Proceedings of Umeå's 12th Student Conference in Computing Science

USCCS'08

Edited by

Jürgen Börstler, Håkan Gulliksson and Dipak Surie

UMINF-08.08
ISSN-0348-0542

UMEÅ UNIVERSITY
Department of Computing Science
SE-901 87 UMEÅ, SWEDEN

Preface

Umeå's Student Conference in Computing Science is the highlight of a "conference course" in our Computing Science curriculum. The objective of this course is to give students a forum, where they can actively participate in scientific research and development. The conference format was chosen to provide a realistic environment for the presentation of their research results.

The "conference course" is a practical course for students interested in research and introduces them to

– independently researching an interesting topic;
– using a foreign language (English);
– writing scientific reports on their work; and
– presenting their work at a conference.

This year was the twelfth offering of the course with a total of 30 registered students. Of these 30 students, 27 actively participated in some part of the course and 19 eventually submitted a full paper. Of the 19 submissions, we accepted 16 for publication in the proceedings.

Each submission received at least two independent reviews. We would like to thank all reviewers who helped to review all papers within a very short time frame.

Please check the conference course home page for further information on the course (http://www.cs.umu.se/kurser/5DV054/).

Umeå, May 2008

Jürgen Börstler
Program Chair
USCCS'08

Program Committee

Jürgen Börstler (Program Chair)
Håkan Gulliksson
Dipak Surie

Additional Reviewers

Emanuel Dohi
Johanna Högberg
Jan Erik Moström
Greger Wikstrand

Table of Contents

Biometrics / Quality of Service

Software Development Issues

Designing digital-photo-collection browsers for small-screen devices

Petter Svensson-Silfver

Department of Computing Science
Umeå University, Sweden
dit04ssn@cs.umu.se

Abstract. In recent years, mobile phones have become an everyday product for people all over the world. Almost every mobile-phone model now offers the possibility to capture digital photographs with different kinds of integrated cameras. As a result of this a multitude of digital-photo-collection-browser solutions have emerged on the market. This article investigates the work of two of the most recent researches in the area of development of digital-photo-collection browsers for small-screen devices. As will be revealed in this article, designers should be aware of the way we humans think about and organize our photo collections as much as being aware of how to cue the user of the size of the collection and how to avoid overcrowding the screen.

1 Introduction

Ever since the beginning of the 21st century, hand-held devices such as mobile phones, personal digital assistance (PDA) and pocket personal computers have undergone immense changes. The progression of computational power, connectivity and memory capacity has created new possibilities for using and interacting with these devices. As a result of this technical evolution, new features for these devices have materialized. One of the most interesting feature is the ability to capture digital photographs. In the midst of the emergence of this new functionality, a necessity for an easy way to explore collections of digital photographs has arisen. Unfortunately today, most user-interface solutions for small-screen devices are the same as the ones used in applications for personal computers (PC) [1]. This is somewhat of a problem since the way of interacting with a small-screen device and a desktop PC greatly differs [2]. To expound the problem from a more technical point of view one can examine the Nokia N70 (N70)—one of the most popular mobile phones today according to recent sales statistics [3]. The N70's display has a resolution of 176x208 pixels which can display up to 262144 different colours [4]. Most importantly, the N70 has a built-in 2.0 mega pixel digital camera with a maximum photo-capture resolution of 1600x1200 pixels. With its built-in memory of 22 megabytes and a memory-card slot that can hold up to 2 gigabytes, the N70 has a storage capacity of over 5 000 photos. Consequently, there is a need of a high-performance browser for exploring a digital-photo collection containing more than 5 000 photos on a device with a screen the size of a matchbox.

1.1 Goal

This article presents a study of literature on how people use and think about their photo collections and common problems for user-interface design when displaying large quantities of data on a small-screen devices. Research questions are then formed on the basis of the literature study. The research questions will be used in an investigation of work made by Harada et al. [5] and Patel et al. [6, 7]. Their research are two of the most recent in the area of user-interface development of digital-photo-collection browsers for small-screen devices. The results of the investigation will then be used to reveal different aspects for developers to consider when designing digital-photo-collection browsers for small-screen devices.

1.2 Article structure

The ways photo collections and digital-photo collections are most commonly explored are explained under the headline *From prints to digitals.* Also the first two research questions are constituted under this headline. In *Large quantities on small screens* common problems and solutions for displaying large quantities of data on a small-screen device are described. In this section the last two research questions are formed. *Digital-photo-collection browsers for small-screen devices* describes the browsers implemented by Harada et al. [5] and Patel et al. [6, 7]. *Results* answers the research questions with regard to Harada et al.'s and Patel et al.'s work. *Discussion* confers the results of the investigation, and conclusions are finally drawn in the section *Conclusion.*

2 From prints to digitals

People typically explore their photo collections alongside storytelling and reminiscing. This means that people most often search their photo collection trying to find specific photographs to help supplement their recollection [6, 8]. Rodden and Wood have found that people in search of a single photograph initially try to recall the event to which the photograph is related (e.g., a wedding or a birthday) [9]. According to the results of studies and surveys carried out by both Rodden [8] and Frohlich et al. [10], this phenomenon is correlated to the fact that people predominantly do not organize their photo collections at all. Still, when people do organize their photographs the most common manner is to organize them into albums for special events [8]. Rodden and Wood also found that people, when not remembering the event, try to recall the date on which the photograph could have been captured [9]. Coupling the two search criteria reveals two important features that can help alleviate the user when exploring a photo collection; (1) it should be ordered by events and (2) the events should be accessible in a chronological order [9, 11].

Rodden and Wood [9] state that when people search for photographs there seem to be three common activities: (1) locating a set of photographs bound to a specific event; (2) locating an individual photograph from the entire collection and (3) locating a set of photographs sharing a common property or element, but not necessarily captured at the same event. For this to be feasible, the collection has to be ordered. However, the practice of managing thousands of digital photographs on a small-screen device is an extremely time-consuming chore. One of the most widespread solutions employed by manufacturers today is to let the users organize their digital-photo collection on a more suitable platform, like a PC, and then reload the collection into the small-screen device [5, 6]. Another approach investigated is to automatically organize the photographs by pre-processing them [5]. This is made possible by extracting the time of capture from the metadata that a digital photograph carries. By using the metadata researchers have been able to develop techniques to order a digital-photo collection without human intervention.

Here, the first two research questions on which to investigate the work of Harada et al.'s [5] and Patel et al.'s [6, 7] can be formulated:

- ***Q1:*** How does the user interface of the browser make use of the fact that events are the most natural way to think about photographs?
- ***Q2:*** In what way does the browser support organizing the digital-photo collection?

3 Large quantities on small screens

Much can be said about characteristics of well-designed applications for small-screen devices, such as concerns about consistency, feedback and hierarchical placement [12, 13]. However, one of the major challenges arises when trying to display larger sets of data [14], e.g., in a digital-photo-collection browser. As Zwick et al. put it: "The length of a book can be determined by the thickness of its spine, however the size of a digital application is far more difficult to estimate" [15]. This pinpoints a major problem with designing a digital-photo-collection browser for a small-screen device, namely, how to help the user understand the size of the data structure [15]. Another problem is how to help the user maintain an overview of current relative position inside the data structure. If an application somehow approaches these two problems it can help the user to comprehend the so-called *invisible space*, which is the near surroundings of the present location depicted on the screen [15]. This could for example be the next screen when clicking on a hyper link.

Rodden and Wood state that a digital-photo-collection browser for a PC easily can support the user by displaying a large number of thumbnails (i.e. a reduced-size version of a photograph), since people are familiar with their own photographs [9]. This is unfortunately not durable on a small-screen device since you run a high risk of visual clutter. Still, thumbnails are the most common solutions among manufacturers today [6]. An alternative to the thumbnails is speed dependent automatic zooming (SPAZ), which is believed to have great potential for small-screen devices although originally developed for desktop PCs

[16]. SPAZ starts off by showing a small part of the data at normal zoom level (i.e. 100%). The level of zoom is, as the name SPAZ implies, dependent on the speed of the scrolling. As the user starts scrolling the zoom level decreases and more data becomes visible. As Jones and Marsden states: “The subjective effect is like ‘flying’ over the document, where the faster you scroll the more your altitude increases” [16].

At this point, the final research questions on which to to investigate the work of Harada et al.’s [5] and Patel et al.’s [6, 7] can be fomulated:

- ***Q3:*** In what way does the browser inform the user on how large the digital-photo collection is and where he or she is spatially positioned in the data?
- ***Q4:*** How does the user interface solve problems concerning visual clutter?

4 Four digital-photo-collection browsers for small-screen devices

Below follows a description of the functionality of four different digital-photo-collection browsers. The first two browsers, implemented by Harada et al. [5], make use of metadata pre-processing to automatically order the photographs in the collection. The second two browsers, implemented by Patel et al. [6, 7], make use of SPAZ technique when displaying the photographs in the collection.

4.1 The Baseline browser and the Timeline browser

Harada et al. implemented two types of digital-photo-collection browsers for PDAs [5]. Both interfaces make use of a type of clustering algorithm originally presented by Graham et al. to automatically organize the digital-photo collection by pre-processing of the metadata [11]. The cluster algorithm make use of the observed fact that people take photographs in a burst [5, 11]. What this means is that people tend to take a lot of photographs during an event, like a wedding or a birthday (i.e., a cluster of photographs), but rarely take photographs in between such events. Graham et al. found that by recursively comparing relative activity in such a cluster they could identify even more specific clusters within a cluster [11]. For example if the first cluster of photographs represents a wedding, two clusters in that cluster could be the ceremony and the reception.

The first browser Harada et al. implemented is called the Baseline browser (see Fig. 1) [5]. It is based upon the usage of folders and thumbnails. There are two versions of the Baseline browser. In the Baseline Manual (BM) the digital-photo collection is ordered by using the organization already applied to the photographs by the user on a PC. The Baseline Automatic (BA) automatically organize the photographs by pre-processing the metadata on a PC using the cluster algorithm. Each folder in the BM lists all of the folders and photographs that the folder consists of (see Fig. 1 A). In the BA, only the sub-folders that the folder consist of are listed. A folder is represented by a button showing three sample photographs from the folder, the number of photographs in the folder, the number of sub-folders in the folder and the name of the folder. In the BM version the names of the folders are given by the user, and in the BA version the names of the folders are replaced with the time span in which the photographs in the folder were captured. Tapping on a folder button takes the user one step down in the hierarchy, displaying the thumbnails and sub-folders in the tapped folder. If the user at anytime chooses to tap on one of the thumbnails in the present folder, the user will be presented with a preview screen where the chosen photograph is enlarged and displayed (see Fig. 1 B).

In this view, four thumbnails are visual at the top of the screen. The thumbnails represent the two photographs next and previous in order to the enlarged photograph. If the user at this time taps on the enlarged photograph, that photograph will fill and cover the entire screen (see Fig. 1 C). Tapping on the photograph at this point redirects the user to the four-thumbnails view (see Fig. 1 B). There is at all time a series of red arrowheads visible at the top of the screen to show where in the hierarchy the active folder is located. The back button at the bottom left of the screen takes the user one step up in the hierarchy, and the home button at the top left corner takes the user directly to the highest level of the folder hierarchy (see Fig. 1 A).

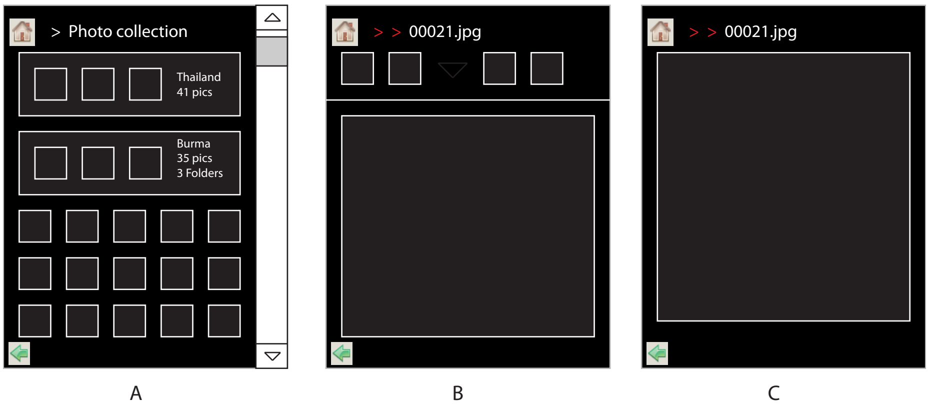

Fig. 1. The Baseline browser (adapted from Harada et al. [5])

Harada et al.'s second browser, the Timeline browser (TB) (see Fig. 2), is strictly based on organizing the digital-photo collection using the earlier explained cluster algorithm [5]. The TB interface is divided into three different columns, where the column in the middle shows a vertical timeline. Each column on the side of the timeline shows a number of album buttons similar to the folder buttons used in the Baseline browser. Unlike the folder buttons, the album buttons only show one sample photograph instead of three and the folder names is set to be the time span in which the photographs within the folder were captured. If the user draws a circle around the photograph on the album button with a stylus, he or she is allowed to change the photograph on the album button for a quick peek of what photographs are inside. No more than ten album buttons are shown in the timeline view to avoid clutter (see Fig. 2 A). Originating from each and every album button there is a white line connecting the album buttons to the vertical timeline. The reason for this is to show the temporal information of capture for the photographs within that album. At the intersection between the white line and the timeline is an additional vertical line. This line indicates the time span in which the photographs in that album were captured. Either by tapping on one of the album buttons or dragging the stylus on the vertical timeline, albums with an even more detailed time span can be shown in the timeline view (see Fig. 2 B). This is made possible by using the recursive nature of the cluster algorithm.

Finally, the user will tap on a album that contains nothing but photographs. When this occurs, he or she is presented with a grid of thumbnails (see Fig. 2 C). The thumbnails are laid out in chronological order where the first photograph taken on a new day is marked with its corresponding date. By tapping the up and down buttons at the bottom of the screen the user can navigate through pages of thumbnails in the entire collection. Next to these buttons the current thumbnail page is shown. If the user at this point taps on a thumbnail, the related photograph is enlarged and displayed similarly to the last stage of the Baseline browser (see Fig. 1 C). The home and back buttons work in the same manner as for the Baseline browser. The button to the right of the back button (see Fig. 2 A–B) is a shortcut that at anytime can take the user from the timeline view to the grid of thumbnails. The thumbnails are then presented starting from the date at the top of the timeline. On the top of the screen, showing at all time, is the time span for the currently showing photographs or album buttons.

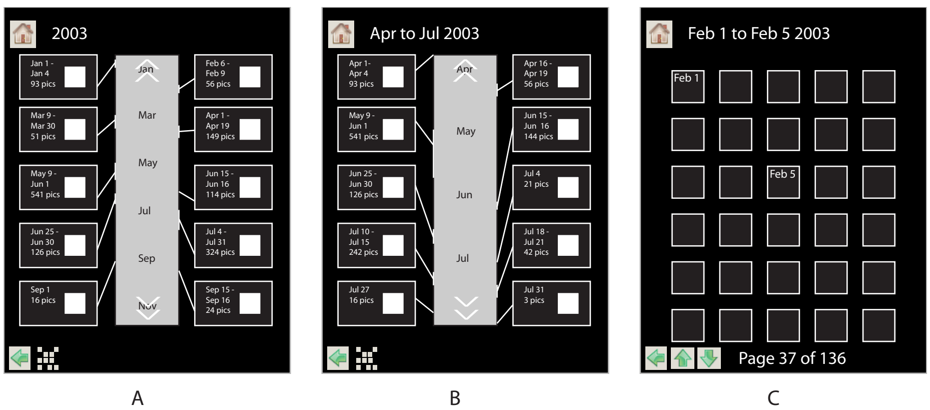

Fig. 2. The Timeline browser (adapted from Harada et al. [5])

4.2 The AutoZoom and the GestureZoom

Patel et al. try a different approach from Harada et al. to solve the problems that concerns the exploring of digital-photo collections on small-screen devices [6, 7]. Instead of evaluating what kind of pre-processes that can be used, they concentrate on how information should be displayed and what interaction techniques are more suitable for the task. To do this, Patel et al. developed two types of digital-photo-collection browsers for PDA's with touch-screens, both deriving from the same layout principle and using a kind of SPAZ technique (see Fig. 3). The layout principle is that all photographs in a collection are presented in a vertical list (or as Patel et al. calls it *a strip*). The width of the strip is the same as the width of one photograph. By dragging a stylus in relation to the vertical centre of the screen, the strip is scrolled either upward or downward. This means that dragging the stylus above the middle makes the strip scroll downwards and dragging below the middle makes the strip scroll upwards. The speed at which the strip scrolls is dependent on how far from the middle the stylus is placed. All the photographs in the strip are chronologically ordered, with the most recent captured photograph at the top of the strip. When scrolling through the collection, the path to the photograph currently under the stylus is shown at the top of the screen. This is done by retrieving information about where in the structure on the disk the photograph is located (i.e. the names of the folders and subfolders).

Patel et al.'s first browser is called AutoZoom (AZ) [6, 7]. In the AZ, the zoom level at which the photographs in the strip are displayed is interrelated to the level of speed in scrolling (see Fig. 3 A-C). Still, the zoom level is locked when the height of the strip is less than, or equal to, the height of the screen. When the stylus is released, the browser animates a smooth stop and restores the native zoom level to show the photograph at the current position in the strip (see Fig. 3 A). The background colour of the strip alters between dark and light grey to portray the shifting of months. Years are portrayed by changing between black and blue (see Fig. 3 C). Patel et al. call this *calendar overview*.

The only difference between the AZ and the second browser, the GestureZoom (GZ), which Patel et al. implemented, is that instead of automatically directing the level of zooming, the user controls it [6, 7]. This is done by taking into consideration the position of the stylus relative to the horizontal centre of the screen (i.e. the further to the left or right from the horizontal centre the stylus is positioned, the less is the zoom level).

5 Results

This section presents the results of the investigation in regard to the research question *Q1-Q4*.

Harada et al.'s browsers make use of events in the user interface either through user (BM) or time-span (BA and TB) labeled folders [5]. All three browsers load the photographs to the PDA from a PC. This means that the

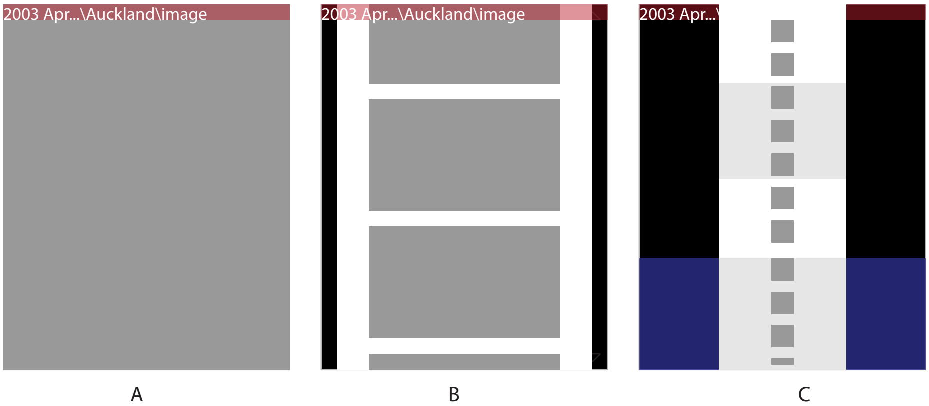

Fig. 3. The AutoZoom and GestureZoom (adapted from Patel et al. [6, 7])

organization of photographs in the BM's collection is made on the PC by the user. However, in the BA and TB the organization is made by preprocessing the metadata using the cluster algorithm. Both the BA and the BM indicates the size of the collection by two cues; (1) showing how many photographs a folder contains and (2) using a scroll list in the thumbnail view. To show the user where he or she is currently located in the collection, the BA and the BM use the red arrowheads. The TB uses two cues to advise the user of the size of the collection; (1) showing how many photographs an album consists of and (2) displaying the page number of the current showing thumbnail page. The current position in the collection is revealed by the time span text on top of the screen of the TB. The BM, BA and TB have a maximum number of thumbnails (5x6) that are shown at the same time in their thumbnail view to avoid visual clutter. No information is found in Harada et al.'s work that gives reason to the size of the grid. The TB also has a maximum number of album buttons showing at the same time in the timeline view (i.e. 10 album buttons).

Patel et al.'s browsers use events in their interface by showing the path to the folder in which the currently showing photographs are located [6, 7]. Both browsers load the photographs to the PDA from a PC, which means the organization of the collection is made by using the structure already applied to it on the PC. The AZ and the GZ uses no cues to advise the user of the size of the collection. Some indication is made of the user's position in the collection by showing the path to the current showing photograph. Since the user decides the zoom-level for both the AZ and GZ, visual clutter is no factor for neither browsers.

6 Discussion

All four investigated browsers were based on the principle that events are the most intuitive way for people to think about photo collections, although some of the browsers went to greater extent than others. For example Patel et al.'s browsers, although using events through their so-called calendar overview, were more supporting a mental model of time then events [6, 7].

Patel et al. state that a cluster algorithm may not be suitable to use, since the order and arrangement of photographs are subjectively correlated [6, 7]. What they mean is that it is not likely that any algorithm today automatically can extract clusters that coincide with what the user perceives as events. On the contrary, Harada et al. found through user experiment that the TB interface performs at least as well as the BM interface [5]. Since the TB uses automatic organizing and the BM uses manual organizing, this result is quite exhortatory. An interesting question to ask oneself when developing digital-photo collections for small-screen devices is; what is more time consuming for the user, organizing the collection or browsing the collection? Since manually organizing a digital-photo collection on a small-screen device is both time consuming and a disliked chore among photographers, maybe automatic organizing is the only option in the future. A fascinating new area in automatic organizing for manufacturers to explore is what possibilities the global positioning system (GPS) technologies in small-screen devices will bring. How much better can a cluster algorithm perform if it can extend its inputs by also taking spatial information as a parameter? Today, the most common way among manufacturers to support organizing is to let the user organize their collection on a PC and then upload it to the PDA. However, adding another platform means adding another step to the task which can easily lead to user frustration [17].

Harada et al.'s way of using a scroll bar in their interface is a small but very resourceful way of helping the user to understand the invisible space. Unfortunately, it is unclear if it is a proportional scroll bar. Using a proportional scroll bar can improve the estimation of the collection size even further [15]. Patel et al.'s interface uses no explicit cues to indicate the size of the collection. The reason, according to their work, is so they can investigate their so-called calendar overview. Although this is more than an acceptable reason to omit a scroll bar, maybe they a bit hastily overlooked the importance of helping the user to understand the invisible space of the application.

Harada et al. got the comment that the thumbnails in their browsers were too small from the majority of the subjects [5]. This is always a major problem when designing a thumbnail-based browser since the preference in thumbnail size is individually correlated. Both Patel et al.'s browsers works their way around this problem since the user easily can change between zoom levels with the stylus [6, 7]. They found in their research that every subject had a unique preference on zoom level when they hovered over the photographs. Maybe an option for interfaces that uses thumbnail is to enable the user to decide the size of the thumbnails and the grid so that it suites him or her. Without going any deeper, designers should be aware when resorting to a tailorable solution. Such a solution can bring about many other unforeseen circumstances, e.g. if the user is able to manipulate this, why should he or she not be able to alter that? Another relatively unexplored but still highly interesting area is that of expanding the interface to make use of a third dimension to avoid visual clutter. A third dimension would almost certainly demand more from the user in a cognitive perspective, but maybe adding depth to the screen is not such a bad idea in dealing with the ever so present limitation in screen size.

7 Conclusion

Harada et al. has shown through their work that a digital-photo-collection browser using automatic organization coupled with the right kind of interface can compete with the efficiency of a digital-photo-collection browser using manual organization [5]. Patel et al., although they never compared their interface to another solution, showed that there is great potential in extending the way we think about and interact with small-screen devices [6, 7]. Both Harada et al.'s and Patel et al.'s browsers were only tested using collections containing roughly 1300 photographs [5–7]. For future work, designers should also keep in mind the growth of memory capacity and connectivity. Below is a list that summarizes different aspects to consider, when designing a digital-photo-collection browser for small-screen devices:

- **Events are the most natural way for people to think about photographs.** make use of events when designing the structure of the interface
- **Organizing photographs is a tedious task.** support an intuitive way for the users to manage and organize their collection
- **Users are sensitive to visual clutter.** if resorting to thumbnails, support the user to change the thumbnail environment so that it suits him or her
- **The size of the collections keeps on growing.** be sure that the solution is scalable and supports collections containing large amounts of photographs
- **Size and spatial position is hard to perceive in a large data structure.** help the user to understand the invisible space by giving cues about how big the data structure is and where he or she is positioned in that structure

References

1. Xie, X., Liu, H., Ma, W.Y., Goumaz, S.: Learning user interest for image browsing on small-form-factor devices. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems. (2005) 671–680
2. Patel, D., Marsden, G.: Customizing digital libraries for small screen devices. In: Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries. (2004) 234–238
3. MobRef.com: Mobile phones and devices reference guide (2008) `http://www.mobref.com/statistics/`, accessed 2008-02-22.
4. Nokia.com: Nokia nseries (2008) `http://www.nseries.com/nseries/v3/media/sections/products/tech_specs/en-R1/tech_specs_n70_en_R1.html`, accessed 2008-02-22.

