot will open the corresponding application.

2.3.2.1 Google Now

One new innovation in newer versions of Android is Google Now. Google Now aims to help its user through voice recognition, access to Google's search technologies, and context awareness (Google, 2012). Google Now is accessible through its own gesture on Android versions from 4.1 and up on both tablets and phones.

The technology is at the moment limited to a set of specific kinds of information. These are organised into individual cards, whose contextual information can be configured by the user. Examples of cards are weather, traffic information, appointments, flights, restaurant reservations and events. These cards retrieve information from the user's personal accounts, such as Google Calendar, in addition to using information about time and location in order to retrieve information from the Google search engine. In addition to the cards, Google Now also incorporates voice controls, which lets users look up information on the internet and control a limited subset of phone features through using voice commands and to some extent natural language questions.

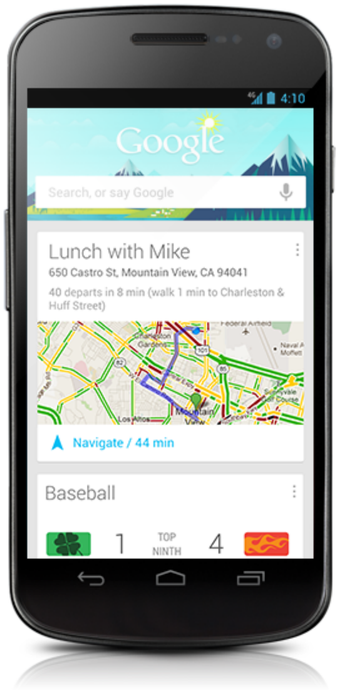

FIGURE 2.8: Screenshot showing Google Now (Google, 2012)

2.3.3 Windows Phone 8

Windows Phone is Microsoft's new mobile operating system, replacing Windows Mobile. According to IDC, Windows Phone and Windows Mobile have a joint market share of 2.5% of sold units in 2012 (IDC, 2013). Windows Phone features unique interface elements. Windows Phone applications are organised in several layers of information. The outermost level is the application tile in the home screen. These tiles are capable of displaying important information from within the application. For instance an e-mail application could display the number of unread messages directly on its tile. When opening applications you have several levels of information. The first screen will often display general information from the different parts of the application. When clicking through to new levels you will get more specific and detailed information. One can say that information in Windows Phone applications bubble up through the different levels, getting less detailed the higher in the hierarchy they are displayed. This lets users gain a quick overview of the status of their applications through glancing at the home screen.

2.3.4 Ubuntu Touch

Ubuntu Touch is a new contender in the mobile operatings system market. Created by Canonicle, Ubtuntu Touch is the mobile version of the Ubuntu Linux operating system. Canonicle's explicit aim is to provide a unified platform for all devices, including desktop computers; media centres; tablets and phones. Canonicle also regards smartphones as fully featured computers, which has some interesting implications. Due to this philosophy, a Ubuntu powered phone will adapt to different screens and input methods. For instance if docked to a desktop setup the GUI will adapt to display a desktop environment instead of the mobile GUI.

The Ubuntu Phone interface introduces some novel interactions, notably through the active edges. Active edges means that you through dragging your finger from an edge will be able to do certain things. When dragging from the left you will access an application menu, where you can select an application to open. When dragging from the right you go back to the previously used application and when dragging from the top you access a search menu. In their official presentation of the system Ubuntu names information accessibility and multiple-application usage as some of their main points of interest when creating the interface.

2.3.5 KDE Plasma Active

One effort into implementing ACC in consumer devices was the KDE Plasma Active. KDE Plasma is an activity-centric system for tablet devices, where the user could organise applications in groups on the tablet desktop (KDE, 2012). The project also talked of an activity-aware browser and other features. KDE Plasma Active has not yet reached any consumer devices, and the project seems to have gone dormant.

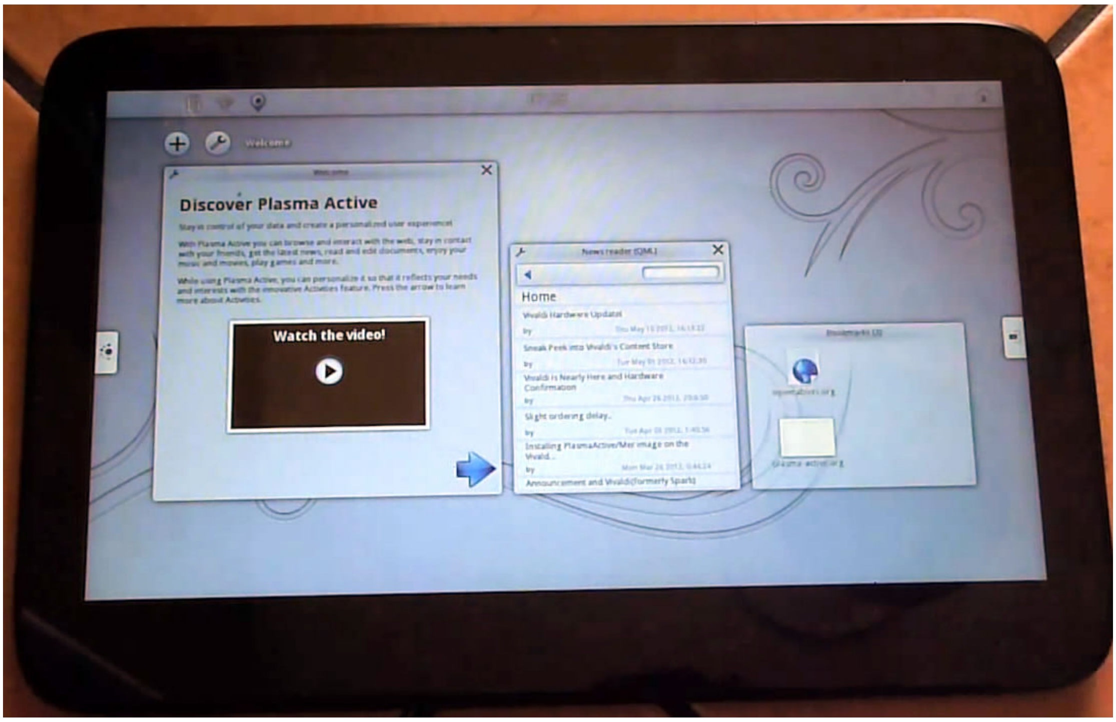

FIGURE 2.9: Screenshot from a video demonstrating KDE Plasma Active 3 (KDE, 2012)

2.3.6 Summary

The current smartphone market is dominated by a few actors, namely Apple's iOS and Google's Android. In addition there are some contenders, such as Microsoft's Windows Phone 8 and Canonicle's Ubuntu. These systems are mostly very similar with applications being segregated from each other, and seemingly designed to work independently of each other. In other words, the industry is still operating within the application-centric paradigm. The smartphone industry is however continuously changing, with innovations being released continuously.

The industry seems to focus on innovating within a few areas. One area is the usage of multiple applications, where both Samsung and Sony have released new functionality, namely "multi-view" and "small-apps", within the last six months. Another area of continuous innovation is context-awareness. The most notable example of recent innovation within this area is Google Now, which is designed to provide information based on context.

In general the industry seems to have identified three problematic areas in current smartphones. One is information comparison and tool sharing, exemplified by the innovations allowing the use of multiple applications and functionality for quickly switching between applications. Another problem area is context-sensitive information, such as weather forecasts and traffic information, which Google has tried to solve with its Google Now application. The last area is input, which the industry is trying to alleviate through speech recognition.

2.4 Problems and Potentials

As mentioned by Jones (2008) the different PIM tools available today have led to information fragmentation. Within the current generation of smartphone operating systems this is especially true, since each application basically constitutes its own PIM structure. In fact, it seems like many applications are trying to deliver a specialised PIC for the domain that they cover. In addition smartphones try to cover a plethora of different uses, such as communications, personal organisation, entertainment and quick referencing. These factors combine to create a situation where PICs are confined to single applications resulting in a fragmented and unmanageable smartphone PIS. The fact that current smartphones neglect to provide functionality for creating user personalised and inter-application PICs also contributes to a situation where the perceived utility of smartphones probably does not coincide with the actual capabilities.

Research shows that smartphone users go through a set of identifiable contexts each day, and that usage varies by contexts (Karikoski & Soikkeli, 2011; Falaki et al., 2010). This is something that especially Google has tried to incorporate into the Android operating system through Google Now. Judging by the data gathered provided by Øvrebø (n.d.) it seems like the introduction of context sensitive information has increased the user experience for some users, and that said users now wish for more context-sensitive information in other applications. The fact that usage patterns change in different contexts combined with a possible increase in perceived usability might imply that context-awareness is useful, and might further imply that applications should at least in some cases comply to contextual cues. However the general contexts identified in research and the contexts implemented in Google Now might be too general. I will propose a hypothesis that the usefulness of contexts will increase if users are able to more clearly define and configure contexts specific to themselves.

While far from conclusive for the general population, numbers gathered by Øvrebø (n.d.) indicate that some users perceive a need to combine tools, as shown through the requests for "integrated note taking" and "todo-list in iCal". There have been some efforts in the industry to create functionality to meet such requests. For instance "multi-view" in Samsung's flavour of Android and "small-apps" in Sony's flavour of Android exemplify recent efforts. These systems let users compare information and use tools from two different sources. The downsides are that the functionality is very limited, and that it is impossible to save the state of the tools. A user cannot save a multi-view of two applications in order to quickly access the same multi-view at a later occasion. When looking at the usage data gathered by

Falaki et al. (2010) this might be a problem. Users seem to check quick snippets of information most of the time, and while multi-view provides some opportunity to check information from several sources at once it does not fit into the quick interaction sessions that were found due to the fact that state cannot be saved.

At the time of writing activity-centric computing has almost exclusively been applied to desktop operating systems and distributed interaction scenarios. For instance ROOMS (Card & Henderson, 1987) and UMEA (Kaptelinin & Boardman, 2007) are examples of ACC applied to desktops, while for instance J. E. Bardram (2007) looks at distributed multi-user scenarios. I believe that ACC provides a valuable frame of thought for solving the previously mentioned PIM issues. I believe that introducing the concept of activities, in the form of meta-applications would serve as a good middle ground between the current state of affairs and and a completely activity centric approach. I believe that the ability to both continue with current usage patterns in addition to facilitating other usage patterns through enabling the creation of PICs would serve to increase both the perceived and the actual usefulness of smartphones.

Context-awareness serves to get information at the right time, and activity-centric computing provides a concept in which you can gather the right information and tools. Combined they serve to fulfil, or at least approximate, an ideal PIM implementation.

Chapter 3

Methodology

One prevalent paradigm within Information Science research is design science. Design science is a paradigm in which one seeks "to extend the boundaries of human and organisational capabilities through creating new and innovative artefacts" (Hevner, March, Park & Ram, 2004). IT artefacts are broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems) (Hevner et al., 2004). Fundamentally it is a problem-solving paradigm, focusing on the development and performance of novel artefacts to solve problems. This makes it well suited to situations in which one explores how abstract concepts can be applied to solve or alleviate concrete problems, such as in this thesis.

Design within design science can be said to be two things, a set of expert activities performed and a resulting product in the form of an artefact. Through evaluating the artefact one can gain insights both into the artefact itself and into the design process used in order to create said artefact. In turn these insights can help evolve both the process and future artefacts. In general this design-feedback loop is performed several times throughout a design science research project before a final design artefact is generated (Hevner et al., 2004).

Design science research is fundamentally different from routine design in that design research aims to solve a problem through novel practices, and thus gaining new knowledge, while routine design aims to solve problems through applying existing knowledge and practices. In other words one can say that the key aim of design research is to contribute new knowledge through solving a problem, while the key aim of routine design is to apply existing knowledge to solve a problem.

Hevner et al. (2004) proposes 7 guidelines to design-science research:

Design as an Artefact Design-science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation.

Problem Relevance The objective of design-science research is to develop technology-based solutions to important and relevant business problems.

Design Evaluation The utility, quality, and efficacy of a design artefact must be rigorously demonstrated via well-executed evaluation methods.

Research Contributions Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.

Research Rigour Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artefact.

Design as a Search Process The search for an effective artefact requires utilizing available Process means to reach desired ends while satisfying laws in the problem environment.

Communication of Research Design-science research must be presented effectively both Research to technology-oriented as well as management-oriented audiences.

The guidelines are however not necessarily completely compatible with design science research within the discipline of interaction design. For instance it is hard to achieve a level of research rigour in which two interaction designers would reach the same result. However there have been efforts to alleviate this. One example is Zimmerman, Forlizzi and Evenson (2007)'s model. This model involves four "lenses" which both inform the design process and the evaluation process for an HCI-angled design science research project.

Process

The first of the four lenses in Zimmerman et al. (2007)'s model is the process. While one in some research disciplines can achieve a high level of consistency, that is not necessarily true for research within interaction design and HCI. While there are no expectations that designers through replicating the same process will arrive at the same result, Zimmerman et al. (2007) argues that one through looking at the rigour at which methods were applied and through looking at the rationales behind decisions can judge the quality of a interaction design research contribution. Therefore it is natural that enough detail is provided about the design process that it can be reproduced, and that the rationale behind choice of methods is provided.

