Continuity of Service in Design for a Specific Platform
Combining service- and interaction design perspectives in a multiple platform environment

Linus Nilsson
2006-11-24
LIU-KOGVET-D--06/20--SE

Master thesis in Cognitive Science
Linköping University, Department of Computer and Information Science

Tutor: Stefan Holmlid
Acknowledgements

First of all I want to thank Marina Björnberg for telling me to apply for a thesis work at the office where it all happens instead of just talking to everyone about one of my favorite inventions.

Then I would like to thank Peter Erhard for accompanying me across the sea for this thesis adventure and I would like to thank Rodrigo Madanes, Duncan Lamb and Kristjan Jansen for giving us the chance to do so.

I also would like to thank my academic tutor Stefan Holmlid for his patience with my impatience.

And finally I would like to thank Emma Persson for her endless support, the task being academic or just to cheer me up.

Linus Nilsson

Linköping, September 2006
Abstract
This thesis presents a design work in which one platform specific User Interface (UI) is redesigned to fit into the multiple platform range of a software application. Among other design methods a service design analysis and a platform general UI framework are used. The thesis aims to study the impact of these methods.

The service design analysis starts with a workshop which outcome is used as reference point during the design process and later in the analysis of the results. The general framework is extracted from existing mobile implementations of the software and is used as a guideline in the design of the platform specific solution.

The results presented show that the service design perspective added high level priority tools to the process, and there are indications that this method can be even more valuable in earlier stages of adaptation of software to new platforms. The general framework is shown to have contributed on a more detailed level and there is indication that this method can facilitate the work of obtaining continuity between platforms.
1 Continuity of Service in Design for a Specific Platform.......................... 1
  1.1 Scenery of the presented work.................................................................. 1
  1.2 Focus of the thesis...................................................................................... 2
    1.2.1 Research questions .................................................................................... 2
  1.3 Intended reader.......................................................................................... 2

2 Theoretical background ............................................................................3
  2.1 Service design ............................................................................................. 3
    2.1.1 Service Package.......................................................................................... 3
    2.1.2 Service concept.......................................................................................... 4
  2.2 Interaction Design ...................................................................................... 5
    2.2.1 Inter-usability.............................................................................................. 5
    2.2.2 Multiple-User Interface ................................................................................. 5
    2.2.3 Graceful degradation..................................................................................... 6
  2.3 Design methods .......................................................................................... 7
    2.3.1 Service design.............................................................................................. 7
    2.3.2 Interaction design ........................................................................................ 8

3 Method .....................................................................................................10
  3.1 Identification of the service offer............................................................ 10
  3.2 Extraction of a general framework ........................................................ 11
    3.2.1 Identification of features of current interest...................................................... 11
    3.2.2 Division of analysis into levels................................................................. 12
    3.2.3 Division of platforms into classes.............................................................. 12
    3.2.4 Walkthrough of existing implementations.............................................. 12
  3.3 Design of a specific solution .................................................................... 13

4 Results ......................................................................................................14
  4.1 The identified service offer...................................................................... 14
    4.1.1 Communication features – overview ...................................................... 14
    4.1.2 Service concept – overview ................................................................. 15
    4.1.3 Core services.............................................................................................. 15
    4.1.4 Peripheral services..................................................................................... 16
    4.1.5 Support services.......................................................................................... 17
  4.2 General and specific design solutions..................................................... 18
    4.2.1 Making a voice call: Core service and focus on Physical interaction........ 18
    4.2.2 Create user account; Peripheral service and focus on Flow/Steps.......... 21
    4.2.3 Contacts list: Support service and focus on Graphics/Screen size....... 24
  4.3 User testing of the specific design solution............................................. 25

5 Discussion ................................................................................................27
  5.1 Discussion of methods.............................................................................. 27
    5.1.1 The service design workshop................................................................. 27
    5.1.2 Extraction of the general framework ................................................... 27
  5.2 Discussion of results.................................................................................. 29
1 Continuity of Service in Design for a Specific Platform

The availability of small mobile computing devices is increasing in today’s society. With this follows that mobile versions of former desktop applications can now be accessed by a larger public than before, and cross platform user experience gets more and more important. A platform is in this thesis defined as the combination of hardware and operating system.

A service provider wants her clients to be able to access her service from anywhere, and to migrate software applications to as many platforms as possible is a way to accomplish this. While doing this platform migration the service provider wants to ensure that the service stays one and the same so the clients feel at home with it. For the sake of interaction design and usability it is a question of preserving the user experience when the conditions are changing, and sometimes changing drastically.

One target platform for service providers is mobile devices with their own operating system, such as mobile phones, PDA’s (personal data assistant) and Pocket PC’s. These devices have their own design and interaction models, into which any third party application must blend to create a smooth user experience.

As interaction designer there are then constraints coming from both directions, and they are potentially conflicting. The task is to design the User Interface (UI) of the application in such a way that the user can make use of her experience both from using the specific application on other platforms and from using other applications on the specific platform.

A first step to take in order to achieve this is to address the software side of the problem and make sure that there is a core of consistency between the different implementations of the application. The thesis aims to describe a design process at this stage. The work presented has used a service design perspective. Concepts from this discipline are used as tools to compliment a traditional interaction design approach.

1.1 Scenery of the presented work

The over all scenery for the design work presented in this thesis is that of a communication software company. The software began as a voice call over internet application and does now feature video calls and Instant Messaging (IM), text messaging over GSM, etc.

The actual software was first made for desktop use, and developed for one platform only. Then followed other desktop platforms and today the software has been implemented on different mobile devices already, and more are to come. There is a need to unify the designs for future development.
In order to address this issue a project was initiated in which existing mobile implementations were to be examined cross wise to see to what extent they differed from each others and to determine in what direction the design should move in the future.

Along side with this project one of the mobile versions was up for a redesign. Among other factors new platform guidelines was a driver. As participant in both projects I was able to use the outcome of the unifying project in the work of the redesign.

1.2 Focus of the thesis

This thesis presents a design work in which a platform specific UI is being redesigned for a multiple platform application. Among other design methods a service design analysis and a platform generic UI framework are used. The thesis aims to study the impact of these methods on the design process.

1.2.1 Research questions

Research questions that the thesis aims to answer are:

- How can a service design perspective add value to an interaction design process of designing UIs for multiple platform software?
- In a situation where there are existing products and a new one is to be designed, how can a design process be driven?