5. Harada, S., Naaman, M., Song, Y., Wang, Q., Paepcke, A.: Lost in memories: Interacting with photo collections on pdas. In: Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on. (2004) 325–333
6. Patel, D., Marsden, G., Jones, M., Jones, S.: Improving photo searching interfaces for small-screen mobile computers. In: MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services. (2006) 149–156
7. Patel, D., Marsden, G., Jones, S., Jones, M.: An evaluation of techniques for browsing photograph collections on small displays. In: MobileHCI'04: Proceedings of the 6th International Symposium, Springer (2004) 132–143
8. Rodden, K.: Evaluating similarity-based visualisations as interfaces for image browsing. Technical report, University of Cambridge, Computer Laboratory, Technical report (2002)
9. Rodden, K., Wood, K.: How do people manage their digital photographs? In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press New York, NY, USA (2003) 409–416
10. Frohlich, D., Kuchinsky, A., Pering, C., Don, A., Ariss, S.: Requirements for photoware. In: Proceedings of the 2002 ACM conference on Computer supported cooperative work, ACM Press New York, NY, USA (2002) 166–175
11. Graham, A., Garcia-Molina, H., Paepcke, A., Winograd, T.: Time as essence for photo browsing through personal digital libraries. In: Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries, ACM Press New York, NY, USA (2002) 326–335
12. Kärkkäinen, L., Laarni, J.: Designing for small display screens. In: Proceedings of the second Nordic conference on Human-computer interaction, ACM New York, NY, USA (2002) 227–230
13. Norman, D.: The design of everyday things. New York: Doubleday (1990)
14. Chittaro, L.: Visualizing information on mobile devices. Computer **3**(2) (2006) 40–45
15. Zwick, C., Schmitz, B., Kühl, K.: Designing for Small Screens. Ava Publishing (2005)
16. Jones, M., Marsden, G.: Mobile Interaction Design. John Wiley & Sons (2006)
17. Preece, J., Rogers, Y., Sharp, H.: Interaction design. Apogeo Editore (2004)

Tailorable applications for mobile devices

Christian Olander

Department of Computing Science
Umeå University, Sweden
dit04cor@cs.umu.se

Abstract. Advancements in mobile technology have led to an increased supply of devices and applications for both professional as well as non-professional use. Users of these devices have different skills and requirements, and the context in which the device is being used makes some levels of tailoring a necessity. Two important questions are how and to what extent tailorability should be provided on a mobile device. To answer these questions this paper presents a study of different motivations for providing tailorability in software systems, followed by a brief overview of the technical aspects to be considered when implementing such a service. Results show that apart for low-level customization of the interface, some situations require more complex tailoring activities. While complex, these activities can still be carried out by users with no experience in programming, by using an appropriate metaphor-based GUI.

1 Introduction

Applications for mobile phones, PDA:s and other handheld devices include more and more advanced features as the devices computational power increases. This increase of features makes it more difficult for designers to present interfaces and applications appropriate for all users in all situations. A solution to this problem could be to give users the possibility to tailor their own applications at run-time to suite their particular needs. The majority of mobile devices on the market today support low-level tailoring activities such as function keymapping and other interface choices [1], but ubiquitous computing environments [2] or applications that support collective activities require a higher level of tailorability [3]. This paper will discuss the challenges and opportunities of tailorable systems on mobile devices. The aim is to find out in what ways interacting with a mobile device enhance the need for tailorability, and to what extent this service should be provided and how.

To be able to present a discussion of tailorable applications on mobile devices, this paper first gives an introduction of tailorability and different motivations behind it. Then technological aspects of tailorability are presented, followed by a section focusing on the interaction with a mobile device.

2 Tailorability

The process of tailoring, customization and personification of computer systems and their interfaces has been discussed in several research articles [3–7]. This paper makes no difference between these three words and they are all used to describe the same activity. Blom and Monk [4] describes personification of a system as a process that changes the functionality, information content or distinctiveness to increase its personal relevance to an individual. Most importantly the authors stress that the effect of the changes should persist across sessions. If this were not so, Blom and Monk point out that almost anything we do with a system could be regarded as personalization. The process of changing the interface or the system behind it can be system- or user-initiated, the former is when the system recognizes that tailoring can be performed and presents this to the user [4]. In this article the changes will be considered as an activity of tailoring if they do not take place until the user authorizes them.

In her PhD thesis [5], Mackay distinguishes customization activities from normal use, which she describes as normal interactions with the software within the constraints defined by the software design. An example is given where a user interacts with a spreadsheet. If the user repeatedly specifies that a column should be 10 characters wide he or she is engaging in normal use. Changing the default value of the column width is customization. Mackay [5] further points out that tailorable software should not be confused with flexible software. Software is considered flexible if it offers the user a variety of ways to complete a task. When faced with multiple choices to complete a task, Mackay states that users often prefer a particular method over the others, and thus establishing a preferred pattern of use. Simply establishing patterns of use is not to tailor an interface or a system, there must be some use of a customization feature that continues to affect the software or the user's interaction with it in the future [5].

2.1 Work related motivations

Mobile devices and technology are used different than their stationary counterparts. People can now be literary anywhere at anytime and use an application on a mobile device [8]. Therefore context is an important factor when discussing tailorbility. Considering the dynamic nature of the environment, changes of lighting and noise levels for example, basic tailoring activities such as modifying text size and sound volume must be provided at runtime.

Appelt et al. [6] describes a general problem for interface design: The more functionality a system provides, the more buttons, menues, icons etc. have to be presented in the interface. An effective interface has easy and quick access to the systems functionality and data, but this will give a rather cluttered look considering the screen size on a handheld unit [6]. The authors claim that experienced users may not have a problem with a cluttered look as it can be the most efficient way to invoke functionality. For inexperienced users, elimination of features of no relevance to them means achieving a simpler interface and may eventually improve their effectivity in using the system [6]. Basically the authors state that different users have different needs. The solution to this problem according to the authors is to provide user interfaces that can be tailored by the end users to match their skills and work environments.

In an extensive study of Unix users at MIT's Project Athena [5], patterns of customization were found among the participants involved in the study. One of these patterns found was that users who customize like to maintain the same environment, even when the software changes. According to the article, the users will either try and tailor the new software to behave and look like the old or refuse to use the new software at all. Antother finding by Mackay [5] was that the most common reason for customization occurred when the users discovered that they were doing something repeatedly and chose to automate the process.

Newman et al. [2] describe a future scenario where computation is ubiquitously embedded in the environment, and in the objects we carry with us. With that, the authors imply that users will have a greater multitude of devices, and the range of interactions and interconnections between different kinds of devices will explode. In such a future, the authors believe that users will want to create configurations and combinations of these devices that is hard and maybe impossible to foresee. For such an environment with embedded computation, challenges rise for the interface designers. Newman et al. believe a semantic neutral, tailorable interface is preferable in system that operates in this dynamic environment [2]. Other research within ubiquitous computing focusing on the home environment has emphasised the need for allowing the end-users create and configure ubicomp applications to suite their particular needs [9–11]. The aim of these projects is not to present the user with a solution and for every need and task, but give the power to the user when integrating the home with advanced technology.

In areas such as group collaboration and shared workspaces, research focuses on building flexible UI:s for synchronous collaboration that can be dynamically customized to suite the given situation [1]. Groupware further increase the need to quickly configure software to adapt to a new meeting session or the introduction of new artifacts in a shared workspace.

2.2 Socially related motivations for tailorability

Furthermore there are also socially related motivations for tailorability as well as emotional motivations. In a 2-part study by Blom and Monk [4], a group of frequent Internet users where taught to tailor a commercial web portal, and discussion groups where then formed to let the users talk about their experiences. The second part of the study concentrated on mobile phones and a group of Finnish high school students. This group of students discussed the personalization process and motivations behind it. One of the results of the study was a checklist of desirable effects for assessing the need to include specific features for personalizing appearance.

– Ease of use.
– Being able to change the aesthetics of the design to fit personal preferences.

- Being able to recognize ones own copy of the product or system.
- Being able to reflect personal identity.
- Being able to reflect personal membership of some group.
- Feeling familiar with the system or product.
- The system of product feels personal.
- Feeling in control of the system or product.
- A feeling of ownership of the system or product
- Being fed up with the appearance of the product or system and wanting to repersonalize it.
- Having fun with the product or system.
- Making the user happy.
- Being emotionally attached to the system or product.
- Being able to accommodate current emotional state.

As the checklist shows, there are not exclusively functional or work related motivations for tailorability. Beyond the cognitive effects such as ease of use and being able to recognize ones own copy of the system, socially related motivations such as reflection of personal identity and reflection of membership of a group can be found. Further the authors states that purely emotional motivations should not be overlooked by software designers. Examples of emotional motivations are feeling familiarity with the system, and in control over it. The concept and importance of emotional design was introduced by Norman [12].

Mugge et. al [13] has studied the effects on users personalizing a product's appearance, and concludes that when a person invests energy in the product, and the product is used to express his/her self, this positively affects the bond or degree of attachment experienced by the user. This can be used for example to increase a product's lifetime [13].

3 Technical aspects of tailorability

3.1 Architectural level

A commonly used infrastructure for providing tailorability is a component-based approach [3]. A component can be seen as a program building block that can be combined and reused with other components to form an application. Examples of a component include a button in a GUI or something with even larger functionality such as a small program itself [14]. These components or software modules can be used in a variety of ways for different purposes [2]. A strenght with using a component-based approach is that a user might not need to know all the details of the specific modules in order to use them. All the users need is a way to access the services of the components through the interfaces. The visibility of the source code behind the software modules can reach from no accessibility to full accessibility of the code [3].

The FreEvolve platform [3] uses a client-server model. The platform runs on a server with one or more clients. The different component definitions are stored on the server and are are instantiated and connected during start up on their respective target machines. The platform provides an API (Application Programming Interface) with a complete set of component-based tailoring operations for the already running application. The systems uses components they call FlexiBeans, which are based on JavaBeans and implemented in Java. These FlexiBeans are stored on the server and can be retrieved and used in run-time.

Davis et. al [15] presents HYP, a system for application tailoring on mobile devices. They are using Java 2 Platform Micro Edition (J2ME), mainly because that platform is targeted to consumer electronics and embedded devices. This solves certain issues like screen size, when using the high-level API [15]. The authors do stress the point that J2ME is not the entire solution. They explain that though many devices are capable of using J2ME, most programs need to be packaged for their own environment outside of the mobile client. One of their approaches for providing tailorability is letting a translator search a structured XML document and generate J2ME source code. Java servlets are then used to compile the code and package it for the specific device. By running the application tailoring on a server the results will be shown at run-time. This system requires the user to be comfortable in editing an XML-document.

Both FreEvolve platform [3] and the HYP system [15] use a client-server approach which makes the user dependent on a connection to the server at all times. This can be a limitation considering changing connection possibilities for mobile devices.

3.2 Interface level

A system that provides tailoring activities benefit from having an easy an intuitive interface which supports exploration and is easy to learn, but it must also be efficient for all users regardless of their previous technical skills [6].

An example of an interface and technique that requires some programming experience to be effectively used is a command-line interface, where the user types textual commands to instruct the system. This could be a user interacting with a device running Unix and using its pipe system. The Unix pipe system [16] is a interprocess communication channel consisting of three main pieces: a streaming data source, a data pipe, and an endpoint which accepts the data. A user typically types a command string which indicates the programs to use as source and endpoint for the pipe. For example, in a Unix command line window, a user could type ”cat article.txt” | grep ”author” to pipe the contents of the article.txt file to a the small program ”grep” that displays any lines containing the string ”author” . When using different commands with the pipe system a user can develop powerful combination of different commands. Though the pipe system is powerful [16], it does but require some technical skill and is not a good choice for a novice user not familiar with a command-line interface, which is the case for many users that have only used graphical user interfaces.

Python an is an object oriented scripting language that provides a windowing system which can be used with small amounts of programming [2]. To make it easier for users inexperienced in programming some scripting languages allow users to compose predefined UI components into new programs. The ability to record actions into scripts as also provided by some environments [2].

The use of pipes and scripting languages as Python are powerful when used correctly, the logic and syntax of these textual languages can be hard to understand for a user with little or no skill in programming. To make tailoring less complex one can use a visual language instead of a textual to express applications [17]. This is called Visual Programming (VP). VP is often based around direct manipulation [17]. Direct manipulation depends on visual representation of the objects and actions available instead of a complex syntax, and the effects of the actions can be seen immediately [18]. In a study by Shneiderman the users reported several positive feelings when using a direct manipulation interface [18]. Some of them are listed below.

– Competence in performing tasks.
– Ease in learning the system originally.
– Enjoyment in using the system.
– Desire to explore more powerful aspects of the system.

In practice, many of the existing systems are using metaphor-based GUI interfaces to make the process of tailoring an application easier for end-users who have little or no experience [19]. For example, work by Humble et al. [20] uses a jigsaw puzzle GUI metaphor in which different components and actions are represented by puzzle piece-shaped icons. By snapping these icons together a user can build an application. The editor used in their system uses two distinct panels, one for showing the available components that can be used and an editing canvas. The components (puzzle pieces) can be dragged onto the canvas, which represents the "work area" for connecting pieces together and visualizing their tailoring activities.

The CAMP system [19] uses natural language for application tailoring since its user interface is based on a magnetic poetry metaphor. Magnetic poetry sets consist of small, flexible individual magnets, each of these magnets has a word printed on it. These magnets can be combined into different "poems" or statements. By using a constrained vocabulary the authors believes that it is clear to the users what their choices are, and what aspects of the system they can play with or configure [19]. An evaluation of CAMP made by the authors showed that inexperienced users could quickly and easily make complex statements, but they recognized that the CAMP interface cannot scale to display and parse an exhaustive vocabulary [19].

Another interface for building applications is the use of wizards or question-based configuration [2]. This style of interaction provide the user with a sequence of screens asking the user for input which can be used to describe what the user wants to achieve. Typically this choice of interaction is used when configuring a newly installed software [2]. An example of a system is the earlier mentioned HYP system [15], which allows users to create applications for ubiqutous environments on a mobile phone, specifying actions and conditions by navigating through screens of choices. This style of configuration is very constrained, but has the advantage that it generally requires very little technical skill [2].

4 Interacting with a mobile device

As mentioned earlier in this paper, a mobile device used in different environments is more context-sensitive compared to traditional desktop computers, and they are often used by only one individual and are therefore also more personal [21]. These differences makes users more likely to tailor the device and its applications to match their personal preferences as well as adapting to variations in the environment. A mobile device can be used and operated in different settings, and users can share their tailored applications or components via Bluetooth or other techniques for wireless data exchange. This can develop a tailoring culture by just by letting the users interact with each other. Maclean et al [7] stress the importance of building a community to enhance the positive effects that can be derived from learning and sharing between users, especially with varying technical skill levels.

With a wide range of mobile devices on the market today, there is no standardization of input-techniques. Desktop applications can typically rely on the presence of a mouse and a keyboard, but on mobile devices the interaction facilities are often device-specific [22]. On many devices there are no full alpha-numeric keyboard, which makes textual input slower and more frustrating compared to a full keyboard. Paelke et al. [22] also points out that alpha-numeric input are provided by indirect means such as handwriting character recognition and virtual keyboards. According to the authors additional problems arise and these techniques are often far less usable than traditional keyboards. Devices such as the Blackberry includes a smaller keyboard, but the keys are small and some users must learn to type with both thumbs. Data entry and error rates suffer from this approach [8]. Considering run-time textual customization of an application such as scripts or command-line piping on a mobile device, the experienced user must feel that the tailoring can be done effectively, and the slower textual input is a problem that need to be solved. Auditory input is becoming more popular as the voice recognition technology continues to improve [23], but privacy issues and changing environments makes must be considered, and even a perfect voice recognition system is not the best solution in all situations [22].

GUI:s on desktop computers often utilize a pointing device, typically the mouse, but also different pads or graphic table [22]. On a mobile device, different pointing devices have their limitations. A touch screen requires a large screen when operated by fingers, or an additional component such as a stylus pen for high precision. For tailoring using direct manipulation and drag-and-drop such as the CAMP system [19], a pointing device is needed. A variety of devices on the market today does not use a pointing device for navigation and manipulation of the GUI at all. This is a limitation since a system needs to be adjusted depending on the interaction-technique provided.

The size, resolution and color capabilities are the limitations of the screen on mobile devices. All are usually less than those on ordinary desktop computers [8]. Kamba et al. [24] conclude that the design of user interfaces for mobile devices must balance the need to make the devices (and with that the screens) small, and the need to keep the screen size as large a possible to show enough information so the device is useful and effective. The authors call the latter a ”functional” limitation. As for tailoring environments, especially direct manipulation programming, there is often two distinct panels as in [20, 19]. A work area and a panel containing the different sets of objects that can be manipulated and placed onto the work area. The objects in the interface must be large enough to be distinguished from each other to other type of information content displayed on the screen, which can be difficult on a a small screen with low resolution.

5 Results

Section 2.1 and 2.2 presented different motivations for customization. The results are presented in Table 1 using specific examples of tailoring tasks.

Table 1. Common tailoring activities.

Task	**Level of tailoring (low, medium, high)**
Changing text size and noise level	Low
Customize interface to avoid clutter	Low to medium
Adjust interface for group collaboration purposes	Medium to high
Configuration of a smart environment	High
Automate repeated tasks	High
Express personal identity and ownership of the system	Low
Exploring different features of the system	Low to high

Key findings regarding the technical aspects of tailorability and the limitations of mobile devices are listed below.

- A component-based infrastructure are often used to provide tailoring activities.
- A client-server based system makes the user dependent on a connection, but computation can be distributed to more powerful hardware.
- The use of Visual Programming makes high-level tailoring less complex for inexperinced users.
- Many of the existing systems that support complex tailoring uses a metaphor-based GUI with positive results.
- The size, resolution and color capabilities are the limitations of the screen on mobile devices. All are usually less than those on ordinary desktop computers.
- The lack of standardization of input-techniques on mobile devices makes it more difficult to design appropriate GUI:s for multiple platforms.

6 Discussion and conclusions

Interaction between humans and their mobile devices in a social setting enhance the need for tailorable applications. Findings presented by [4] include that tailoring activities could be a tool for the user to express membership of a group, and also to show personal identity within that group. In a setting such as a schoolyard or an office environment, these aspects of run-time tailorability should not be overlooked. Looking into a future scenario, research findings within ubiquitous computing implicate that smart environments requires end-user tailored solutions to facilitate everyday activities. In a future such as the one envisioned by [2], people use mobile technology to create these solutions.

When using mobile devices in the workplace, effectiveness when working together to achieve a common goal is essential. Shared workspaces for group collaboration requires these flexible applications [1]. Users within a tailoring culture benefit from being able to meet in person with other users, which is a specific strength of mobile devices as they can be used almost anywhere. In this way, users can easily share their applications, as well as get instant feedback and help from more experienced users. The downside of a tailoring culture is the possibility that some experienced users may not leave any room for other less experienced to learn and contribute with their own solutions. Being presented to technologically advanced programs can be intimidating to a novice user and have a negative effect on the tailoring culture.

When providing the ability to tailor applications on a mobile device, developers must balance the need of different users. The use of scripts and other sorts of textual customization may provide great flexibility and control to an experinced user, but it will be highly unlikely that a novice user will see it as an efficient tool. Using a metaphor-based GUI to tailor applications will not have the expressive power of textual programming [25], but considering the environments and motivations presented in this paper, might well be enough to meet the need of all users. Basic activities such as changing a ringtone or the size of text on a mobile phone are easily done using a few buttons. More complex tailoring activities such as configuring a smart home [9] or manipulating a shared workspace requires more interaction, but can still be done effectively with a GUI. However, the enviroment to tailor an application must still appeal to a more experinced, serious user, which according to [25] can find the semantics of clicking and dragging limited. A user with interest and experience in programming might want to manipulate and optimize code behind the components or software modules. Considering the input-limitations this will probably be done on a traditional desktop computer during design-time, and then transferred to the mobile device.

The limitation of the screen and resolution is always a problem for applications on mobile devices and is not specific for environments that supports tailoring. The lack of standards over different platforms [15] makes it harder for a component-based approaches, and users must be aware that components may not be compatible with their system. Compared to ordinary applications on a mobile device such as a calendar or a calculator, a system that is to be used to tailor other applications demands no less of the screen or the technique used for input. One of the hardest challenges is to provide the users with the appropriate tools needed to explore and play around with system. This exploration of benefits in tailoring activities can hopefully lead to personal advancements [5] regardless of technical skill. That is why further research in this area could bring focus to building systems that highlights the positive benefits from a tailoring community, and find proper metaphor-based GUI:s that makes it easy for all kinds of users to start tailor their applications.

References

1. Marsic, I., Dorohonceanu, B.: Flexible User Interfaces for Group Collaboration. International Journal of Human-Computer Interaction **15**(3) (2003) 337–360
2. Newman, M., Sedivy, J., Neuwirth, C., Edwards, W., Hong, J., Izadi, S., Marcelo, K., Smith, T., Sedivy, J., Newman, M.: Designing for serendipity: supporting end-user configuration of ubiquitous computing environments. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques (2002) 147–156
3. Won, M., Stiemerling, O., Wulf, V.: Component-Based Approaches to Tailorable Systems. End User Development, Kluwer Academic (2004)
4. Blom, J., Monk, A.: Theory of Personalization of Appearance: Why Users Personalize Their PCs and Mobile Phones. Human-Computer Interaction **18**(3) (2003) 193–228
5. Mackay, W.: Users and customizable software: a co-adaptive phenomenon. PhD thesis, Massachusetts Institute of Technology, Sloan School of Management (1990)
6. Appelt, W., Hinrichs, E., Woetzel, G.: Effectiveness and efficiency: the need for tailorable user interfaces on the Web. Computer Networks and ISDN Systems **30**(1-7) (1998) 499–508
7. MacLean, A., Carter, K., Lövstrand, L., Moran, T.: User-tailorable systems: pressing the issues with buttons. Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people (1990) 175–182
8. Tarasewich, P.: Wireless Devices for Mobile Commerce: User Interface Design and Usability. Mobile Commerce: Technology, Theory, and Applications. Idea Group Publishing, Hershey, PA (2002)
9. Barkhuus, L., Vallgårda, A.: Smart Home in Your Pocket. Adjunct Proceedings of UbiComp (2003) 165–166
10. Dey, A., Hamid, R., Beckmann, C., Li, I., Hsu, D.: a CAPpella: programming by demonstration of context-aware applications. Proceedings of the SIGCHI conference on Human factors in computing systems (2004) 33–40
11. Hague, R., Robinson, P., Blackwell, A.: Towards Ubiquitous End-User Programming. Adjunct Proceedings of UbiComp (2003) 169–170
12. Norman, D.: Emotional Design: Why We Love (or Hate) Everyday Things. Basic Books (2004)
13. Mugge, R., Schifferstein, H., Schoormans, J.: Personalizing Product Appearance: The Effect on Product Attachment. Proceedings of the International Conference on Design and Emotion Ankara, Turkey (2004)
14. Dewan, P., Choudhard, R.: Flexible user interface coupling in a collaborative system. Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology (1991) 41–48

15. Davis, V., Gray, J., Jones, J.: Generative approaches for application tailoring of mobile devices. Proceedings of the 43rd annual southeast regional conference-Volume 2 (2005) 237–241
16. Stevens, W., Rago, S.: Advanced programming in the UNIX environment. Pearson Education,Inc (2005)
17. Hague, R., Blackwell, A., Robinson, P.: End-User Programming in the Networked Home. Proceedings of the 1st Equator Workshop on Ubiquitous Computing in Domestic Environments (2001) 46–67
18. Shneiderman, B.: Direct manipulation for comprehensible, predictable and controllable user interfaces. Proceedings of the 2nd international conference on Intelligent user interfaces (1997) 33–39
19. Truong, K., Huang, E., Abowd, G.: CAMP: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home. Proceedings of Ubicomp 2004 (2004) 143–160
20. Humble, J., Crabtree, A., Hemmings, T., Åkesson, K., Koleva, B., Rodden, T., Hansson, P.: Playing with the Bits-User-configuration of Ubiquitous Domestic Environments. Proceedings of the Fifth Annual Conference on Ubiquitous Computing, UbiComp2003, Seattle, Washington, USA (2003) 12–15
21. Gong, J., Tarasewich, P.: Guidelines for Handheld Mobile Device Interface Design. Proceedings of DSI 2004 Annual Meeting (2004)
22. Paelke, V., Reimann, C., Rosenbach, W.: A visualization design repository for mobile devices. Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa (2003) 57–62
23. Melin, P., Urias, J., Solano, D., Soto, M., Lopez, M., Castillo, O.: Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms. Engineering Letters **13**(2) (2006) 108–116
24. Kamba, T., Elson, S., Harpold, T., Stamper, T., Sukaviriya, P.: Using small screen space more efficiently. Proceedings of the SIGCHI conference on Human factors in computing systems: common ground (1996) 383–390
25. Eisenberg, M.: Programmable applications: interpreter meets interface. ACM SIGCHI Bulletin **27**(2) (1995) 68–93

Aesthetics and initial perceived usability in a mobile user interface

Fatima Boukarchid

Department of Computing Science
Umeå University, Sweden
kv02bfa@cs.umu.se

Abstract. This paper explores the relationship between aesthetics and usability in mobile graphical user interfaces. This was done by investigating the effect of aesthetic appeal for a certain aspect of usability that occurs before usage of an interface - the initial perceived usability. Two mobile graphical user interfaces with different aesthetic properties were developed for a user test with the help of aesthetic principles from previous research, one being "beautiful" and one "ugly". Both interfaces had identical functionality and graphical components. The results generated from the pilot study with 20 participants divided into two groups suggest that aesthetically appealing interfaces are more likely to be considered easy to use than less aesthetically appealing interfaces. More test subjects, i.e. a larger sample size, are needed to make a reliable statistical analysis and to make any firm conclusions.

1 Introduction

The continuous advance of technology has lead to humans being more and more surrounded by technology in everyday situations. Computers of today have become more portable and are frequently used. Since these kinds of products gradually have gained more functions and thus become more complex, increased demands have been brought forward concerning the usability of the products. It has become more important for the products to be simple to use and consequently usability has been recognized as an important factor in creating graphical user interfaces (GUIs), because if an interface is too complex and difficult to interact with the user will lose interest for the product. Therefore, usability has traditionally been viewed as the single most important factor in the success of a design in the area of human-computer interaction (HCI), and hence, aspects of usability have been a primary source for interface design [1, 2]. In the area of usability the focus has been mainly towards efficiency (e.g. performance time and error rates) while other factors have been more or less overlooked [3, 2]. This is unfortunate given that important factors like aesthetics, emotion and expectation have been highly disregarded. It might appear strange since properties such as these intuitively are understood to play important roles in the success of a design. However, these factors have increasingly become noticed. It has been realized that efficiency is something that no longer can be considered to be the only factor that affects user satisfaction [1, 4, 2].

Research has been done to determine the role of aesthetics in usability, i.e. how it affects the perceived usability. In a study by Kurosu and Kashimura [5] a correlation was discovered between perceived usability and aesthetics, a relationship that was significant during initial perceptions of the system. This correlation was later confirmed by Tractinsky [3, 6], who also showed that this relationship, even if not equally strong, lasts after system usage. The motivation for this article comes from this previous research. If there is a connection between the visual appearance of a GUI and its perceived usability, aesthetics should be considered in GUI design. Furthermore, if there is a correlation, perhaps the principles of aesthetics should somehow be integrated in the area of HCI. Consequently, this paper will be dealing with the influence of aesthetics in the design of graphical user interfaces for mobile phones. Mobile phones have been chosen because they are commonly used throughout the world and reflect the technological development towards increasingly portable units. An experiment will be performed comparing the GUIs of two mobile phones where the main research question is: if a user interface is considered aesthetically appealing by the user how does it affect the user's initial perceived usability in an interface?