Invention

A interaction design contribution should also constitute a significant invention. In order to demonstrate this an extensive literature review has to be conducted, that places the work and details aspects that demonstrate the novelty of the concept and how the contribution advances the research field (Zimmerman et al., 2007).

Relevance

There can be no expectations that two designers will reach the same conclusion given the same problem. Therefore Zimmerman et al. (2007) argues that one instead of validity should use relevance as a benchmark for interaction design research. In order to do this an interaction design researcher should frame the work within the real world in addition to articulating the preferred state their design attempts to achieve. The researcher should also support why this state is to be preferred.

Extensibility

The final criterion as presented by Zimmerman et al. (2007) is extensibility which he defines as "(...) the ability to build on the resulting outcomes of the interaction design research: either employing the process in a future design problem, or understanding and levering the knowledge created by the resulting artefacts." This implies that the research conducted should be described and documented in such a way that others can utilise the knowledge derived from the work in their own projects or research.

3.1 Prototyping

In order to explore the design questions posed a series of prototypes were created. Prototyping is typically used in order to test concepts without overcommitting to the implementation of said concept. There are several types of prototypes, and a plethora of prototyping techniques, which depending on situation can be used to clarify or answer questions or contribute to the collection of requirements.

'It is often said that users can't tell you what they want, but when they see something and get to use it, they soon know what they don't want (Sharp et al., 2007)[530]'

In general one can split prototypes into two categories: high-definition and low-definition prototypes. The level of definition is related to how closely the prototype resembles the finished product. Low-fidelity prototypes are prototypes that are not similar to the final product in appearance and medium. They are created in order to explore questions related to a concept and are thus created in a format which is easy to work with, like for instance cardboard. High-fidelity prototypes on the other hand, are prototypes that are very similar to the final product in appearance and medium. Within the realm of human-computer interaction prototypes often come in the form of low-fidelity prototypes consisting of interaction diagrams, modelling the proposed interaction, and cardboard mock-ups, which are basically drawings resembling the screens of the finished product. Combined these can be used to perform test sessions, thus enabling interaction designers to quickly test concepts without the cost of creating and implementing a fully functional system.

In this thesis prototypes were employed in order to reify possible solutions related to problems posed as a part of the exploration of the second, third, and fourth research questions. Due to the limited resources available, the highly advanced and technical nature of implementing high-fidelity prototypes of smart phone concepts and the need to quickly iterate, low-fidelity prototyping was chosen. Several prototypes were created through the work on this thesis in order to explore possibilities and ideas and in order to reify concepts in order to gather data in the form of user feedback gathered through focus groups and a user test. The prototypes were created in different levels of fidelity, varying from the purely conceptual models of the CATA-framework to more concrete visualisations. The latter prototypes, each created using a iterative process of user stories; sketches; wireframes and computer imaging, consisted of computer generated screens combined with explanations of how each interaction would function. The first finished prototype, dubbed Activity Ant 1 (chapter 5), was a vertical prototype detailing the creation process of activities and contexts. The second prototype, dubbed Activity Ant 2 (chapter 6), consisted of a series of mock-ups and explanations of an alternative activity format and information sharing for said activities.

3.2 Focus Groups

Focus groups were used to gather data in order to explore the fourth research question ("How can the concepts and potentials of activity-centric computing be conveyed to users?"). A focus group is a form of group interview often conducted within marketing, political campaigning and social sciences research. The general assumption being that people generate opinions through communicating in a social context. Group sizes vary, but Sharp et al. (2007) suggests a size of 3 - 10 participants in addition to a trained facilitator. Focus groups should consist of a representative sample of the target population. Within the context of this thesis that means people who use smartphones.

Focus groups have both positives and negatives. One positive is that one through employing focus groups can gather multiple viewpoints quickly, thus; while one must point out that focus groups are qualitative in nature; they do give a slight quantitative quality to a mostly qualitative field. They also serve as a good resource for requirements gathering and as a source of fresh ideas. They do also serve as an arena where developers can discover points of consensus and conflict within their user mass. Focus groups are however prone to being dominated by dominant characters. One must also be aware of other human factors, such as faulty memories and lies (to put oneself in a better light), which can discolour the results of a session. Focus groups are a good way to explore the feelings and judgements of a few people, they are however not a substitute for usability tests since they are not suited to measure performance issues and real behaviours (Rubin & Chisnell, 2008).

Conducting a focus group session is a way of quickly gathering multiple viewpoints. One can say that, while mostly qualitative, focus groups have a slight quantitative quality as well. Within HCI, focus groups are often conducted in order to gather requirements and to act as a form of think-tank in order to collect fresh ideas. They can be considered part of a "proof of concept" review. Questions posed in a focus group are intentionally deceptively simple, in order to let the subject be open enough to encourage the participants to forward their opinions. And, while a general goal and questions are created in advance, focus groups provide an open enough forum to freely follow unanticipated issues if raised.

Focus groups were chosen as a data gathering technique due to the limited time necessary to set up and perform sessions. They were also chosen due to their free format and intrinsic ability to let participants propose their own questions, issues and concepts. In essence, the focus groups were used both to gather data related to users' understanding of the concepts proposed, and in order to gauge how they would receive and use such systems. The focus groups also served as a platform on which I and the participants could come up with real world applications and to propose concrete changes to the prototypes.

Chapter 4

Design Process

In this chapter I will present the design process that led to the two prototypes. The prototypes are presented in chapter 5 (Activity Ant 1) and chapter 6 (Activity Ant 2).

In general a design process within HCI consists of a set of basic activities. Sharp et al. (2007)[17] names the following four basic activities:

1. **Identifying needs and establishing requirements for the user experience.**
2. **Developing alternative designs that meet those requirements.**
3. **Building interactive versions of the designs so that they can be communicated and assessed.**
4. **Evaluating what is being built throughout the process and the user experience it offers.**

The main goal of the design process was to explore the feasibility of ACC and CA within the confines of functionality present within the current generation of mobile operating systems. This poses an interesting problem in that the goal was to create something novel within something existing. Thus the natural first step in the design process was to gain domain knowledge about the current state of affairs, as detailed in section 2.3 and section 2.2.

The next step was to imagine use cases for context-awareness and activity-centrism, and thus create a foundation on which a set of requirements could be defined. This was done through creating scenarios. Inspired by the contexts "everywhere else" and "office" found by Karikoski and Soikkeli (2011), the below scenario was created. This was the first user scenario created in order to inform the design process, and involves a typical afternoon activity: grocery shopping.

> Bob is a student who lives at Fantoft and likes to create shopping lists for his groceries.
>
> Through the system he creates an action "Go Home". Inside this action he defines the following information needs: find a route home, provide contact details to the local taxi company and display a bus schedule. When Bob is away from home he can click the "Go Home" action which will then display the given information in an interface.
>
> Through the same system Bob defines an action "Do Grocery Shopping". This action he defines as show closest super market and show notes in the category groceries. When clicking the "Do grocery shopping" action he will be displayed a map which guides him to the closest super market and when there the application switches to show his note depicting his grocery list.

In order to get some early feedback from potential users, these scenarios were discussed in informal settings. Through these discussions a pattern emerged; without being prompted users would start creating scenarios related to their own lives. This indicated two things: (i) people I spoke to seemed to be able to understand the concepts quite quickly and (ii) they were able to apply the concepts to their own lives. As previously stated these conversations were informal and spontaneous in nature, and were therefore not recorded. I have however tried to reproduce an anonymised version of an actual usage scenario as I remember it below.

> Bill is a student who is involved in the boards of many volunteer organisations in his spare time.
>
> Due to his role in these boards Bill often has to attend meetings. Each of these meetings has a stated agenda with related documents.
>
> Bill creates an activity for meetings. In this activity he displays the agenda and related notes. He also includes a notepad.
>
> Bill also adds information about each meeting, including time and location to his calendar.
>
> Bill's smartphone sees that Bill has to go to a meeting, so it gives him a notice.
>
> The smartphone notices that Bill has arrived at his meeting and activates the activity containing relevant information and the notepad.

One interesting aspect of the above user scenario is that, while not an exact fit for a typical office context, it still confides to a general office context. While it is impossible to draw any definitive conclusions from scenarios created as part of a limited set of informal discussions it does seem like users would like their phones to adapt to some contexts. There also seems to be some indication that these contexts would have to be user defined, exemplified again by the above user scenario where a typical office activity (meetings) would be held at a time and location outside a typical office (due to the fact that the scenario is for a student and the meetings for volunteer activities are often held after typical office hours).

4.1 The CATA-Framework

These scenarios were used in order to inform a brainstorming session where ideas as to how context awareness and activity-centric computing can be combined and how these could use information from existing applications. The result of this session was a series of drawings that ended up forming the basis of the CATA-framework.

The CATA-framework is a abstract framework that aims to model the interaction between system components that combined constitutes a CATA system. The framework consists of four concepts: contexts, agents, tasks and activities that when combined forms the foundations of a context-aware and activity-centric mobile PIM system. The framework was created in order to inspire further work on creating the two tangible prototypes presented in this thesis: Activity Ant 1 (chapter 5) and Activity Ant 2 (chapter 6).

The CATA-framework consists of four main parts: (i) contexts, (ii) agents, (iii) tasks and (iv) activities.

Contexts refers to contextual information available. Contexts are used to determine which situation the phone is in, which in turn can be used in order to affect the information displayed or in order to trigger notifications to the user. Contextual information is available through sensors and through phone activity. Examples of sensory information available is time, location and direction. Examples of contextual information available through activity is simply information about what the user is doing and has been doing on the device.

Agents are small programs that gather and uses contextual data. Agents can be implemented through using the services in Android. One example of an use for an agent is to check the phones current context against a list of contextual triggers. This would be done for instance in Activity Ant 1 in order to see if a task should be triggered.

Tasks are descriptions of a particular task that a user wishes to perform. A task consists of two parts: (i) a specific state that an activity should start in and (ii) contextual triggers in which the task should start. An example of a state would be a specific to-do-list that should be displayed in an activity's todo-application. An example of a contextual trigger could be "phone is outside of grocery store between 16:00 and 20:00".

Activities are PICs consisting of information and tools. Activities are created by the user. The state of an activity, i.e what information and tools that are displayed, can be overridden by a task.

The concepts of CATA can quite easily be tied to existing system resources and functions. Contexts can simply be tied to existing systems through existing sensors and programming interfaces. Agents can simply be implemented through using services in Android. Services are basically programs that continuously run, even when their parent application is closed. Such a program would, within the CATA domain, simply have to check the current context of the phone, and see if any tasks have registered a trigger for that particular context. Activities are the hardest part to implement in existing systems. However, as Activity Ant 1 (chapter 5) shows, it is not impossible, and as Activity Ant 2 (chapter 6) shows, it does not necessarily require existing applications to be rewritten. Tasks are a bit harder to implement. Context triggers can be stored in a form of data storage that can be checked by an agent. Activity states however would in the case of Activity Ant 1 and 2 require that the applications that serve data to the activities can receive states that they should start in.

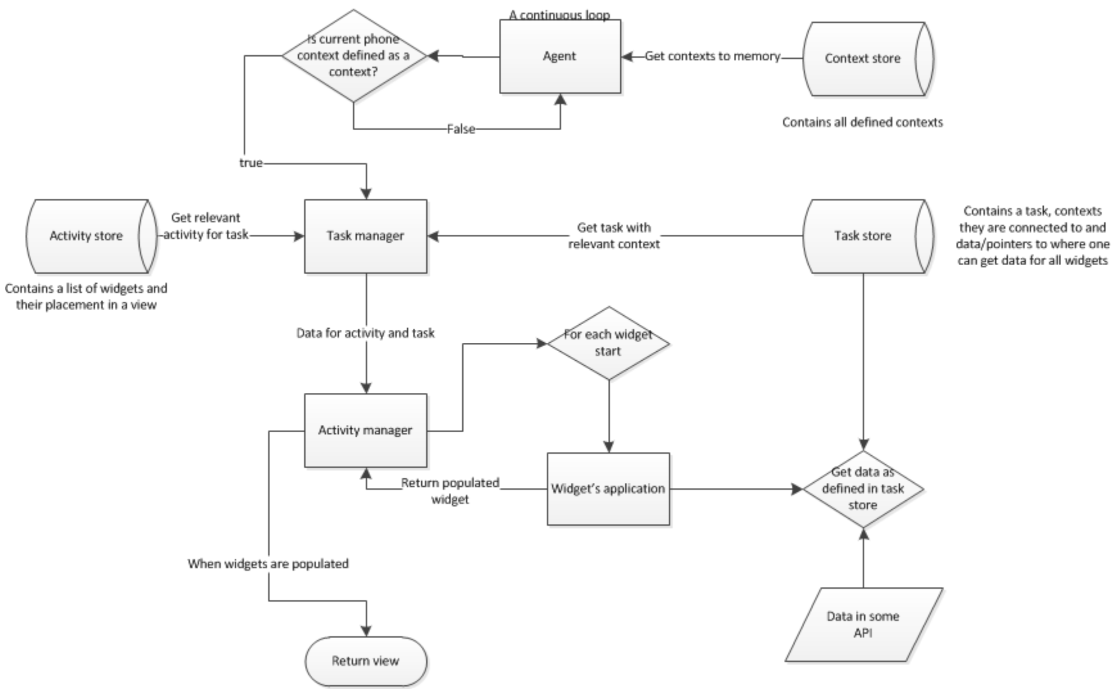

FIGURE 4.1: The CATA framework

The CATA-framework can be implemented in different ways depending on how activities receive information and tools from the system. In the prototypes in this thesis the information and tools are sent through the existing applications, thus forcing an implementation of CATA which relies on these applications. An alternative implementation would be to separate information from applications, which would entail an alternative implementation of CATA. fig. 4.1 shows how CATA could be implemented in Activity Ant 1.