1.3 Intended reader

This thesis is written as a final thesis for the Cognitive Science programme at Linköping University. The intended reader has a similar background. This includes experience within the human-computer interaction field and familiarity with interaction design and design methods. For readers who want to get an overview of interaction design, see for example Löwgren and Stolterman (2004) and for design methods see Jones (1992).

I would also consider it beneficent to have had some contact with the field of service design, but it is not crucial for the understanding of this thesis. I will use a limited set of concepts from this discipline, which will be presented in the next chapter. For a broader introduction to service design, see for example Edvardsson, Gustafsson, Johnson, Sandén (2000).
2 Theoretical background

In this section some key terms and research are described. A presentation of relevant service design concepts is followed by interaction design oriented approaches of how to address cross platform functionality.

2.1 Service design

In this section I introduce the service design concept of a Service Package and a Service Concept. These theories will be used in the work presented in this thesis.

2.1.1 Service Package

Grönroos (1990) presents what he calls the basic service package. He takes on a management perspective and states that for this purpose it is important to divide the service offer into three different groups of services: core, facilitating and supporting services.

Grönroos (1990) defines the core service as the reason for being on the market and exemplifies with the transportation for an airline company. He also states that a service offer may include multiple core services, which in the airline example would be both shuttle service and long-distance transportation.

The facilitating services are defined as ways to enable the consumption of the core service and Grönroos (1990) continues his airline example by describing the check-in services as such required services.

The third group of services contributing to the basic service package, the supporting services, do not support the core services, but the experience of using the service as a whole (ibid). According to the author the point is to increase the perceived value of the service and might also be used to differentiate the offer from competitors on the market. Grönroos finish his airline example by describing in flight services such as food serving as a supporting service.

Grönroos (1990) continues his description of the basic service package with a discussion concerning the differences between the facilitating and the supporting services. He states that it is not always evident what is what, and exemplifies this with a recap of the in-flight meal; on a short flight this might be viewed as a supporting service while it is rather a facilitating service on a long-distance flight.

Despite, or maybe because of this situation, the author claims that from a managing point of view it is important to separate the two service bundles. The reason lies in the definition: facilitating services enables the core service and are therefore mandatory. The implementation of them might very well contribute as a mean of competition as well, but the difference lies in that the supporting services are there as a mean for competition only (ibid).
A service package is rarely independent of physical products, as an example you need a telephone to use telephone services. This is called a product-service system and has been defined as “a marketable set of products and services capable of jointly fulfilling a user’s need” by Goedkoop, van Halen, te Riele and Rommens (in Mont, 2002). This definition makes it interesting to use a service design perspective even though the subject of interest is not a purely service offer.

2.1.2 Service concept
Edvardsson (1996) uses Grönroos (1990) model of the basic service package in his description of the service concept. Edvardsson follows a quality perspective and states that in order to obtain a high quality service the different needs of the customer should map to the different levels in the service offer.

He defines the primary needs as the origin of the customer’s wish, and the secondary as needs that emerges once a primary choice of how to satisfy the wish has been made. Edvardsson (1996) exemplifies this with a person that wants to get in touch with someone in a foreign country. In the example the person chooses the phone as her way of communication, which is then viewed as the core service. Secondary needs that have to be met then become such things as getting the persons phone number, and if abroad the country code prefix.

In his model Edvardsson calls the facilitating services peripheral services, this label will be used in the rest of this thesis.

![Figure 1. Model of the service concept.](source: Edvardsson 1996 p. 82.)
2.2 Interaction Design

In this section I present the concepts of Inter-usability, Multiple-User Interfaces and Graceful Degradation. These theories will act as references to the work presented in this thesis.

2.2.1 Inter-usability

Denis and Karsenty (2004) states that while studying usability on mobile devices can improve the user experience on each and single device in a multi device system, there is a need to look into the transitions between devices to make the whole system work.

The same authors compare the ease of use when switching between devices in a multi device system with the technical concept of inter-operability. Inter-operability is defined as the possibility of using software on different computer platforms, and Denis and Karsenty want to introduce the term inter-usability to cover the usability part of that concept. When switching between devices in a multi device system the user should be able to reuse his or her knowledge of available functions and how tasks are performed. The ease of these transitions is what the authors want to call inter-usability.

To achieve good inter-usability Denis and Karsenty (2004) claims that a service must give the user a feel of continuity in two dimensions: knowledge and task.

The knowledge dimension deals with the static enablers of the service – what functions are available on this particular device and how do I perform tasks considering the available ways of interaction?

The task dimension concerns the performing of series of tasks from different devices. I start a task on one device and finish it on another, or I make changes to a task carried out on another device – what was it I was doing, and what is the current status?

I would like to exemplify good inter-usability with the media player application iTunes, designed for continues usage on a desktop computer and the portable IPod.

2.2.2 Multiple-User Interface

The term Multiple-User Interface (MUI) refers to a UI that allows a user or a group of users to access the same service from different computing platforms (Seffah, Forbrig, Javahery, 2004).

With computing platforms the same authors refers to a combination of hardware and operation system, such as a desktop computer, a PDA or a mobile phone. The term is used in the same sense in this thesis.

Related terms also presented are horizontal and vertical usability (Seffah et al., 2004). The first one refers to cross platform usability with the goal of consistency. The latter term refers to platform specific usability, taking local
constraints into account and exploring the platform specific capabilities. This term is comparable to inter-usability, introduced by Denis and Karsenty (2004). Both horizontal and vertical usability are according to Seffah et al (2004) of great importance when building a MUI.

According to the same authors device manufacturers often develops guidelines to external developers, to ensure the vertical usability, and they now see an upcoming trend for software developers to do the same for the horizontal guidelines, as in SUN’s java machine.

Seffah et al lists a number of usability factors critical when developing a MUI:

*Interaction style* Since different platform has different input devices the MUI has to cope with different interaction styles.

*Information scalability* Screen size issues forces smaller devices to be more restrictive regarding amount and richness of content.

*Task structure adaptability* A mobile device might have to structure tasks differently. Typically tasks are split up in smaller steps and thus creating a deeper structure.

*Visual meaningfulness* A mobile device might have to use different graphical elements due to screen size.

*Platform constraints and capabilities* Under this header the authors summarizes and highlights differences in bandwidth, computing powers, screen size and input devices between platforms.