The subsequent sections of this paper begin with Section 2 that contains background information to the subject. This section gives brief introduction to usability and aesthetics. Thereafter it presents some previous research in the area. The following Section 3 introduces the methods used in the experiment comparing two mobile phone GUIs. Next in Section 4 the results are presented. In Section 5 these results are discussed and in Section 6 we present our conclusions.

2 Background

2.1 Usability

In HCI usability has traditionally been viewed as a primary factor in the success of a design and has therefore also been an important influence on guidelines for interface design [1, 2]. Despite some evident agreements concerning the notion, usability has been interpreted and defined in many different ways, and has thus been assigned a variety of explanations largely depending on the author. For instance, Nielsen describes usability as a set of different attributes that is a small part of a larger context. Nielsen describes this context as the system acceptability, which essentially correspond to the question "whether the system is good enough to satisfy all the needs and requirements of the users and other potential stakeholders, such as users' clients and managers" [7]. Another definition of usability is stated by Dix et al. [8]. Here usability is explained as the result of three main factors; learnability, flexibility and robustness, that all are further divided into subfactors.

It is obvious that these definitions of usability differ from each other, and that a uniform explanation of the actual meaning of usability is lacking. Thus a standard definition has been brought forward, ISO 9241-11 [9], in which usability is defined as follows: *"Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."*, which are further defined as follows:

- Effectiveness: *"Accuracy and completeness with which users achieve specified goals."*
- Efficiency: *"Resources expended in relation to the accuracy and completeness with which users achieve goals."*
- Satisfaction: *"Freedom from discomfort, and positive attitudes towards the use of the product."*
- Context of use: *"Users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used."*

The biggest advantage with this definition is that it provides a broad, unified and concrete explanation of what usability really is. Usability is seen as a measurable unit consisting of a vast amount of aspects such as system perspective, user perspective and context of use. The definition also considers user satisfaction and therefore acknowledges factors like aesthetics, emotion and expectation.

However, the usability that is to be tested in this study is usability before usage of the system, i.e. the initial perceived usability. It could be argued that usability is not measurable before system usage, but Tractinsky refrains from this point of view by pointing out the extensive influence first impression has on attitude formations [6]. Furthermore Tractinsky reasons that there is no reason to assume that such attitude formations does not apply to the HCI domain. This claim is supported in a study made by Hiltz and Johnson [10] described by Tractinsky [6]. In this study it was discovered that "if computers were perceived initially as difficult to use, users were more likely to express dissatisfaction with the interface of the system after four months of use".

2.2 Aesthetics

The founder of aesthetics as a discipline was a German philosopher by the name of Baumgarten (1714-1762). He considered that the science of philosophy should be complemented by something he called "inferior cognition" described to be mediated by the senses. The Greek name for aesthetics used by Baumgarten "aisthanesthai" stands for "to perceive". As the name implies, the definition does not only include what is perceived by vision but rather the full spectrum of human perception. Baumgarten claimed in one of his works "Aesthetica" that the "perfection of sensory awareness" only could be found in the perception of beauty and thus aesthetics became the theory of beauty [11]. Aesthetics has thereafter been a definition with a wide variety of meanings largely depending on different schools of thought [2]. Ambivalence in the expression has had negative effects on the comprehensiveness of the definition. Depending on the approach and the interpretation, aesthetics as a term is used for different areas such as beauty, arts, nature, knowledge and perception [12]. Visual aesthetics concerning beauty and the study of beauty [2] is the definition of aesthetics that will be used in this study. Thus, this definition is most relevant for GUIs which are clearly visual.

The general assumption of aesthetics is that it is strictly subjective and not measurable. However through research in this area several steps has been taken in an attempt to concretize aesthetics. Tractinsky [2] describes this research as being roughly divided into two separate directions; the experimental approach and the exploratory approach. It is mainly what the analysis is based upon that separates these approaches from each other. The exploratory approach is concerned with evaluation of complete and ecologically valid stimuli (such as building, landscapes and works of art) and is thus concerned with the whole. What is of interest in this approach are subjective opinions of aesthetics. The experimental approach follows a bottom-up methodology. Here analysis of aesthetics is based on smaller parts that constitute a foundation for research of the effects on the human aesthetic response. One of the most significant purposes of the experimental approach was to identify laws concluding human aesthetic preferences, using a scientific course of action. Tractinsky further describes how there has been some criticism to the experimental approach, based mostly on the absence of consideration for the whole and substantial focus on isolated parts. This research assumption contradicts the laws of gestalt theory [13] where the whole is considered to be of greater importance than the sum of its parts.

One dimension of perceived aesthetics recognized from research within the exploratory approach is classical aesthetics [2] (further explained in 2.3). It is considered to have a particularly strong relation to perceived usability. Thus it is of special interest in this study and will be used to assist in the design of the GUIs that will be used for the user tests.

2.3 Previous research

Aesthetics has increasingly become recognized as a part of usability. An example of early research aiming to explore the relationship between usability and aesthetics is a study conducted in Japan by Kurosu and Kashimura [5], which highlights the importance of usability in the initial perception of a product. Their view was that initial perceptions of usability, which they called "apparent usability", affected the possibility to sell the product. Therefore the affecting variables of the initial perceptions were examined. The relationship between the experience of beauty and apparent usability was also examined, using different layouts of automatic teller machines (ATM:s). The results showed a high correlation between these two factors, something that suggested that there was a connection between aesthetics and usability. To confirm this result Tractinsky [6] performed a similar study in Israel. The purpose was to eliminate all potential method bias and to verify that no cultural differences of the test subjects had affected the result. Tractinsky's opinion was that the subjective experience of aesthetics clearly was a cultural factor and therefore the effect of aesthetics on usability also should vary between cultures. According to him the Israeli culture valued aesthetics less than the Japanese culture. Tractinsky argued for these differences by proclaiming that the Israeli culture was known for its "action orientation" and that the Japanese culture was known for its "aesthetic tradition". Therefore the relationship between aesthetics and usability should not be as strong in this new study. The result from the new study was surprising. Opposite to the expectations, the correlation between beauty and usability was a lot stronger. In a more recent study Tractinsky et al.[3] confirmed that the relationship between aesthetics and usability was not only valid before usage of the system, but also after usage of the system. According to Dion et al. [14] as cited by Tractinsky [3] the relationship between aesthetics and usability was similar to the social psychological phenomenon "what is beautiful is good", i.e. people ascribe social desirable characteristics to other people who they find attractive. It is assumed that attractive people are more social, happier and more successful than unattractive people. In the article two suggested explanations to the phenomenon by Dion et al. was presented. The first explanation is that there exists a stereotypical behavior, where beauty is associated to other human characteristics. The other explanation is "the halo effect". This explanation suggests that since human beauty is the most obvious and apparent human property, it is noticed first and thus it affects later perceptions of the individual.

Studies such as those described earlier have acknowledged aesthetics and clearly influenced the way it is increasingly being recognized as an important factor in the area of usability. Several subsequent studies have followed, some of which are concerned with putting together principles of aesthetics. One example of such research is a study by Lavie and Tractinsky's [2]. Here the current absence of measurement for aesthetics is pointed out as a possible limitation for research within this area. Using this as motivation Tractinsky et al. has tried to develop a "measurement instrument of perceived web site aesthetics". Based on the aesthetic exploratory approach the user experience of web sites was researched. In the study two dimensions of aesthetics that users experience were defined, classical aesthetics and expressive aesthetics. Classical aesthetics refers to the aesthetic concepts from the antiquity to the 18th century. This kind of aesthetics could be described in a five-item scale with the attributes "aesthetic design", "pleasant design", "clear design", "clean design" and "symmetrical design". Expressive aesthetics is a wider concept that reflects the designers' ability to think creatively. This dimension could be described in a scale with the attributes "original design", "creative design", "fascination design", "use of special effects" and "sophisticated design". In this study Tractinsky et al. also exposed that classical aesthetics is more strongly connected to perceived usability than expressive aesthetics. The classical dimension of aesthetics is therefore the dimension that will be used in this study for the design layout of the mobile GUIs.

Another example of a research trying to map and involve aesthetics into usability is a study conducted by Sutcliffe [15]. In this study heuristics determining attractiveness for web based GUIs were introduced. These heuristics were used together with existing heuristics for usability to form a new method for web site evaluation. Sutcliffe wanted to incorporate attractiveness and aesthetics as an obvious part of HCI, rather than just leaving this aspect for the graphical designer to handle. The proposed heuristics were a step towards this goal. It could both be used by designers to help build attractive GUIs and as an evaluation criterion for existing user interfaces. The heuristics for attractiveness were applied in a user evaluation of three different airline websites, to determine the role of aesthetics for website usability. Some of these guidelines used in the study concerned use of color, symmetry, structure for layout and depth of field. For example, according to the applied heuristics, color use should be balanced and there should not be more than 2-3 fully saturated colors. Also a symmetrical visual layout should be used and the use of curved shapes in contrast to rectangles conveys an attractive visual style. The depth of field heuristics claims that usage of layers in an image stimulates interest and can thus contribute to attain a peaceful effect. From the evaluation using these heuristics Sutcliffe concluded that explicit consideration of aesthetics could affect judgment of the quality of websites (e.g. usability problems in attractive web sites might partly be forgiven). Because of such evident effects it was further concluded that it is important to be able to assess usability and attractiveness together, and that the study was a step in the right direction.

There have also been other subsequent studies were it has been revealed that the relationship between aesthetics and usability is not as simple as it perhaps seems. For example, De Angeli et al. [1] evaluated two web pages, one of them being metaphor-based (displaying site content using animated interactive metaphors) and the other one menu-based. These two pages were evaluated on several of factors, of which two were usability and aesthetics. The analysis of these two factors was among other things based on questionnaires measuring subjective opinions concerning usability and questionnaires measuring perception of classical and expressive aesthetics. The result showed that both web pages had received equal ratings on the classical aesthetic scale, but that the metaphor-based web page gained a higher rating on the expressive scale. Yet, the menu-based web page had received higher usability scores. According to De Angeli et al. this indicated that the relationship between aesthetics and usability probably is more complex than what has been indicated in earlier studies by researchers like Tractinsky. Thus, this result implies that more research in the area is needed to acquire more substantial knowledge regarding the relationship between usability and aesthetics.

3 Method

3.1 Participants

The participants in the study were 27 students of Umeå University in Sweden. They were divided into two groups, the first group had 14 participants and the second group had 13 participants. The number of participants was eventually reduced to 20 participants. This lead to 10 participants left in both groups. The overall age of these two reduced groups ranged between 21 and 29. The mean age was 23.5 years for the first group and 25.1 years for the second group. The distribution between the genders was 40/60 (percent men/women) in the first group and 70/30 in the second group after the reduction. The entire group of participants used mobile phones on a daily basis.

3.2 Equipment

The equipment consisted of a questionnaire and two mock-up images of a touch screen mobile phone, each displayed on a computer screen. The first image (Figure 1) was the "beautiful" GUI and was the actual user interface of a HTC Touch mobile phone.

Fig. 1. The "beautiful" GUI is presented to the left and the "ugly" GUI is displayed on the mobile phone to the right.

This interface was selected as the beautiful interface because of how it corresponded to the scale of classical aesthetics (mentioned in Section 2.3), which according to Tractinsky [2] was particularly closely related to usability. It had a symmetrical, clean, clear and pleasant design. The second image (Figure 1) was an image of the "ugly" GUI. This interface had exactly the same functionality as the other interface, but had been made unattractive with the help of the heuristics for attractiveness by Sutcliffe [15] (e.g. the heuristics concerning usage of color, symmetry and depth of field described in Section 2.3). Some of these heuristics corresponded well with the aesthetics of the first interface and therefore the heuristics were reversed to create this second ugly interface, because if the heuristics were made for attractiveness they would have the opposite effect reversed. It could be claimed that classical aesthetics and the heuristics are not valid for mobile phones because these aesthetic principles were made for web pages. Web pages have larger GUIs, are used for different purposes and are possible viewed in a different light. But in some aspects mobile phones are similar to web pages, for instance they both use computer based graphics like icons and images to presents their content. Therefore, the more general principles addressing the graphical style were applied in a reversed manner on the mobile interface.

3.3 Procedure

Data was collected individually with each participant of the test sitting in front of a computer. In the beginning of the test the participants were told to imagine that they were in a store to buy a touch screen mobile phone and that the interface displayed on the computer was the interface of this phone. Thereafter they got to watch the GUI on the computer screen. The participants in the first group were presented with the “beautiful” interface and the participants in second group were presented with the “ugly” interface. They were also given the questionnaire. In the questionnaire they were first asked to rate the interface after how aesthetically appealing they found it on a Likert-scale ranging from 1 to 5, were 1 corresponded to ugly and 5 to beautiful. Next they were asked to rate the interface after perceived ease of use on a scale from 1 to 5, where 1 one corresponded to difficult to use and 5 to easy to use. The participants in the first group who had rated the “beautiful” interface beneath 4 on the scale were removed from the test group. The participants in the second group who had rated the “ugly” interface above 2 were also removed. This was done because the basic condition for determining the effects of an aesthetically appealing GUI on the initial perceived usability is that the interface is actually considered aesthetically appealing. The opposite pertains to the "ugly" GUI. If there had been only one test group these removals could have been a problem for the validity, but since there are two groups in this study this is not a problem. Furthermore, participants who had previously used a HTC Touch phone were also removed from the test group to avoid a biased result. In total these removals lead to 10 participants left in each test group.

4 Results

The results from the questionnaire used in the user tests are presented in Figure 2. Mean values for the questions were calculated and presented side by side for each of the two evaluated graphical user interfaces. The mean value of the aesthetic appeal for “Beautiful” GUI is 4.1 compared to 1.5 for the “Ugly” GUI. Similarly the usability for the “Beautiful” GUI is 3.6 compared to 2.4 for the “Ugly” GUI. The response frequency for the different possible values on the Likert-scale is presented in Figure 3.

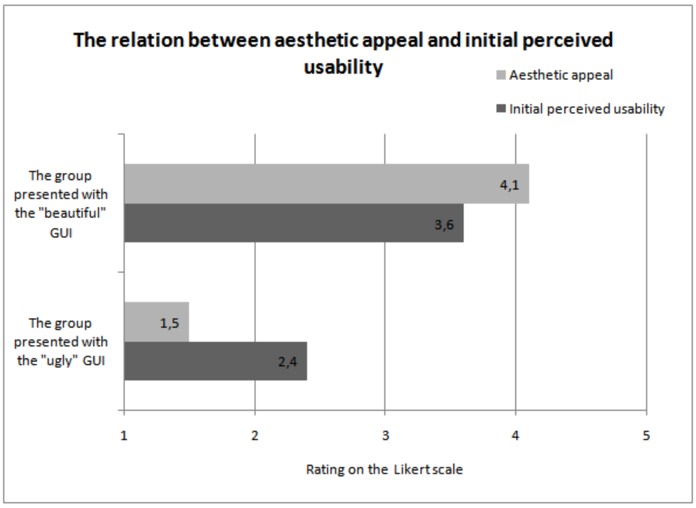

Fig. 2. The diagram describes the investigation of the relation between aesthetic appeal and initial perceived usability. The y-axis presents the two GUIs and the x-axis presents the mean value of the measured factors for each GUI.

5 Discussion

Because of a relatively small number of test subjects in the user test (sample size 10 for each test group) it can be reasoned that the study does not provide a satisfactory foundation for a detailed statistical analysis of the results. However, by using mean value calculations as a way to present the collected data (Figure 2) an indication of a trend emerged. As previous studies by Tractinsky and also by Kurosu and Kashimura [3, 5, 6] have confirmed, there seems to be a relationship between aesthetics and initial perceived usability. The mean value for the initial perceived usability in the evaluation of the “beautiful” interface was 3.6 and for the “ugly” interface only 2.4. This result clearly indicated a difference in the participant’s perception of the usability between the two different GUIs and thus it seems like this relationship between usability and aesthetics also is valid for mobile phones. Though, the range of the ratings for the initial perceived usability in the “ugly” interface is quite wide (between 1 and 4 on the Likert-scale) as displayed in Figure 3 and therefore it would have been better if the sample sizes for the test groups had been larger. This would have provided a more certain

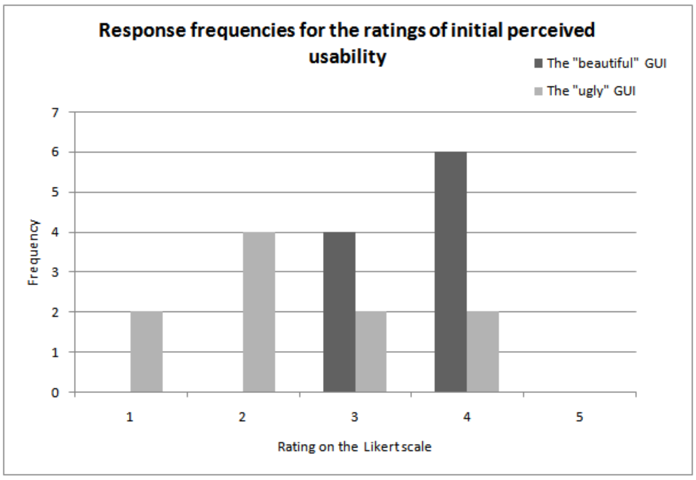

Fig. 3. The diagram describes the response frequency for the ratings of inital perceived usability for all 20 participants. The y-axis presents the number of responses for the different values on the Likert-scale presented on the x-axis. The color of the bars determines which GUI the responses concern.

result and a better foundation for more thorough statistical analysis. Hence, if the test is to be replicated in the future it is strongly recommended to use a large test group. The low rating of the aesthetic appeal for the "ugly" GUI (mean value 1.5) can be seen as an indication that Sutcliffe's heuristics [15] that were used for the GUI design did affect the results in the way they were anticipated to. This implies that there exist more generally applicable principles for aesthetics in GUI design that account for different kinds of GUIs. It also suggests that the heuristics for attractiveness can be used in a reversed manner to attain an opposite effect. These findings have however not been a primary goal in this study to investigate; the principles were merely used to assist in the design of the GUIs for the user tests. Something that could have affected the results of the study was the homogeneous test group of participants. They were all students with a narrow age range only between 21 and 29, which makes the results more difficult to generalize. Thus more extensive user testing with a greater span of age and occupation should provide a more reliable result.

6 Conclusion

The result from this study is an indication of the relationship between aesthetics and initial perceived usability as shown by researchers indeed also accounts for mobile phones GUI. Though the test groups in this study were too small to say something definite about the results and the homogeneous test group makes the results difficult to generalize. For that reason it is necessary to conduct future research with larger and heterogeneous test groups to receive a more reliable result. Also more research should be done in the future to fully explore how aesthetical principles can be generalized to different kind of GUIs, like mobile phones.

References

1. Angeli, A.D., Sutcliffe, A., Hartmann, J.: Interaction, usability and aesthetics: what influences users' preferences? In: DIS '06: Proceedings of the 6th conference on Designing Interactive systems. (2006) 271–280
2. Lavie, T., Tractinsky, N.: Assessing dimensions of perceived visual aesthetics of web sites. Int. J. Hum.-Comput. Stud. **60**(3) (2004) 269–298
3. Tractinsky, N., Katz, A. S., and Ikar, D: What is beautiful is usable. Communications of the ACM **13**(2) (2000) 127–145
4. Hartmann, J., Sutcliffe, A., Angeli, A.D.: Investigating attractiveness in web user interfaces. In: CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems. (2007) 387–396
5. Kurosu, M., Kashimura, K.: Apparent usability vs. inherent usability: experimental analysis on the determinants of the apparent usability. In: CHI '95: Conference companion on Human factors in computing systems. (1995) 292–293
6. Tractinsky, N.: Aesthetics and apparent usability: empirically assessing cultural and methodological issues. In: CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems. (1997) 115–122
7. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
8. Dix, A., Finlay, J.E., Abowd, G.D., Beale, R.: Human-Computer Interaction (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2003)
9. International Organization for Standardization: Iso 9241-11: Ergonomic requirements for office work with visual display terminals (vdts) - part 11: Guidance on usability (1998)
10. Hiltz, S.R., Johnson, K.: User satisfaction with computer-mediated communication systems. Manage. Sci. **36**(6) (1990) 739–764
11. Saw, R., Osborne, H.: Aesthetics as a branch of philosophy. The British journal of aesthetics (1960)
12. Welsch, W.: Undoing aesthetics. Sage Publications Ltd, London (1997)
13. Eysenck, M.W., Keane, M.T.: Cognitive Psychology: A Student's Handbook. Psychology Press (UK) (2000)
14. Dion, K., Berscheid, E., Walster, E.: What is beautiful is good. Journal of Personality and Social Psychology **24**(3) (1972) 285–290
15. Sutcliffe, A.G.: Heuristic evaluation of website attractiveness and usability. In: DSV-IS '01: Proceedings of the 8th International Workshop on Interactive Systems: Design, Specification, and Verification-Revised Papers. (2001) 183–198

Usability in Game Design Literature

Anders Engqvist

Department of Computing Science
Umeå University, Sweden
dit04aet@cs.umu.se

Abstract. Traditionally usability has not been a concern in game design. Game designers made games for themselves instead for the people that bought and played the games. This has changed in recent years and game designers have adopted a more player- approach. A few studies have examined the benefits and the importance of usability evaluation of games. The studies have concluded that games can benefit from being designed for usability. This paper gives an insight of the notions of usability among game designers by analyzing game design literature. Five books were analyzed in this study. The analysis showed that not much was explicitly written about usability. All authors recommended the designer to prototype the game and evaluate it iteratively. Not much was however written about different methods that could be used to evaluate the game. From the data collected in the analysis, a list of game design heuristics were compiled.

1 Introduction

Usability in game development has been quite overlooked for a long time [1]. But in recent years there have been attempts to integrate usability in game design. There is now growing evidence that usability methods are useful in game design [2, 3].

When video games became popular, more and more books on how to design them found their ways to the bookshelves. Several universities now offer courses in game design. This paper will take a look at the opinion of usability and player-centred design in game design literature. It will examine the usability guidelines that the authors of the literature use, and also investigate the suggested methods on how to evaluate games. Finally the paper presents a list of usability heuristics for the evaluation of video games. The heuristics are compiled from the data and guidelines found in the literature.

There are two reasons for studying the literature in game design. The first reason is that game design literature is written by people working in the game industry. The second reason is that this literature is used in schools teaching game design i.e. future designers will probably take part of the information in some of this literature. The chance that usability will successfully be integrated in the game industry increases if the people working in the industry have an understanding of usability.

This paper begins with an introduction to usability and usability in video games. Then it will give an overview of the related work in the field. The remainder of the paper presents the analysis of the literature and then introduces the compiled list of game usability heuristics. The paper ends with a short conclusion and discussion.

2 Background

2.1 Usability

The term usability was introduced to replace the more vague term "user friendly" [4, 5]. The aim with usability is to improve interactive systems and user interfaces. Usability in itself is an abstract concept; it is hard to measure, so therefore it is broken down to more measurable attributes. These attributes fall into two categories: the user's objective performance and the user's subjective satisfaction with the system. The exact definition of usability varies among different standards and authors [6]. This is because different authors and groups do not have the same view on how to measure usability. According to Nielsen [4, p. 26] a usable system should be:

- **Easy to learn**: A user should be able to start working with it after a short period of introduction.
- **Efficient to use**: A user should be able to have a high degree of productivity after the initial learning.
- **Easy to remember**: A user should not have to learn it over again after a period of absence.
- **Impossible to make errors with**: And If an error occurs the user should have the opportunity to recover from it.
- **Satisfactory to use**: A user should like using it, be subjectively pleased with it.

These attributes can all be measured and evaluated formally. There are also Several different ISO standards that define usability [6].

There are different ways to evaluate usability. These can be divided into usability testing, inspection and inquiry [7]. Usability testing provides direct information on how users are using a system and what problems they have. In usability testing a user is performing some tasks while being observed. Often the user is encouraged to say what comes to mind during the interaction (Think Aloud).

The most common usability inspection method is Heuristic evaluation [8]. Heuristics for usability are properties and principles that positively affect usability. Heuristic evaluation is performed by an expert evaluator which spends some time alone with the system and tries to find occurrences where a particular principle is broken. There are several different heuristic principles available that an evaluator can use.

Usability inquiry is about learning the likes, dislikes and needs of the users. This can be achieved by observing them, talking to them or interviewing them.

Observation takes place in the environment where the system is, or shall be, used and has a different purpose from usability testing. Interviews can be performed verbally or in the form of questionnaires. Focus groups is another method that can be used early in the design process. In a focus group people discuss their attitudes towards a product, concept or other. A focus group consists of end users and a moderator. The moderator's job is to guide the discussion and propose questions. Focus groups can also be used for brainstorming and generating new ideas, but are not good for objective evaluation [9].

Many of the usability methods requires some form of working prototype, but it is important to consider usability early in the development cycle, otherwise changes can be costly and hard to implement [8]. Usability issues that require changes in the architecture of the software are more expensive to change than changes in the user interface. Therefore it is important to begin prototyping and testing early.

The first step in usability engineering is to know the users [4, p. 73]. This includes both the end users and all others that are affected by the system. This can be done by the usability inquiry methods described above. Another method is to involve users in the design process, which is called participatory design. The purpose with involving users is not to let them come up with ideas for the design, but to provide feedback to the designers. They can raise questions that the designers themselves never had thought about. The grade to involve users may differ, from having a meeting once a week where the designers present their results to let the user be a part of the design team.

A prototype is a mock up of a part of the system, usually the user interface. The reason to build prototypes of the interface is to be able to test it without having a whole functional system. It is much cheaper to evaluate a prototype than a fully implemented system. In iterative design not only one prototype is build and evaluated, but many. This is to ensure that changes made to the interface have the desirable effects. When a prototype has been evaluated a new one can be constructed with possible solutions to the issues found in the former [4].

2.2 Video games

There are some similarities between video games and productivity software [1, 10]. They both have some sort of user interface and input devices and provide feedback to the user's actions. They also require initial learning and that the user form a mental model of the system.