4.1.1 Designing Activity Ant 1

Originally I had planned to create a high-fidelity prototype in the form of an application in order to show the feasibility of context-awareness and activity-centric computing on mobile devices. However, after having looked at the components in the CATA-framework and how they could be implemented in the Android operating system, it soon became clear that this was too large of an undertaking. So, instead I opted for low-fidelity prototypes. This had the added benefit that it enabled me to cover more concepts and go through more iterations.

The first prototype, Activity Ant 1 (presented in chapter 5) was largely inspired by three existing systems: the ROOMS system (section 2.1.4), the widget-home-screen concept in Android (section 2.3.2) and the live tiles of Windows Phone 8. The general concept was to recreate the information hub that the Windows Phone 8 home screen is through employing widgets and grids to create activity-meta-applications. Each application is split into logical parts, which can be combined to form new "meta-applications". In essence, the idea is that this route to combining information does not require the information itself to be changed, since their respective applications are still in charge of storage, retrieval, presentation and computation of said information.

AA1 tried to take into account several aspects of phones and phone usage. As previously mentioned a study conducted by Falaki et al. (2010) showed that most smartphone interaction sessions were short in nature and limited to one application.

AA1 tried to expand the possibilities of these quick interaction bursts by creating collections of information and tools displayed in a single view. In essence the general idea was to enable the user to create views with specific information required in different scenarios.

AA1 also tried to take into account states and contexts through tasks. The general idea was to let users define both situations in which they wanted to be reminded to use specific activities, and the state in which the widgets in said activity should start.

The resulting prototype resembles the Haystack system (presented in section 2.1.4). It however differs in one central aspect: instead of reworking the entire operating system stack into a system based on semi-structured data and connected views, AA1 uses the applications to provide information to the user. This difference makes AA1 more closely resemble the existing systems, and allows applications to continue in almost the same manner today while simultaneously allowing users to combine information sources almost as they would in the Haystack system. I believe the AA1 way is more likely to succeed since it allows companies that essentially live of their information keep control. I also think it would be more likely to succeed since it does not require an as fundamental rewrite of mobile software.

Activity Ant 1 is described in further detail in chapter 5.

4.1.2 Designing Activity Ant 2

Activity Ant 2 was created through looking at the data gathered through the testing of Activity Ant 1, and had the explicit goal of solving usability problems and incorporating feature requests from the debriefing part of the user test and the discussions at the focus group session.

One of the major points of feedback from the testing of AA1 and the focus group session was that the concept of combining several small widgets on one screen is flawed. Users thought it would prove unusable due to the limited screen real-estate of a small phone in addition to the widgets being too small to provide sufficient information for them to actually be useful. One participant in the user group mentioned the multi-view feature (shown in fig. 2.7, lets a user display two applications simultaneously) on his phone, and noted that he never used it due to the small screen. Another problem that was not addressed in AA1 was how to share data between activities. Another feature requested by the focus group participants was quick access to different tools, such as search and music player controls.

With this in mind the widget idea was scrapped, and the general idea turned to how one could create an application wrapper. The general concept behind such a wrapper would be to facilitate the creation of application collections with functionality to quickly switch between said applications. In addition a secondary goal was to create functionality for collecting and transporting information between activities. In essence AA2 was designed to resemble the functionality of the GroupBar system (section 2.1.4) in addition to providing interactions to ease switching between applications.

AA2 was created through a brainstorming and sketching session where different interfaces were looked at for inspiration. One aspect of activity-centric computing that was not necessarily captured by AA1 is task-interruption. When looking at the smartphone usage scenarios it is quite clear that the interface has to be suited for constant and sudden interruptions. One example of a typical interruption in current smartphones is detailed in the following scenario:

> Trude is adding appointments to her calendar. In order to do this she copies information from emails in her email client to her calendar application.
>
> In the middle of adding appointments Trude receives a phone call. She decides to answer this phone call.
>
> After finishing the phone call, Trude returns to adding appointments. Both her email application and her calendar applications were closed due to her answering the call, and have lost their current state.

> Staring at the start-screen Trude feels frustrated, and decides to add appointments on her laptop instead.

AA2 is even closer to the current state of mobile computing than AA1, and I strongly believe that it would be even easier to implement. AA2 at the same time lets a user group the applications needed to perform an action, and provides tools to quickly switch between said applications. The functionality of AA2 is already present. A user can start a set of applications one after the other, and quickly switch between them. However, the strength of AA2 is that the user can quickly switch between groups of applications instead of just single applications.

AA2 is described in further detail in chapter 6.

4.1.3 Medium and Tools

Originally I wanted to implement the prototypes using the web stack, namely JavaScript; CSS and HTML, in combination with Phonegap, which is a system for starting web-applications as a native application. This would have let me test the concepts at hand in an actual device over time, which would have provided valuable user data over time in addition to a general impression of how the concepts works in the real word on an actual device. However, it quickly became clear that, while confined by currently available technology, the concepts proved to advanced to be implemented in the short timespan available for the work on this thesis. Therefore it was decided that the prototypes would be low-fidelity, consisting of computer generated screens printed on cardboard.

In order to create the prototypes a number of tools were used. Interaction models were created using Microsoft Visio. Designs were created through sketches with pen and paper, wireframed using printed grids, and digitised using Adobe Fireworks.

In order to record the user testing and focus group sessions a Canon EOS 5D Mark II was used. QSR Nvivo 10 was used in order to transcribe the contents of the filmed sessions.

Chapter 5

Activity Ant 1

This chapter presents the first prototype, Activity Ant 1 (AA1). The design process that led to this prototype is presented in section 4.1.1.

AA1 explores the hypothesis that fragments from applications can be assembled into new applications (named activities), and that this would both be useful and be perceived as being useful by test subjects. In essence, all applications would be divided into logical parts, which could be recombined into new meta-applications. These application fragments, called widgets, are capable of displaying information from their "parent application", that is the application that they originated from and get information from. They would also be able to display tools from these applications. These new meta-applications (applications made of applications) essentially constitute specialised PICs.

5.1 Concepts and interactions in Activity Ant 1

Several concepts combine to form Activity Ant 1 (AA1). Some of these are refinements of concepts from the CATA-framework, while others are unique to AA1.

5.1.1 Widgets

Widgets in AA1 are heavily inspired by their widget counterparts within the Android operating system. A widget is a self-contained sub-application which can be placed into any view alongside other widgets. A widget is designed to adapt to given space within a view, just like modern websites designed to adapt to a range of different screen resolutions. A widget can either ask its parent application for content, or receive content from a task.

5.1.2 Activity Shells

The activity shell consists of a collection of references to widgets, and their placements in a grid.

With activity shells a lot of information from different sources can be displayed in one single view. This was done in order to facilitate the short interactions and single application usage patterns found by Falaki et al. (2010). Users can with this concept create views containing excerpts from a number of sources, which can be quickly accessed and viewed in order to watch the state of several systems at once.

5.1.3 Contexts

A context in the domain of a mobile phone is basically the combination of time, locations (through GPS) and what you are doing on your phone at the moment. In some phones there is also the possibility of adding elements to the context through using other build in sensors like gyroscopes, accelerometers and light sensors.

Through analysing the context and the tasks you have defined the mobile phone would be able to remind you of things and maybe in some cases automate things. One example of a reminder would be to make a notification reminding you to get to your meeting. An example of automation would be to ready a route home when it detects that you are in a unfamiliar place late in the evening.

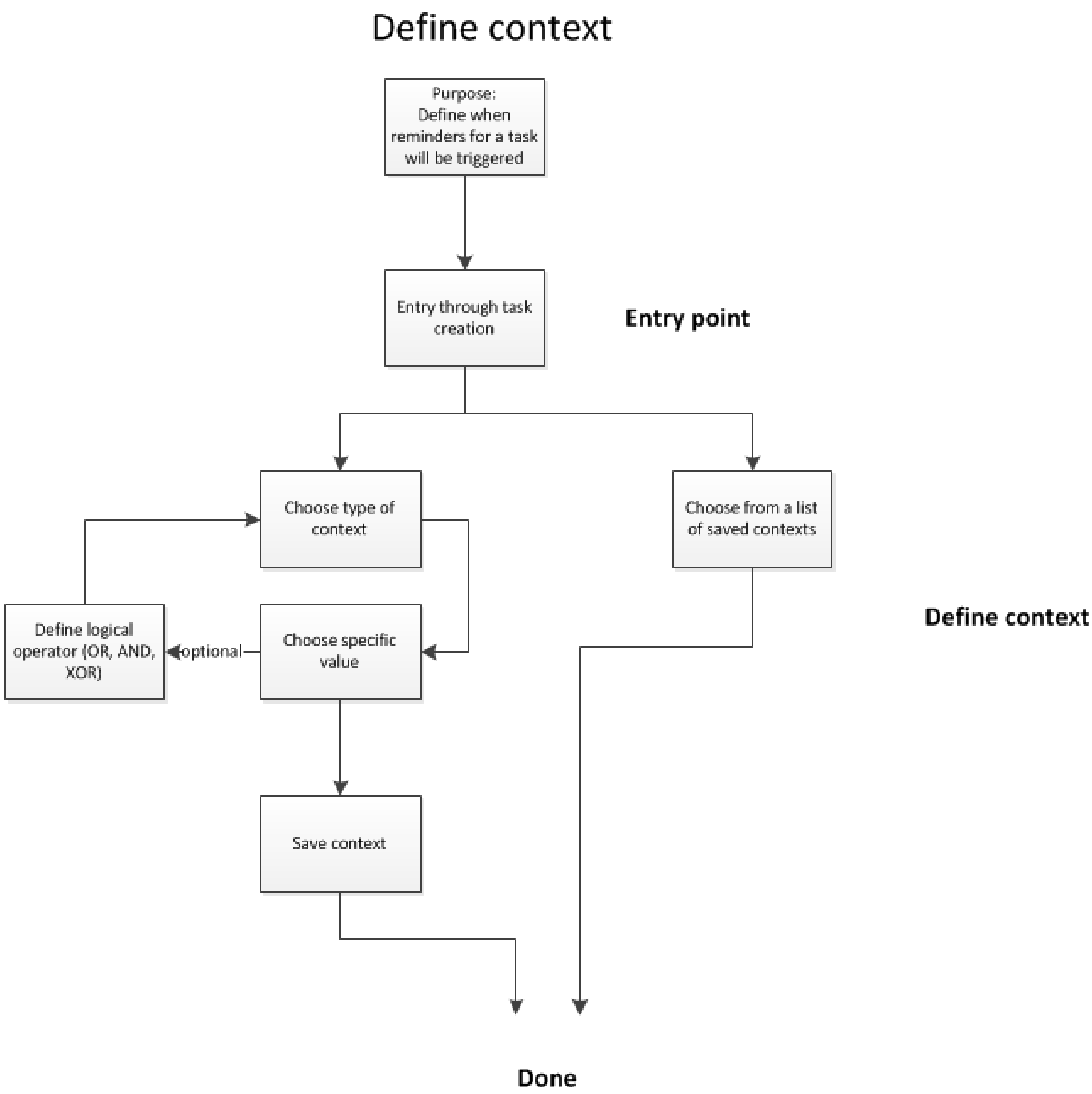

FIGURE 5.1: Interaction for creating contexts in AA1.

In AA1 contexts are created using sensor output and logical operators. This means that a user can combine different sensor inputs into one main context. For instance a user can define a context "at home AND time between 10:00 and 12:00" with the AND functioning as a logical AND. Said context would only be active if both conditions are fulfilled. In addition to AND users are able to use OR. XOR and negation are not part of AA1.

Logical operators were chosen due to their intrinsic ability to combine and transform complex equations into a boolean, true or false, truth value. This allows users to combine any number of sensors to form an unequivocal context that is either active or not active. Negation was not included due to my perception that they are harder to use correctly, and thus would make potential contexts harder to understand for users.

An alternative to using logical operators is to predefine contexts, and only let the user configure the specific parameters. This is a solution that is used for instance in Google Now, but was not tested as a part of AA1.