*Behavior* Consistent behaviour between platforms in use is important, e.g. that global settings set locally should carry over to all platforms.

*User awareness of trade-off* This point states that users might very well be willing to somewhat degraded experiences, as the lack of some features, as long as the changes are made clear and the core service is available.

*Service availability* In this last point Seffah et al. (2004) state that not all assets of a feature needs to be available on all platforms, e.g. pictures can be displayed in black and white or not at all, text can be abbreviated etc.

Seffah et al claims these characteristics to be used as guidelines when designing and evaluating a MUI, which we will come back to later on in this essay.

### 2.2.3 Graceful degradation

Florins and Vanderdonckt (2004) introduces the term of Graceful Degradation (GD) as a concept for designing UI’s for multiplatform systems. The concept is made for systems existing on platforms with different constraints, such as computing capabilities, screen sizes and ways of interaction.

The starting point of a GD process is the system as it works on the less constrained platform of the system’s device range. Adapting this original UI to the more constrained platforms forces a degradation of the UI, hence the
second part of the name of the concept. The first word comes from the fact that
the model has as its focus to keep up the continuity of the user experience,
thereby making the degradation graceful.

The approach of GD is a model based one, and is to be used on three of four
abstraction levels identified in the Cameleon framework, presented in Calvary

- Tasks and concepts – this is the highest level in terms of abstraction
  and concerns the tasks the user should carry out and what objects that
  needs to be manipulated in order to do so.
- Abstract interface – on this level tasks and objects are turned into
  presentation units such as windows and panes that support the
  execution of connected tasks.
- Concrete interface – this is the last level of concern for the designer and
  includes layout changes to and transformations of graphical objects.
- Final interface – at this stage the UI is implemented in source code.

Since the final interface is not a concern of the designer in this model, the GD
rules apply to the three more abstract levels. On each level there is a different
set of GD rules that apply. Naturally changes on a more abstract level result in
bigger differences in the final interface than a low level change, so in order to
ensure maximum continuity the authors propose to prioritize the
transformations in the reversed order, compared to the list above. Only when
transformations on lower levels are not possible to use or do not give the
needed effect should the designer go on to higher levels.

2.3 Design methods

To better understand how and at what levels service and interaction design
operates I will describe some methods below.

2.3.1 Service design

Zeithaml and Bitner (2003) present service blueprinting as a way to describe a
service, drawing on earlier research done in the field (Shostack, 1984, 1987).
They claim that it is critical to the success of a service that employees,
customers and managers all know what the service is in terms of steps and
flows and can see their role in its delivery. The inclusion of customers in the
process is important. Zeithaml and Bitner (2003) state that a service blueprint
is particularly useful at the design and redesign stages of service development.

A full presentation of service blueprinting is out of scope for this thesis. The
goal is instead to show the reader at what levels the service design process
operates, and that it deals with services as processes rather than objects.
Traditionally service blueprints have been used for services that are delivered by a person, but Zeithaml and Bitner (2003) present a model for Technology-Delivered Self-Services, shown in Figure 2 and briefly explained below.

![Figure 2 Structure of a blueprint for Technology-Delivered Self-Service. Source: Zeithaml and Bitner 2003 p. 234.](image)

Physical evidence is the feedback the customer receives after each contact with the service. On the customer actions level all steps the customer performs when consuming and evaluating the service is described. The line of interaction represents the direct interaction between the service providing organisation and the customer. The onstage technology describes the interface between the technology and the customer. The line of visibility shows how much of the service activities the customer can see. Finally the support processes contains those hidden activities that are needed to make the delivery of the service possible.

### 2.3.2 Interaction design

Design activities has been said to begin once a set of requirements has been established (Preece, Rogers, Sharp, 2002). It is beyond the scope of this thesis to go into details on design activities, but I will describe some aspects of it, namely platform guidelines as one aspect of requirements, and prototyped based design.

The work with requirements in an interaction design situation concerns first and foremost the collection and understanding of user needs, according to Preece et al. (2002). Other aspects are technological limitations. As mentioned under the MUI section, device manufacturers often write UI guidelines for external developers to follow. These guidelines act as yet another requirement to be met by the designer. In general such guidelines govern the look and feel of the application and describe how to solve typical problems of layout and
interaction. For an online example see Nokia’s UI and User Experience Guidelines.

Coming back to Preece et al. (2002) and their description of the design process, the authors claim that a design emerges iteratively through a repeated cycle of design, evaluation and redesign. They also states that users must be involved in this process to achieve a good result. In order for the users to evaluate the design there is a need for interactive prototypes of the designer’s idea. Such prototypes can, still according to the same authors, be anything from a paper-based storyboard through to a complex piece of software. What they all have in common is that they are process-oriented rather than static. Beyond being a tool for user testing Preece et al. (2002) also promote prototypes as enablers of discussion together with other stakeholders, within the team and for the designer to test out ideas for him or herself.
3 Method

The gathering of data for this thesis has been performed as a design work. There are two main reasons for this. First I wanted to be able to discuss the results on a hands-on and first person experience level. Secondly design methodology can be described as a reflective activity, in which a practitioner constantly needs to shift between being a doer and an observer (for example Löwgren and Stolterman, 2004). As a consequence of this I consider design work suitable as data when conducting research.

The remains of this chapter is divided into three sections: Identification of the service offer, Extraction of a general framework and Design of a specific solution. In each section the methods used to perform the respective task are described.

3.1 Identification of the service offer

Concerning the theoretical background for the service offer, presented in the previous chapter, I think that it is problematic that Edvardsson (1996) uses the combination of Peripheral and Supportive services to answer to Secondary needs. Following the definition of peripheral services, as made by Grönroos (1990), they are more closely related to the core services, while the supportive services might very well be independent services and answer to somewhat different needs.

I would say that secondary needs, following Edvardsson’s definition of it, should be met by peripheral services, but since supporting services are meant to increase the chances of the service package to be chosen among others, they are rather answers to related needs, that the user might and might not feel after he or she identified the primary needs.

I will use Edvardsson’s model of the service offer in this thesis, but I will treat the secondary needs as composed by different levels in order to keep the difference between peripheral and support services introduced by Grönroos (1990).

For the analysis of the service offer, four employees of the software company joined for a workshop. Three of the participants were interaction designers, one being the first to work within the company as interaction designer. The fourth member of the group was the creative director of the company, responsible for the branding of the company, including the graphical look and feel of the software, and one of the first employees to join the company.