The difference is that while a game primarily is used for fun and amusement, productivity software is there to help the users perform a task. Even if one of the usability attributes is that a system should be pleasant to use, this is not the main goal with productivity software. Another difference is that while a player is free to choose what game to play, this is not always the case with productivity software. In workplaces the system to use is not necessarily chosen by the people that are eventually going to use it. Games are also made to be challenging, something that productivity software try to avoid [1, 10].

While a game is designed to offer a challenge to the player, it is the game play that should offer that challenge, not the interface. Balancing the game to provide the right amount of challenge can be hard. The game should not be to easy to complete, but also not too hard. In either way the player might get tired and stop playing. During evaluation, it might also be hard to know which heuristic that applies to an identified problem. Barendregt et al. [11] have come up with a procedure for evaluators to distinguish between usability problems and fun problems. It is a chart that the evaluator can follow to determine the type of the problem. If an evaluator can determine the source of a problem it is more likely that he can suggest an appropriate solution.

2.3 Usability in video games

Given the differences between video games and productivity software, can usability methods be successfully implemented in game design? There is a growing number of studies indicating that the answer is yes. Usability expert evaluation and testing have been shown to provide novel and useful information in game development [3].

Some game companies have also started to incorporate usability methods in their work. Microsoft Game Studios have been using different usability methods for several years now in their playtest laboratories [12]. One method used there is the RITE (Rapid Iterative Testing and Evaluation) method [13]. RITE resembles "normal" usability testing. The difference is that RITE consists of several play sessions after each other. As soon as an issue is identified it gets fixed in the prototype before the next session. In that way designers can see if the fix handled the issue or if it introduced new problems. Another method is the Consumer Playtest method [10], but it is more related to finding "fun" problems than finding usability problems.

Clanton [14] identifies three levels where usability issues can be found in a game. These are game interface, game mechanics and game play. Game interface is equivalent with the user interface, the menu, the style of interaction etc. Game mechanics is the physics of the game, the bounds, constraints and possibilities the game character has. Game play is the challenge and goal of the game.

3 Related Work

There have been some attempts to create heuristics and guidelines aimed especially at game design. Federoff [15] compiled a list of game heuristics by game literature studies and by observing and interviewing five people from a game developing team. She also found that the team had difficulties defining the concept of usability.

Fabricatore et al. [16] presented a set of game design guidelines for action video games. The guidelines were divided into prescriptions and recommendations and were based on player studies and interviews.

Desurvie et al. [17] came up with a set of heuristics they called Heuristics for Evaluating Playability (HEP). HEP was tested and compared against user testing methods and the result showed that HEP could be used early in the design phase as a complement to user studies.

Sotamaa [18] examined how players were represented in game design literature. He found that players often are divided into casual and hardcore gamers. Hardcore gamers spend much time and money on games. For them, gaming is a lifestyle. Casual gamers play for amusement and fun and do not spend as much time and money on games. Another popular approach is to divide players into novice players and experts.

In the game design literature there are different opinions about if, how and when to involve players in the design process. Some argue that the players do not really know what they want, so involving them early in the design is just a waste of time and money. Others argue that if involving players only when there is a working version of the game, then it would be too late to make any adjustments [18].

4 Procedure

Five books were analyzed in this study. As an aid in the selection process some requirements were established. To capture the recent changes towards player-centred design it was important that the books were not too old. To encourage diversity the same writers name should not appear on the cover of more than one book. The focus of the books had to be on design and not on programming. The final requirement was that the books had to be used as literature in a game design course. The analyzed books are as follow:

- Adams, Ernest and Rollings, Andrew (2006) *Fundamentals of Game Design* [19].
- Fullerton, Tracy and Swain, Christopher and Hoffman Steven (2004) *Game Design Workshop: Designing, Prototying and Playtesting Games* [20].
- Oxland, Kevin (2004) *Gameplay and Design* [21].
- Salen, Katie and Zimmerman, Eric (2003) *Rules of Play: Game Design Fundamentals* [22].
- Schuytema, Paul (2007) *Game Design: A Practical Approach* [23].

The book by Adams and Rollings had extra chapters online, which also were included in the study. The books in this study used different angles when approaching game design; from a “hands on” approach to a more academic and analytic view. The books are all written by authors who are working in the game industry, and some authors also teach game design at university level. These books are also used as literature in game design courses at universities in Sweden.

There are specialized books about game interface and game environment design, but those were not included. The general game design books often have a chapter or more dedicated to these issues.

5 Analysis

The analysis begins by examining what the authors write about usability and player-centred design. Then follows the game design process and the authors view of prototyping and iterative design. The analysis ends by examining methods on how to evaluate games.

5.1 Notions of usability

According to Adams and Rollings [19, p. 61] large developers are more and more using usability experts to test and improve the game interface. Fullerton et al. [20, p. 290] highly encourages the designer to utilize usability engineers if the designer has access to them. The ideal situation is if a usability specialist is part of the team from the beginning. Many game designers mistake usability testing for focus testing and marketing and bring in the usability specialists too late. A usability specialist as a part of the team offers a great opportunity for the game designer. By working with the usability specialist the designer can learn a great deal of how players interact with the game. The designer can also learn how to break down the game into parts, identify issues and solve them [20, pp. 337-338].

5.2 The player-centred approach

Game designers have traditionally made games for themselves. This opinion is common among the authors. The new approach is to design games for the audience instead of the designer.

Adams and Rollings [19, pp. 38-40] define the player centric approach in which the designer has two obligations to the player: *The duty to entertain* and *the duty to empathize*. The primary function of a game is to entertain and the designer must imagine that he is the player in all design decisions. This means that the designer must identify and define the player, his likes, dislikes and motivations.

According to Adams And Rollings there are two common misconceptions among game designers: *I am my own typical player* and *The player is my opponent*. The first misconception is that the game designer assumes what he likes, the customers also like. Many game designers are young males and consequently most games are targeted for young males. It is not reasonable though, to assume that everyone likes games targeted for that group. The second misconception dates back to when games were played in arcade machines. Then the goal for the designers was to make players keep feeding the machines with coins. This was achieved by making the games hard to play more than a few minutes without losing. The arcade model has later been used in games where it is less appropriate. It suggests that the designer's job is to be the opponent of the player which contradicts the player centric approach.

Oxland [21] proposes that the designer must study the audience. This can be done by going to a game shop and talk with customers, asking them what they like and taking notes of what they buy. If possible the designer can volunteer to work in the shop. Throughout the book Oxland reminds the reader to always design for the audience. The designer should go to them and ask them what they like in a game.

According to Fullerton et al. [20, pp. 2-3] the role of the game designer is to be an advocate for the player. The designer should look at the game through the player's eyes and not get lost in the graphics and effects. It is important to involve players in every stage of the design process.

5.3 The game design process

Schuytema [23, pp. 12-13] identifies the three stages of the game development cycle; pre-production, production and post-production. During pre-production developers brainstorm and look at other games for inspiration. In this stage the design document is taking form. The design document is the blueprint of the game. The designer uses the document to communicate the game to the construction team and stakeholders. During production the game is implemented. Testing and refinement is also performed and the design document is updated. The post-production takes place when the game has been released. It involves making patches and additional downloads, evaluating the reception among players and possible planning the sequel.

Adams and Rollings [19, pp. 52-60] define the three stages of the design process: concept, elaboration and tuning. The concept stage defines the game and the target audience, the elaboration stage adds details through prototyping and playtesting and the tuning adjusts details without changing too much. It is somewhere during the elaboration stage the game will go into full production. Later in the process, only minor changes can be made.

Fullerton et al. [20, pp. 10-16] use seven stages in the design process. The first stage is brainstorming, where several ideas are generated. In the second stage the ideas are tested on a physical prototype. The third stage is to present the idea to secure funds for further development. In the fourth stage a software prototype is built and tested. Step five involves writing the design document. In step six actual development starts. Step seven is quality assurance. The game has to be thoroughly prototyped before moving on to the development stage. Many times games are designed during the development, when changes are costly to implement. Even before making a software prototype, the designer has to have a deep understanding about the gameplay. This understanding comes from building plenty of physical prototypes and testing them [20].

5.4 Prototyping and iterative design

Prototyping and iterative design are mentioned in every book examined. Adams and Rollings [19] advice the game designer to build, test and iterate. Build prototypes to test ideas before incorporating them into the game, and then throw the prototypes away.

Schuytema [23] suggests that the team should prototype to test the game dynamics. The prototype is a functional portion of the game and should be tested before incorporating features into the design document. The designer should also make a prototype of the in-game interface early so that the development team can start testing the interface as early as possible.

Oxland [21] suggests that balancing the game should begin early. The prototypes should be built with no artwork and crude models of the terrain, objects and characters in the game. This saves time and money because the artwork designers can come in later in the process and no artwork has to be thrown away when the game changes. The prototype should be tested on colleagues and end users. The interface should be prototyped and tested on players in an iterative way.

According to Salen and Zimmerman [22, pp. 12-13] a game should be prototyped and playtested not later than 20 percent of the way into a project schedule. Early prototypes are not pretty, but are playable versions of the game. Prototyping is used to test the mechanics of the game. Furthermore is iterative design important because it is not possible to anticipate play in advance. According to Salen and Zimmerman most digital game designers do not follow an iterative design process.

Fullerton et al. [20] distinguish between the physical and the software prototype. The physical prototype is constructed out of paper, cardboard and other materials that the designer deems fitting. By building a physical prototype the designer can test the game mechanics. Even action games like first-person shooters can benefit from being prototyped. The software prototype is used to further explore the design of the game. There are several ways to build the prototype and many different tools can be used. Fullerton et al. suggest that the designer can use level editors as tools to learn how to prototype the game. Some games come with a level editor that let's the user build their own levels. They usually have some sort of drag and drop feature that makes it easy to build a customized level. Fullerton et al. uses the iterative design approach at every aspect of game design, from the concept stage to quality assurance.

5.5 Heuristics and guidelines

Guidelines targeted at game design can be found in every book. Often these guidelines are found in the text, presented as warnings or advices. Many guidelines relate to the usability of video games and concern areas such as the game interface and mechanics. Section six in this paper contains a list of heuristics compiled from these guidelines.

Throughout the book Schuytema presents lists of game design guidelines which he calls design atoms. Schuytema defines an atom as a "core design guideline or rule that applies to nearly every interactive game, no matter what the platform or genre" [23, p. 147]. These atoms have evolved and have been used over dozens of years by the designers. The atoms are divided into groups like game, interface, AI, inventory, level and mission design atoms.

5.6 Evaluation

All writers agree that testing the game/prototype is important. Adams and Rollings, Oxland and Schuytema all acknowledge the importance to test and evaluate the game, but do not go into describing how to test it. The designer is encouraged to bring in others to test the game, because the designer is too close to the game to recognize issues. Testing is important throughout the development and the designer can test on both the design and development team and players. Oxland [21, p. 302] mentions a special kind of testers that play the game while looking for flaws in the interface, gameplay and flow of the game. This differs them from testers that play to find bugs and unusual behaviour in the game. These testers can not only tell what is not working, but also why it does not work and how to fix it.

Fulerton et al. and Salen and Zimmerman encourage the designer to invite end-users to play the game. The designer should observe the players and pay attention to what they do when they get frustrated, stuck or bored. Observe how players interact before, after and between games and write down everything they tell. According to Fullerton et al. [20, p. 3] many designers bring playtesters in too late, if at all. Fullerton et al. [20, p. 10] also explicitly states that the sooner the designer can bring players into the process, the better.

Fullerton et al. describes how a test with players can be performed. They distinguish between playtesting and usability testing. Usability testing is to systematically analyze how users interact with a system by recording mouse movements, eye movements, navigation patterns etc. Playtesting is performed to gain insight on how players experience the game and is performed during the entire design process. At first playtesting can be performed by the designer with the team, later with friends and family. Finally, when the prototype is stable enough, the designer can playtest with the target audience. testing can be performed on the target audience. In a playtest session people play the game with the designer. The mistake many designers make is to start explaining the game and its features. Instead they should let the players explore and experience the game themselves and only guide them when they have problems. There are some methods that can be used during a playtest session to structure the test. One-on-one testing involves one player and the designer sitting behind and asking questions as they play. Group testing involves several players at once. The players can also fill in feedback forms when they are done playing. In an interview the designer sits down face to face with a player and asks questions from a predefined list. A discussion can be performed either one-to-one or in a group and is less structured than an interview. Some techniques from usability testing can also be used. One of them is to ask the players to think out loud while they play.

Another thing to test is player accessibility, i.e. if the player can pick up the controls and play without help. Testing for accessibility is related to usability testing, the difference lies in who performs the test. Usability testing is performed in a usability lab by experts while accessibility testing is performed by the designer. Before the accessibility test begins the designer prepares a script with tasks for the player. These tasks cover the most critical parts of the game.

During the play session the designer usually sits behind a one-way glass in an adjacent room and observes the player. Data is gathered during the session to be analyzed afterwards. It is also important to use players that have not played the game before [20, pp. 290-292].

6 Game Usability Heuristics

Even if there are similarities between games and productivity software, there is a need to develop a new set of heuristics targeted at game design [15, 17]. In table 1 a list of game usability heuristics is presented. The heuristics are based on the guidelines and advices found in the analyzed literature. Often similar guidelines can be found in several books; these have been grouped and presented as one general heuristic. Guidelines that apply to specific genres have been excluded from the list. The left column in the table presents the name of the heuristic and the right column presents the description.

Table 1: Game usability heuristics

Heuristic	**Description**
Clear goal	The player needs to know what to do next. The player must be informed about the victory conditions. Alert the player if the goals change.
Success feedback	The player should be able to tell how good he is performing, if he is making progress and if he is in danger of loosing.
Player action feedback	The player needs to know what he can do and what he can not do. Can the player tell if an action has been accepted and how the player action affected the game?
Player frustration	Player frustration should be minimized. When the player fails the cost should not be frustration. Are save points placed so that the player does not have to replay large sections of the game when he loses? Alternatively; is there an auto save function? Default settings should be used where possible. The player should be able to skip non interactive parts. Parts that are repetitive and not fun should be abstracted or automated.
Consistency	The Interface should remain consistent. The game world should act in a consistent way.

Table 1: Game usability heuristics

Heuristic	**Description**
Easy to learn	The game should be easy to learn. The interface metaphor should help the player form a good mental model of the game. It is important that the game and the content is understandable for the player. In-game tutorials can be used to teach the player, both at the start of the game and when new features are introduced. Use tried approaches when designing the interface.
Easy to use	The player should struggle with the obstacles that the designer intended, not the interface. The number of steps required to perform an action should be no more than three button presses. Items and icons should be easy to recognize and should not require that the user clicks on them to identify them. The interface should not ask the player to remember. Provide meaningful information. Protect the player from making errors. Offer undo if it does not affect the game play. The player should be able to customize the controls and the interface.
Clear rules	The player needs to understand the game rules. Inform the player of the rules, not just in the manual.
Player should never feel lost	The player needs to know where he is and where he can go. Provide a world and an interface that always let the player know where he is.
Avoid dominant strategies	Avoid strategies that are superior to all other strategies.
Visibility of game status	Do not hide vital information from the player. The status of the game should always be clear, unless it is part of the gameplay.
Aesthetic	Strive for harmony, elegance and beauty. Everything in the game has to be there for a reason. Let form follow function.

7 Conclusion and Discussion

Two out of five books explicitly mention usability. Both of them give the impression that usability is something that concerns experts and is something that only large game studios can afford. Every book has guidelines that relates to usability, but these seem to have evolved outside of the usability field of research.

There can be much to gain if the designers create games with usability in mind from the start. This means that the designer knows how to design for usability and avoid common pitfalls. This can not replace evaluation but it can avoid that the same issues occur several times. Designing for usability also avoids the danger that usability is regarded as something that can be "glued on" when the game is near completion. Some issues like an ambiguous word in a menu are relatively easy to fix, but if the issue lies deep in the structure and mechanics of the game large parts of the code may have to be rewritten and artwork remade. The later in the process such a usability issue is found the more costly and less likely it is to be repaired.

Prototypes are built to test and evaluate the game, but there seems to be a lack of methods on how to evaluate the prototype. Heuristic evaluation could be a way to test the prototype in an early stage. The evaluator needs to have some experience to be able to apply the heuristics correctly, but inexperienced evaluators can also find many usability issues with the method [4].

All books stress the importance of testing the game, but only one mentions exactly how such a test should be performed. It is important to have some knowledge on testing with end users; otherwise the data gathered can easily be biased and more or less useless. A test also produces different data depending on how it is designed. These two factors speak for that the designer needs to have some knowledge when testing the game with players.

Different usability methods identify different aspects of usability and work best at different stages of the development process. It is worth studying if these traditional usability methods need to be redesigned when applied to games. Another approach is to design entirely new methods for the evaluation of video games.

Further research is needed to verify the list of game usability heuristics. This can be done by having game designers and usability specialists use the list when performing heuristic evaluation. The evaluators can also add heuristics, edit the list and grade the heuristics according to relevancy.

References

1. Jørgensen, A.H.: Marrying HCI/Usability and computer games: a preliminary look. In: Proceedings of the third Nordic conference on Human-computer interaction, New York, NY, USA, ACM (2004) 393–396
2. Christensen, L.J., Jørgensen, T.T.Y., Jørgensen, A.H.: Developing a hybrid of MMORPG and LARP using usability methods: the case of Takkar. In: Level Up, Proceedings of DiGRA 2003 Conference. (2003)
3. Laitinen, S.: Do usability expert evaluation and test provide novel and useful data for game development? Journal of Usability Studies (2006) 64–75
4. Nielsen, J.: Usability Engineering. Academic Press (1993)
5. Bevan, N., Kirakowski, J., Maissel, J.: What is usability? In: Proceedings of the 4th International Conference on HCI. (1991)
6. Seffah, A., Donyaee, M., Kline, R.B., Padda, H.K.: Usability measurement and metrics: A consolidated model. Software Quality Control **14**(2) (2006) 159–178
7. Folmer, E., Bosch, J.: Architecting for usability: a survey. The Journal of Systems & Software **70**(1-2) (2004) 61–78

8. Holzinger, A.: Usability engineering methods for software developers. Communications of the ACM **48**(1) (2005) 71–74
9. Fulton, B., Medlock, M.: Beyond focus groups: Getting more useful feedback from consumers. In: Proceedings of the Game Developers Conference. (2003)
10. Davis, J.P., Steury, K., Pagulayan, R.: A survey method for assessing perceptions of a game: The consumer playtest in game design. Game Studies: the international journal of computer game research **5**(1) (2005)
11. Barendregt, W., Bekker, M.M., Speerstra, M.: Empirical evaluation of usability and fun in computer games for children. In: INTERACT-03, Proceedings of the IFIP 8th International Conference on Human-Computer Interaction. Volume 3. (2003) 705–708
12. Pagulayan, R., Keeker, K., Wixon, D., Romero, R., Fuller, T.: User-centered design in games. Human Factors And Ergonomics (2002) 883–906
13. Medlock, M.C., Wixon, D., Terrano, M., Romero, R.L., Fulton, B.: Using the RITE method to improve products; a definition and a case study. In: Humanizing Design, Usability Professional's Association 2002 Annual Conference, UPA (2002)
14. Clanton, C.: An interpreted demonstration of computer game design. In: CHI '98 conference summary on Human factors in computing systems, New York, NY, USA, ACM (1998) 1–2
15. Federoff, M.: Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games. PhD thesis, Indiana University (2002)
16. Fabricatore, C., Nussbaum, M., Rosas, R.: Playability in Action Videogames: A Qualitative Design Model. Human-Computer Interaction **17**(4) (2002) 311–368
17. Desurvire, H., Caplan, M., Toth, J.A.: Using heuristics to evaluate the playability of games. In: CHI '04 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM (2004) 1509–1512
18. Sotamaa, O.: Perceptions of player in game design literature. In: Situated Play, Proceedings of DiGRA 2007 Conference. (2007) 456–465
19. Adams, E., Rollings, A.: Fundamentals of Game Design. Prentice Hall (2006)
20. Fullerton, T., Swain, C., Hoffman, S.: Game Design Workshop: Designing, Prototyping, and Playtesting Games. CMP Books (2004)
21. Oxland, K.: Gameplay and Design. Addison-Wesley (2004)
22. Salen, K., Zimmerman, E.: Rules of play: game design fundamentals. MIT Press (2003)
23. Schuytema, P.: Game Design: A Practical Approach. Charles River Media (2007)

Designing Web Sites for Senior Citizens

David Åberg

Department of Computing Science
Umeå University, Sweden
dit04dag@cs.umu.se

Abstract. Currently, the reliance on Internet based services for communication and supplying of information are rapidly increasing. Consequently a growing number of senior citizens will need to make use of these services to avoid social exclusion. Thus it is important that web sites are designed to be an accessible resource for older people. However, many older people are experience difficulties when using the web. One of the problems is that an older individual differs in many ways from the mainstream computer user. Age related decline of cognitive and physical abilities are affecting the performance of web site interaction. This paper considers how age related impairments may be addressed through appropriate web design. A summary of recommendations is conducted from a review of some of the most commonly used guidelines for web design. Furthermore the drawbacks of relying on guidelines as the one and only solution for providing accessible design are discussed.

1 Introduction

In the recent decade Internet has become a mainstream technology just as television did in the latter half of twentieth century. More and more services are being moved on-line with the expectation that individuals are experienced Internet users. Governments across the globe are relying increasingly on computer based mediation to communicate with citizens. Despite the positive aspects of the web there is one group in the society where Internet usage is relatively low, i.e., among people over age 65 [1].

The members of this group, many of whom are housebound or located far away from family and friends could benefit from consuming Internet based services [2]. There are many areas where Internet can help older adults retain a good quality of life. The providing of shopping, entertainment and communication online has the potential to enhance the day to-day life of many older people. Senior citizens could be able to keep in touch with their social contacts and maintaining recreational pursuits without leaving home [3, 4]. Without taking advantage of the benefits with Internet, older adults are in danger of further social exclusion. In order to remain as an active part of the society it is important to have access to the information provided. Although advanced age does not restrain computer and Internet use, there are some unavoidable age-associated declines of physical and cognitive nature which affect the use of web sites. If web sites are expected to be an accessible resource for the majority of the population they must be easy to use despite of changes due to age [5], currently this is not the case. In a evaluation of 36 web sites designed for older adults it appeared many of them did not even comply with some of the most fundamental requirements [6].

An essential prerequisite for an accessible web is to make sure that designers possess knowledge about non typical users and the issues that they are facing. This paper starts out with a dissection of the differences between senior citizens and the regular computer users in terms of cognitive abilities, physical abilities and computer experience. After the differences have been sorted out the paper proceeds with a summary of design guidelines to have in mind when building web content directed towards an older target group. Finally the drawbacks of a guideline-based design approach are investigated.

2 Expected age-related differences

It is obvious that older people differ from younger in many ways. With increasing age the overall performance of body functions deteriorate. This can be a source of trouble when dealing with computers. Older people are facing a greater challenge when trying to learn new computer applications [3]. Some of the issues that appear when seniors come in contact with computers can be directly related to individual cognitive declines, while other problems can be of a more physical nature. Just as young people older are not a homogeneous group, in fact a fit 65 year old is very different from an 80 year old person suffering from dementia. Although there is a possibility to divide older users into various subgroups [7], this paper focuses on the general differences between an older person and the typical computer user. Below follows a summary of expected age related differences.

2.1 Cognitive differences

Cognition is referred to as the human processing of information and involves a variety of processes on different levels. Cognitive abilities are the brain-based skills and mental processes that are needed to carry out any task. The cognitive abilities increase gradually from birth to young adulthood and are then either maintained or decline to old age, depending on the specific ability [8]. It is well established that the perceiving of computer mediated information requires several cognitive skills [9]. Christensen chooses to categorize the cognitive abilities in three main groups: crystallized Intelligence, cognitive speed and memory. Crystallized Intelligence is referred to as the knowledge and skill acquired by an individual over a life time. Crystallized abilities have shown to increase up to high ages, and signs of decrease are not likely to appear among younger individuals [8].

A reduced cognitive speed is one of most known and confirmed aspects of cognitive decline. This ability is especially affected when age increase. A lower processing speed has a bad influence on an individuals capability to attend to, select, process, remember and respond to received stimuli. Each one of these abilities can be related to the use of web sites [10]. The general slow down of cognitive speed becomes more obvious in complex environments where the processing demands are heavy [11]. Therefore can cluttered and complex interfaces cause trouble for users with cognitive speed deficits [12, 13].

The memory capacity is significantly decreased with age; this affects different parts of the memory to different degrees. Regarding short time memory the deficit is most noticeable [8]. The changes can be identified by decreased memory efficiency and reduced success of retrieval [14]. Seniors with memory declines are likely to become disoriented and confused when browsing the web [12, 9]. As web navigation tasks become more complex the demands on working memory significantly increase [15].

2.2 Visual impairments

The human visual perception is constantly declining starting from young adulthood. Due to the slow rate the effect of these declines tend to become noticeable first in the early forties [16]. At this age the human lens successive decreases the elasticity to adapt its focal length, which reduces the ability to focus on close targets [17]. With a regular computer setup the distance from eye to screen typically falls into the range where the vision of older people is blurred. Consequently small font sizes and icons may become fuzzy and hard to discern. Contrast sensitivity is the ability to separate between luminosities of different levels. Comparing 99 subjects in the age between 20-69 showed a significant age-related decline in contrast sensitivity [18]. The sensitivity for color is also reduced with age. Research has indicated age-related decrease in relative sensitivity to short-wavelength light. While the studies resulted in no evidence for age-related changes in red-green isoluminance values, red-blue isoluminance values showed a significant decrease [19]. These kinds of declines affect the ability to separate text and other information from the background of a web site.

Furthermore, older adults appear to be less able to adapt to changes in brightness levels and more sensitive to glare [16]. Rapidly shifts in brightness between different screens can constitute a problem for these individuals. A reduced field of view related to aging has also been reported. Older people find it harder to localize targets in the periphery of their visual field. This effect is sometimes explained as a lowered cognitive ability to process information in a cluttered scene rather than a physical change in the eye [20]. In any case seniors might not detect critical information if it is located far off the center on the web site.