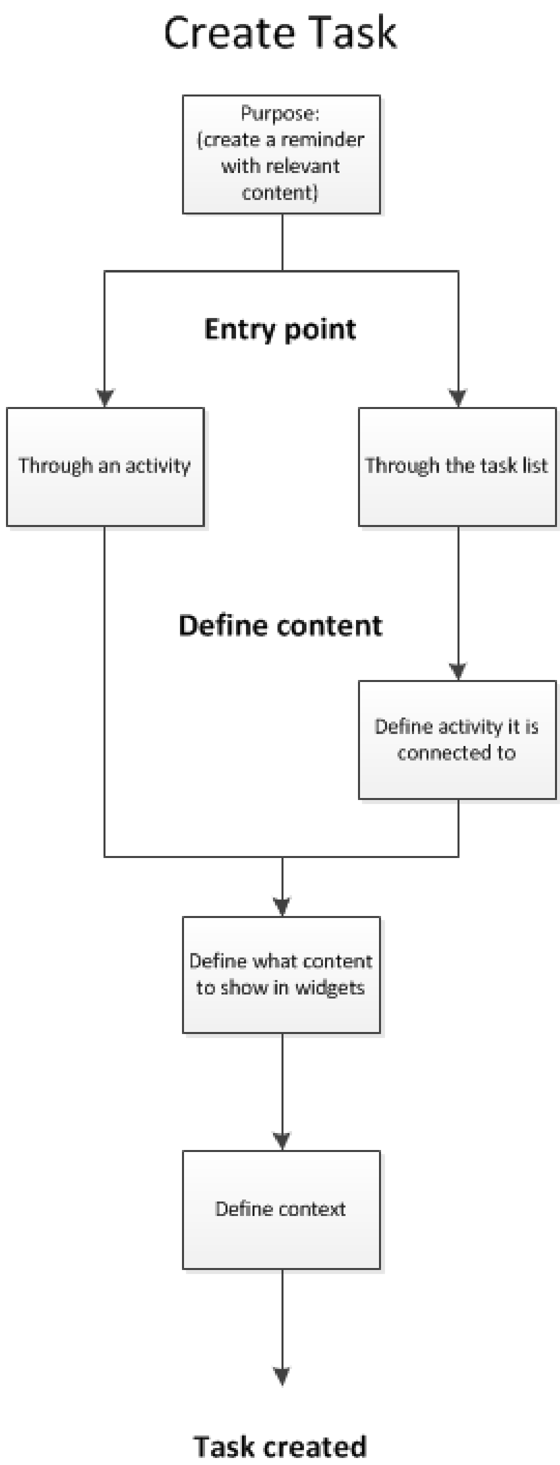

FIGURE 5.2: Interaction for creating tasks in AA1.

5.1.4 Tasks

Tasks in AA1 come in two formats: tasks for the smart phone and tasks for the human. A task given to the smart phone might be for instance to calculate routes on a map at given times (for instance right before you leave for work).

Tasks given to the human user could for instance be to remember to charge the phone.

A task contains several important parts. Each task is connected to one or several contexts, it contains information needed to populate (or override) widgets in activity shells (for instance lists to populate checklists) and they are connected to an activity. Basically the tasks are the glue that combines the different components together in order to start an activity.

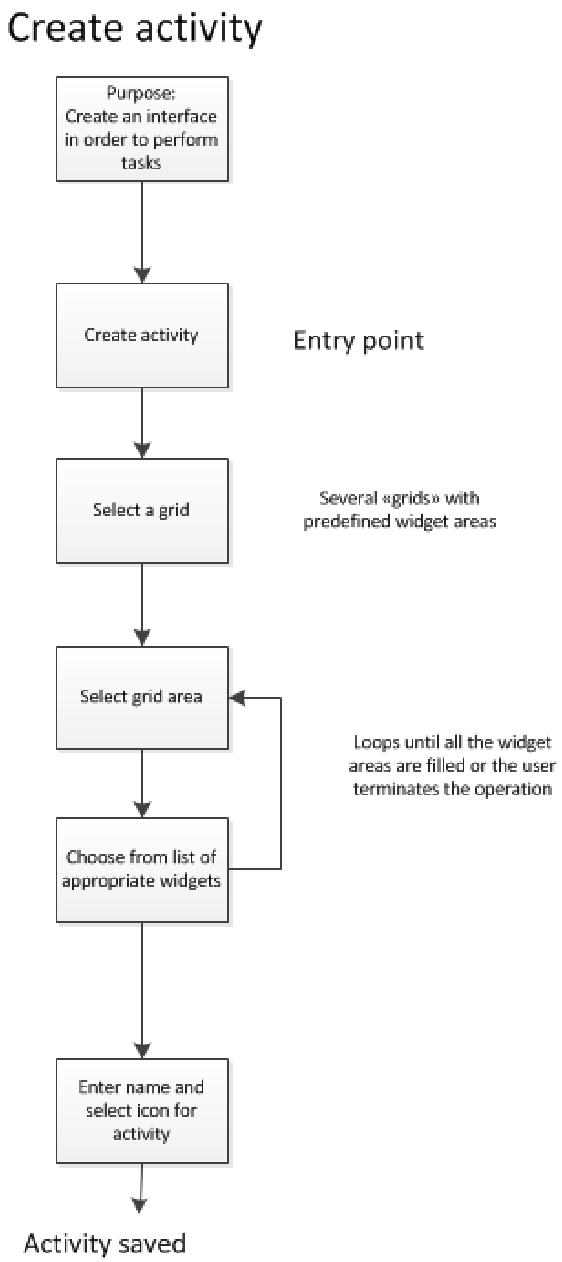

Figure 5.3: Predefined grid interaction alternative for creating activities in AA1.

Content overrides are content blocks that can substitute the default content defined in the activity. For instance if the default content for a widget is to display "Master thesis to-do" a content override can make the widget display the "Create examples for master thesis to-do" instead. This functionality lets users define a specific state to start an activity in.

5.1.5 Activities

Activities are the central hubs
of AA1. Activities contains an activity shell, and tasks are targeted at an activity. In addition the activity contains a default state for each of its connected widgets, describing which information that should be displayed if the activity is started without a task's content override.

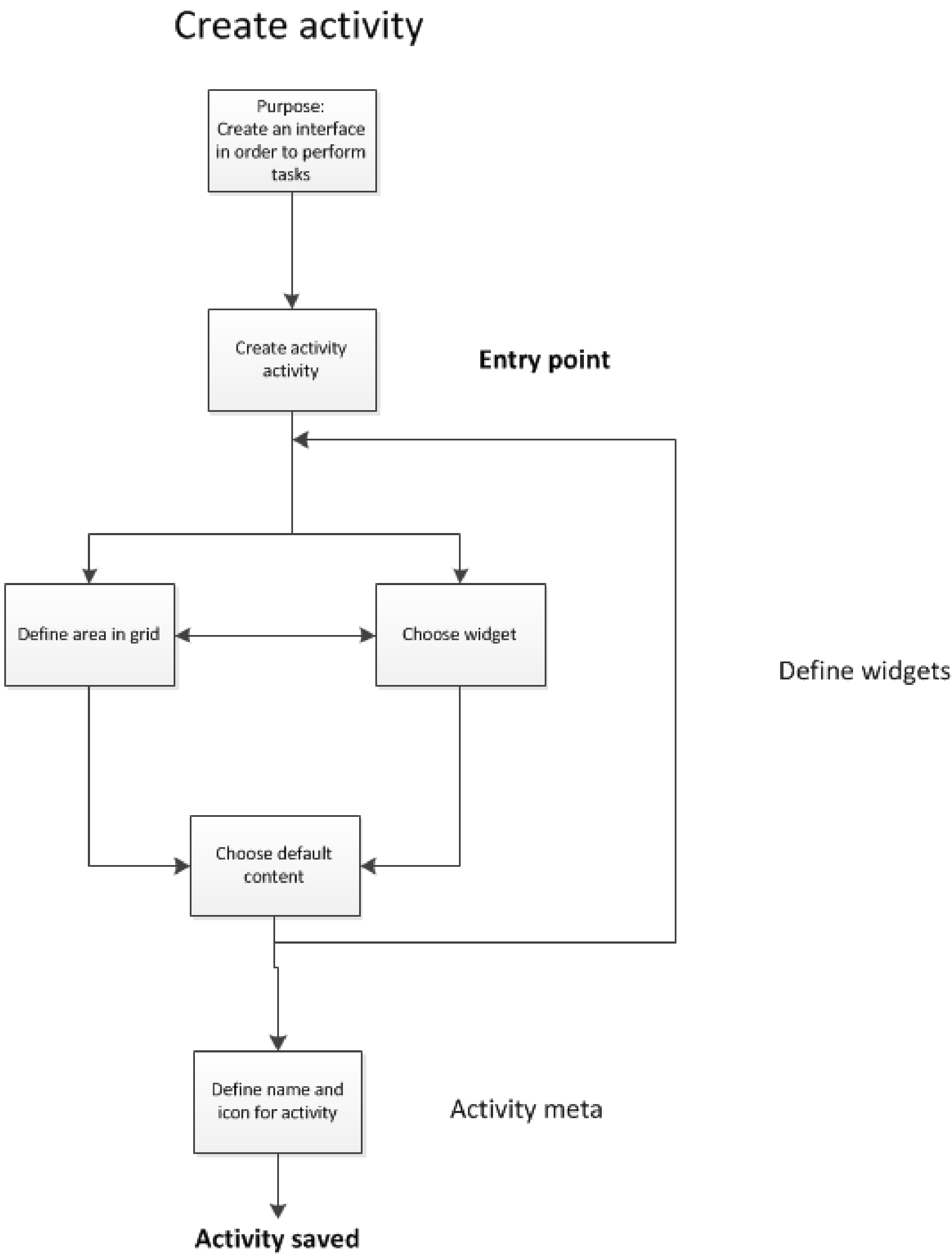

FIGURE 5.4: Dynamic grid interaction alternative for creating activities in AA1.

In AA1 two alternative activity creation processes were modelled and tested. Figure 5.4 displays the interaction for creating an activity where the activity shell contains a fully dynamic grid, meaning that the user can define the exact size and placement of widgets. fig. 5.3 shows interaction for an activity shell where the grid has been pre-defined. This means that the user can choose a grid containing some widget areas and then place widgets into these predefined areas.

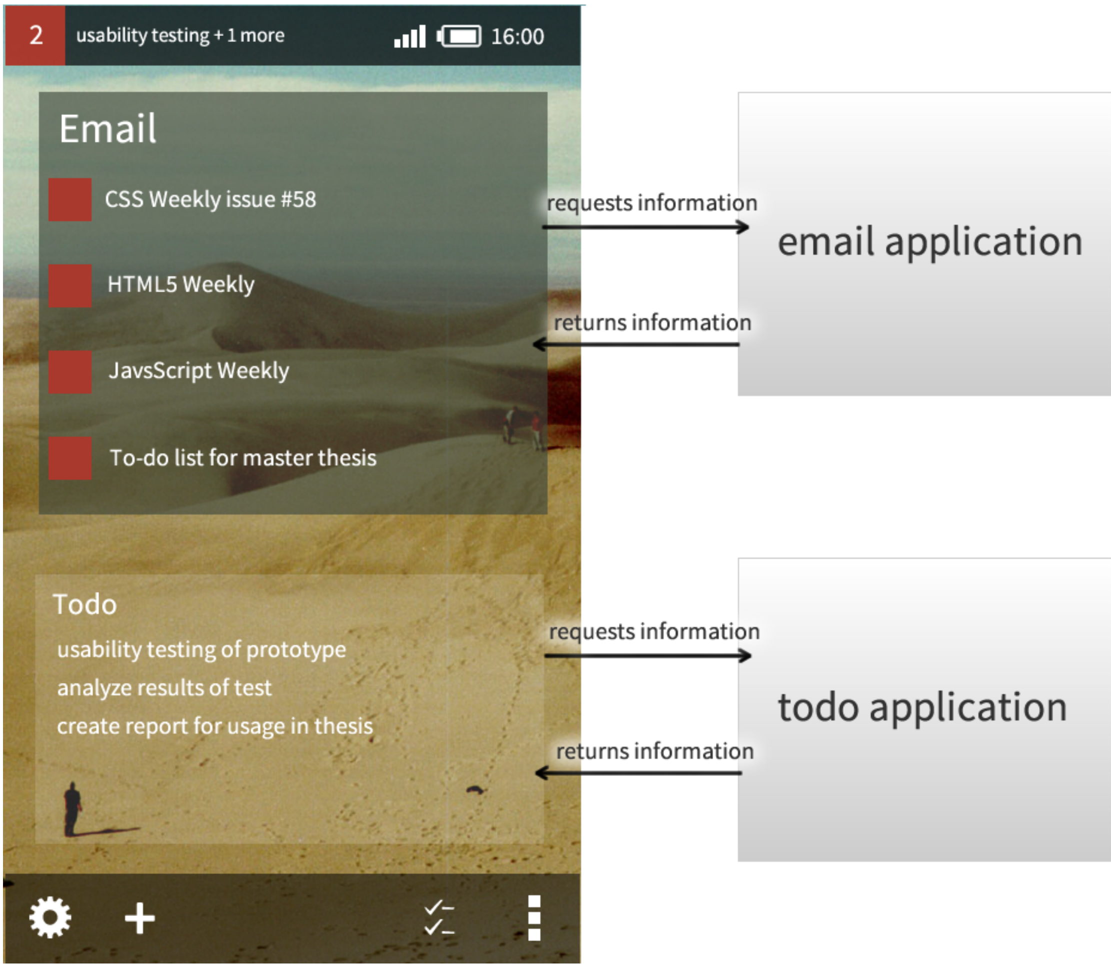

FIGURE 5.5: Actvity Ant 1.