Two of the participants had previous experience of the service design perspective, the other two got a brief introduction to the field in general and to Edvardsson’s model in particular. A print out of a picture similar to Figure 1 was present during the workshop.
Before looking into the service offer behind the application an overview over the many communication features included in the software was created in terms of a chart.

The work process was then such that each participant wrote down user needs and features of the software on post-it notes and put them on a table. Through common discussions these needs and features were then sorted to form a number of charts, displaying the instantiations of the terms used in Edvardsson’s model. User needs were sorted into primary and secondary needs and features were sorted into core, peripheral and support services to match these needs.

The outcome has been subject to further refinement done by me and another participant after the workshop. The charts, both the feature overview and the ones describing the service concept are displayed in the results chapter. The other participant mentioned is Peter Erhard, and his conclusions on the outcome of the workshop can be read in Erhard (2006).

3.2 Extraction of a general framework

A number of steps where taken in order to extract the general framework from existing implementations, each one have their section below.

3.2.1 Identification of features of current interest

As a start up for the work with a general framework a feature based analysis was made. The goal of this analysis was to identify features of current interest on the mobile platforms, due to available technology and business goals. I did not participate in this session, but the outcome was the following list:

- Sign in and account creation
- Call
  - Incoming
  - Outgoing
- Contact list
- History and event handling
- Myself
  - Online status
  - Account management
  - Profile
- Add contacts
- Chat
- Options
3.2.2 Division of analysis into levels
To facilitate the analysis of the identified features the work was divided into three levels:

- Flow/steps
- Physical interaction
- Graphics/screen size.

And as a compliment to the latter point the issue of information showing and information hiding was considered.

The idea was that these levels would work as starters, and that the project team would see if there was a need for any more levels along the way. It was assumed that the flow/steps axis would be the easiest to look into and that it was the most important one to ensure consistency.

3.2.3 Division of platforms into classes
To simplify the walkthrough of the implementations the mobile platforms were divided into groups with similar capabilities and constraints. Each class was represented by one specific platform during the walkthrough.

- Platforms dedicated to the software, Dedicated
- Platforms with mobile operating system running the software as an application, with smaller screens and no touch screen, Light mobile
- Platforms with mobile operating system running the software as an application, with bigger screen and touch screen, Heavy mobile

The classification was guided by the traditional division between dedicated and application based platforms concerning product development at use in the software company at the time of the project.

3.2.4 Walkthrough of existing implementations
Screen shots from the three class representatives were collected and a walkthrough of the features of current interest was made. From these walkthroughs a general design framework was extracted for each feature, and in the result chapter three of them are presented: making a voice call, creating a user account and viewing a contact card.

The reason that these three were chosen was that they have a natural focus on the three axis of analysis: making a voice call on Physical interaction, creating a user account on Flow/steps and viewing a contact on Graphics/screen size.

In addition to this the service offer analysis of the software presented in the result chapter identified them as implementations of a core service, a peripheral service and a support service, respectively.
3.3 Design of a specific solution

The redesign mentioned earlier concerned a platform in the *Heavy mobile* device class. The project was run together with external consultants who did the main part of the interaction design while I acted as reviewer and co-designer. At the end of the project my role became more active due to resource constraints from the consultants part.

The work was iterative with several reviews of wire frames and mock-ups\(^1\), and we also had meetings with the developer team to adapt the scope of the software release as we went on. No user tests were conducted until a beta version of the code was ready, a fact that will be subject to discussion later on in this thesis.

Since I was a member of both the general mobile and the specific platform design projects the processes could feed each other. I also used the service offer analysis of the software as a reference point to look at and reflect upon during the time of the two projects.

---

\(^1\) The word mock-up is here used for a graphic, image or paper illustration made to describe screens and user interaction of the software application.
4 Results

This section is divided into three parts: The identified service offer, General and specific design solutions and User testing of specific solution.

4.1 The identified service offer

Below are presented the charts from the service offer analysis. The first one categorises among the communication features provided by the software, the ones further down show the division of the service concept into primary and secondary needs and core, peripheral and support services.

4.1.1 Communication features – overview

The analysing group found that the communication features could be grouped along two axis: one going from targeted to broadcasted communication, the other from asynchronous to synchronous. The following model was created:

![Diagram of communication features]

Figure 3 Communication features

The horizontal axis, going from Asynchronous to Synchronous, was defined as the assumed mean time between sending and receiving of a message. The vertical axis, going from Targeted to Broadcasted, was defined as a combination of the number of recipients and to what extent the sender can choose the recipient of a message.
Shared groups, Mood messages, Picture and Presence are features that merit some explanations in terms of functionality. In brief Shared groups is a way to introduce contacts to each other and the other three features let a user choose how to appear in the contact list of his or her contacts; Mood messages is a text the user can edit, Picture is an image the user can choose to add and Presence display the online availability of the user.

One feature that stands out in the chart is File transfer which can be viewed as a nestled form of communication, versus the rest that are of a more direct form of communication.

Once the features were grouped in this overview the workshop participants argued that the historical core of the application lay mostly in targeted and synchronous communication features, with a recent move into other areas.

4.1.2 Service concept – overview

The outcome of the analysis was that the primary user need can be summarized as Getting in touch with someone which is met by Primary communication services, and that the secondary needs can be summarized as Contact points and Enrichments of experience met by Launching pad services and Secondary communication services, respectively.

![Figure 4 Overview of the service concept](image)

The customer needs will be kept condensed as a way of delimiting the scope, whilst the different part of the service offer will be developed in the following paragraphs.

4.1.3 Core services

The workshop identified mostly synchronous and targeted communication features as implementations of core services to meet the primary need of getting in touch with someone. There were two exceptions to this, the nestled
communication feature of File transfer was excluded and the asynchronous and targeted communication feature of SMS was included.

- Synchronous and targeted communication
  - Voice call - 1:1 and conference
  - Video call – 1:1
  - Text – 1:1 chat, Multichat
- Asynchronous and targeted communication
  - SMS

Figure 5 Core services

The service of voice calls includes the mediation of computer-computer calls, computer-phone calls, and phone-computer calls.