2.3 Hearing impairments

Comparing all of the changes occurring with age hearing is probably the most expected and accepted. While communication through sound is not that common in current web sites, hearing loss may affect attempts to design speech based interfaces to compensate for other problems. The degenerative hearing loss related to aging is referred to as presbycusis. This hearing impairment affects the perceiving of tones at all frequencies but it is most noticeable as a reduced ability to hear high pitched sounds. Humans can generally hear sounds with frequencies between 20 Hz and 20 kHz. Normal aging implicates losing about 1 kHz in the upper range for every 10 years. Interfaces that use sound to get the users attention will need to use lower frequency sounds for older users. Presbycusis act on both ears equally. This type of impairments affects the understanding of speech in both quiet and noisy environments. Presbycusis causes a significant problem for over a third of the older adults [21].

2.4 Motor impairments

When aging both morphological and neurological declines affect the muscles [22]. This is leading to slower response times when performing complex motor tasks. Some web places rely on complex maneuvers like dragging to execute certain tasks. These activities are extremely difficult for older users [15]. Also performing repetitive fast movements like double-clicking with a computer mouse may be difficult for people of older age. Increased age is related to more slowness in general. A Stiffening of the joints and arthritis causes difficulty with finer movements and motor coordination [23]. Older people commit significantly more errors than younger when executing high precision tasks with a computer mouse. Clickable surfaces and navigation elements with small areas can constitute a challenge. Walking menus that leads from one pull-down menu to the next requires precise movements, which may be problematic for seniors [15].

2.5 Computer experience

The computer experience among senior citizens is in general relatively low [1]. Experience is known to have a huge impact on the performance during computer based tasks [24, 10]. Having web browsing experience for instance leads to sharper performance when retrieving information from web sites. Experienced users are also faster in terms of search times. There is a lot of variability in the design of web sites and inexperienced users are more likely to loose track when navigating [10]. The various technologies to develop web sites are evolving at a fast pace. Inexperienced computer users are not likely to take advantage or be aware of the latest updates and freshest technology [13]. Experience can possible compensate for other declines that older people suffer from when dealing with computers[10].

3 Guidelines and standards

Following guidelines and standards has become a trademark for quality work in the industry of web design. Designers and developers lean towards different guidelines, standards and automated tools to certify the accessibility for their creations. The number of guidelines and standards that claims to ensure usability is growing fast. There are both general directives and those concentrating on web design for older people. Guidelines can be treated in many different ways, they can serve as handy summaries of HCI research, provide designers with reliable research based advices, be used to distribute information or to a raised awareness among designers. Although used as a direct influence in the design process is probably the most common way [25].

3.1 World Wide Web Consortium

The W3C guidelines are intended to be used as de facto standard for building interoperable, robust, and easily accessible web places. WAI, Web Accessibility Initiative, is one of four domains of the W3C. The ambition of WAI is to provide an accessible web by focusing in particular on technical aspects. Specifications describe how to properly use HTML, CSS and other associated technologies. The idea is to allow for any web browser to present information in the most appropriate way for the user. By correctly using a descriptive mark-up language the content of a web site can be accessed through a variety of web browsing technologies, for example by screen readers and mobile browsers. When developers declare that a web site is standards-compliant they usually mean that it is correctly built and reviewed according to the web technology specifications of W3C [13]. The member states of the European union have agreed upon that all governmental web sites must be compliant with the guidelines from the W3C [26].

3.2 Research-Based Web Design and Usability Guidelines

The U.S. Department of Health and Human Services has produced a set of design and usability guidelines for web developers, based on actual user testing and research. The guidelines were developed to provide web designers with evidence based assistance in design decisions. Compared to others these guidelines are less technical and more directed towards practical guidance. Topics like web accessibility, navigation, page layout, titles and headings, graphics, animation and searches are addressed. The guidelines are in the form of practical advices, sets of do's and don'ts. These advices should not be considered prescriptive, but as guides to give designers starting points. Every guideline is graded by the authors to give information about its relative importance and strength of evidence [9].

3.3 24-Timmarswebben

In Sweden it is a fundamental principle that public web sites ought to be a resource that is available for all irrespective of age, sex, impairments, and cultural background. The contact with public authority should be a supple experience suited for the needs of both citizens and companies. The framework "24-timmarswebben" is a tool to help fulfilling these obligations. The framework consists of 147 guidelines for developing web pages and e-services towards the public sector. Although the target for "24-timmarswebben" are authority web sites the expectation is that the guidelines should appeal in many other areas of web development. The guidelines included in the framework deal with many of the issues that older people must confront when browsing the web [12].

4 Summary of design recommendations

Much of the information presented on web accessibility and usability cover many types of disabilities and not only the specific needs of older people. Below is a summary of design recommendations distilled from the guidelines mentioned in the previous section. The initial sets of guidelines are very comprehensive, covering many areas of accessibility. However, they may be perceived by web designers to be too many to adhere to. The resulting recommendations are a smaller set of 22 guidelines focusing on the specific issues for older people. The guidelines are split into 5 distinct category headings corresponding to the deficits that were identified in section 2.

4.1 Cognitive impairments and web design

Decline in cognitive ability may influence the capacity to make effective use of web services [9]. Older adults are known to perform less well than younger adults in a variety of cognitive demanding tasks [15]. The following guidelines address issues related to cognitive changes due to increased age.

- Avoid using movements in the interface, users with cognitive speed deficits may not manage to read or comprehend the messages quickly enough. Moving objects can distract from the rest of the content so much that users are unable to access it [12, 13].

- If the information presented demands for moving, blinking, scrolling, or auto-updating objects allow them to be paused or stopped. Ensure that the user is in control of all time-sensitive content [13]. At the design stage, there is a need to carefully consider the speed of walking banners and flashing text.[13].

- Users often follow a set pattern when approaching a new web site, starting out with scanning the site for link, titles and headings. To not slow down this process the amount of prose text on a web page should be limited [9].

- Do not require from the user to adopt to new interaction patterns just to manage executing certain tasks [9]. As users get accustomed to specific sequences of action, make sure that a similar pattern of actions can be applied in all situations.

- It is of importance to try to keep the number of items on each page down to avoid overload of the working memory [9]. Cluttered web pages with an immense amount of objects cause a decreased performance when scanning for information.

- Equip web sites with an information overview that displays the hierarchy of the web site. A site map can assist users recall and as a result they are less likely to become disoriented and confused when navigating [12, 9].

- Information required to perform an action should be presented so it is visible when executing the action [9]. Do not assume that users are able to remember information from previous pages visited on a web site.

- To help users understand what kind information that is expected from input components display default values whenever it is possible. A suitable default value is often the most frequently selected item or alternative [9].

4.2 Visual impairments and web design

Visual decline is generally considered as the most affective impairment on web site interaction, after all the web is a visual medium. Numerous variants of physical changes take place in the eye when becoming older [16]. The following guidelines attempts provide support for these issues through appropriate web design.

- Every non-text element should be supported with a textual description that captures the information. This allows for visually impaired accessing the information trough alternate techniques like screen readers and such [13, 9].

- An accessible web place should have good readability, with well a considered typeface and text layout [13]. The font sizes ought to be at least 12-points or larger providing for a comfortable reading experience [9]. Remember to following mark-up standards and allow for parameters like text size to be overridden by the users web browser configuration [13].

- It is important to use sufficient contrasts and make sure that the difference between background and text always is more than 50 percentage points [9, 12]. High contrast elicits reliably faster reading performance and does not discriminate users with decreased contrast sensitivity. For the same reason be careful with the use of background images, they should be simple and rarely used behind text [9].

- Essential properties and functions shall work independent of the user's ability to perceive certain colours. Make sure that all information communicated with colour is also available without colour [9, 13, 12]. If colour is the only aspect that distinguish some information, people how are unable to differentiate between colours are missing out on this information [13].

4.3 Motor impairments and web design

Seniors are expected to have limitations in motor abilities. Since hierarchical user interfaces like web sites are usually mouse driven, it is predicted that older users will have a harder time navigating. While there are assistive technologies for those who are unable to use a mouse, the mouse is by far the most used input device among seniors. Support for motor impairments can be provided by complying with the following guidelines.

- Extend the area of clickable surfaces and navigation elements as much as possible. This is to compensate for a decline in precision with mouse interaction [12].

- Design menu systems with a point-and click function instead of mouse over to select items. The point-and click technique is preferred by users and have proven to reduce the number of errors committed [9]. Complex mouse maneuvers should in general be avoided.

- A common approach to build a secure web page is to let the page time out automatically. To support the slower response time of older people let the users be aware if a page is designed to time out. When the users are prepared they can request for more time [9].

4.4 Hearing impairments and web design

Auditive information and communication are not that common in current web sites. Although with the increasing use of multimedia on-line this may change. The following recommendations address problems that web site users with hearing declines may face.

- All visual or auditory information should be supplied with equivalent textual alternatives. The strength of a textual alternative is that the users are given the control over the way they access the information. That way it can be rendered using a variety of technologies [13].

- Interfaces that use sound to get the users attention will need to use lower frequency sounds for older users [9].

4.5 Computer experience and web design

One interpretation of differences in performance on web sites interaction is that older adults have less computer experience and less Internet experience than younger adults [1]. The following guidelines address issues related to lack of computer experience.

- Be consistent with navigation, structure and layout across a web site. Make sure the users are aware of where they are at present and retain a good overall structure of information across the site [9, 13, 12].

- Proper assistance is essential to offer for inexperienced users. These users require extended support when browsing web sites. This is of particular importance when dealing with many first time users [9].

– The language on a web site should be simple and clear. Use a language that is easy to understand and avoid terminology that inexperienced computer users not are accustomed to [9].

– An accessible web page can not be dependent on newer technologies that are not widely accepted [13]. Inexperienced computer users are not likely to be up to date with the latest technologies

5 Can guidelines guarantee accessibility?

To develop and adopt web content for older users, designers need a fair amount of knowledge about the needs of the target group. There are many ways of providing such information to designers, the most common are currently through guidelines [27]. Guidelines can be hard to interpret and use consistently. There are some inherent contradictions where accessibility guidelines may be in conflict with usability [27]. A risk factor when publishing guidelines is that people are using them without grasping their underlying purpose. Guidelines are often simple statements derived during immense research [9]. These statements provide the interface designer with a diffuse version of the result from the original experiments. To fully take advantage of the information one must understand the background and context of the guidelines [28]. Many web designers originate from a technical graphic-design background, at the same time accessibility is a complex area of HCI research. Hence there is a lack of essential expertise to make use of guidelines among web developers [29]. Generally speaking, to build successful interfaces the designer has to take on a reflective approach and a holistic view. When using more of a checklist approach there is a danger of encourage developers to only focus on the interface areas acknowledged in the guidelines [27].

The growing popularity for building accessible web sites has created a necessity for tools that can check for compliance with the guidelines [27]. The process of evaluating a web site with an automated tool is simple; hence there is a great temptation to use these tools. Tools are good at quickly detecting errors, but it is the skill of designer that limits how well the errors are corrected [29]. The automatic tools can for example verify simple conditions, like that if a web site claims to be compliant with the W3C guidelines there must be an alternative textual description for every image [13]. But the tools can never decide whether the textual description is valuable for the user. An alternative text like "001.jpg" does probably not mediate the motive of its related picture [29].

6 Discussion

Future generations of older adults will have different needs and skills from today's older adult population, due to their prior computer experience [10]. However, the declines and impairments of aging are unavoidable. Interface design directed towards an older target group is a challenging task and designers have to look further than to what suites the traditional computer user. One important thing to have in mind is that seniors differs from young adults, they have their own special characteristics and needs. Even a fit older individual suffers from age related cognitive and physical declines [7], see section 3. Some of the impairments alone do not matter so much, but together they can have a deep impact on web site interaction, especially when combined with a low computer experience.

This paper has presented a summary of design guidelines to have in mind when building web content directed towards an older target group. This set of guidelines is supposed to provide a good support for the age related declines older individuals are facing. They can also serve as a starting point for a raised awareness in usability and accessibility questions. The difficulty is that a holistic approach towards web designing is required to apply these guidelines [27]. The risk when presenting guidelines is that web developers follow them without reflecting on the actual problems. Guidelines require that developers fully understand the requirements and the idea behind each guideline; this should be in the interest of every professional. Ignoring the needs of the oldsters implicates missing out on a huge group of potential users.

Web standards are another strategy to face accessibility problems that should be mentioned in this context. Web standards are considered either as compliment to accessibility guidelines or as a standalone solution [13]. Using standards implicates that a web site can be accessed through a variety of web browsing technologies. When following web standards users are given the control over the way they access web content. This flexibility is generally considered as a benefit for the accessibility of a web site. Although having the freedom of choice adds to the complexity. Older users are not likely to take advantage of advanced client side features, which reduces the positive effect with building standard compliant web sites [5].

While guidelines and standards are valuable, they should not be considered as the one and only norm which developers need to consider when designing for accessibility [27]. To involve the end-user in an iterative design process can be at least as helpful as any written advice [9, 12].

7 Conclusion

It is obvious that the World Wide Web has an unavoidable influence on many peoples life. It is also clear that the World Wide Web has a potential to increase the social network and enrich the day to day life of older people. Although problems with functional impairments brought on by the process of aging implies that older senior citizens are facing a lot of issues when accessing the web. A variety of age related declines that affects web page interaction among seniors have been identified and presented in this paper. Many of these issues could be addressed trough proper web design and the use of accessibility guidelines. A summary of guidelines to deal with the specific issues has been constructed from some of the most well-known resources providing information concerning the design of accessible web pages. Web developers adherence to technical standards and resource-based guidelines can facilitate older peoples contact with the web. The very existence of a set of guidelines will have the effect to put forward the useful discussion around usability and accessibility. However, to ensure accessible web sites following design guidelines are not enough, a more holistic approach is required. This paper puts forward the argument for encouraging designers to understand the background and context of guidelines before applying them.

References

1. IT-kommissionen: It och äldre, it-kommissionens rapport (2002) `www.itkommissionen.se/doc/379.html`, accessed 2008-05-26.
2. Wright, K.: Computer-mediated social support, older adults, and coping. The Journal of Communication **50**(3) (2000) 100–Ũ118
3. White, H., McConnell, E., Clipp, E., Bynum, L., Teague, C., Navas, L., Craven, S., Halbrecht, H.: Surfing the net in later life. Journal of Applied Gerontology (1999)
4. Juznic, P., Blazic, M., Mercun, T., Plestenjak, B., Majcenovic, D.: Who says that old dogs cannot learn new tricks? a survey of internet/web usage among seniors. New Library World **107**(1226) (2006) 332–345
5. Ellis, R., Kurniawan, S.H.: Increasing the usability of online information for older people: S case study in participatory design. International Journal of Human-Computer Interaction **12**(2) (2000) 263–276
6. Hart, T.A., Chaparro, B.S.: Evaluation of websites for older adults: How Şsenior friendlyŤ are they. Usability News (2004)
7. Gregor, P., Newell, A.F., Zajicek, M.: Designing for dynamic diversity: interfaces for older people. In: Assets '02: Proceedings of the fifth international ACM conference on Assistive technologies. (2002) 151–156
8. Christensen, H.: What cognitive changes can be expected with normal ageing? Australian and New Zealand Journal of Psychiatry (2001) 768–775
9. U.S. Department of Health and Human Services: Research-based web design and usability guidelines (2008) `http://usability.gov/pdfs/guidelines.html`.
10. Priest, L., Nayak, L., Hamilton, S.: Website task performance by older adults. Behaviour and Information Technology **26**(3) (2007) 189–195
11. Bashore, T.R., Ridderinkhof, K., van der Molen, M.: The decline of cognitive processing speed in old age. Current Directions in Psychological Science **6**(6) (1998) 163Ũ–169
12. Verva: 24-timmarswebben (2008) `http://www.verva.se/web/t/Page____1154.aspx`.
13. W3C: World wide web consortium (2008) `http://www.w3c.org`.
14. Morrissey, W., Zajicek, M.: Remembering how to use the internet: An investigation into the effectiveness of voicehelp for older adults. In: Proceedings of HCI International. (2001)
15. Morrell, R.W.: Older Adults, Health Information, and the World Wide Web. Lawrence Erlbaum Associates (2001)
16. Straub, K., Weinschenk, S.: The gradual graying of the internet (2003) `http://www.humanfactors.com/downloads/jun03.asp#screenfonts`, accessed 2008-04-04.
17. Glasser, A., Campbell, M.C.W.: Presbyopia and the optical changes in the human crystalline lens with age. Vision (1997) 209–229

18. Nio, Y.K., Jansonius, N.M., Fidler, V., Geraghty, E., Norrby, S., Kooijman, A.C.: Age-related changes of defocus-specific contrast sensitivity in healthy subjects. Ophthal. Physiol **20**(4) (2000) 323–334
19. Nguyen-Tri, D., Overbury, O., Faubert, J.: The role of lenticular senescence in age-related color vision changes. Investigative Ophthalmology and Visual Science **44**(8) (2003)
20. Sekuler, B, A., Bennett, J, P., Mamelak, Mortimer: Effects of aging on the useful field of view. Experimental Aging Research **26**(2) (2000) 103–120
21. Weinstein, B.E.: A primer on hearing loss in the elderly. Generations (2003) 15–19
22. Jensen, B.R., Laursen, B., Sandfeld, J.: The effect of aging on performance and muscle activity during computer use. In: BCS HCI Workshop on A New Research Agenda for Older Adults. (2002)
23. Smith, M., Sharit, J., Czaja, S.: Aging, motor control and the performance of computer mouse tasks. Human factors **41** (1999)
24. SJ, C., J, S.: Age differences in the performance of computer-based work. Psychol Aging **8**(1) (1993) 59–67
25. de Souza, F., Bevan, N.: The use of guidelines in menu interface design: Evaluation of a draft standard. In: Proceedings of IFIP INTERACT'90: Human-Computer Interaction. (1990) 435–440
26. Verva: Krav på offentliga webbplatser (2008) `http://verva.se/verksamhetsstod/webb/vl24/anv/krav/`, accessed 2008-04-11.
27. Kelly, B., Phipps, L., Howell, C.: Implementing a holistic approach to e-learning accessibility. Canadian Journal of Learning and Technology **30**(3) (2005)
28. Zajicek, M.: Successful and available: interface design exemplars for older users. Interacting with Computers **16** (2004) 411–430
29. Diaper, D., Worman, L.: Two falls out of three in the automated accessibility assessment of world wide web sites: A-prompt v. bobby. People and Computers (2003)

The Importance of Facial Expressions in Animated Agents in Spoken Dialogue Systems

Petra Hagström

Department of Computing Science
Umeå University, Sweden
kv03phm@cs.umu.se

Abstract. Even though graphical user interfaces are the most commonly used interface in technical applications, they are not the only option. One way to go is to use speech interfaces instead. But, speech systems of today are not always as robust as desirable. To increase the robustness of the system an animated agent can be added. Thus follows that not only the system itself but also the users' interaction with the system becomes even more complex. Also, many questions of how to design and implement the agent arise. This article focuses on how important facial expressions and other non-verbal conversational functions are for the experience of the system. The article gives an account for two already implemented systems, The August System and the Rea System, and also discusses how important embodiment is for the believability of the system. It has been shown that with respect to reactivity and believability of the system it is very important that non-articulatory movements are used to accentuate the auditory speech in the agent. Also, in evaluations of Embodied Conversational Agents (ECAs) in the context of multimedia education, it has been found that redundancy (i.e. gestures duplicate pieces of information conveyed by speech) increased verbal information recall, ratings of the quality of explanations, and expressiveness of agents along with higher likeability scores for the agent as well as a more positive perception of their personality.

1 Introduction

For a long time, graphical user interfaces have been the dominating interface in almost all technical applications, such as computers, cell phones and ticket machines. When it comes to computers, the most common input devices are a mouse and a keyboard. But typewriting requires a lot of training and learning. Also, illiteracy is a widespread problem around the world. Globally speaking, therefore typewriting remains one of the least widely spread communicative skills [1]. Also, graphical user interfaces could cause a lot of trouble for those motorically disabled or people with visual impairment. There are other options though. Speech technology is one of them. Gustafson et al. state "[s]peech technology promises to offer user-friendly interfaces for dialogue systems" [2]. One advantage of an operational speech interface is that the user is not limited in his/her input. The user does not have to browse through numerous menus or click on one link after the other when trying to access a certain point in the interface. According to Bell [1], “If the goal of a human-computer interface is to allow efficient and quick transfer of information, spoken dialogue is an excellent medium from the point of view of the human interlocutor”. But designing a speech system interface is not easy. Including an animated agent makes it even more complicated. Facial motions have been an area of research in many different disciplines. The face is the most complex and effective tool in human conversation, and convey for example information about the content of speech and a person’s emotional state [3]. The purpose of this paper is to investigate how important facial expressions are in an animated agent in a spoken dialogue system. It has its focus on two different speech systems, the August system and the Rea system.

The paper starts out with an overview of human face-to-face interaction. What kind of visual cues do we use in our daily face-to-face communication? How do we know when it is our turn to speak? How important are facial expressions for our understanding of different utterances? In this section we will look closer into all these tiny little things that come natural to humans, but have turned out to be very difficult to implement into speech systems.

The next section contains a brief overview of speech systems and how far the technology has progressed today. It also gives an account for already implemented systems, the August system and the Rea system. An attempt has been made to avoid an overboundance of technical details in this work. Brief overviews of the different systems are included to give readers a better understanding of the systems and how they make use of facial expressions. For more information about the systems, see [1, 2, 4–8].

Section 4 defines an embodied conversational agent (ECA). Cassell [6–9] has defined several so called “conversational models” and discussed how these can be implemented in an ECA. In section 5 the focus is on these conversational functions, such as; What conversational skills can be embodied and how do you implement human behavior? It also demonstrates how embodiment has been implemented into the speech systems in question. The last part of the paper discusses how important the conversational skills are for interaction with the agent. How does it affect the interaction and the overall experience of the agent? It includes the results of others as well as the writer’s thoughts about embodiment.

2 Human face-to-face interaction

To humans, speech comes naturally. Speech is an excellent mean for reasoning and problem solving and is commonly used, for example, for social and bonding purpose [10]. Conversations are considered to be an orderly event, governed by rules. But not two conversations look exactly the same. Not two people use exactly the same language. Also behaviors differ from person to person, as well as from conversation to conversation [7]. Thinking about it, a dialogue is a very complex phenomenon. Spoken human-human face-to-face interaction includes facial expressions, head-movements, gaze, and eye blinks for example, but also more complex behavior like non-verbal sounds as hums and grunts, mid-utterance, pauses and a flexible turn taking [10]. Also, the pitch and the melody of our voice is used to clarify what we are saying [8].

Have you ever thought about how you invite another person to a conversation? You will probably turn to face the potential interlocutor, look him or her in the eyes. You might even give a little smile. While engaged in the conversation, how do you know it is your turn to speak? In fact it does not only depend on the content of the conversation. Determining whose turn it is to speak and when the turn should be given away includes many factors, among them for example intonation and gaze. Using the voice is not the only way to interrupt a speaker, also gestures are used by the listener to indicate he or she wants the turn [6]. Figure 1 shows some examples of how communicative functions are mapped to particular behaviors. All these human behaviors, which we hardly pay any attention to while talking, turn out to be very complicated to implement into a speech system.

Communicative Functions	**Communicative Behavior**
Initiation and termination	
Reacting	Short glance
Inviting contact	Sustained glance, smile
Distance Salutation	Looking, head toss/nod, raise eyebrows, wave, smile
Close salutation	Looking, head nod, embrace or handshake, smile
Break away	Glance around
Farewell	Looking, head nod, wave
Turn-taking	
Give turn	Looking, raise eyebrows (followed by silence)
Wanting turn	Raising hands into gesture space
Take turn	Glance away, start talking
Feedback	
Request feedback	Looking, raise eyebrows
Give feedback	Looking, head nod

Fig. 1. Examples of conversational functions and their behavior realization, redrawn after Cassell et al. [6].

3 Speech systems

Spoken dialogue systems are developed to provide a mean for people to interact with computers using spontaneous speech [1]. According to Bell [1], a spoken dialogue system can be described as a "system that recognizes natural spoken input, interprets its meaning in the current context, decides which action perform based on that interpretation and subsequently generates an appropriate answer" (see Figure 2).

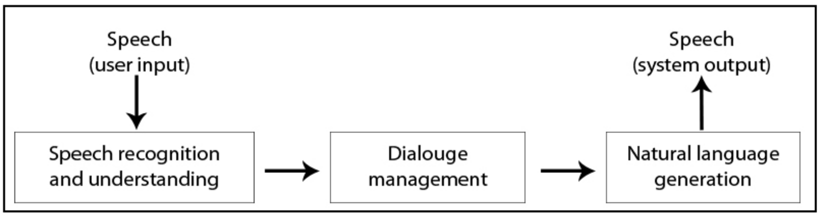

Fig. 2. A spoken dialogue system, redrawn after Bell [1].

It has turned out to be very difficult to model conversations artificially using a computer system. However, the field of automatic speech recognition (ASR) has progressed over the past few decades. The advances in technology have made it possible to develop systems with relatively high performance capable of recognizing large-vocabulary continuous speech. Thanks to the technology advancements the systems are user-independent and also works sufficiently even in noisy environments [1]. The ASR systems are usually built by using statistically-based recognizers. To improve the quality of the recognizers they are first trained acoustically on very large domain-independent corpora, and thereafter primed on a smaller set of domain-specific data. However, as Bell points out, the ASR systems do not solve the problem of language *understanding*, since the only thing they do is to transcribe the speech signal into a sequence of words. For many applications, such as dictation, this might be good enough, but spoken dialogue systems need more advanced techniques. In these more complex systems ASR is only an input component for higher levels of analysis [1].

Today, single-domain systems with restricted vocabularies and grammars have been successfully implemented for commercial purposes. These systems are usually accessed by telephone and are specifically designed to provide information about a narrow task, such as information about bank statements, weather forecasts or time tables [1]. The freedom of the users' spoken input is often very limited to minimize the risk of errors and increase the systems accuracy. Many of us recognize an instruction such as "To confirm your flight, say confirm" as an example from these telephone-accessed systems.

There are also some currently available commercial systems where the user is allowed to use speech input in a more spontaneous manner. These systems are called key-word spotting, because the system picks out the important words in the dialogue. However, users are still restricted to a narrow and well-defined task [1].