5.1.6 Applications

Applications within AA1 have the responsibility of handling information for it's connected widgets to use. An application, as downloaded from an application store, would have several parts. One part is an activity-shell which contains all the widgets of the application. This way applications could be delivered in the same manner as today, and we could see for instance a Facebook application that while different still looks the same as today. The second part is an API-layer, which handles information for the widgets connected to the application. Widgets request information from their parent application, and the application's underlying API responds with the requested information. Essentially, the applications are responsible for storing and retrieving information.

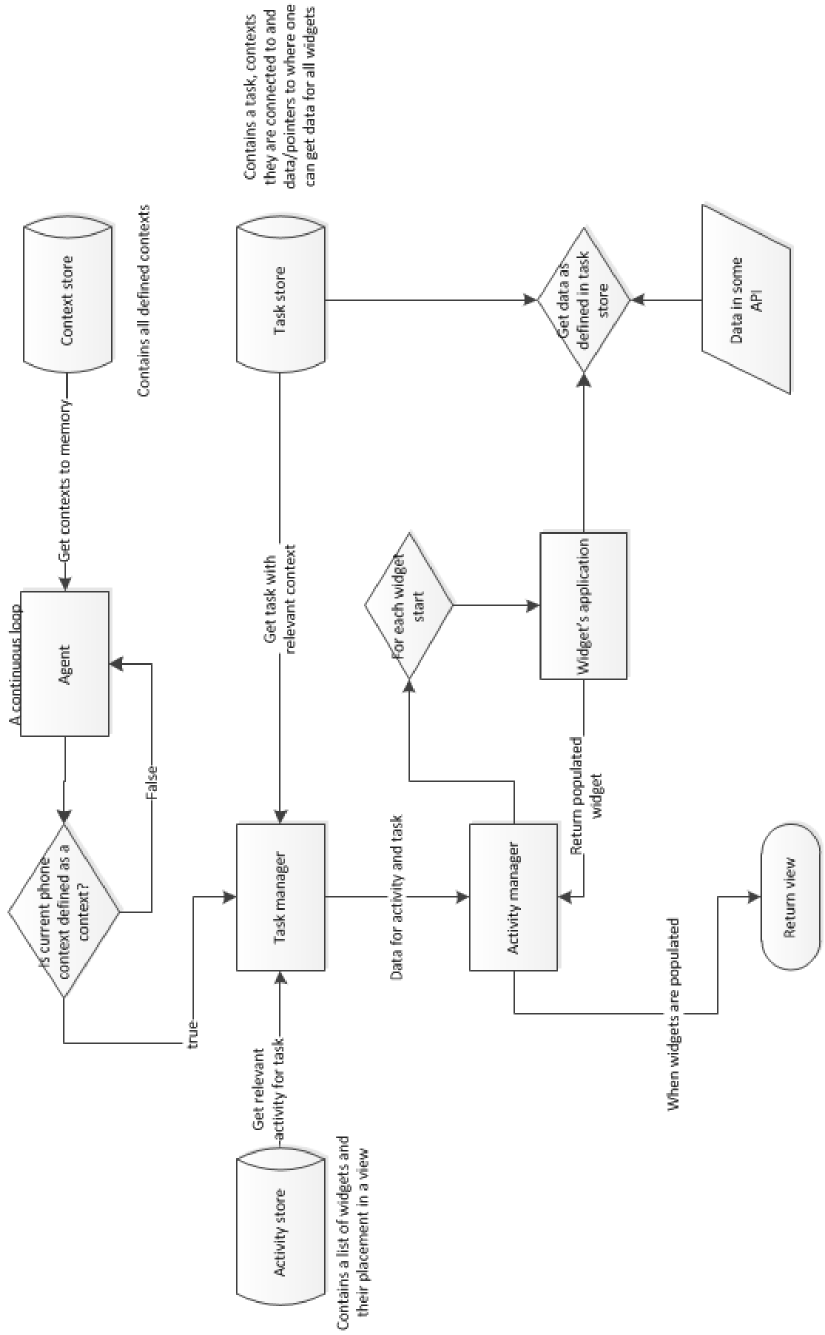

FIGURE 5.6: Activity Ant 1 connection between computational components

5.2 Testing AA1

In order to assess the success of AA1 two tests were performed. The first one, a usability evaluation of a vertical prototype, was performed in order to see users' reaction and to identify usability issues. The second test, in the form of a focus group, was performed in order to gauge users' response to the concepts and in order to get feedback to inform the further design process. This section will present these two tests.

5.2.1 Usability Test

The first test performed on AA1 was a usability test performed by pairs of information science students. The test was performed in order to identify usability problems, and to implicitly see how long it took a technology informed user group to figure out the system.

The prototype tested was vertical in nature, and consisted of cardboard cutouts of screens detailing activity creation and task creation within the AA1 paradigm.

For activity creation two different approaches were tested. The first approach presented the user with a grid in which the user could select sections to place widgets in. The second approach gave the user a set of differently sectioned grids to choose from. This was done in order to test whether the users would prefer the completely free or the semi-constrained approach to placing widgets in a grid.

5.2.1.1 Goals

The main purpose of conducting the test was to see if the participants both (i) understood the concepts and (ii) could see the relevance and usefulness of said concepts in a real world setting. A secondary purpose of conducting the test was to see if my particular implementation of the concepts were (i) understandable and (ii) appropriate for the medium (the medium being smartphones). A third purpose was to test a completely open grid, where users can determine the exact size of each widget, versus a predefined grid, where the user can choose between some predetermined sizes.

Objectives

1. Assess the efficiency and understandability of the proposed interfaces for creating activities and tasks.
2. Identify confusing and unclear screens.
3. Identify unnecessary steps/screens.
4. Gather suggestions and ideas for future development.

5.2.1.2 Test Setup

Since this was an early test more geared towards idea development than actual usability testing, and due to the non-existing budget and the nature of the project, it was decided to draft students as test subjects. The test subjects were all recruited from master level information science. In total 6 subjects were recruited. The test subjects were all highly technically proficient, with experience from both using and creating technology. All the test subjects did at the time of the test own and use Android smartphones.

During the test session the participants were asked to do the following tasks using the cardboard prototype:

1. Create an activity ≪work≫ containing your work to-do list and your work calendar. The to-do list should take up the upper half of your screen and the calendar the bottom half of your screen.
2. Create a recurring task which repeats each morning between 8 and 10am when you arrive at work. This task should be tied to the work activity.

3. Create an activity «shopping» which contains a map and a checklist. The map should show «Rimi»-stores and the checklist should be the previously defined «groceries». The map should take up the upper half of the screen and the checklist the lower half of the screen.

4. Create a recurring task which repeats each saturday morning when close to home. The task should be connected to the «shopping» activity. You want to display all grocery stores, so you will have to override the default map you defined when creating the activity.

The test participants were grouped into pairs. The pairs were asked to cooperate to solve the tasks. In general the "think aloud protocol" was encouraged. More specifically they were asked to openly discuss the screens they were seeing and their thought process when solving the tasks. They were also encouraged to share any relevant ideas, both during the test and afterwards. After going through the test tasks the subjects were debriefed. In the debriefing they were allowed to talk about their general impressions in addition to come with any suggestions for improvements that they might have.

The subjects were filmed during the test, and the films were later transcribed. Each test lasted for approximately one hour.

5.2.1.3 Test Material

The AA1 prototype is in essence a series of interaction diagrams. For the purpose of testing them a series of concrete screens were designed, printed and glued to cardboard. These screens were designed to represent a vertical prototype, each group of screens representing one concrete task that the users were asked to perform during the test.

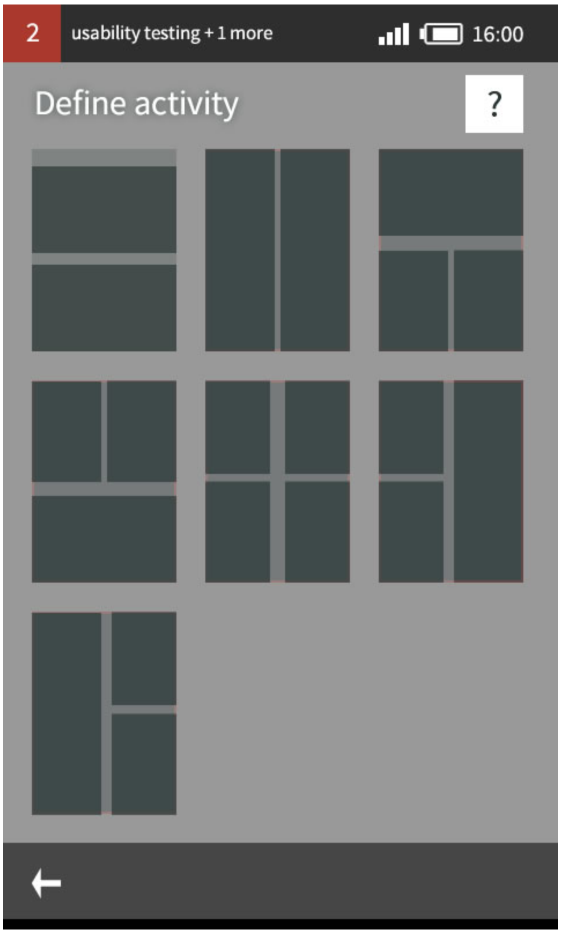

FIGURE 5.7: Choose grid layout in the strict activity creation variant.

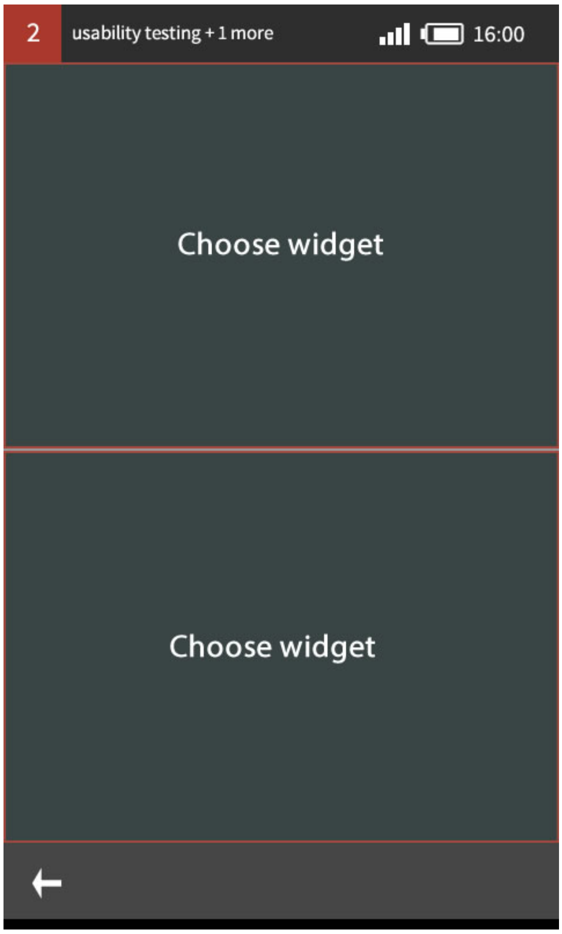

FIGURE 5.8: Choose grid area to place widget in.

5.2.1.4 Results

In general the subjects were able to perform all tasks relatively quickly. The users had relatively few errors, and were able to use the interface without asking many questions. All the three pairs were quicker when using the semi-constrained grid, and when asked answered that they preferred the constrained version. All the three pairs were able to create and configure tasks, but noted that they believed the context creation process, and the underlying logics-based concept, would prove too complex for most users.

In the post-test debriefing conversations all the three pairs questioned whether the concept of placing widgets on a screen would work in real life scenarios. The main argument was that the amount of information snippets would quickly become too large when compared to screen size, resulting in a cluttered and confusing user experience. A second argument was that the amount and complexity of information displayed in widgets would probably be too low for many use cases. One pair noted that the terminology used in the prototype was confusing, and that this severely impacted the usability of the prototype.

5.2.2 Focus Group

In addition to the usability test discussed in the previous section a focus group session was organised in order to discuss the concepts and implications of AA1. The discussion was performed in a session with 4 participants plus one moderator.

5.2.2.1 Goals

The focus group session had several goals: (i) to see if the participants were able to understand the concepts, (ii) explore which uses the participants could imagine, (iii) inquire into whether the participants could imagine themselves using AA1 or a general CATA-based system and (iv) gather feedback and ideas for a new iteration. In order to achieve the goals a set of questions was created:

1. Do users understand the concepts?
2. Which uses do users imagine?
3. Which ideas do users have to improve the concept?
4. Would the participants actually use AA1 if it was included on their smartphone?

5.2.2.2 Test Setup

In total there were five people participating in the session: four participants and one moderator. The four participants were all students. Two participants had backgrounds in computer science, while the other two had backgrounds in the Humanities and social sciences.

The group was introduced to the concepts through some rough sketches and scenarios. It was decided not to use the higher fidelity screens from AA1 in order to avoid discussions regarding irrelevant design choices on the prototype such as for instance icon placements and font choices. After the general introduction the group was allowed to ask questions to clarify anything they did not understand.

Afterwards the group was asked some general questions, along the same lines as the ones above. However, the group was allowed to somewhat dynamically form their own discussion, and interesting topics that were not a part of the original were allowed.