4.1.4 Peripheral services

The part of the secondary needs that arises from the will to use the core services are met by the peripheral services. In the overview these needs and services where summarised in Contact points and Launching pad service. Here is a list of services that also address the related need of Account management:

- Launching pad for communication
  - Contact names in Contact list
  - Presence indicators in Contact list
- Account Management
  - Sign in and Account creation
  - Manage paid services
4.1.5 Support services

The part of the secondary needs that are not directly related to the use of the core services were summarized as *Experience enrichments* needs above, to be met by *Secondary communication* services.

- Synchronous and targeted communication
  - File transfer
- Asynchronous and targeted communication
  - Voicemail
- Asynchronous and broadcasted communication
  - Mood messages
  - Picture
- Synchronous and broadcasted communication
  - Public voice conferences
4.2 General and specific design solutions

Each of the chosen features/services is presented separately. First comes the general framework presented in priority tables and wire frames\(^2\). Then follows a presentation of the new platform specific design.

4.2.1 Making a voice call: Core service and focus on Physical interaction

General design

Vertically aligned frames in the figure below are alternatives to one and another. The wire frame format used will be further explained in following sections.

\(^2\) The term wire frame is used for a simplified and schematic graphic made to describe screens and user interaction of the software application on a basic level, compared to mock-ups that are more detailed.
### Stepwise Call Journey with Focus on Physical Interaction:

<table>
<thead>
<tr>
<th>1. Starting point</th>
<th>2. Contacts</th>
<th>3. During Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle screen</td>
<td>Contacts</td>
<td>Call</td>
</tr>
<tr>
<td>15:23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Contacts**

- Andres
- Arno
- Beatrice
- Carter
- Emma

**Start Point**

- If the software acts as an application the user should start in Contacts. If the software acts as operating system the user should start on an Idle screen with soft key\(^3\) access to Contacts.

**Contacts**

- Navigation in the contact list should be available on hard keys\(^4\) or contextual touch control such as clickable contacts and scroll bar, or both.
- Make call should be available through as many controls as possible, following the priority list below:
  1. Hard key
  2. Contextual touch control
  3. Toolbar touch control
  4. Direct soft key
  5. Menu item

---

\(^3\) A soft key is a button which functionality changes dependent on the current situation. Cell phones of today often have physical soft keys below the screen, with the current functionality displayed on the screen just above the button.

\(^4\) A hard key is a physical button with fixed functionality. Most cell phones have a hard key for making a call.
During call

- End call should be available on as many controls as possible, following the priority list below:
  1. Hard key
  2. Contextual touch control
  3. Toolbar touch control
  4. Direct soft key
  5. Menu item

Specific design

As mentioned before one of the drivers for the redesign was a new generation of platform guidelines. One of the more prominent differences was the removal of the bottom placed toolbar with room for a number of action bound icons. In the new guidelines two soft keys are used instead. The most probable action should be on the left soft key, and a Menu option on the right soft key.

![Figure 9 Call screens for Specific design](image)

Stepwise Call journey with focus on physical interaction:

- Starting point
  - The user starts on the contacts tab
- Contacts
  - Navigation remains on hard keys and direct manipulation on touch screen
  - A call can be launched from:
• Hard key
• Call icon on card
• Left soft key
• Tap and hold menu

➢ During call
  ▪ A call can be ended by
    • Hang up icon on call tab
    • Left soft key

In addition to the call icon, controls for chat, voice mail and links to ordinary phones were introduced on the contact card. The latter two shows only if the selected contact features these functions. We will come back to this in the user testing section, and in the discussion.

4.2.2 Create user account; Peripheral service and focus on Flow/Steps

General design

In order to describe a flexible flow that can be used with different screen sizes a table with wire frames were used. In the figure below each wire frame is a step, and the numbered table cells are tasks, containing one or more steps. Vertically aligned steps are alternatives to one and another whereas horizontally aligned steps all should be implemented. If all horizontal steps of a task can be put in one screen they should, but different tasks should not be put on the same screen even if possible.
Figure 10 Sign in flow for general design

Stepwise Sign in flow, one bullet per task:

- Sign in or Create new account
  - Sign in or Create new account OR
  - Sign in or Create new account
    - Fill in existing User Name and Password
    - Remember password
- Choose User Name and Password
  - Enter User name, 6-32 characters
  - Enter Password, Minimum 4 characters
  - Repeat Password
- Read and agreement of End User Licence Agreement
- Creating account
Signing in

Start screen (idle screen/contacts)

The first task description merits some extra attention since it contains alternatives, and therefore an extra level in the bullet list. This is to be understood as an optional step, that should only be included if it fits in the Sign in or create new step. If not, the step is to be excluded, and is then to be part of the Sign in to existing account task path only, which is not presented here.

Specific design

The following screens present the platform specific design:

![Specific design screens]

Figure 11 Stepwise sign in flow of specific design

The graphics below the actual screens shows a button that appear when user name and password have been filled in and the screen when scrolled down, respectively.

Create new account journey, one bullet per frame:

- Sign in or Create new account
  - Fill in existing User Name and Password
  - Remember password
  - Create new account
- Choose User Name and Password
  - Enter User name, 6-32 characters
  - Enter Password, Minimum 4 characters
  - Repeat Password
  - Agree to End User Licence Agreement
Read End User Licence Agreement – not shown in the figure.

Creating account

Contacts

4.2.3 Contacts list: Support service and focus on Graphics/Screen size

General design

The general framework for what information and controls to include in the contact list is presented in the table below. The screen size and available controls decides how many of the steps that can be implemented. It is not always desirable to expand a selected card, again depending on screen size. To address this issue a limit is set so that a selected card should only expand if the screen size is big enough to show at least six collapsed contacts along side with it. This limit, as the rationale of the whole design, will be subject to discussion further on.

Table 1 Contact card priorities

<table>
<thead>
<tr>
<th>Priority 1</th>
<th>Priority 2</th>
<th>Priority 3</th>
<th>Priority 4</th>
<th>Priority 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unselected contact</strong></td>
<td>Presence/Contact type icon - Full name</td>
<td>Mood message snippet</td>
<td>Promotion</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Indicators – Video, Voicemail</td>
<td>-</td>
</tr>
<tr>
<td><strong>Selected contact</strong></td>
<td>Presence/Contact type icon - Full name</td>
<td>Practical Indicators - Phone numbers available</td>
<td>Picture - Mood message/Phone number links</td>
<td>Controls – Call, Chat</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Location – Time zone</td>
</tr>
</tbody>
</table>
Specific design

Figure 12 Contact card in specific design, with and without mood message

- Non selected contact
  - Presence indicator / type of contact icon
  - Full name
- Selected contact
  - Presence indicator / type of contact icon
  - Full name
  - Picture
  - Call button
  - Chat button
  - When available:
    - Mood
    - Telephone links
    - Voicemail button

4.3 User testing of the specific design solution
The user testing of the specific design was done by external user experience consultants. The test was carried out on 12 persons, five of them had experience of the application but not of the platform, five had experience of the
platform but not of the application and two people had experience from using both the platform and the application.