3.1 The August System

The August system is a multi-modal spoken dialogue system in Swedish (see Figure 3). It was developed at the CTT (Center for Speech Technology at KTH). The system is featuring an animated agent, named after the 19th century author August Strindberg, and therefore the talking head is also made to resemble

Strindberg. August was placed at the Stockholm Cultural Centre, downtown Stockholm, where any visitor had the opportunity to interact with the system [1, 2, 4]. Instead of handling one complex domain the August system was constructed to handle several simple domains [5]. The simplest configuration presented information about restaurants and other facilities in Stockholm, about KTH, the research at CTT and about the system itself. August also had basic knowledge about the life and works of August Strindberg [2]. Since the system was placed in a public location, that is, in a noisy environment, a push-to-talk button was used to ensure the system captured only the utterances that were directed to the system.

User		System
What is your name?	→	I call myself Strindberg, but I don't really have a surname
When were you born?	→	Strindberg was born in 1849
What do you do for a living?	→	I can answer questions about Strindberg, The Royal Institute of Technology and Stockholm
How many people live in Stockholm?	→	Over a million people live in the Stockholm area
Do you like it here?	→	People who live in glass houses should not throw stones
Yes, **that** was a smart thing to say	→	Hello, I come from the department of Speech, Music and Hearing
The Royal Institute of Technology	→	The information is shown on the map
Thank you	→	You are welcome
Goodbye	→	Perhaps we will meet soon again
Yes, **it** might be that we will	→	Strindberg was married three times

Fig. 3. A conversation between August and a user, redrawn after Bell [5].

3.2 The Rea System

Rea ("Real Estate Agent") is a computer generated humanoid [6–8]. The system can sense the user passively through cameras and audio input. The visual agent has an articulated graphical body. Rea is capable of speech with intonation, facial display and gestural output (see Figure 4). The user stands in front of a large projection screen on which Rea is displayed. There are two cameras mounted on the top of the projection screen, which track the user's head and hand positions in space. To capture speech input, users do wear a microphone. As the name implies, Rea is a real estate agent and therefore handles the domain of real estate. Various models of houses appear on the screen behind Rea where she can show the user different features.

4 Embodied Conversational Agents

Embodied conversational agents are animated heads or animated heads with a body, which are defined as having the same properties as human face-to-face conversation. According to Cassell [8] embodied conversational agents are not only interfaces in which human bodies appear lifelike or believable in their actions and their reactions to human users; "Embodied conversational agent interfaces

Mike approaches the projection screen while Rea is gazing about idly. As Mike moves within range of the two cameras mounted on the screen, Rea turns to face him and says:
Hello. How can I help you?
Mike. I'm looking to buy a place near MIT.
Rea nods, indicating that she is following.
Rea. I have a house to show you.
A picture of a house appears on the screen behind Rea, who blinks and looks at the house and then at Mike.
Rea. It's in Somerville.
Mike. Tell me about it.
Rea looks up and away while she plans what to say.
Rea. It is big.
Rea makes an expansive gesture with her hands.
Mike brings his hands up as if he is about to speak, so Rea does not continue, waiting for him to speak.
Mike. Tell me more about it.
Rea. Sure thing. It has a nice garden.
Rea sketches a curved gesture with her hands, indicating the garden extends along two sides of the house.
Mike. How far is it?
Rea. It is five minutes to the Porter Square T station.
Rea makes it clear it is five minutes on foot from the T station by making a walking gesture with her fingers.
Mike. How big is the house?
Rea. It has four bedrooms, three bathrooms...
Mike interrupts Rea, who stops speaking immediately.
Mike: Wait. Tell me, where is the master bedroom?
Rea: I'm sorry, I didn't catch that. What did you ask me?
Mike: Where is the master bedroom?
Rea. It's upstairs.
Rea points up.
Mike. Where is the master bathroom?
Rea. It's next to the bedroom.
Rea brings her hands together to indicate the relationship between the bedroom and the bathroom.
And the house tour continues.

Fig. 4. A conversation between Rea and a user, redrawn after Cassell [8].

are specifically conversational in their behaviors and specifically humanlike in the way they use their bodies in conversation." Cassell further states four human abilities required in the system to define it as an ECA.

- Recognizing and responding to verbal and nonverbal input.
- Generating verbal and nonverbal output.
- Dealing with conversational functions, such as turn taking, feedback, and repair mechanism
- Giving signals that indicate the state of the conversation, as well as contributing new propositions to the discourse.

In an ECA, information from several modalities must be integrated to one representation of speaker intention. In that sense they are similar to multimodal systems [7]. According to Cassell referring to Steedman, what the multimodal systems are missing is a concept of non-verbal function with respect to conversational functions. That is, there is no notion of "speaking turn" or "information structure". Still, there are features in these systems that are of interest for researchers on ECAs since they highlight areas of potential difficulty, such as the problem of recognition errors in speech and gesture [7].

5 Conversational models

To get an animated agent to act in humanlike ways, there are certain behaviors that needs to be implemented. Cassell [8] has defined five properties of human conversation that needs to be modeled in a system for people to perceive it as humanlike in its conversational patterns.

- *Function rather than behavior*
  When building a model for handling conversation, the emphasis has to be on identifying the high-level structural elements that make up a conversation. Some of these elements were shown in Figure 1. They are for example, conversation initiation, turn-taking, feedback, contrast and emphasis, and breaking away. The distinction between function and behavior is important since one set of behaviors can produce a variety of different communicative effects, depending on the circumstances. Thus, the preferable approach is to decipher different communicative functions and thereafter look at what behaviors are used to express these functions.
- *Synchronization*
  In human conversation, behaviors that represent the same function or emphasize the same communicative goals occur in synchrony with another. For example, the meaning of a head nod in an utterance is interpreted in different ways depending on where the nod is put. Even a time difference of 200 msec is making a difference of the interpretation. Therefore, synchronization, and not only the behaviors themselves, is an important aspect when modeling a system.

- *Division between propositional and interactional contributions*
  Cassell refers to social scientists when stating that contributions to the conversation can be divided into "propositional" and "interactional" information. "Propositional information, which corresponds to the content of the conversation, includes meaningful speech, as well as hand gestures and intonation used to complement or elaborate on speech content". Examples of propositional information are gestures indicating size, as in "It was this small" - using the fingers to show how small it was - or using rising intonation to indicate a question, as in "You had ice cream?" . According to Cassell "Interactional information consists of cues regulating the conversational process and includes a range of nonverbal behaviors". It could be quick head nods to indicate one is following or showing one is giving up the turn by bringing one's hand to the lap and turning to the listener. Regulatory speech, such as "Huh?" and "Go on" is also part of the interactional information. Propositional functions shape the actual content of the conversation, while the interactional discourse functions account for making the conversation smooth and fluent. Both functions could be fulfilled through either verbal or nonverbal behavior.

- *Multithreadness*
  Propositional behaviors are usually longer in duration than their interactional counterparts. The variety of time scales involved is a striking characteristic of human conversation. When delivering a contribution to a conversation the speaker could go on for several minutes, but a pause as short as 500 msec is long enough to show the listener that the speaker needs some feedback to that the listener is following. When engaged in a conversation, only parts of the conversational behaviors are planned. The longer ones, such as deciding what to say, often belong to the category of planned behaviors. Shorter behaviors, such as producing feedback, are most of the time carried out unconsciously and therefore simply reactive. Thus, multithreadness is an important part of an ECA, where lots of information has to be processed at the same time.

- *Entrainment*
  While engaged in a conversation, participants synchronize their behaviors with one another to make up a smooth and efficient dialogue. Raised eyebrows, gaze and head nods are examples of behaviors entrained to synchronize turn taking. Entrainment is a prominent aspect of human-human conversation. Rea is not yet able to entrain her nonverbal behaviors to those of the listener. But users that interact with Rea do entrain to her after only one or two conversational turns by beginning to nod and turn their heads in synchrony with Rea's head movement. It is not only conversational behavior that is entrained. According to Carlson et al. [10] the linguistic behavior is also coordinated i human-human dialogue. This process is called lexical entrainment.

5.1 Embodiment in the August System

When designing the August system, the aim was also, apart from implementing convincing lip-synchronized speech, to implement a rich and believable non-verbal behavior [4]. To achieve this goal, a library of communicative gestures of varying complexity and purpose was created (see Figure 5). The parameters used to signal prosodic information, such as stressed syllables and phrase boundaries, are primarily eyebrow and head motion. But also other gestures and motions were implemented to communicate non-verbal information such as emotions, attitude and turn taking. Each gesture can be invoked at any point in time, either in-between or during utterances. If there is a conflict, articulatory movements will always supersede the non-verbal gestures [4].

Motion involved in gesture	Typical usage
Eyeblinks	Word boundaries, emphasis, idle random blinking.
Eye rotation	Thinking, searching
Head nodding	Emphasis, turn taking
Head turning	Spatial references (e.g. "the bathroom is over there"), attitude
Eyebrow raising	Mark words in focal position, stressed syllables, questions, emotions
Eyebrow frowning	Thinking, disagreeing
Twisting of mustache	Attitude e.g. proudness, aggressivness or flirting
Emotional expressions	Semanticaly motivated

Fig. 5. Examples of typical usage of some of the gestures in the August system (courtesy of Lundeberg and Beskow [4]).

To accentuate auditory speech with non-articulatory movements has turned out to be complicated. If words in focal positions are associated with eyebrow movements, the eyebrow motions could become predictable and the agent will look nervous. Therefore, in the August systems, a combination of eyebrow and head movement is used [4]. Also, six different emotions were implemented into the August system to be able to display different moods of the agent. A number of utterances in the system were assigned appropriate emotional cues, often paired with other gestures. In addition, to make the agent more believable, a set of listening and thinking gestures were created (see Figure 6).

As stated earlier, turn taking is an important aspect of human conversation. When designing the August system this was taken into account. Thus, visual cues for turn taking were implemented. For example, to indicate the end of a question phrase, August will raise his eyebrows or slightly tilt his head. August is also able to use visual cues to further emphasize the message, by for example turning his head to show directions [4].

Fig. 6. Examples of August's facial expressions listening (left) and thinking (right) (courtesy of Lundeberg and Beskow [4]).

5.2 Embodiment in the Rea System

When Rea acknowledges the presence of a user, she engages in a greeting behavior, turning to face the user and smiling. She also uses verbal output to acknowledge the user, in response to user's verbal greeting (see Figure 7). The same also applies for farewells [6]. Rea uses eye gaze, body posture, hand gestures, and facial expressions to contribute to, organize, and regulate the conversation [7]. In addition, she can also track some of her human interlocutor's behavior and understand how he/she uses these modalities. For example Rea tracks who has the speaking turn, and therefore only speaks when she has the turn. Rea always allow verbal interruption. If the user begins to speak while Rea is speaking, she yields the turn right away. If the user starts to make gestures the system interprets this as the user wants the turn, therefore ceasing to speak when the ongoing sentence is completed [6]. When Rea wants the turn when the user is speaking, she looks at the user, indicating that there is something she wants to say. When she is done speaking, several behaviors are used to indicate she is ready to give the turn back to the user. To show the user that he/she is expected to speak Rea looks at the user, drops her hands, and raises her eyebrows [7].

State	Output Function	Behaviors
User present	Open interaction	Look at user. Smile. Toss head.
	Attend	Face user.
	End of interaction	Turn away.
	Greet	Wave. Say "hello".
Rea speaking	Give turn	Relax hands. Look at user. Raise eyebrows.
	Signoff	Wave. Say "bye".
User speaking	Give feedback	Nod head, paraverbal ("hmm").
	Want turn	Look at user. Raise hands.
	Take turn	Look at user. Raise hand to begin gesturing. Speak.

Fig. 7. Rea's output behavior, redrawn after Cassell et al. [7].

Rea has several different ways to give feedback, using different modalities. She can use paraverbal utterances, such as “Mmm” or a shorter statement like “okay”. Other ways are to give a head nod or raise her eyebrows. Depending on the current conversational state and the conversational function she is trying to convey, Rea generates different speech, gestures, and facial expressions. This is done by modeling behavioral categories as discourse functions where Rea is able to combine multiple modalities in both input and output. Thus, depending on what is appropriate at the moment, Rea can choose any of several modalities when deciding on giving feedback [7].

6 Why is embodiment important

There is an ongoing debate as to whether embodied conversational agents are the appropriate way to go when expanding voice recognition systems. According to Cassell [8] critics such as Ben Shneiderman questions what function building humanlike computers really serve. Shneiderman points out that anthropomorphized (ascribing complex human skills) interfaces never succeeded in the past. However, these interfaces have not been given conversational embodiment, that is, giving the interface the appearance and the **function** of the human body in conversation. As Casell [8] states “Simply building anthropomorphized interfaces that talk (but don’t use their talk in humanlike ways) sheds no light on the debate about embodiment”. In the early 1990s researchers started to develop agents with bodies and autonomous embodied characteristics, often called autonomous synthetic characters. The researchers building these systems put focus on general interactional social skills, not on conversation. They found, sometimes to their surprise, that the best way to produce believability and lifelikeness was not always through the modeling of life; “They found themselves turning to insights from Disney animators and other artists about caricaturization and exaggeration as a way of getting users to suspend disbeliefs and attribute reality to interactive characters” [8]. However, in the development of embodied conversational agents this drama-based approach has not made any significant impact. Instead, researchers find they have to turn to theories of human-to-human interaction and investigate all the tiny details of conversation to ensure their interfaces share the conversational skills of human users [8].

Facial expressions have been studied in many different disciplines, for example in computer graphics, linguistics and psychology [3]. For example, de Gelder and Vroomen [11], have conducted experiments showing that people combine data from two sources (tone of voice and facial expression) to arrive at a judgment about the person’s state of mind. They concluded a person cannot ignore the facial expression while asked to only judge the tone of the voice. They also showed this yields for the other way around, a person cannot ignore the tone of voice when asked to judge the facial expression. Results like the one from de Gelder and Vroomen indicate that facial expressions probably also are a very important aspect in animated agents.

There is also an experiment conducted by Granström et al. [12] exploring the relevance of both acoustic and visual cues for signaling feedback in a conversation. The study focused on “negative” and “affirmative” feedback. The results showed that both visual and acoustic signals play an important role in a feedback utterance. Four out of six parameters turned out to have significant influence on judgement, where smile and fundamental frequency were the most prominent. Also eyebrow- and head-movement had significant influence. Eyeclosure and delay did not show any significant influence, but still seemed to contribute to the judgement.

By using a MPEG-4 compatible talking head, interaction between facial expressions and articulation can be captured [13–16]. Infrared (IR) reflecting markers are attached to the subject's face and 3D coordinates for each marker are registered. Several motion corpora have been collected using this technique, where the subjects for example have been asked to imitate different moods or varying the focal accent position in sentences. The recorded motion data can be converted into MPEG-4 facial parameters (FAP), which are used to model the talking head by using the collected motion data as training data for the synthesis algorithm [17]. But the collected data can also be used to analyze different aspects of human communication. Of special interest for this paper are the findings of Granström et al. that a word in focal position results in greater dynamics in all facial movements compared to a non-focal pronunciation, and also in some instances the expressive state (mood) has a stronger influence on articulation than do different vowels [13].

Although gestures are not the main focus of this article, it is still interesting to point out Buisine and Martin [18] investigation on the effects of speech-gesture cooperation in animated agents. In their research, they focused on two types of speech-gesture cooperation, redundancy, where gestures duplicate pieces of information conveyed by speech, and complementarity, where the gesture conveys a new piece of information which was not conveyed by speech (e.g the agent points at something without referring to it by name when speaking, e.g “went from here to here”). Buisine and Martin found, in the context of multimedia education, that redundancy increased verbal information recall, ratings of the quality of explanations, and expressiveness of agents along with higher likeability scores for the agent as well as a more positive perception of their personality. Of even more interest for this paper, they also found the only variable that was influenced by the agent's appearance was likeability. One of the agents, Marco, was always the agent (three different agents were used) receiving the highest likeability scores, despite whatever speech-gesture strategy he was selected to use. When analyzing comments from the participants, it showed the key feature for Marco's high likeability scores was his wide smile. Buisine and Martin's results are along the same line as Lundeberg et al. who states: “Having the agent accentuate the auditory speech with non-articulatory movements is found to be very important with respect to reactivity and believability of the system” [4].

6.1 Two methapors

Edlund et al. [19] state the research is often guided by a wish to achieve more natural and human like interaction. "But natural is a fuzzy term at best, and a spoken dialogue system does not automatically become better just because a component that is in some manner human-like is added". They propose that users may, knowingly or subconsciously, interpret spoken dialogue systems in more than one way. "More specifically, there is more than one metaphor people use in order to make sense of spoken human-computer dialogue." They call the two metaphors the interface metaphor and the human metaphor. In the interface metaphor "the spoken dialogue system is perceived as a machine interface - often but not always, a computer interface". In this metaphor the user interacts with the system in a way that for example would be equivalent to pick a choice in a scroll down menu. In the human metaphor, "the computer is perceived as a human-like creature. Much like a television set which is not part of the film it shows, the computer is not itself represented in this metaphor". Edlund et al. think it is important that researchers and developers are aware of how they want their system to be perceived by its users and strongly suggest to avoid mixed metaphors.

7 Discussion

As Granström and House state [13] "...understanding the interactions between visual expressions, dialogue functions and the acoustics of the corresponding speech presents a substantial challenge". This quotation summarizes the challenges speech systems researchers are facing when trying to implement embodiment into a system. To study the interaction and increase the understanding will be a tedious work. Almost every aspect of human conversation will have to be studied in every tiny detail, and preferably during different circumstances. The workload becomes even greater when realizing that it is not only the articulatory movements visible on the outside of the face that has an impact on speech acoustics. Also, internal parts such as the tongue, palate, jaw, and teeth have to be taken into account. To get realistic articulatory measurements from internal parts, techniques such as MRI (Magnetic Resonance Imaging), EMA (ElectroMagnetic Articulography), and EMG (ElectroPalatoGraphy) are used [20].

In Buisine and Martin's study, the agent with the widest smile got the highest likeability scores. This result shows that facial expressions are really important. If people have a high likeability of the agent, they will probably be more prone to interact with the agent and perhaps also interact with the agent for a longer time. For educational purposes the likeability variable might be especially important, since students have to listen to the agent for a long time, and might find the lecture more interesting if they have a high likeability of the agent.

Also trustworthiness is an important aspect of speech systems. Cassell and Bickmore [21] state "... trustworthiness affects how we act and converse with one another. When we do not trust, we neither believe what others say to us nor can we learn from them". Therefore, trustworthiness might be the single feature that is of greatest importance, apart from the conversation it self, in an embodied speech system. According to Cassell and Bickmore [21], of high significance when trying to make a system trustworthy, is to ensure that all communicative modalities deliver the same message .

Also, synchronization is of great value. A system where different modalities do not act in synchrony with another will not appear believable. People might not feel comfortable interacting with the system. Even though they might not be able to point out that their unease is due to poor synchronization, they will probably still sense that something is not exactly the way it is supposed to be. Thus, it will probably also be easier to misinterpret the system. Entrainment is another aspect that will make the system more robust. If the system is able to entrain to each user, the risk of errors in the conversation will probably decrease since the user and the system will be at, or at least closer to, the same level.

It is important though, that embodiment is not the only part focused on. The speech technology has to make the same progress. People may become confused and overestimate how and in what ways they can communicate with the agent if it is too humanlike in its appearance. If they believe it to be a "human" and therefore interacts with it in the same way as they would with other humans, they will probably get frustrated if it turns out the agent does not understand them or gives speech responses that are not appropriate.

Another important aspect is to think about where the system is going to be used. Talking heads could be a useful application for, for example hearing- or visually impaired, educational software, language learning, entertainment, experimental setups, and high quality text-to-speech applications such as news readings. Realistic embodiment will be more important in some of these applications than others. It is important that the systems are implemented to fit in the environment where they are going to be used and not become complex and sophisticated only because technological advancements make it possible.

There is no doubt that facial expressions are very important when interacting with an animated agent. They contribute with important information that sometimes cannot be accessed in any other way, even though also linguistic features are very important. In some cases facial expressions might be even more important in animated agents than in human face-to-face interaction, since technology limits what could be expressed by linguistic features. Facial expressions might for example be helpful to improve the interaction with consideration to error-correction, but are also important for the believability and trustworthiness of the system. Also, facial expressions may be important because of the simple reason that they do make the face look alive, and therefore make the interaction with the agent more interesting.

8 Conclusion

This paper has focused on facial expressions and other non-verbal communicative behaviors in animated agents in spoken dialogue systems. Such behaviors are important for the conversational functions, as for example initiation and turn taking. It is also argued that such behaviors enhance the trustworthiness and believability of the system. In future research it might also be an approach in the aim to make speech systems more robust. The paper gives an account for two already implemented systems the August system and the Rea system, and also contains argumentation for embodiment as well as a discussion about current research and what has to be taken into account in future research.

References

1. Bell, L.: Linguistic Adaptations in Spoken Human-Computer Dialogues - Empirical Studies of User Behavior. PhD thesis, Department of Speech, Music, and Hearing, KTH (2003)
2. Gustafson, J., Lindberg, N., Lundeberg, M.: The august spoken dialogue system. In: Proceedings of Eurospeech99. (1999)
3. Wang, A., Emmi, M., Faloutsos, P.: Assembling an expressive facial animation system. In: Proceedings of the 2007 ACM SIGGRAPH symposium on Video games. (2007)
4. Lundeberg, M., Beskow, J.: Developing a 3d-agent for the august dialogue system. In: Proceedings of AVSP99, ESCA Workshop on Audio-Visual Speech Processing. (1999)
5. Bell, L., Gustafson, J.: Interaction with an animated agent in a spoken dialogue system. In: Proceedings of Eurospeech99. (1999)
6. Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsson, H., Yan, H.: More than just a pretty face: Conversational protocols and the affordances of embodiment. Knowledgebased Systems **14**(1-2) (2001) 55–64
7. Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjálmsson, H., Yan, H.: Embodiment in conversational interfaces: Rea. In: Proceedings of CHI. (1999) 520–527
8. Cassell, J.: Embodied conversational interface agents. Communications of the ACM **43**(4) (2000) 70–78
9. Cassell, J., Tartaro, A.: Intersubjectivity in human-agent interaction. Interaction studies **8**(3) (2007) 391–410
10. Carlson, R., Edlund, J., Heldner, M., Hjalmarsson, A., House, D., Skantze, G.: Towards human-like behaviour in spoken dialog systems. In: Proceedings of Swedish Language Technology Conference. (2006)
11. de Gelder, B., Vrommen, J.: The perception of emotions by ear and by eye. Cognition and Emotion **14**(3) (2000) 289–311
12. Granström, B., House, D., Swerts, M.: Multimodal feedback cues in human-machine interactions. In: Proceedings of the Speech Prosody 2002 Conference. (2002) 347–350
13. Granström, B., House, D.: Inside out - acoustic and visual aspects of verbal and non-verbal communication (keynote paper). In: Proceedings of the 16th International Congress of Phonetic Sciences. (2007)
14. Granström, B., House, D.: Multimodality and speech technology: Verbal and non-verbal communication in talking agents. In: Proceedings of EuroSpeech 2003. (2003) 2901–2904

15. Beskow, J., Cerrato, L., Granström, B., House, D., Nordenberg, M., Nordstrand, M., Svanfeldt, G.: Expressive animated agents for affective dialogue systems. In: Tutorial and Research Workshop on Affective Dialogue Systems, ADS'04. (2004)
16. Beskow, J., Granström, B., House, D.: Focal accent and facial movements in expressive speech. In: Fontetik 2006, Working papers 52, General Linguistics and Phonetics, Lund University. (2006)
17. Beskow, J., Nordenberg, M.: Data-driven synthesis of expressive visual speech using an mpeg-4 talking head. In: Proceedings of Interspeech 2005. (2005)
18. Buisine, S., Martin, J.C.: The effects of speech-gesture cooperation in animated agents' behavior in multimedia presentations. Interacting with Computers **19**(4) (2007) 484–493
19. Edlund, J., Heldner, M., Gustafson, J.: Two faces of spoken dialogue systems. In: In Interspeech 2006 - ICSLP Satellite Workshop Dialogue on Dialogues: Multidisciplinary Evaluation of Advanced Speech-based Interactive Systems. (2006)
20. Engwall, O.: Tongue Talking - Studies in intraoral speech synthesis. PhD thesis, Department of Speech, Music, and Hearing, KTH (2002)
21. Cassell, J., Bickmore, T.: External manifestations of trustworthiness in the interface. Communications of the ACM **43**(12) (2000) 50–56

Mobile computer technology as an aid for memory

Andreas Wikström

Department of Computing Science
Umeå University, Sweden
dit04awm@cs.umu.se

Abstract. This paper studies in what ways mobile computer technology can be used to help people with age-related dementia. Dementia is a fairly common illness in todays western societies therefore the paper considers some of the different approaches used in other studies including tracking, prompting to remember tasks and lifelogging. It also presents some of the challenges and problems with designing a mobile device as an aid for memory. These include the challenges with designing for people who have troubles with learning as well as bad eye-sight and other age-related impairments. It also points out the importance of clear presentation of information and the problems with further cognitive decline in the users. The paper concludes that mobile computer technology has potential as an aid for memory and finally gives some pointers to what must be considered when designing such a device.

1 Introduction

The purpose of this paper is to research in what ways mobile computer technology, such as a mobile phone or a PDA, can aid memory for people with impaired memory, particularly for people with degrading memory due to age (such as people suffering from Alzheimer's disease). It is also interesting to see what aspects are important when designing such a device. This is of some interest because of the aging population in western societies, among whom dementia is a quite common illness [1]. It has been estimated that over 4,6 million people in the EU suffered from dementia in the year 2000 [2]. There is also an exponential increase in the frequency of dementia with age and this does of course result in both social and economic difficulties [3]. It is however very difficult to determine exactly how common dementia is since the methods for analysis and the definitions vary [1].

Incorporating an aid for memory into a mobile device could possibly be a good solution because mobile phones, PDA:s or other techniques does not only provide the user with computational power they are also easy to carry. Thus they are able to aid the memory in many different situations. This is important because you do not know in advance when you need help remembering something.

The paper starts with a brief introduction to age-related dementia and Alzheimer's disease and its consequences for memory loss. Section 3 through 6 constitute the main part of the paper investigating how to effectively design a mobile aid for memory. Section 3 explains functionality that could be used in such a device. Section 4 delves deeper into the usability of a mobile device aimed at helping people with age-related dementia. Some of the challenges with developing such a device are presented in section 5. In section 6 some of the more general problems (such as stigmatization and security issues) one might need to think of will be presented. The paper then discusses the findings and possible future research in section 7 and finally in section 8 the conclusions are presented.