5.2.2.3 Results

In general a few themes were discussed during the session.

Smartphone Usage
One topic discussed was current smartphone usage. Two of the participants had tried to use their tablets for writing articles. However, they both agreed that there were several problems with how their tablets work when trying to research and write articles. The largest problem was touch keyboards, which they felt was not sufficiently accurate and ergonomic for their use. The other limiting factor was research, especially when paraphrasing or citing elements from sources, due to limited functionality for comparing and storing information. The two other participants stated that they mostly used their phones for "the usual things". By that they meant communication through voice calls, text messages and social media applications. They also stated that they often used their phones for surfing the web and to play media resources.

When asked if they could recall any situations in which they use multiple applications, the general consensus was that they had tried to use multiple applications at once, but that they would rather use their laptops. This was due to screen size and how windows can be manipulated.

How would you use AA1?

Another topic discussed was how the focus group participants imagined themselves using AA1 or a similar system. The participants came up with a couple of concrete use cases, one for a pubcrawl and one for new students.

> Siri is going on a pubcrawl with her friends in a foreign town. Together with her friends she has planned the evening, and has placed markers for different bars in a map.
>
> Siri creates an activity consisting of the pub map and a to-do list, where the to-do list is a list of pubs that the group of friends wants to visit during the evening. Siri also creates a set of tasks which are set to trigger at set intervals in order to inform the group that they are going to the next pub. These tasks also contain information about the route to the next pub, which is displayed in the map.

> Lars is a new student, and is unfamiliar with his new university and town.
>
> Lars goes online and finds an activity designed by the university to help new students familiarise themselves with their new surroundings.
>
> Lars opens the activity. Within the activity there is a map with markers for relevant locations to the university. There is also an information tile which displays different information snippets relevant to student life.
>
> When Lars is at a location covered by the activity, the activity changes to display information about the location and about extracurricular activities that takes place at the location.

> Bob likes to exercise. He creates an activity creating tools such as a timing clock, output from a heart sensor and his workout routine.

> Throughout the workout the activity adapts to display information relevant to the exercise at hand. When he goes running the phone displays a heart monitor in addition to speed and a map of the route he has planned to run. While at the gym the activity adapts to display the workout regimen planned, such as the number of repetitions for different exercises.
>
> When Bob finishes his workout, and the phone notices that he has left the workout area, a notification is triggered and a workout logging utility is shown along with statistics from the workout.

What the above scenarios show is that the participants were able to understand some of the possibilities that an AC and CA system has. It also shows that they are able to quickly apply the possibilities to real-world scenarios.

Setting up and finding activities

Another question raised in the session was whether the participants could envision themselves actually setting up activities. To this the participants responded that they were unsure. They did however mention that some users (such as users covered by the scenarios above) would probably be interested in setting up activities.

One participant noted that he was unsure whether he would find AA1 useful. He pointed out that he had access to similar functionality on his smartphone in the form of multi-view and that he had never used this functionality.

The participants also noted that they would like to be able to get pre-made activities from an online repository. They were unsure as to whether they would like to share activities they had set up themselves with others.

5.2.3 General findings

In general the objective results of the user test and the focus group was positive. The test participants were able to use the prototypes without much trouble, and it was clear that they grasped the concepts. The focus group attendees demonstrated that they understood the concepts and potentials through proposing user scenarios.

However, participants in both sessions questioned the usability of having a large amount of information from different sources on one screen. The focus group attendees were also unsure if they themselves would use AA1, they did however state that they saw potential uses for other user groups. One participant stated that ït is like those apps that are nice to have, but that you never use.̈

Chapter 6

Activity Ant 2

This chapter presents the second prototype, Activity Ant 2. The design process that led to this prototype is presented in section 4.1.2.

Activity Ant 2 was a more concrete prototype than AA1, consisting of a set of interactions and screenshots depicting an activity. The general purpose of the prototype was to iterate upon the feedback and findings from AA1. AA2 focuses solely on activities, and does not iterate upon the context-aware aspect of AA1.

When testing AA1 I found that users were generally sceptical to the concept of assembling several information snippets into new meta-applications. They were however positive to the concept of creating PICs with tools and information. This lead to the idea that users might be more interested in creating collections of applications than they were in creating new applications.

Where AA1 aimed to gather information and tools from several sources into one view, AA2 tries to simply gather applications as they are into an easy to navigate subset of applications. AA2 also aims to provide tools for navigating the activities, and functionality for sharing information between applications. The main goal of AA2 is to act as an ACC-based application wrapper that fits right into the current mobile operating systems. In essence AA2 is an effort to introduce activity-centric PIM tool without disrupting the fundamentals of current smart phone operating systems and without requiring a massive level of adjustments by application developers.

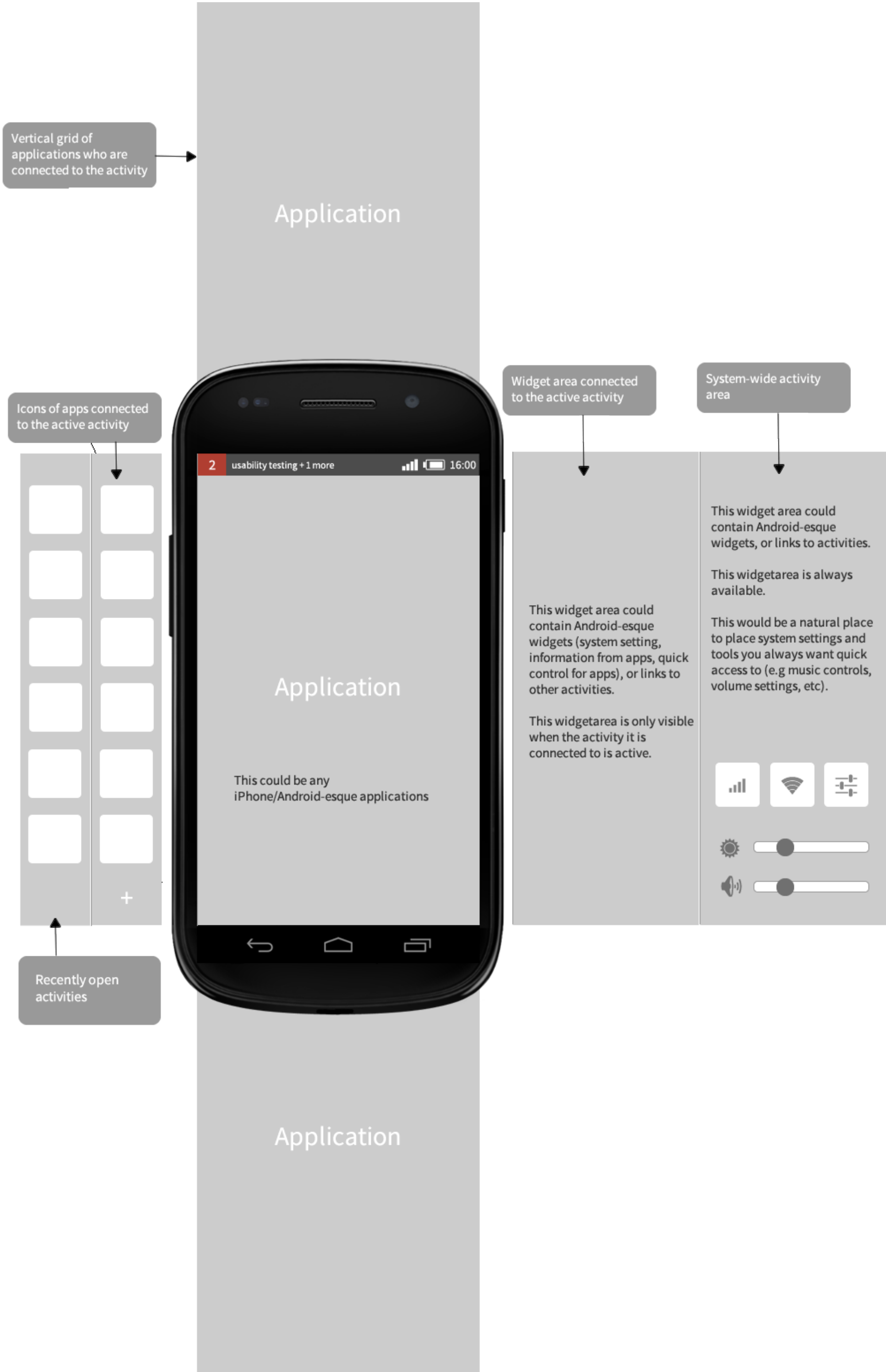

FIGURE 6.1: Activity Ant 2 overview

AA2 consists of several different elements organised in columns, each column with its own specific purpose.

6.0.3.1 The application column

The main column in activities in AA2 is the application column. This columns consists of rows, each row composed of the view of a single application.

All activities in AA2 contain a collection of applications. These applications function independently of each other and are simply started/invoked by the activity (which functions as a wrapper). The applications work as they normally would, controlling their own data scheme and views.

In order to switch between applications a user simply has to use a two-finger scroll gesture, which will result in the column scrolling up or down in order to display applicable views.

6.0.3.2 Application icon columns

To the left of the application column there are two columns of application icons. These serve as a quick navigation tool between different activities and different applications within the currently active activity.

The leftern-most column consists of icons for recently used activities. This is in order to facilitate quick navigation between activities.

The right column consists of icons for the active applications. These are placed chronologically equally to the applications in the application column. Thus the uppern-most icon represents the uppern-most application, etc. This column serves several purposes. The first purpose is quick navigation in large activities; clicking an icon will take the user to the applicable application. The second purpose is to quickly reorganise the activity; through tapping, holding and then dragging the icon the user is able to reorganise the application column. The third purpose is to invoke the dual-application view (described in section 6.0.3.5). The fourth and last purpose is to provide an easy way of sharing information between applications (described in section 6.0.3.4).

6.0.3.3 Widget columns

The widget columns are designed to provide the user with convenience tools. There are two columns; one column connected to the active activity and one system-wide column. Both columns accept application icons and widgets. This means that a user can for instance place icons of frequently used activities in addition to quick system settings, such as a toggle internet connection button, in the system-wide widget column in order to quickly navigate and manipulate the phone. In the same vein a user can place the icons of activities frequently used together in the activity widget bar in order to quickly navigate between related activities.

6.0.3.4 The Clipboard: Sharing information between applications

One factor that users found annoying when using traditional mobile operating systems is the sharing of information between applications. AA2 proposes two solutions: an explicit clipboard and a ”drag and drop” gesture.

When selecting text or other items in a view on traditional mobile operating systems you are presented with the choice to copy the selection. You can then later paste this information into other applications. While sufficient in many cases this does not fulfil usage scenarios when it is interesting to copy multiple selections.

When making a selection in AA2, a clipboard appears. Users can then drag their selection to this clipboard, and have it available until they remove said selection from the clipboard. The clipboard allows several selections to be stored. When the clipboard is empty it disappears.

In AA2 it is also possible to drag a selection to an application icon. The selection will then be sent to the application in question, which will then have to handle the input. This interaction is similar to traditional desktop operating systems

FIGURE 6.2: The Activity Ant 2 clipboard

where one can often drag elements from one application to another, or from an application to the desktop or a folder.

6.0.3.5 The dual application view: Comparing information between applications

One common usage scenario is when one needs access to information from two sources at once. In AA2 you can open two applications within one view. This is done through dragging an application icon onto another in the application column. The two applications will then appear in the same view, and the user can easily compare or share information between these two applications.