The tests showed a major problem in the design which constitutes of users not paying any attention to the soft keys and instead searching for interaction points above the tabs, which often were not present or visible enough.

The consequences of the soft key problem were biggest in the first time user experience, when the user got presented with a screen stating ‘You have no contacts’ (the test contact shown in the specific design screens had not been implemented in the beta version). Many users explored the more visual tabs before finding the menu on right soft key containing Add a contact.

**Creating a user account**

The tests found some missed details which caused unnecessary annoyance among the users, such as lack of information about needed uniqueness of user name when registering a new account.

In the End User Licence Agreement window users searched for an accept button, or at least an on screen exit button. The screen conformed to the platform dialog guidelines, with the closing OK-action in the top right corner, which users did not find immediately.

**Contact card and making of call**

The contact card, where controls had been placed on the screen, worked better in that users started to look there, but they did not immediate understand which control to use for making a call for example. Many tried to use the phone number links to make a computer-computer call, and the voicemail button also drew attention from the call button. There was thus a problem of visual prominence between the controls.

In the next chapter the results will be discussed.
5 Discussion

In this chapter I will discuss the impact of the service design approach and the use of a general UI framework. The main part of the discussion is held in the sections Discussion of methods, Discussion of results and Theoretical comparisons and the chapter ends with Conclusion and suggestions for Future work.

5.1 Discussion of methods

In this section the service design workshop and the extraction of the general framework will be discussed.

5.1.1 The service design workshop

As mentioned in the methods chapter the participants of the workshop that produced the analysis were all employees of the software company. As a consequence they had very good knowledge of the application and also of the ideas behind it. The service design literature traditionally addresses managers in charge of service development, from my point of view a work task comparable what the participants are involved with at the software company. I would therefore claim that they were suited for analysing the service offer side of the service concept analysis.

When it comes to the analysis of the needs that the service offer wants to meet, there can be some doubts concerning the method. The participants can not be viewed as representatives of the common user group. To achieve a more accurate picture of the needs involved a user research would be a more appropriate tool.

Another aspect of the workshop participants was that only two of them had previous experience from the theoretical model used. My view is that their professional experience of designing the application compensates for the lack of theoretical experience. A last reflection on the participants is that it would have been interesting to involve some people from higher up in the product management department, who have more insights in the development roadmaps.

My impression of the work process as such was that it worked well, but that the result would have benefited if there would have been more than one session. Now only two of the participants took part in the refinement of the charts, which might have put too much of their personal views into account.

5.1.2 Extraction of the general framework

The division of platforms into classes were necessary to make the extraction work efficient. The screen shot walkthrough got manageable that way. I would though argue that the role of classes is greatly diminished once the general framework has taken shape and the task is to move from general to specific.
The framework should be able to cope with the differences among platforms without forcing a classification of them. This statement urges a tighter relationship between the dedicated and application based platforms during development than the current work process admit.

To visualise flows in the general framework wire frames were used. The format of the wire frames was described briefly in the result chapter, but I will elaborate a bit more here. In Figure 13 a generalized overview of the wire frame format is shown, and its components are described below.

```
  x
  1
  y
  
  z
  2
  
  m
  3
  n
  o
```

Figure 13 Schematics of wire frame format

First of all the wire frames were used to capture flows and they are to be read from left to right. Boxes with black lines and letters represent steps and are placed inside of grey boxes with numbers which represent tasks. A task is thus made up of one or more steps. Within a task vertically placed steps, as step x and y in the figure, are alternatives to one and another, whereas horizontally placed steps, as m, n and o in the figure, are all to implemented - in the order they appear.

The relationship between the wire frames and a specific design is that the latter may fit all steps of a task into one screen if the screen size admits it, but should not fit more then one action even if it is possible.

The format was developed out of a need to find a balance between flexibility and guidance. The wire frames served their purpose in that we could treat flows from all platforms within one format. To allow alternative steps within a task was a way to keep even quite different solutions within the same framework.

My belief is that the concept is sound and works well when working with platforms with similar capabilities and constraints. With increasing differences between concerned platforms I think the method would need further development. If for example the number of alternatives grows the simplicity diminishes. The method will be subject of further discussion when the role of the general framework is discussed further down.
5.2 Discussion of results
In this section the use of the service design approach and the general framework are being discussed. The discussion addresses both the actual products of them, that are the service offer and the general design, and the impact the use of these artefacts had on the design process.

5.2.1 The identified service offer
Before discussing the actual service offer, the categorization of communication features, earlier shown in Figure 3 will be discussed.

![Communication Features Diagram](image)

Figure 14 Communication Features. Previously presented in Figure 3.

Looking at where along the axis the different features were placed, the most questionable one in my eyes is that the instant text messaging features of One-to-one chat and Multi chat are positioned on the right half of the axis that runs from asynchronous to synchronous. The choice of representing this quality as a continuum rather then as a binary choice leaves room for interpretations. If communication is viewed upon as messages sent back and forth, my interpretation is that the longer period between messages the more asynchronous is the communication. With such an interpretation these features can indeed move quite far on the axis, but in the analysis we chose to put it where we thought the heaviest usage of them lies. So the assumption was that most users use these features to have conversations that are quite compact in time.
The goal of creating the chart in the first place was to discover groups among the features that would help us identify the service offer. This was partially achieved as most of the core services were found among the synchronous and targeted communication features. As shown in the result section there were also exceptions to this, however.

When looking at the identified service offer one issue to discuss is the relation between core and support services. For example one-to-one calls and standard conference calls have been identified as core services while broadcasted conference calls has been labeled a support service, following Grönroos (1990) definitions. In my view this has to do with the historical evolution of the communication services offered by the software. The enabling of broadcasted conference calls happened only recently and the users are not thought to have included the urge to have public conversations into their primary needs just yet, which is Edvardsson’s (1996) criteria for being a core service. This might very well change in the future.