2 Background

Dementia is a description of symptoms that affect the brain, and in particular causes the gradual death of brain cells. This leads to the loss of cognitive abilities such as memory, reasoning, planning and even behavior [4]. The symptoms get worse when the disease causing dementia progresses. The cause of dementia can vary widely, but the most common one is Alzheimer's disease (AD).

The progression of AD can be divided into three phases, the early stage, the middle stage and the late stage [5]. In the early stage people with AD can usually understand their diagnosis and participate in decisions regarding their future care. Some of the symptoms can be mild forgetfulness, difficulties with learning new things and following conversation, getting lost or not being able to follow directions and withdrawal from usual activities. In the middle stages people can have a tendency to wander and become lost. Symptoms include forgetfulness about personal history, disorientation, confusion, needing assistance for daily tasks (i.e. dressing, using the toilet etc.) and language difficulties. In the late stages people are unable to take care of themselves and need assistance at all times. Some of the symptoms in the late stages are severe speaking difficulties, loss of control over bladder and bowels and difficulties with eating and/or swallowing [5].

Caregivers for people suffering from impairment of the episodic memory (IEM), which is a symptom of dementia, use a lot of energy helping out with remembering past events (i.e. supporting the episodic memory) [6]. Most commonly this is done by giving small cues, or details, to a memory when engaged in a dialog to help the person suffering from IEM to remember more of that particular event. If the event cannot be recalled the caregiver gives another small cue and so on until the event can be recalled. However, the caregiver only does this when he or she believes there is a reasonable chance for the person with IEM to recall that event. An event that cannot be recalled only results in frustration for both parts. It is also common for caregivers to let incorrectly recalled events go as it would simply be to much work to correct. Also because of the strain on the caregivers they do not always give the cues in an incremental fashion, but sometimes just give the person with IEM all the details of an event. When the caregiver continues with giving cues there is however some level of satisfaction for the person with IEM when they can remember an event given just the right amount of cues.

Another problem for the caregivers is the frequent repetition of questions which is common for people in the early and middle stages of AD [5, 7]. There are several reasons for this behavior, but the dominant is the occurrence of short-term memory loss. According to Hawkey et al. [7], the techniques used by caregivers are mostly to just answer the questions or refer the person with AD to a source for information (such as telling the person to look at the clock if he/she asks for the time). However they sometimes, due to stress, react in different ways. The reactions can for example be to ignore the person with AD, yell, get impatient, leave the room or not believe that the person with AD does not know what he or she asks for. People with AD sometimes also want to confirm information that they gather by themselves with the caregiver, thus not relieving the caregiver from having to answer the question. Hawkey et al. [7] for example report of one case where a woman would go between her calendar and her husband repeatedly apparently not trusting her ability to understand the information.

3 Functionality in existing memory aids

In studies by Inglis et al. [8], one of the participants who was suffering from impaired memory said "Imagine a memory which is outside you and responsive to you but does not control you". This pretty much sums up what a mobile aid for memory would have to be able to do, but there are several different ways and approaches to try to achieve this.

Regarding technical aids for memory they are mostly focused on helping the people suffering from impaired memory by prompting them to remember tasks, appointments and such in a compensatory way [8]. There are also restorative methods which focus on the rehabilitation of the memory, but currently these do not seem to use technology to any great extent. Also, since compensatory methods show greater potential they should probably be the methods of choice [9].

There are several different approaches for helping people with impaired memory, one being a device called Guide Me [10] where the idea is to provide caregivers with the possibility to supervise the person with impaired memory (AD in this case). Mainly this is supposed to help if the person with AD suffers from blackouts or tends to wander. The Guide Me is a fairly simple solution consisting of two devices, one attached to the person with AD and one handheld device for the caregiver. The device attached to the person with AD has only one button that, if pressed, sends out a signal to the caregivers' handheld device. The caregiver can locate the person with AD via tracking of the attached device and the whereabouts of the person is displayed on a map presented on the caregivers' handheld device.

The following feedback was gathered from testing the Guide Me in simulations [10]:

- The caregiver does not require location information of the patient at all times

- The caregiver wants to know if the patient is having a blackout and wandering the streets
- The patient did not want to be intrusively contacted, for example by incessant phone calls, whenever the partner wanted to know the patient's whereabouts
- Both parties agreed that there should be no excessive communication. The communication should be subtle and only when needed
- In emergency situations, the initiative of tracking should lie upon the caregiver
- The patient should carry or wear the transmitting device at all times since blackouts can occur anytime
- The device worn by the patient should not be designed as a stigmatizing tag

To solve the problem with people getting lost Wu et al. [11] have constructed a system that relies on the user for input of information about the situation or space he/she is in. That information can later be retrieved if the user gets lost. The authors' address the potential difficulties with having the users input all data by themselves, but since they target people with anterograde amnesia they concluded that this was not a problem. The explanation for this is that people suffering from anterograde amnesia most commonly are aware of when they could be at risk of becoming lost. This kind of awareness of ones memory issues is however not all that uncommon for people with age-related dementia (such as AD) either [7].

Morris et al. [12] list some of the social barriers for people with declining cognitive abilities. The barriers mentioned are difficulties following conversation, forgetting names and faces and fears of imposing (i.e. fear of being a burden on caregivers etc.). Regarding these barriers the writers continue to suggest conceptual features which would help. Their features include "pace control" to enable replaying parts of a conversation that has been to fast to comprehend, memory cues to help remembering people and faces and "opportunity hunting" to find good opportunities for people to connect with other people (such as caregivers, family members etc.). These could all be valuable things to incorporate into an aid for memory. Further explaining the use of their social memory aid Morris et al. suggest that the people with declining cognitive abilities should select a tangible photograph and then questions and clues about that person would be presented on a familiar interface such as a TV screen. This is based on ubiquitous technology such as radio-frequency identification-tags and -readers. Portable interfaces were also being explored when the article was written. This to be able to give cues and information independent of the situation the person is in.

A device developed by Szymkowiak et al. [13] suggests that the most important information for an elderly with impaired memory is to be able to know what tasks should be carried out on a given day. These tasks can range from simple tasks like brushing ones teeth to more advanced tasks like meeting someone for tea. Their device is a PDA which can show reminders about the tasks and also has the possibility to create or modify tasks. Their device is also accessible remotely via the Internet so that a caregiver can modify and add tasks. The device has more functions like a diary and the ability to see a calendar on which a day can be selected and show the tasks scheduled for that day. The focus however lies on listing the activities for the current day and supplying the user with reminders for each activity. The reminders are supplied by prompting the user to perform a task and the user should then acknowledge the prompt with the touch of a button. Acknowledgment can come in two forms, either to not be reminded again or to be reminded again at a later moment, this to handle situations where the user cannot or do not want to perform the task at the given moment. Another study where reminders have been used is Wade and Troy's article [14] in which mobile phones were used to prompt people to remind everyday activities. This worked by simply having an automated system call the mobile phone and present a prerecorded message reminding the user to perform a certain task. Although this study focused on people with impaired memory due to brain damage rather than dementia, they still saw vast improvements on the users' abilities to remember tasks like taking their medication, brushing their teeth and making meals. This is similar to some of the problems people with age-related dementia can have and thus supports the argument that prompts can be a usable aid.

Life-logging could also be an interesting approach for aiding memory as it basically tries to capture and store information about everything a person experiences. There are several devices used for life-logging, one being the "SenseCam" which automatically captures events in the daily life of the user. It can for example take photographs and capture temperature and movement. The device is worn around the neck and of course contains a camera as well as several other sensors [15].

This section has explained a few devices aimed at helping memory. Listed below are the primary functions of these devices:

- Supervision of people who might become lost.
- Supplying user-generated information about the current situation so that the user will not become lost.
- Aiding the possibility to overcome different social barriers.
- Prompting to remember tasks (two different devices).
- Lifelogging.

4 Usability

Regarding the ease of use of a device, which is important especially if the user has no experience of using the device, Inglis et al. [8] got several comments. These included that the interface should have big and clear buttons and prompts and that the system should be "nice and friendly". This can be related to general design aimed at older people, where such things as bad eyesight might complicate the usage of any device. To solve the problem with bad eyesight auditory feedback has been shown to support usability [16]. There are however several other factors one has to consider when designing for the elderly. Not only eyesight but also intellectual skills (such as attention, concentration and problem solving), hearing and motor skills might be affected by age [17–19]. Also, even though cognitive abilities do decline, it is impossible to say to what extent or how fast they decline. Even knowing which abilities will decline is difficult since it varies between individuals to a great extent [20].

A general solution to make older adults able to use a device is to try to keep it simple. This can for example be done by presenting the user with limited options and very simple search spaces. One can also choose to leave out features that might not be necessary or are too complex [21].

In their studies, Loh et al. [10] got comments about keeping the functions down to only the required ones as well as having minimal aesthetics. This along with solid and robust design made their device perceived as reliable and it also conveyed the functions effectively. There is however a conflict between having many and advanced features and keeping the device simple which must be considered. This also conflicts with scalability since a simple solution might not be possible to scale up to support more features [22].

Large text-sizes and buttons are of course also something that can be used to keep the interface and features clearly visible. Having large buttons can also make it easier for the user if he/she has trouble with his/her fine motor skills [23]. It is also important to keep the administration of the interface itself to a minimum (i.e. by trying to avoid scrolling, resizing etc.). Having good feedback is also important to clearly state that a task has been successfully (or unsuccessfully) completed since older adults can have a tendency not to expect to succeed when interacting with an unfamiliar device [22].

Most of these guidelines are quite general and are of course similar to design guidelines that are not aimed specifically at older adults, but nonetheless it is important to have these things in mind. Even more general guidelines are to be consistent when presenting the user with information and functions so that similar things appear in similar ways. It is also essential to provide important information as close to the top of any hierarchy as possible and to provide feedback of what functions that are currently available [21].

5 The challenges

Worth to take notice of regarding the Guide Me [10] is that it apparently was not possible to expect the person with AD to actually use the signal button when a blackout occurred since he/she would probably be too confused to use the device. This kind of problem poses the question of to what extent one can rely on a person with degrading memory to actually be able to use a new device and different researchers all have their own way of handling the problem.

In the research by Wu et al. [11] the user required quite a bit of training to be able to use the device and according to Inglis et al. [8] this is not optimal since required training should be kept to a minimum. This can of course be supported by the fact that one of the early symptoms of AD is the problems with learning new things [5]. Morris et al. [12] suggest presenting information on familiar devices, such as a TV-screen, which would alleviate the problem of learning a new tool. The writers also mention mobile interfaces, focusing on wearable interfaces such as hearing aids and glasses. They do not however mention the use of a mobile phone or any other more technologically advanced devices, which makes sense since it is not certain that someone suffering from age-related dementia has any experience of using such a device. Szymkowiak et al. [13] reflects upon the ease of use of their device and decided not to use any menu structures but rather show the most important functions directly on a default screen. This was done to minimize the need for training and the load on working memory. They also mention the benefits of a touch screen as the interface can be adapted for the target users in many different ways. One way in which they take advantage of the touch screen is to use large and clearly visible buttons in contrast to the often tiny buttons on mobile devices.

If one aims to help the user remember events, experiences, people etc. it is important that the cues given are distinctive and personally significant [6]. Lifelogging has the potential to remember almost everything someone experiences and thus can give a lot of cues. It does however suffer from creating a vast amount of data for the user to go through. The amount of data that can be generated is not necessarily needed to support the episodic memory. Instead it demands that the user sorts through all this data to find the cues that really are helpful. This can be quite problematic since people with memory problems often also have limited cognitive resources for sorting through this kind of material. The caregivers can thus be burdened with having to go through all the data to find good cues [6].

When it comes to memory aids one also has to consider whether or not the user actually can remember to use the aid. For example there were concerns about the users' ability to remember to wear the Guide Me [10] as it was a clip-on. To solve this problem it was suggested that the device could be permanently worn, perhaps in the form of an implant. This is possible if the device is a localization tool and does not rely on the user to interact with it, but if the user needs to interact with the device this solution gets more troublesome.

6 Problems

When considering a device for tracking people who might become lost there are some problems that should be considered. In the simulations of the Guide Me [10] some of the comments gathered pointed out that the caregiver did not need to know the location of the person with AD at all times but only when he/she was lost. But since the person with AD was too confused to use the device to send out a distress signal he/she would have to wear the device at all times. This means that the caregiver do have knowledge about the position of the person with AD at all times and must try to figure out whether or not he/she is lost based on his/her position.

Since the caregiver knows where the person with AD is at all times there might be a concern about personal integrity, this however was not mentioned as a problem in the article. Worth pointing out is that the study was conducted with only a few participants and thus such concerns might have been overlooked.

There were however comments about that the communication through the device between the person with AD and the caregiver should not be intrusive which of course can be related to the problem with integrity. There were also comments about stigmatization which is important to avoid, especially since it is quite common that people are embarrassed of their disease and thus might feel ashamed of using a visible aid [10, 14, 11]. The mobile phone and other commonly used mobile devices do have some potential in this area since they do not imply any illness. Wade and Troy [14] do point this out in their article but mentions it mainly as something positive for younger people, but the benefits also for older people should not be neglected.

Regarding more compensatory ways of aiding memory like the possibility to get information about people and events for example from photographs or similar items (like the system in the article by Morris et al. [12]) there might be a concern that the memory could get even worse when relying too much on the system. This could also potentially be a problem with devices that give the user reminders about tasks that should be performed (such as the devices in the articles by Szymkowiak et al. [13] and Wade and Troy [14]). Since there are studies showing that both physical and cognitive activity lowers the risk of dementia [24, 25] there might be reasons to believe that relying to much on a system and thus not being activated, at least not cognitively, to the same degree an aid might actually worsen the illness. It is not however certain in any way that memory will decline even further due to the lack of training as the device might actually provide training rather than prohibit it. One person in the study by Wade and Troy [14] did not need to be reminded any more after a few weeks of being prompted to perform a certain set of tasks since he had learned what each message would say. This is interesting to note even though the study focused on people with memory problem due to brain damage rather than age-related dementia. The same principle could possibly hold true even for people with age-related dementia. Perhaps not totally relieving the person with dementia from having to use an aid, but it could maybe train the person so that he/she could be more self-reliant. This is however not proven in any way.

There are problems with introducing technology not just because it might be difficult for the user to learn how to use it, but also because it might become more difficult for the user over time. For example, several participants with AD in the study by Hawkey et al. [7] did no longer know how to operate the remote to the TV even though they had been able to do this before. There were more examples of this kind of decline in the ability to use tools and devices that they had previously been able to use including the radio and telephone. It is therefore important to have the future decline of the users' abilities in mind when designing an aid for memory.

A problem with using a mobile phone or other mobile device is the possibilities to use it for other things. For example, one of the participants in Wade and Troys study [14] could not stop himself from using the phone to make calls and was worried that he would build up a substantial bill that he could not afford. He was therefore forced to stop using the phone.

Security risks are also a problem that one should have in mind, perhaps especially so when using lifelogging techniques. Since lifelogging stores information about every situation a user experiences [15] it is vital that the information is only accessible by the user and perhaps a caregiver, once again referring to the personal integrity of the user. It is also problematic to simply have a password to protect the information since it can be difficult to remember a password even for people without impaired memory. So some other, more sophisticated, technique would be needed.

7 Discussion

Going through the different approaches of aiding memory it is clear that mobile computer technology has a lot to offer in terms of functionality and availability. Devices used for tracking, prompting to perform tasks or gathering information about everyday life are some of the solutions available. Some of the positive features which such devices offer is of course the mobility but also the connectivity (whether by simple telephone, access to networks, GPS and so on), the social acceptance and, if the device has for example a touch screen, the possibility to adapt it to its target users.

What is also clear is that there is apparently not yet any complete solution that solves all problems for people with age-related dementia. This can be seen in section 3 where all of the devices aims at solving one single problem (except for life-logging that aims at remembering everything). That is not to say that such a device is necessarily what should be aimed for since different people have different needs and symptoms as well as ability to use technological devices.

I would suggest that one should focus very specifically on one symptom or need and solve that problem. Preferably in a way that does not require the user to learn any new skills since one symptom of AD is the problem with learning new things [5]. Because of this, one could possibly draw the conclusion that it is more difficult for a person with AD to remember how to use, or even bring with them, a device he/she has not used regularly before than it is to remember something that he/she has used regularly for a long time. This suggests that it could be more interesting to include an aid for memory in a mobile phone or PDA, not for the people currently suffering from age-related dementia, but rather for the people that now uses the mobile phone in their daily lives and might be in need of an aid for memory in the future. However, one cannot simply ignore design or usability issues when targeting older people since even though coming generations may be more used to using mobile devices there will still be people who are not used to such devices for many years to come. People will also get the same symptoms as older people of today, including bad eye-sight, declining motor skills and so on. These and other aspects are further developed in the article by Hawthorn [22].

One will also have to consider the memory issues. Even though a person might be used to using for example a mobile phone, he or she might not be able to learn how to use a new program for their phone if he/she has already started having memory issues. Also, one would probably not want to use such a specific program for helping the memory if one does not have memory problems. Therefore the possibility that people will learn a tool in advance is quite low. There is however the possibility that such techniques as lifelogging does evolve into something that ordinary people use as most such devices are not aimed specifically at people with impaired memory. This has the potential to dramatically decrease how disabling age-related dementia might be.

To make a mobile device work as an aid for memory there are some requirements that needs to be fulfilled. The compensatory information must be presented in a clear way that actually aids the user, it might for example not be enough to present what a person needs to do but also why he/she needs to do it [8].

Developing the idea of a clear presentation, it is not only important that the information is clearly visible or audible for the user, but it must also mean something if it supposed to be a cue for memory. Simply put, it must give the user information about what the user needs information about. The device should also require little or no effort or learning to use. It is also interesting to note that a mobile device is most likely best suited to aid a person in the early and maybe middle stages of age-related dementia and thus it should be introduced as quickly as possible. This is because the aid should be focused on prolonging the users' ability to stay independent and there is probably no realistic way for a mobile device to achieve this for someone who is already in the late stages. Other things that one needs to think about is the general design issues when designing for older people, security and personal integrity issues (especially if the device aims at tracking the user or make us of lifelogging techniques), stigmatization and the possibility that the users cognitive abilities might decline even further.

There is no easy way to solve all of the problems, so the most important thing should be to be very clear on what one wishes to achieve when designing an aid for memory and also who the aid is being designed for. Further research might be focused on more specific problems to give a better view of how certain symptoms might be relieved. It is also interesting to research if, in what way and to what extent there is a risk for an aid to worsen the illness because of lack of training or if it can be beneficial to use an aid as it might encourage or support training in itself.

8 Conclusion

This paper has researched in what ways mobile computer technology, such as mobile phones or PDA:s, can aid memory. It has also shown that there is potential in such a device, but that there is not yet any one device that solves all problems. Some of the different approaches for such devices have been summarized as well as some of the challenges, problems and other things one needs to think about when designing an aid for memory. Since the paper does not focus on any single approach for solving memory issues, but rather on a collection of different approaches [10–15], the conclusions are quite general. They can thus be seen as aspects one must consider when designing a new aid for memory based on mobile computer technology.

The most important things to consider are listed as follows:

- The device should require little to no learning to be used.
- The compensatory information must be presented clearly.
- The device should not be introduced too late.
- The device should aim at prolonging the users ability to stay independent.
- One must consider other effects of age than only memory.
- Security and personal integrity issues must be considered.
- Stigmatization must be avoided.
- Further decline in the users cognitive abilities must be considered.

References

1. Boersma, F., Eefsting, J., van den Brink, W., Koeter, M., van Tilburg, W.: Prevalence of Dementia in a Rural Netherlands Population and the Influence of DSM-III-R and CAMDEX Criteria for the Prevalence of Mild and More Severe Forms. Journal of Clinical Epidemiology **51**(3) (1998) 189–197
2. European Communities: Health statistics key data on health 2002 (2002)
3. Fratiglioni, L., De Ronchi, D., Agüero-Torres, H.: Worldwide prevalence and incidence of dementia. Drugs & Aging **15**(5) (1999) 365–375
4. Robinson, R.: Dementia (2005) `http://www.healthatoz.com/healthatoz/Atoz/common/standard/transform.jsp?requestURI=/healthatoz/Atoz/ency/dementia.jsp`, accessed 2008-04-09.
5. Alzheimer Society of Canada: Alzheimer's disease: Progression of the disease, three stages (2005) `http://www.alzheimer.ca/english/disease/progression-3stages.htm`, accessed 2008-04-09.
6. Lee, M., Dey, A.: Providing good memory cues for people with episodic memory impairment. Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility (2007) 131–138
7. Hawkey, K., Inkpen, K., Rockwood, K., McAllister, M., Slonim, J.: Requirements Gathering with Alzheimer's Patients and Caregivers. Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility (2005) 142–149
8. Inglis, E., Szymkowiak, A., Gregor, P., Newell, A., Hine, N., Shah, P., Wilson, B., Evans, J.: Issues surrounding the user-centred development of a new interactive memory aid. Universal Access in the Information Society **2**(3) (2003) 226–234
9. Robertson, I.: Setting goals for cognitive rehabilitation. Current Opinion in Neurology **12**(6) (1999) 703–708
10. Loh, J., Schietecat, T., Kwok, T., Lindeboom, L., Joore, P.: Technology applied to address difficulties of Alzheimer patients and their partners. Proceedings of the conference on Dutch directions in HCI (2004)
11. Wu, M., Baecker, R., Richards, B.: Participatory design of an orientation aid for amnesics. Proceedings of the SIGCHI conference on Human factors in computing systems (2005) 511–520
12. Morris, M., Lundell, J., Dishman, E.: Catalyzing social interaction with ubiquitous computing: a needs assessment of elders coping with cognitive decline. Conference on Human Factors in Computing Systems (2004) 1151–1154

13. Szymkowiak, A., Morrison, K., Gregor, P., Shah, P., Evans, J., Wilson, B.: A memory aid with remote communication using distributed technology. Personal and Ubiquitous Computing **9**(1) (2005) 1–5
14. Wade, T., Troy, J.: Mobile phones as a new memory aid: a preliminary investigation using case studies. Brain Injury **15**(4) (2001) 305–320
15. Sellen, A., Fogg, A., Aitken, M., Hodges, S., Rother, C., Wood, K.: Do life-logging technologies support memory for the past?: an experimental study using sensecam. Proceedings of the SIGCHI conference on Human factors in computing systems (2007) 81–90
16. Leonard, V., Jacko, J., Pizzimenti, J.: An exploratory investigation of handheld computer interaction for older adults with visual impairments. Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility (2005) 12–19
17. Zajicek, M.: Interface design for older adults. Proceedings of the 2001 EC/NSF workshop on Universal accessibility of ubiquitous computing: providing for the elderly (2001) 60–65
18. Levy, R.: Aging-Associated Cognitive Decline. International Psychogeriatrics **6**(01) (2005) 63–68
19. Smith, C., Umberger, G., Manning, E., Slevin, J., Wekstein, D., Schmitt, F., Markesbery, W., Zhang, Z., Gerhardt, G., Kryscio, R., et al.: Critical decline in fine motor hand movements in human aging. Neurology **53**(7) (1999) 1458–1461
20. Wilson, R., Beckett, L., Barnes, L., Schneider, J., Bach, J., Evans, D., Bennett, D.: Individual differences in rates of change in cognitive abilities of older persons. Psychology and Aging **17**(2) (2002) 179–193
21. Kärkkäinen, L., Laarni, J.: Designing for small display screens. Proceedings of the second Nordic conference on Human-computer interaction (2002) 227–230
22. Hawthorn, D.: How universal is good design for older users? ACM SIGCAPH Computers and the Physically Handicapped (2003) 38–45
23. Hutchison, D., Eastman, C., Tirrito, T.: Designing User Interfaces for Older Adults. Educational Gerontology **23**(6) (1997) 497–513
24. Laurin, D., Verreault, R., Lindsay, J., MacPherson, K., Rockwood, K.: Physical Activity and Risk of Cognitive Impairment and Dementia in Elderly Persons. Archives of Neurology **58**(3) (2001) 498–504
25. Verghese, J., Lipton, R., Katz, M., Hall, C., Derby, C., Kuslansky, G., Ambrose, A., Sliwinski, M., Buschke, H.: Leisure Activities and the Risk of Dementia in the Elderly. The New England Journal of Medicine **348**(25) (2003) 2508–2516

Common errors in human-computer dialogues

Joakim Norberg

Department of Computing Science
Umeå University, Sweden
dit04jng@cs.umu.se

Abstract. Natural spoken language is very complex and dialogue systems between humans and computers are hard to design. This paper addresses some of the common errors that can occur in such dialogues. Dialogues can be handled very differently; some systems have limited amount of world knowledge and a finite set of questions and responses while other more advanced systems can be able to participate in very complex conversations and know when to speak and when to listen. The more advanced a dialogue system is, the more important is it to predict and prevent errors. This can be done through cue phrases, pattern recognition, visual input and by analyzing pitch, prosody and speed of the spoken language. If an error does occur, the system should try to repair and minimize the damage to prevent cyclic errors.

1 Introduction

Human-computer spoken dialogues describe the exchange of spoken input and output between a human and a computer system. The aim of this article is to examine some of the most common errors occurring during a human-computer spoken dialogue. What can be done to eliminate these errors to improve the quality of the dialogue and how do current speech systems deal with or work around them? There are several problems when a computer system has to recognize a user's input. The computer has to separate background noise from the user's words, correct hesitations and restarts, decide whose turn it is to speak and decide if silence from the user is just a pause, hesitation or if the intent is to let the computer speak.

In situations where one cannot use the hands and need auditive feedback, for example a situation where a doctor need to write a journal but cannot use a keyboard due to sterile environmental regulations and the doctor also might want some advice on setting an accurate diagnosis. Hence a good computer system is needed that can interpret the input, calculate tone, pitch, prosody etc. and search for cues so that an appropriate output could be given at the right time.

As another example automatic telephone services are used where one sometimes have to choose between tens of choices and memorize them in order to press the correct alternative on the phone buttons. Here a dialogue system is indeed much more convenient and is used today with varying quality.

It is a common understanding that the best way to design and build a human-computer dialogue system is to study the dialogues between humans [1–4]. The human mind is however much more advanced than any current computer system and adds emotions to dialogues, reads between the lines, understands irony, sarcasm and much more. By extracting cue phrases and visual cues, process the prosody, pitch and speed of the spoken, the system will become closer to a human actor.