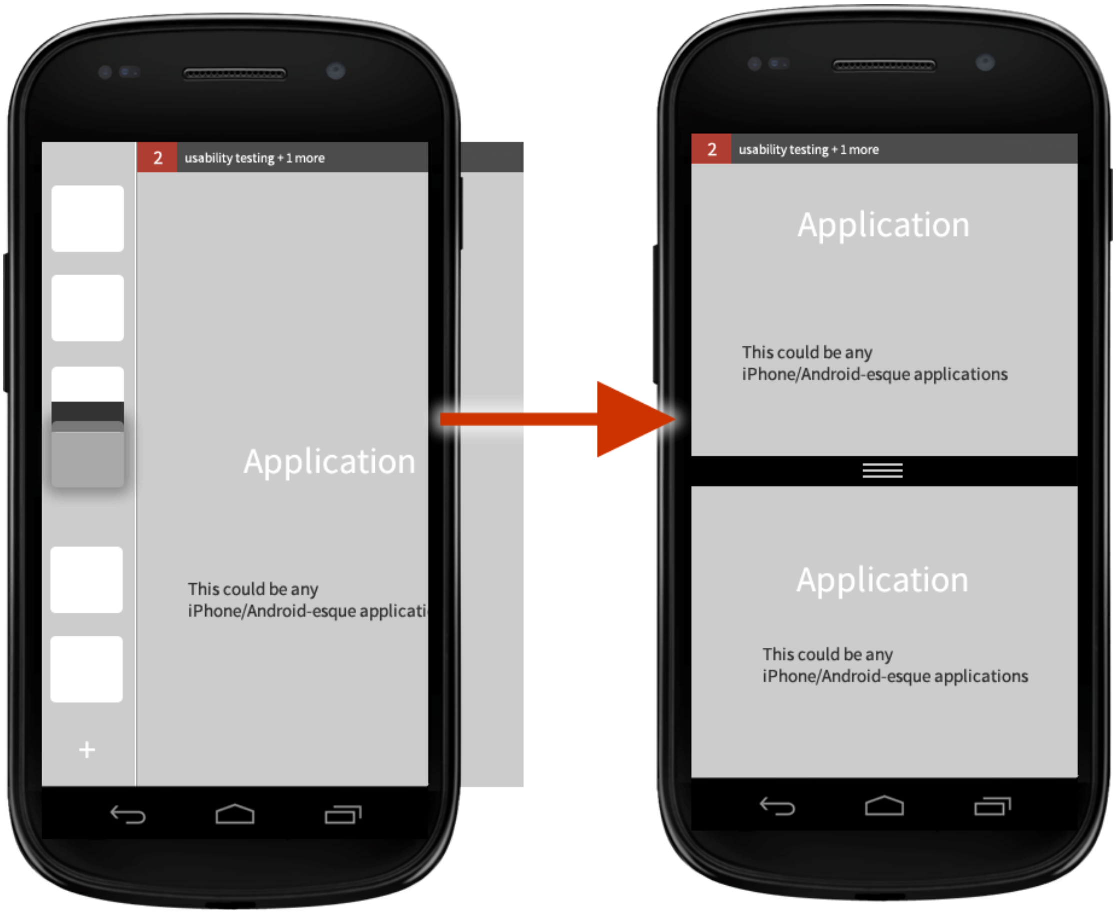

FIGURE 6.3: The Activity Ant 2 multi-view

6.0.4 Test: Focus group

In order to gauge user response and to discuss the concepts and implications of AA2 on mobile usage a focus group session was organised. In this session a total of seven people participated: six students and one moderator.

In order to see some user reactions to the concepts a focus group session was arranged. The group consisted of six participants in addition to a moderator. All of these users were technically inclined users with experience from both software development and modern smartphone operating systems.

A prototype in the form of some computer sketches was presented along with a general explanation of how the different concepts would work in a higher definition prototype.

6.1 Focus Group

In order to gauge users' perception of AA2 and its usefulness a focus group session was conducted. In this session a total of eight persons attended: seven participants and one moderator. The participants were all students with technical backgrounds, and most of them had been a part of either the usability test or focus group session conducted to test AA1.

6.1.1 Goals

The focus group session had several goals: (i) to see if the participants were able to understand the concepts, (ii) explore which uses the participants could imagine, (iii) inquire into whether the participants could imagine themselves using AA2 or a general CATA-based system and (iv) gather feedback and ideas for improvements.

6.1.2 Test Setup

In total seven people participated in the session: six participants and one moderator. The participants were all students with backgrounds in either information science or computer science. Most of the participants had participated either in the focus group session or the usability test of AA1. There are two reasons why these subjects were chosen. The primary reason was availability. The students all had open schedules and were able to participate on short notice. The second reason was experience and technical inclination. All the participants have experience with technology, and most can be considered early adopters of new technology. This was considered to be positive due to the novelty of the concepts of AA2.

The participants were gathered in a room. There they were introduced to AA2 through a series of screens and explanations. After this they were given the opportunity to ask questions in order to fully understand the concepts.

Afterwards the group was asked to discuss some open-ended questions. They were however encouraged to deviate from the concrete questions as long as it was relevant to the prototype at hand. In total the session lasted for about 45 minutes.

The session was recorded using a camera, and the film was later transcribed.

6.1.3 Results

While the session was originally planned to explore the goals presented in section 6.1.1 it quickly turned into a combined brainstorming and discussion session, where several ideas for improving AA2 were proposed and discussed by the participants.

6.1.3.1 Current smartphone usage

During the session the participants were asked to talk about what they currently use their smartphones for. A clear pattern was that most of the participants used their phones for simple tasks, including note taking, text messages, voice calls, social media applications, calendar, music and radio, and web browsing. All of these activities are limited to the scope of one application except note taking, which one of the participants noted that he used as a tool to collect information from other applications.

In general it was abundantly clear that while the participants were all highly technically inclined, they could not explicitly recall any uses of their smartphones outside of the scope of single applications.

6.1.3.2 Information Management in AA2

After presenting AA2, one of the participants raised some highly relevant questions. Would it be possible to pin documents to the application column? This lead to a discussion regarding file management within AA2. Two issues were presented: how can you navigate to files to tie to an activity, and how do you organise documents that are tied to an activity?

One suggestion that was raised was to simply include a file manager. In this way users would be able to navigate to relevant documents and maybe save them to a folder specific to the activity.

6.1.3.3 Ad-Hoc Activities and States

When asked about whether they could imagine themselves actually creating activities an interesting idea was proposed: ad-hoc activities. The general concept is that one from any application would be able to create a temporary activity through adding other applications and tools to the AA2 grid. One concrete use case discussed in the focus group was travel planning, reformulated into the scenario below.

> Linda is on a bus and is reading about travel destinations in a web browser.
>
> She finds a destination, and adds a ticket application to the activity.
>
> She orders tickets, and adds a calendar application to the activity. She adds the details to the calendar application and removes the ticket application.
>
> She then adds a web browser with information about travel destinations, a trip advisor application and a mapping application to the activity. In these applications she looks up and records sights she wants to see.
>
> Before exiting the bus, Linda saves the activity as ”summer vacation planning”, so that she can continue the activity later.

A related issue that was discussed was activity states. In essence when using an activity the different applications within will be in different states. For instance in the scenario above the web browser will have different websites opened, and the map application will have different locations looked up. The state of an application is highly relevant for the action performed. The activity dictates the general action-domain, while the state dictates the specific action being performed. For instance the activity saved in the scenario could be used to plan vacations in different locations, while the actual state of the activity as saved by Linda would be tied to a specific location.

6.1.3.4 Piping

Another interesting point discussed was the possibility of introducing piping to AA2. Piping is a technique used in for instance the Unix shell. In essence piping lets information be processed in a serial manner by a set of processes. Information is sent as input to the first process, the output of this process is then sent to the next process in the chain as input.

The general idea as proposed in the session would be to allow users to create a piping series. When starting a pipe information would automatically be sent to the first application in the pipe-series. When the user is done with treating the information in the current application the resulting information will be sent to the next application in the pipeline. One example, as proposed by a participant would be an picture pipe, reformulated as a scenario below:

> Bob has created a pipeline for pictures, that involves sending pictures through several specialised applications.
>
> Bob takes a photo using his favourite camera application. The application then prompts Bob as to whether he wants to send the photo through a pipe. Bob selects his predefined photography pipeline.
>
> The photo is sent to a cropping application. Bob selects the area he wants to crop, and clicks save.
>
> The cropped photo is sent to an application specialising in fixing red eyes on people in photographies. Bob clicks the fix red-eye button, is content with the result and clicks save.

> The resulting edited photo is then sent to a file manager. Bob selects the folder he wants the edited photo to be saved in. The pipeline is done and exits. Bob is sent back to the camera application.

The piping concept has interesting implication for PIM. With a piping system a user could for instance automatically add information from one activity to another activity.

6.1.3.5 General feeling

The participants in the focus group had a good grasp of what activity-centric computing is, and which possibilities the AA2 system has. They were also able to propose some concrete uses that they would have for the system. The group was split in their view as to whether they could imagine themselves using AA2, but they generally agreed that it would be a nice feature to have on their phones. The ones who did not currently see a need for the system did note that they would have to try the system in action over time in order to properly conclude, and that they could see that others might have uses for the system.

Chapter 7

Results and Discussion

This chapter will discuss the findings and present some general thoughts about ACC's and CA's potential to advance mobile PIM.

In general the goal of this study has been to take some first steps into exploring how ACC can be implemented in a mobile scenario, and to gather knowledge about this process and the potential of ACC to advance mobile PIM.

As the design process advanced it became increasingly clear to me how scenarios can be used to present ideas. The possibly most surprising thing was that my test-participants and discussion-partners were in accord to this idea to the degree that they unprompted used scenarios to convey ideas and to test their understanding of the concepts discussed. This epiphany has lead to scenarios having a central role throughout the study, being used both to provide examples and to present results.

7.1 Research question 1

Are there problems or unfulfilled potential with current mobile personal information mangament systems?

In order to answer this question I performed a literature review, where I looked at smartphone usage patterns, and the current state of the art of the smartphone industry. The ideal PIM system would allow users to have access to the right information at the right time in the right format and complexity. This is true for smartphones in most of the current quick 4-second interactions where users are simply checking a single application for a single piece of information. It is also true when users are for instance using a single application for its intended usage. However it is not true when users need to compare information from several sources or use tools from different applications. This is a problem on WIMP systems with large screens where users can have multiple applications open at the same time on the same screen. It does not work as well in a mobile context where only one application is open at a time. In essence the current generation of smartphones is application-centric to the degree that it is nearly impossible to compare information from two sources without using some form of note application, which in turn constitutes a poor man's database.

Some efforts have been made in order to alleviate this precarious situation. For instance the multi-view application from Samsung, that lets users open two applications at the same time. Another example is the small-apps system from Sony that lets users pin simple applications, such as a notepad, to the front of the view while using other applications. A third example is the "send to" functionality found in iOS and Android which lets users open some information in relevant applications. Another effort to solve the information fragmentation problem is Google Now which tries to provide relevant information for the context of the user. Yet another effort is Google Voice and Siri, which tries to search information.

Some specific PIC scenarios are implemented in modern smartphones. For instance on Windows Phone 8 a quick overview of the status of different applications is provided. However, in most instances, the level of information fragmentation is almost complete. There is no way to assemble information from comparable sources on a phone unless the application creators have specifically allowed it.

Summarised, the current generation of smartphones works well for single application usage scenarios. Information is however tied to applications to the degree that each application in essence constitutes a walled garden of information.

7.2 Research question 2

Can activity centric computing be used in order to fulfil potentials or solve problems with current mobile PIM systems?

The major problem with the current generation of smartphones is information and tool fragmentation. This is caused by how information is strongly tied to their ”parent” applications, and how smartphones have separated themselves from giving users access to handle documents. Through employing concepts from ACC this artificial barrier between comparable information is removed. Instead of having to rely on application developers or smartphone developers to create suiting PICs for us, we can instead create them ourselves based on our exact needs.

As we can see from AA1 and AA2 activity-centrism can be implemented in many ways, each with its own positives and negatives. For instance AA1 is suited to fit scenarios where we need many small information snippets from different sources at the same time, making it suited for the typical quick interactions discussed in section 2.2.3. On the other hand AA2 provides quick access to many different fully-fledged applications through grouping them together, making it more suited to situations where users need to work with the information.

7.3 Research question 3

How can activity-centric computing be implemented in current mobile operating systems?

As shown by the CATA-framework and its relation to current functionality in smartphone operating systems, it is not necessarily a large step to go from the theoretical implementation proposed in this thesis to a functioning prototype in the Android operating system.

There is a correlation between functionality needed and existing functionality. For instance agents can be implemented through running system services on Android, while contexts are already present through output from built-in sensors. The largest difficulties are tasks and activities. As seen in AA2 the introduction of activities do not necessarily imply that current applications need to be rewritten. However, in order to implement tasks, applications needs to be able to receive and compute specific states. This in turn would for most applications require at best some new functionality and at worst a complete rewrite depending on its internal structure.

Through this thesis I present two alternative implementations, each with its own difficulties from a developer perspective. AA1 is most likely the alternative introducing the most problems for developers. Using AA1 as a paradigm for application development would imply a fundamental rewrite of every piece of software for a platform. In essence every application would have to be rewritten to work like web APIs, as opposed to today where everything is handled within a single instance of an application.

While less difficult to implement, AA2 still poses its own problems. The largest problem is probably memory handling. For large activities a large amount of applications would have to run, or be able to start running, quickly. I do however not believe that this would be a large problem, due to the fact that iOS and Android already have measures in place to solve this exact problem.

Both AA1 and AA2 are both based on moulding the front-end part of the mobile stack to allow users to collect information from several applications. This was done in order to propose solutions that fit into the current state-of-art. However, a potentially better solution would be to explore how one could use structured or semi-structured data along with specialised middleware and views in order to let any application access any information that is stored in the system. While

AA1 and AA2 lets users assemble parts to create a computational activity, semi-structured data would potentially allow applications to gather and compare data. The difference is illustrated in the scenarios below.

> Tom structures his life through two calendars. In Google Calendar he adds information about lectures, job interviews and other work-related activities. In his Facebook calendar he stores information about social happenings.
>
> Tom has a GUI-based activity centric system on his phone, and creates an activity containing both calendars, so that he can easily compare them, and use information from both calendars.

> Tom's girlfriend Triss has the same way of organising her calendars.
>
> She has a phone with an ACC system that utilises semi-structured data. Triss creates a calendar activity using a calendar widget. The calendar widget sees that there are events from several sources stored on the phone, and asks Triss which sources she wishes to display. Triss chooses Google Calendar and Facebook. The calendar activity gets all the events, and presents them in a unified GUI.

In the first scenario Tom has to compare the two calendars himself, while Triss' phone does the same operation for her. Changing to a semi-structured data layer would however force most application developers to rewrite their applications. It would also mean that services that earn money through handling information might lose their grasp over their users' information. This in turn could potentially lead to these vendors abandoning a platform.