Another difficulty in the separation of core and support services encountered in this work was when a support service has become so common among competitors that it feels like a central part of the service offer, thus making it tempting to count it as a core service. This case is not quite addressed by any authors in the theory section, even though Grönroos’ (1990) definition of core services as the reason for being on the market gives some aid. An example of such a border case in this work is Picture in the contact list, here kept as a support service. The importance of being restrictive with what to call a core service can be illustrated by the example of the airway company Ryan Air that stripped away things generally known as part of the deal, such as free serving of food, and made a business go back to basics. Since the move of a software application to different platforms often forces the designer to go back to basics, I would say that it is of great importance in this context.

The separation of peripheral and support services can also be complicated, as stated by Grönroos (1990) in the theory section. In this light the fact that Presence indicator got identified as peripheral service while Picture and Mood message was viewed as support services is worth discussing. The three features are part of the contact list, and the reason of the division between them was the result of an analysis of the contact list. The name of a user was viewed to be as basic as a contact gets and makes up a launching pad for communication, making it a peripheral service. To display the on-line presence can be viewed both as a way to inform the decision of launching core communication services and as a secondary communication service per se, as in “I’m here - you’re not alone”. This makes it partly a peripheral and partly a supportive service. Since it is not possible to separate the feature any further it was sorted as an implementation of a peripheral service.

Mood and Picture were viewed as answers to the need of Enriching experience and were therefore sorted as implementations of support services.
5.2.2 Impact of the service offer on the design

When designing the general framework for how to make a voice call the service offer provided the design with the fact that this is a core service and therefore, following Grönroos (1990) of highest priority. This encouraged a design with few steps and prominent controls, but did not provide any detailed guidance.

The general design of the account creation flow followed the same pattern as the making of a call when it came to the impact from the service offer.

My experience from the creation of the priority table over what information to show on the contact card in the general design is that the impact of the service offer was important here. Many features compete for the available space and the fact that they had been classified as implementations of different types of services, with different priority levels attached to them, helped to sort among them. The service design perspective thus acted as a driver in the interaction design process. This experience is inline with Grönroos’ (1990) claim that from a managerial point of view it is important to separate the service bundles.

In my opinion the differences between how much impact the service offer had in the general design process is due to the nature of the analysis. The biggest gain is found when implementations of different services compete for the attention of the user. This characteristic indicates that the method might be useful on higher levels of abstraction, a thought that will be further discussed in the theoretical comparison.

When it comes to the work with the specific design my experience is that the impact of the service offer analysis was of an indirect nature since the major priority decisions had already been made on the general framework level.

5.2.3 The general framework

One interesting aspect of the general design is the list of controls for the actual launching of a call. The design encourages the use of as many as possible, with the theoretical possibility of having five ways of launching a call, out of which three would be visible on the screen. On a small mobile device this might seem overwhelming and a waste of resources. In practise this scenario is not likely, but it highlights that the general framework is better viewed as a toolbox rather then a specification.

The general framework as a tool was visualised with wire frames of steps and actions as well as numbered lists and tables. The wire frames, representing flows, focus on user activities and the process nature of the UI. In this aspect they are following the ways of both the service design methods as presented by Zeithaml and Bitner (2003) and the interaction design methods as presented by Preece et al. (2002). The numbered lists and the tables are more static and are closer to UI guidelines, also presented in the theory section.

When compared to the structure of a service blueprint, as presented by Zeithaml and Bitner (2003), the general framework addresses the onstage
technology, or more precisely acts on the line of interaction. This makes it clear that the work on the general framework can be viewed as part of a service design process.

5.2.4 Impact of the general framework on the design

The general framework had a strong impact on the design of the call flow and the contacts list in the specific solution.

In these situations the framework acted as a toolbox with priorities. Platform specific constraints were checked with the framework wire frames and/or priority lists and the design space available got specified and manageable. In the words of Preece et al (2002) the framework acted as a requirement, preceding the actual design work.

The points made above states that the work with a general framework has eased the work with the specific design. The framework level has been used to work out priority issues and to explore platform capabilities and constraints. The task in the specific design was then reduced to combine the appropriate level of the general design with the platform guidelines. This was probably possible partly because of the fact that the framework was recently created, and that the predecessor to the specific design had been part of that work. Thus many issues in the design had been addressed in the framework.

I believe that in order to have a useful general framework it has to be revised with every new platform that shows up, and this revision needs to become part of the development process across platforms. If the framework does not have a solution to a problem then stakeholders from all concerned platforms should join and add such a solution to the framework, so that it can be reused in future designs. This would indeed address the issue of continuity between platforms.

The parallel development of the general and the specific design in this work indicates that such a continued process could be achievable.

5.2.5 Analysis of the results from the user testing

The results from the user testing are discussed, both in general terms and in the light of the service offer analysis and the usability factors presented in the Multiple-User Interface (MUI) section.

Creating of account

The problem of the missing information concerning uniqueness of user name can be viewed as the result of omitting a peripheral service, with the core service being the creating of an account. The service offer analysis was not that detailed though, so no such guidance was given. This implies that a more detailed analysis could be done, and be fruitful.
First time user experience

In the case of the first time user experience on the Heavy mobile implementation it is safe to assume that a control for Add a Contact visible above the tabs would have helped. The lack of on screen controls can be derived from the general design, which does not include any support for the use of such controls for devices with such capabilities. Expressed in the terms of MUI usability factors there was a lack in the adaptation to a new interaction style, which resulted in suboptimal usage of platform capabilities. In terms of service design perspective this is an example of where a peripheral service has been neglected, which put a core service out of reach for the user.

Contacts card and making of calls

As shown in the result section the user testing showed a major flaw in the design regarding the reliance on soft keys. Part of the mistake was to consider the soft keys as a replacement for the bottom line toolbar in the older version.

In the example of how to launch a call, the controls on the contact card introduced in the new design seemed to have been a move in the right direction when working without a toolbar, but the priority of what to include and what visual prominence the different controls should have should be improved to better reflect the importance of the options available.

Problems occurred with phone number links and the voicemail button, these are only shown if available, and most users do not have them, so it would be interesting to see if they actually could be part of a scaling user experience.

If usage and/or further testing would show that it is not the case, then voicemail is a support service and should be removed altogether, whereas the relation between calling ordinary phones and making computer-computer calls should be further investigated.