Summary

The functionality needed to implement an ACC system in the style of AA1 and AA2 already exists, and is already in place on smartphones. However ACC systems at a deeper level than application wrappers would probably require a rewrite of the entire information and application layer of modern smartphones.

7.4 Research question 4

How can the concepts and potentials of activity-centric computing be conveyed to users?

One interesting phenomenon that was encountered when discussing ACC and mobile PIM with people was that they were generally able to grasp the ideas quickly, especially when coupled with a scenario. Another interesting phenomenon was that people were generally able to create very concrete user scenarios in a matter of minutes after first having been introduced to the concepts. This phenomenon was also present within the usability test and the two focus group sessions performed. All of the participants seemed to quickly understand concepts, and were in many cases able to suggest concrete improvements or pinpoint negatives.

In the usability test of AA1 test subjects seemed to prefer the stricter, less configurable, grid when creating activities. They also noted that the context creation process would quickly become confusing, due to the logical structure used to combine context parameters. This might indicate that users would prefer to have mostly predefined options to choose from as opposed to creating grids and combining contexts completely freely. I would however still argue that the free-form grid and the context creation process would be suited to expert users who has needs outside of the ordinary.

Participants were however often unable to envision themselves using AA1 or AA2 in their daily lives. This may be due to how most smartphone interactions are simply quick information checks or simple communication, and that the participants are unable to imagine themselves using their devices for other tasks. Yet another reason might be the current input mechanisms for phones, which two of the participants in the first focus group session cited as a reason for why they did not use their mobile devices for content creation.

Another issue that was discussed in the first focus group was activity creation overhead. Both prototypes provided that the users would set up activities before using them. This was troublesome in two aspects. Firstly this would require users to think through exactly what tools and information that would be needed to perform some action on beforehand. This in turn makes it impossible to incremently add elements to an activity if such elements are deemed needed. Secondly this makes it hard to create quick activities to solve a problem that is only relevant in one instance. A possible solution to this was proposed by a participant in the second focus group. Instead of creating activities beforehand the suggestion was to facilitate "ad-hoc activities". This would be done through allowing any application to be transformed into an activity. The "ad-hoc activities" could afterwards either remain unsaved, and thus disappear after the user is done with them, or be saved in case the user wants quick future access to them.

Chapter 8

Conclusion

This thesis has looked at how activity-centric computing and context awareness can help advance the state of mobile personal information management. In order to explore this problem, and in order to contribute knowledge, I have tried to answer the following four research questions:

1. Are there problems or unfulfilled potential with current mobile personal information mangament systems?

2. Can activity centric computing be used in order to fulfil potentials or solve problems with current mobile PIM systems?

3. How can activity-centric computing be implemented in current mobile operating systems?

4. How can the concepts and potentials of activity-centric computing be conveyed to users?

In order to explore the first question a literature review, looking at smartphone usage patterns and the current state of the art, was performed. The third question was explored through creating a framework detailing the functionality needed to implement a form of activity-centric computing and mapping the parts to existing functionality in the current generation of smartphones. The second and fourth question were explored by creating two prototypes and discussing their provess in focus groups.

The motivation for conducting this research was a personal frustration with how difficult it can be to handle information on my Android-based smartphone. It has not been a goal to create a fully functioning system, but rather to propose a first step towards a better smartphone user experience.

8.0.1 Findings

The CATA-framework, AA1 and AA2 constitutes a concrete suggestion for implementing ACC on top of the current generation of smartphones. The two prototypes presents two distinct solutions to creating personal information collections.

In Activity Ant 1 a user can collect information snippets, called "widgets", from several applications and reconnect them into a new "meta-applications" named activities. Activity Ant 1 also lets users configure states for activities, called tasks, and contexts in which activities should start in these states. Users were able to understand how to define activities, tasks and contexts, but were generally sceptical to the concept itself.

In Activity Ant 2 users can collect and create groups of applications. Activity Ant 2 also provides a frame around the application collections with functionality for information sharing and comparison, in the form of a clipboard and the ability to view several applications at once. Users generally seemed positive to the concept.

All tests and sessions indicate that the subjects in question comprehend the concept of activity-centrism and context-awareness. They were also able to create scenarios depicting practical uses.

Users were unsure about whether they would use the features presented in the prototypes if included on their phones. As one focus group subject noted ït is like those apps that are nice to have, but that you never use: It is however not possible from the data gathered to conclude that this would in fact be the result of introducing activity-centrism, and I do not expect that a definitive answer can be found, except through actually implementing it and testing it on a large sample over a longer time period.

8.1 Limitations of study

This study was conducted as a design science research project, where the main goal was to explore how ACC and CA can influence mobile PIM. As is the case in many design science projects the wast majority of time, effort and resources was spent on forming and expressing novel ideas through prototypes. This left little time left for planning, organising and conducting tests. So, while the participants in the usability test and the focus group sessions generally seemed to find ACC intuitive and potentially useful, it must be noted that they represent a very limited sample. All of the participants were Norwegian students in their twenties, and the majority were from a technical background. All of the participants were also smartphone users. This might have skewed the results in favour of the concepts tested.

In order to properly test the concepts and implementations proposed in this thesis a larger study would have to be conducted. This study would have to include a representative sample of potential users, a prototype at a much higher fidelity, and testing over a larger timespan.

What this thesis constitutes is an early foray into reifying the abstract concepts of ACC and CA in order to advance mobile PIM from an HCI perspective.

8.2 Reflections

Working through a design science process has been an educational and inspiring process. Through the process I have become increasingly aware of the strengths and weaknesses of low-fidelity prototypes, focus groups and usability tests.

What I see as the inherent strength of low-fidelity prototype is how they allow abstract ideas to be made more tangible. This in turn lets us discuss the said ideas with others, so that we together can evolve the idea to a better one. In total I was able to create two low-fidelity prototypes and was able to present and discuss these with others. This has left me with many new abstract ideas that I did not have time to concretise into prototypes.

I spent a lot of time preparing for, conducting, and analysing data from a usability test. In retrospect I see that I would probably have been better served with simply conducting a focus group session and instead allocate more resources to go through more prototype iterations. This would have let me explore more ideas instead of identifying what was essentially irrelevant usability issues in one prototype.

8.3 Future Research

In this thesis I have been limited to creating low-fidelity prototypes and testing these through discussions in focus groups. In order to really test the prowess of activity-centrism on smartphones a larger and lengthier study would have to be performed. It would be interesting to see how the usage of smartphones would evolve with a properly implemented system.

While I have focused on the personal information management aspect of phones, there is also a collaborative aspect to human activities. It would be very interesting to see how the prototypes could be expanded to explore how activity-centric computing and context awareness could impact how phones helps us collaborate.

Another aspect that would be interesting to explore is how we could separate stored data from applications, and if this could impact how applications are designed both from a computer science and human-computer interaction perspective.

Bibliography

Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R. & Pinkerton, M. (1997, October). Cyberguide: A mobile context aware tour guide. Springer Netherlands. doi:10.1023/A:1019194325861

Balakrishnan, A. D., Matthews, T. & Moran, T. P. (2010, April). Fitting an activity-centric system into an ecology of workplace tools. In *Proceedings of the 28th international conference on human factors in computing systems - chi '10* (p. 787). New York, New York, USA: ACM Press. doi:10.1145/1753326.1753441

Bardram, J. E. (2007). From Desktop Task Management to Ubiquitous Activity-Based Computing. In V. Kaptelinin & M. Czerwinski (Eds.), *Beyond the desktop metaphor: designing integrated digital work environments* (pp. 223–259). The MIT Press. doi:10.1080/10447310802664998

Boardman, R. (2004). *Improving Tool Support for Personal Information Management* (Doctoral dissertation, Intelligent, Interactive Systems Group, Dept. of Electronic and Electrical Engineering). Retrieved from http://www.iis.ee.ic.ac.uk/~rick/thesis/boardman04-thesis.pdf

Card, S. K. & Henderson, A. (1987, May). A multiple, virtual-workspace interface to support user task switching. In *Proceedings of the sigchi/gi conference on human factors in computing systems and graphics interface - chi '87* (Vol. 18, *4*, pp. 53–59). New York, New York, USA: ACM Press. doi:10.1145/29933.30860

Chen, G. & Kotz, D. (2000). A Survey of Context-Aware Mobile Computing Research. *Time*, *3755*(TR2000-381), 1–16. doi:10.1.1.140.3131

Christensen, H. & Bardram, J. (2002, September). Supporting Human Activities - Exploring Activity-Centered Computing. In G. Borriello & L. E. Holmquist (Eds.), *Ubicomp 2002: ubiquitous computing* (Vol. 2498, pp. 37–62). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/3-540-45809-3

Dey, A. & Abowd, G. (2000, January). *CybreMinder: A Context-Aware System for Supporting Reminders* (P. Thomas & H.-W. Gellersen, Eds.). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/3-540-39959-3

Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R. & Estrin, D. (2010, June). Diversity in smartphone usage. In *Proceedings of the 8th international conference on mobile systems, applications, and services - mobisys '10* (p. 179). New York, New York, USA: ACM Press. doi:10.1145/1814433.1814453

González, V. M. & Mark, G. (2004, April). "Constant, constant, multi-tasking craziness". In *Proceedings of the 2004 conference on human factors in computing systems - chi '04* (pp. 113–120). New York, New York, USA: ACM Press. doi:10.1145/985692.985707

Google. (2012). Google Now. Retrieved February 28, 2013, from http://www.google.com/landing/now/

Grønli, T. & Ghinea, G. (2009). Web 2.0 integration in a context-aware mobile PIM application. *Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for*, 1–2. Retrieved from http://ieeexplore.ieee.org/xpl/freeabs%5C_all.jsp?arnumber=5402613

Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. doi:10.2307/249422

IDC. (2013). Android and iOS Combine for 91.1% of the Worldwide Smartphone OS Market in 4Q12 and 87.6% for the Year, According to IDC - prUS23946013. Retrieved February 28, 2013, from http://www.idc.com/getdoc.jsp?containerId=prUS23946013%5C#.US98OTAXFQA

Jones, W. (2008, October). Personal Information Management. *Annual Review of Information Science and Technology*, *41*(1), 453–504. doi:10.1002/aris.2007.1440410117

Kaptelinin, V. & Boardman, R. (2007). Toward Integrated Work Environments: Application-Centric versus Workspace-Level Design. In V. Kaptelinin & M. Czerwinski (Eds.), *Beyond the desktop metaphor: designing integrated digital work environments* (pp. 295–331). The MIT Press. doi:10.1080/10447310802664998

Kaptelinin, V. & Nardi, B. A. (2006). *Acting with technology: Activity theory and interaction design*. The MIT Press.

Karger, D. R. (2007). Haystack: Per-User Information Environments Based on Semistructured Data. In V. Kaptelinin & M. Czerwinski (Eds.), *Beyond the desktop metaphor: designing integrated digital work environments* (pp. 49–99). The MIT Press. doi:10.1080/10447310802664998

Karikoski, J. & Soikkeli, T. (2011, December). Contextual usage patterns in smartphone communication services. *Personal and Ubiquitous Computing*, *17*(3), 491–502. doi:10.1007/s00779-011-0503-0

KDE. (2012). Plasma/Active - KDE Community Wiki. Retrieved April 23, 2013, from http://community.kde.org/Plasma/Active

Law, E. L.-C. (2011, June). The measurability and predictability of user experience. In *Proceedings of the 3rd acm sigchi symposium on engineering interactive computing systems - eics '11* (p. 1). New York, New York, USA: ACM Press. doi:10.1145/1996461.1996485

Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P. & Kort, J. (2009, April). Understanding, scoping and defining user experience. In *Proceedings of the 27th international conference on human factors in computing systems - chi 09* (p. 719). New York, New York, USA: ACM Press. doi:10.1145/1518701.1518813

Müller, D., Law, E. & Strohmeier, S. (2010). Analysis of the Persuasiveness of User Experience Feedback on a Virtual Learning Environment. Retrieved from http://ceur-ws.org/Vol-656/I-UxSED2010-Proceedings-Complete.pdf%5C#page=36

Øvrebø, M. (n.d.). *No Title*, University of Bergen.

Robertson, G., Smith, G., Meyers, B., Baudisch, P., Czerwinski, M., Horvitz, E., … Tan, D. (2007). Explorations in Task Management on the Desktop. In V. Kaptelinin & M. Czerwinski (Eds.), *Beyond the desktop metaphor: designing integrated digital work environments* (pp. 103–138). The MIT Press. doi:10.1080/10447310802664998

Rubin, J. & Chisnell, D. (2008). *Handbook of Usability Testing: Howto Plan, Design, and Conduct Effective Tests*. Wiley.

Sharp, H., Rogers, Y. & Preece, J. (2007). *Interaction Design: Beyond Human-Computer Interaction* (2nd editio). Wiley.

Zimmerman, J., Forlizzi, J. & Evenson, S. (2007, April). Research through design as a method for interaction design research in HCI. In *Proceedings of the sigchi conference on human factors in computing systems - chi '07* (p. 493). New York, New York, USA: ACM Press. doi:10.1145/1240624.1240704