In this project the user testing were performed after an internal beta version of the application was implemented. This limited the possibilities to make changes according to the findings, but some critical changes were made before the public beta release, and others were planned to be done before the gold release. The remaining issues were used as input for the design work for the next version, and a decision to make user testing earlier in the processes next time was made.

Earlier in the discussion it was stated that the impact of the service offer analysis on the specific design was indirect. The analysis of the user testing indicates that user testing of the specific design can give feedback not only to the general framework, but also to the service offer. The relationship between the three levels of design can thus be visualised as in the figure below.
Figure 15 Impact and feedback cycle between levels of design.

The dependencies described in the figure above can be further developed. In the work presented the service offer analysis was dependent on the general framework to influence the specific design. The general framework was not dependent on the service design, but without the service offer analysis the benefit it produced would have been lost. It seems that in order for a service design perspective to contribute to an interaction design process it needs to be applied on a design level that is not to concrete.

5.3 Theoretical comparisons
In this section I discuss the service design approach and the use of a general framework in the light of the interaction design theories presented in the Theoretical background chapter.

5.3.1 The Service design approach
Graceful Degradation (GD) and the theory of Multiple-User Interface (MUI) treats the deciding of what is important to port and not as something the designer should just know while the Service Design (SD) approach described in this thesis presents tools for identifying them. The rules applied to the task and concept level of the GD framework address this to some extent, but it still seems to me as if SD works on a higher level. So I would argue that SD can give the designer a top view of the process with clear goals, while the application of GD rules and to keep the MUI heuristics as a checklist seems to be a better help on the lower and more concrete levels. To put it in yet another format it seems to me that SD can help to answer What and Why- questions, while GD and MUI are more appropriate to answer How questions.
Given this high level of abstraction it would be interesting to see what the service offer analysis could do in product development. After all, the concepts come from the service development literature.

An example would be to consider the move to a complete new context of use, where there is not yet an existing platform to adapt to. Needless to say, this is a situation not explored in the design work presented in this thesis. Nevertheless it seems to me that a service offer analysis might be applicable in such a situation, thanks to its platform independent approach.

5.3.2 The General framework approach
The three levels of analysis chosen for the extraction of the general framework were Flow/steps, Physical interaction and Graphics/screen size. The last one combined with information showing and information hiding. I would say that these levels are comparable to Seffah et al’s list of usability factors for MUIs, described in the theoretical background.

The first two coincides rather perfectly with Task structure adaptability and Interaction style, respectively. The last one, together with the information showing and hiding, is comparable to Information scalability and Visual meaningfulness.

Among the remaining points in Seffah et al’s list we find Platform constraints and capabilities, Behaviour, User awareness of trade-off and Service availability. I would argue that the first and third one is better addressed when moving from general to specific. Behaviour is not addressed by the project described in this thesis. As for the last of the usability factors the toolbox nature of the general framework addresses the Service availability. I would thereby claim that the levels of analysis followed in the project are backed up by Seffah et al’s MUI factors.

When looking at the design work done from a GD point of view, it could be argued that it could have been done using the GD method with the original desktop application used as role model.

The goal of the project, however, was to create a general mobile design from the existing implementations. There was a point made in getting an overview, and reuse work that had been done already. Doing so the process got benefit from some major decisions that had already been made on task and concept level, such as what features to include, instead of restart from the desktop implementation.

From a GD point of view it could also be argued that even if the project starts from existing platforms, it could have started its graceful degradation from the least constrained of the platforms, the Heavy mobile. In this case that platform version happened to be the one that differentiated itself the most from the others and since the goal was to unite the different implementations with minimal redesign such an approach was not desired.
If we instead compare the general framework as used in this work with the GD process, I would say that the use of a general design can be viewed as the introduction of a middle step. Instead of always starting at the least constrained platform the degradation work is done in advance, leaving to each platform design to pick out suitable solutions.

### 5.4 Conclusion

In this section I summarise the discussion through the use of the research questions of the thesis.

- **How can a service design perspective add value to an interaction design process of designing UIs for multiple platform software?**

  This thesis has shown that the design of UIs for multi platform software can benefit from a service design perspective by the use of a service offer analysis.

  The most important benefit is that the prioritised and platform independent view helps the designer in situations where two or more services compete for his or her attention. In situations where there is already only one service of current interest the help in the actual design work is more limited.

  The thesis also shows that the service offer can contribute to the analysis and understanding of usability problems discovered in the user testing.

- **In a situation where there are existing products and a new one is to be designed, how can a design process be driven?**

  A systematic walkthrough of screenshots can be used to extract a general framework from existing platforms. This framework can benefit from, but is not dependent on, the influence of higher level perspectives, such as a service design perspective. The benefit of a higher level perspective is that the designer can iterate between concepts and implementations and thereby keep the focus of the design. In the case of service design the contribution is the concept of services.

  Through the use of a general framework design decisions concerning suitable interaction tools and how to prioritise among them can be made on the general level, clearing the way for the specific design process.

### 5.5 Future work

It would be interesting to look further into situations where UI guidelines from the software and platform are conflicting. The goal of such a study could be to
investigate if the service design perspective can help to minimize these problems.

It would also be interesting to use the general framework in the design of another platform specific implementation of the same software. This would make it possible to investigate to what degree the method can contribute to cross platform continuity, and also look into the aspect of time usage when the framework is reused.

Finally it would also be interesting to use the general framework when designing for a complete new platform, instead of redesigning one as done in this work. A related area of interest would be to investigate if the concept of service offer can help product development when moving software into new business areas.
References


Web sites
Continuity of Service in Design for a Specific Platform: Combining service- and interaction design perspectives in a multiple platform environment.

Linus Nilsson

This thesis presents a design work in which one platform specific User Interface (UI) is redesigned to fit into the multiple platform range of a software application. Among other design methods a service design analysis and a platform general UI framework are used. The thesis aims to study the impact of these methods.

The service design analysis starts with a workshop which outcome is used as reference point during the design process and later in the analysis of the results. The general framework is extracted from existing mobile implementations of the software and is used as a guideline in the design of the platform specific solution.

The results presented show that the service design perspective added high level priority tools to the process, and there are indications that this method can be even more valuable in earlier stages of adaptation of software to new platforms. The general framework is shown to have contributed on a more detailed level and there is indication that this method can facilitate the work of obtaining continuity between platforms.

Nyckelord
- Service Design
- Interaction Design
- Multiple Platforms
- Design Process
- User Interface Adaptation
- Design Framework
- Design Management