The Wizard-Of-Oz method is commonly used when designing a dialogue system and is a method where a human is responding to users utterances through a synthesized voice, pretending to be a computer. This method is useful when one want to know how future users of the system might interact but it is difficult for a human to impose as a computer and give the same responses and understand which recognition problems that may arise in the real-world system [2].

2 Dynamics in spoken dialogues

McTear categorizes spoken dialogue systems into the following three classes [2]:

- **Finite state-based systems** that have a limited set of answers and can only handle a limited amount of input, i.e., they cannot interpret everything the users say. The dialogue is controlled by the system through a number of questions. If the input is ill-formed or unrecognized, a request for the same information is sent again.
- **Frame-based systems** are not as predetermined about what the dialogue will be about but depend on what the user of the system says. They can therefore interpret different responses from the user as long as it is in the domain of what the system knows.
- **Agent-based systems** are systems that can handle complex communication and are also called AI-systems. These systems can predict errors, expect what the user might say, correct ill-formed input, suggest topics etc.

2.1 Turn-taking and pauses

Most of the dialogue systems available today are unable to handle input while producing output, that is, listening while talking. Yankelovich et al. used spoken commands to interrupt their system from speaking when the user wanted to speak [5]. This might be a good way to handle the problem but it is a step away from natural spoken dialogues.

An interactive human-computer dialogue, like with human-human dialogue, should be mixed-initiative [4] where both system and user have control over the dialogue and turn-taking flow smoothly. Finite-state based systems on the other hand are fixed system-initiated [4, 2] since the system has control over the dialogue. Allen et al. [4] claimed that frame-based systems sometimes could be fixed user-initiated where the user provides the system with information until it has a sufficient amount. The authors developed a dialogue system to handle rather complex dialogues by considering its own goal while maintaining the current state in the dialogue.

The pauses that occur are most often a result of the system processing input [5]. These pauses might not be longer than pauses that occur in other non verbal dialogue systems but appear to be longer since the spoken dialogue is more natural and familiar to humans [5]. When a user of a computer dialogue system pauses, either to think or to process the input, the system has to know whether the user is pausing or waiting for the system to speak again. By using visual cues [6] as well as auditive cues from the user, the system has more information on which to make a decision. This will be discussed in more detail in section 3.1.

If no response is given from the user, which could happen if he or she does not know whose turn it is to speak, the system should tell what sort of input it is expecting [7].

2.2 Prosody, pitch and speed

Deaccenting (lowering the pitch of a phrase) is often associated with function words such as prepositions, pronouns and articles [8]. However, one cannot get much semantic information about an utterance by analyzing and associating the intonation of phrases but it is useful to gain structural information [8].

By analyzing the prosody of input, a system might know whether an utterance requires an answer or not. Questions often raise in pitch at the end of the sentence. Other ways to check if an utterance is a question is to look for cue words or phrases, for example ‘please’, ‘would’, ‘will’, ‘why’ etc.

The speed in which the input is presented is an essential factor in error correction. If a sentence is spoken rapidly, the words tend to blend into each other and are therefore harder system to separate and process. On the other hand, a sentence spoken too slowly can be as hard to interpret, if the system is not specifically trained on that.

3 Predicting errors

There are ways in which humans can predict communication errors and prevent them from occurring. Dybkjaer et al. claimed that error prevention is a better alternative than error correction emphasizing that users should be guided through the interaction with the system [7]. Errors will always occur and therefore “interaction guidance should be carefully evaluated” with users throughout the design process to prevent situations where errors may occur [7]. A system that controls the dialogue through questions is easier to develop because it can predict what the user might say, but limits the interaction possibilities [4].

Often the tone of a voice says a lot about the intention of the utterance is, so if a system could interpret the tone it could predict an error and prevent it from occurring.

Pattern recognition is very useful when predicting errors [9, 10, 1]. If a machine could predict the following sentence it is more likely that it is interpreted correctly. This holds for contexts where conversations are similar and the machine can expect the oncoming sequence of words or expect alternatives for it.

It is convenient to compare this to programming language development software where one can begin typing a word and the software fills in the rest of the word or provide one with a set of alternatives.

Fink et al. designed and implemented an prediction system which lowered the sentence error rate from 53 percent to 8. This prediction system also had the ability to ignore and add words to the input to lower the error rate. The authors claimed that sentences or utterances which could have different syntax had the same meaning if they resulted in the same feedback [1].

If a dialogue system logs the history of the dialogue it simplifies for an agent-based system to know what the user's intentions and goals are and therefore determine how the system should interpret the subsequent utterances [2]. By storing the dialogue history the system is also more likely to predict and prevent oncoming communication errors.

Context also affects an utterance and will give hints for the system on what the utterance was about. Consider the phrase 'It is cold here' which in a semantic way might just mean that the temperature is low but it might also, depending on the context of the utterance, mean that someone should close a window or raise the heat [2].

3.1 Visual input/output

In a human-human dialogue we often know whether the listener understands what we are saying by looking at the facial expressions. Through expressions and non-verbal utterances we can gain a lot of information about what the speaker is actually saying and meaning. Wang et al. developed a system that could detect communication errors by analyzing the visual input from the user when producing a reply [6]. By analyzing non-verbal head motion, pose and facial motion using 3D model-based tracking algorithms their system could detect errors during its conversational turn. An example is if the user twitched an eye it could signal an error in the conversation. The authors study involving this system showed that by using visual input their system performed significantly better compared to only audio input [6].

A user in a noisy environment makes it hard for a speech recognizing system to extract the user's spoken words from the background. An example of this is when a user of a GPS-system, that is capable of handling spoken dialogues, wants to listen to the radio when using the GPS. Aoki et al. addressed this problem in the article "Voice activity detection by lip shape tracking using EBGM" where the authors proposed and studied a lip shape tracking system to be able to know when the user speaks and to separate its utterances from background noise [11]. An infrared camera and complex matching algorithms were used to separate the lips of the user from its surrounding [11]. The experiments in their study proved that the tracking system was very efficient. However errors occurred when users uttered words that did not require lips to move much. A lip shape tracking systems are indeed useful when the user is positioned in a fixed place, i.e. car drivers (for GPS navigation using spoken dialogue [11]) or when placed in front of a computer, but will be hard to use in mobile situations.

For a human-computer dialogue to feel more natural it might be preferable to use some visual output. The system become more alive and due to these multimodal channels it might seem to have emotions if facial expressions are handled correctly. If a dialogue system uses an agent as visual output and the agent smiles when a user talks about something he or she likes, the conversation is going to be more relaxed and the user might be more error tolerant.

3.2 Cue phrases

Cue phrases (or cue words) are a set of words that give a hint of what will happen in the oncoming sentence or what the purpose of the sentence is. They signal dialogue structure in speech and can also convey semantic information about a sentence[12]. Litman et al. emphasized that classifying these cue words or phrases is critical when designing a dialogue system or any system that need to recognize and understand spoken input [12]. To extinguish, recognize and interpret these cue phrases is very useful when preventing communication errors and deciding turn-taking.

Hirschberg et al. made several empirical studies on the disambiguation of cue phrases [8]. A word or phrase can be ambiguous in an utterance and it is important in speech recognition when avoiding errors not to misunderstand the true meaning of it. The authors proposed prosodic classification models where the pitch of a phrase where combined with its position and therefore useful for disambiguate and predict cue phrases [8].

4 Error correction

Since errors in human-computer dialogues are unavoidable [13], it is important that a system know and understand when an error has occurred [14, 7], either from the users behavior or a misrecognition by the system, and deal with it according to the current situation.

A user does not know if the computer fully understood what was said if appropriate feedback is not given. How the feedback is presented is a very important part of a human-computer dialogue. If the feedback is not clear enough, the users' correction is even more difficult for the system to recognize.

Litman et al. wrote about cyclic errors, when correction of previous error lead to another error, in the article "Characterizing and predicting corrections in spoken dialogue systems" where they made an empirical study of a train information system [10]. When an error has occurred (a word or sentence is misunderstood or misinterpreted) users often tried to repeat it or make the utterance higher, louder and slower [10, 6] whereas the system might not be trained on that sort of hyper articulation [10].

Bulykoa et al. proposed a solution to this cyclic error problem where they made a threshold on how many of these error cycles that where allowed before the system started the dialogue over [15]. The authors also pointed out that if the system rephrased instead of repeated when users tries to correct, these cyclic errors can be avoided.

Dybkjaer et al. [7] suggested that if a dialogue system fails to recognize a users input it should ask for the utterance to be repeated. Depending on if the input was too low, too high, too fast, too slow, it should ask the user to modify the utterance respectively. The authors called this system-initiated repair meta-communication and means that cyclic error is prevented if the system tries to repair the error before the user does. However, a later study made by Skantze shows that if a system fails to recognize a user's input it should not signal for misrecognition or ask for repetition, but should ask task-related questions. This in order to better understand the subsequent utterances and were proven to be efficient by the authors empirical study [16]. One must however point out that the later study only used human to human dialogues as a model whereas Dybkjaer et al. [7] used a human-computer dialogue to implicate these error recovery strategies.

In an empirical study by Levow they analyzed error correction made by users and tried to identify them in order to recognize them and suggested a method to handle these error correction interactions [13]. The results from the authors study shows that users shift style when correcting an error and that users' correction utterances tend to be longer with longer pauses and a small decrease in pitch. The decrease in pitch was significant when users tried to correct a misrecognition by the system but not when an utterance was rejected by it [13]. The authors thereby developed different decision trees to classify and separate original input from error corrections. The best results were from the decision trees that measured all attributes; duration, amplitude and pitch that were "depending most heavily on durational features" [13].

Interpretation of the users feedback on a systems output is also important for a dialogue system to be able to correct a previous error. Gustafson et al. conducted a Wizard-of-Oz experiment where they analyzed users' utterances to see if their feedback might give clues if the previous statement was understood or not, if it was accepted with gratitude etc. [17]. Contextual and prosodic cues where also considered when the authors interpreted the input as positive or negative. If these cues where interpreted correct the authors meant that serious errors could be avoided in the subsequent dialogue.

4.1 Speech-to-text error correction

According to Karat et al. [9] there are four major classes of possible errors in continuous speech-to-text recognition:

- **Simple misrecognition** in which a single spoken word intended as text is recognized as a different text word.
- **Multi-word misrecognition** in which a series of words are recognized as a different series of words.
- **Command misrecognition** in which an utterance intended as a command is inserted in the text.

– **Dictation as command misrecognition** in which an utterance intended as dictation is taken as a command.

Suhm et al. presented a solution for problems related to speech-to-text error correction by using a multimodal error correction model where users can switch between modalities [18]. For example, if one first speak and then correct misrecognitions or other errors with a supplemental modality. Their user study showed that correction with another modality, like a keyboard or a pen, was faster than repeating verbally. This method is especially useful on small mobile devices with small keyboards and users with poor typing skills [18].

Some speech recognizing systems allow the user first to dictate and afterwards switch to a correction mode, i.e. IBM's VoiceType [9]. Dragon Natural Speaking on the other hand uses so called inline correction where the user through specific spoken commands can alter the dictated text as they go along [9]. The results from the empirical study by Karat et al. where participants used these systems showed that "subjects made many more corrections inline than they did after completion of entry" [9].

The spoken commands, like UNDO, SCRATCH THAT and DELETE, used to correct errors when speaking continuously to a computer in any of the previously mentioned speech-to-text programs should according to the authors in reference [9] be more developed and adapted to natural speech if users are to prefer speech-to-text before keyboard and mouse.

Is speech to text a better alternative for computer inexperienced people than keyboard and mouse? Karat et al. discussed this and drew the conclusion that the methods for error correction are yet too complex and would not be tolerated by these users so keyboard plus mouse is still the easiest way to produce text [9].

5 Discussion

The future dialogue systems seem to diverge from the currently most common finite state-based systems where one can only choose between a finite number of spoken alternatives. Due to rising computational capacity they will probably be more complex and have a higher level of intelligence. We are seeing and will probably see a higher need for alternative input methods since mobility and speed are becoming a demand from the consumers.

Many costumer-service companies may save a lot of money if complex dialogue systems to handle errands from customers were used instead of human personnel. The time spent on line when waiting for personnel to help will be reduced to zero. Therefore customers will probably have a more positive attitude towards the company in question.

Some people might find it difficult or intimidating to speak to a computer system and act differently than with a human employee. Gustafson et al. noticed some interesting differences when people in their test group spoke to a computer system [19]. Men often socialized at first and then began to search for information whereas women started with fact-oriented questions and asked about the system's domain. This is interesting because when designing a dialogue system one have to be aware of that people might not only search for answers in the given domain but also try to socialize and ask questions about the system itself.

Computers are used for educational purposes in many schools today and provide additional support for teachers. Himmelman discussed and studied the benefits of a tutoring spoken dialogue system [20]. Their results showed that students using the dialogue system had a learning gain that was between students who only used material in the classroom and students who where tutored by a human teacher, the later with highest learning gain. One must however point out that there are several benefits from using a tutoring dialogue system; human teachers does not have to be present and it may trigger the student to curiously search for information held by the system.

Litman et al. also evaluated a tutoring dialogue system and emphasized that most important when designing a tutoring dialogue system is for it to understand the semantics of an utterance rather than just transcribe it. The authors next step was to develop the tutoring system to predict and adapt to users emotions [21]. A comparison between spoken and typed human and computer tutoring was made in 2006 by Litman et al. [22]. Their results showed that when changing from text to speech in human-human tutoring the learning gain was higher but did not make much difference when going from text interaction to speech in human-computer tutoring [22].

6 Summary and Conclusions

In this paper we have studied the most common errors occurring during a human-computer dialogue and their respective methods to handle them. A good dialogue system should be able to give appropriate feedback on users input and through analysis of cue phrases and words or even visual cues be capable of running a complex dialogue.

Error prevention is a better option than repairing them when damage is already done. If an error does occur we have discussed error-handling methods proposed by researchers that minimizes the risk of making cyclic errors and sustains the dialogue on a conversational level.

A human-to-human dialogue is extremely complex and usually contains several errors, interruptions, ambiguity, sounds, non-verbal expressions, pauses and restarts but is most often understood by both parts. It is however, as we have discussed in this article, the best model when designing human-computer dialogue systems.

References

1. Fink, P.K., Biermann, A.W.: The correction of ill-formed input using history-based expectation with applications to speech understanding. Comput. Linguist. **12**(1) (1986) 13–36

2. McTear, M.F.: Spoken dialogue technology: Enabling the Conversational User Interface. ACM Comput. Surv. **34**(1) (2002) 90–169
3. Doran, C., Aberdeen, J., Damianos, L., Hirschman, L.: Comparing several aspects of human-computer and human-human dialogues. In: Proceedings of the Second SIGdial Workshop on Discourse and Dialogue, Morristown, NJ, USA, Association for Computational Linguistics (2001) 1–10
4. Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A.: Toward conversational human-computer interaction. AI Magazine **22**(4) (2001) 27–38
5. Yankelovich, N., Levow, G.A., Marx, M.: Designing SpeechActs: issues in speech user interfaces. In: CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1995) 369–376
6. Wang, S.B., Demirdjian, D., Darrell, T.: Detecting communication errors from visual cues during the system's conversational turn. In: ICMI '07: Proceedings of the 9th international conference on Multimodal interfaces, New York, NY, USA, ACM (2007) 323–326
7. Dybkjaer, L., Bernsen, N.O.: Usability evaluation in spoken language dialogue systems. In: Proceedings of the workshop on Evaluation for Language and Dialogue Systems, Morristown, NJ, USA, Association for Computational Linguistics (2001) 1–10
8. Hirschberg, J., Litman, D.: Empirical studies on the disambiguation of cue phrases. Comput. Linguist. **19**(3) (1993) 501–530
9. Karat, C.M., Halverson, C., Horn, D., Karat, J.: Patterns of entry and correction in large vocabulary continuous speech recognition systems. In: CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (1999) 568–575
10. Litman, D., Hirschberg, J., Swerts, M.: Characterizing and Predicting Corrections in Spoken Dialogue Systems. Comput. Linguist. **32**(3) (2006) 417–438
11. Aoki, M., Masuda, K., Matsuda, H., Takiguchi, T., Ariki, Y.: Voice activity detection by lip shape tracking using EBGM. In: MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia, New York, NY, USA, ACM (2007) 561–564
12. Litman, D.: Cue Phrase Classification Using Machine Learning. Arxiv preprint cs.AI/9609102 (1996)
13. Levow, G.A.: Characterizing and recognizing spoken corrections in human-computer dialogue. In: Proceedings of the 17th international conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1998) 736–742
14. Cawsey, A., Raudaskoski, P.: Repair work in human-computer dialogue. In: Proceedings of the 13th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1990) 327–329
15. Bulykoa, K. Kirchhoffa, M.O., Goldberg, J.: Error-correction detection and response generation in a spoken dialogue system. Speech Communication **45**(3) (2005) 271–288
16. Skantze, G.: Exploring human error recovery strategies: Implications for spoken dialogue systems. Speech Communication **45**(3) (2005) 325–341
17. Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granström, B., House, D., Wirén, M.: AdApt-a Multimodal Conversational Dialogue System in an Apartment Domain. In: Sixth International Conference on Spoken Language Processing, ISCA (2000)

18. Suhm, B., Myers, B., Waibel, A.: Multimodal error correction for speech user interfaces. ACM Trans. Comput.-Hum. Interact. **8**(1) (2001) 60–98
19. Gustafson, J., Lindberg, N., Lundeberg, M.: The August Spoken Dialogue System. In: Sixth European Conference on Speech Communication and Technology, ISCA (1999)
20. Himmelman, D.S.: Natural-language processing for computer-supported instruction. Intelligence **12**(4) (2001) 22–32
21. Litman, D., Silliman, S.: ITSPOKE: An intelligent tutoring spoken dialogue system. Proceedings of the 4th Meeting of HLT/NAACL (2004)
22. Litman, D., Rosé, C., Forbes-Riley, K., VanLehn, K., Bhembe, D., Silliman, S.: Spoken Versus Typed Human and Computer Dialogue Tutoring. International Journal of Artificial Intelligence in Education **16**(2) (2006) 145–170

Increasing performance and reducing the visual information overload when using a computer mouse with the help of vibrotactile feedback

Linus Forsberg

Department of Computing Science
Umeå University, Sweden
dit04lfg@cs.umu.se

Abstract. This paper investigates whether adding a new vibrotactile channel for information will improve the user performance in a game-like environment. Many computer games give a lot of visual feedback, to such an extent that they can overload the visual channel of the user. Adding a new vibrotactile feedback channel might make it easier for the users to interpret the information and because of that improve their performance while using the game. A physical vibrotactile actuator has been made and tested on several test subjects with a test application specifically made for this study. The results does not prove the hypothesis, because of the test setup no final conclusion could be drawn.

1 Introduction

1.1 Background

Many computer games today provide their users with tons of both visual and auditive feedback and information. Many of these demands that the user has to take in parallel information, compute it, and then as fast as possible perform a suitable action. In many of todays graphical user interfaces the visual channel is overloaded[1]. This remains as a problem even though developers try to relieve the pressure by also using the auditive channel. Many of the newer games of today also use this auditive channel even more by implementing voice over IP, which in many cases is a must. That leads to the fact that not only is the user given visual and auditive feedback from the game, the user also has to take in linguistic information from other users. However, using visual and auditive feedback is not the only type of channel that could be used for human-computer-interaction(HCI). A haptic channel is also a possibility and, if implemented correctly, can help the users[2].

Haptic feedback is in this area a general term for all types of feedback that you can feel[3]. There are several different types of haptic feedback used in HCI but one that will be mentioned a lot in this study is the tactile one. The tactile feedback is a term for the type of physical feedback that gives a sensation of pressure[3]. The receptors that sense the tactile feedback lies close to the surface of the skin[4]. One type of tactile feedback is the vibrotactile feedback, where the physical actuators are vibrating. The advantage of using vibrotactile feedback in HCI is that it is cheap and easy to implement into systems[5]. Another type of haptic feedback is force feedback. Force feedback provides the user with the feeling of motion, with the help of mechanical solutions [3]. Force feedback can stop the motion of a user by applying force[4].

Lindeman et al also claims that interface designers can make a better use of the capacity of the human information bandwith when using the visual, audiative and physical information channel[5].

1.2 Previous studies

Previous studies has shown that using haptic feedback to reduce visual distraction, indeed decreases the error rate for certain actions in a GUI[3]. This study also showed that the subjective workload as perceived by the users for certain tasks were significantly reduced with the help of haptic feedback.

Although many studies have shown that haptic feedback has increased the performance in many ways, many of them have one thing in common. The actions performed in the experiments has a strong conceptual connection to real world actions. Oakly et. al. made use of haptic devices to simulate gravity in an attempt to improve the performance while doing a targeting test. The error rate for the test was significantly reduced when using some types of haptic feedback[3]. Another study tested whether or not stearing a mouse pointer through a tunnel would be made more effective if hitting a tunnel wall would give a physical feedback. The haptic feedback increased the steering speed of the test subjects in their experiment by 52 percent[6]. In a third study emulating the physical feeling of touching a button in the real world, gave the user vibrotactile feedback while hovering over the button[7]. This design technique for vibrotactile actuators seems to be effective, and Lindeman et al says that designers should take advantage of natural mappings if it is possible[8].

But this is not the only technique for using vibrotactile feedback. Lindeman et al used vibrotactile cueing for increasing the spatial awareness of soldiers clearing a building in a military exercise. Thanks to vibrotactile actuators placed at the torso the time for being exposed to danger, and the amount of spaces cleared was significantly decreased[5]. This result is also very relevant to this study since the actual experiment done by Lindeman et al was performed in a game-like environment.

The haptic devices used in these studies varies from actuators made by the authors to commercial products. The Phantom is a commercial haptic device used in studies mentioned in this article[6] [3]. It is designed as a pen attached to a mechanical arm[4]. The haptic feedback that the Phantom provides is force feedback. The Phantom is used in a range of different areas such as vehicle construction and art[4].

1.3 This article

This article will investigate if introducing a new information channel will improve the users ability to process information in a more efficient way. The information channel will be a vibrotactile actuator attached to the mouse. The purpose of the new channel, is to relieve the users visual channel by transferring some of the feedback through the vibrotactile actuator instead. Thus, the hypothesis is that adding this new tactile information channel will increase the efficiency of a user operating in an environment where the amount of information transfered through the visual channel is very high. The application used in the experiment will use two types of vibrotactile feedback designs. One is to return vibrotactile feedback when hovering over some vital buttons in an attempt to simulate the physical feeling of moving your finger over a button. The other type of situation where vibrotactile feedback will used is to make the user aware of new functions that has become available. An analysis of the results of the experiment will be used to see how making use of vibrotactile feedback will affect the performance.

2 Experiment

2.1 Overview

The experiment consisted of a test in a game-like environment. The application used was made specifically for this study. The tests in the application demanded the user to be aware of several different sources of visual information and quickly make decisions based on the information provided. There were two different versions of the application. The first one only provided the test subjects with visual information. The second one gave the test subjects some information through the vibrotactile actuator. The application rewards the test subjects with points. The points are given after every successful action. If the test subjects fails to perform these actions in a fast manner, the test will be over.

2.2 Software

The application was made in Adobe Flash CS3 Professional. As seen in Figure 1 the GUI is divided into three sections. Section 1 in Figure 1 consists of several bars. These bars randomly decreases, independently from each other, in size during the test. The user has to click on these bars to increase their size, because if one of the bars goes down to its minimum value, the test will be over. By successfully clicking on a bar and increasing it the test subjects are rewarded with performance points. The bars may also, at random, get one out of three markings. These markings prevents the user from increasing the size of the bar. To remove a marking from a bar, the users has click on the button, corresponding the marking at Section 3 in 1, and after that click on the bar. The markings cannot be removed constantly. Each button for removing markings in Section 3 in Figure 1 has a timer that is initiated after it has been used. The timer finishes after a certain amount of seconds, but as long as it is active, the button will not

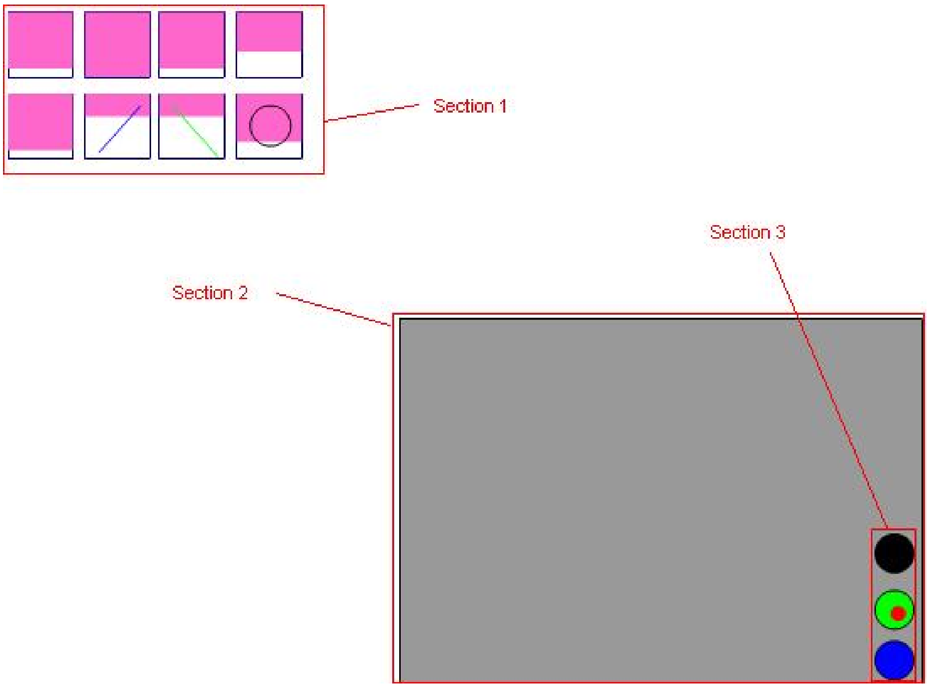

Fig. 1. A screenshot of the test application. The black text and the boxes have been added for demonstration purposes.

be usable. To visualize that a button is inactive, a red dot is drawn above it. The purpose of the middle section shown at Section 2 in Figure 1 is to present distracting visual feedback.

The vibrotactile version of the test application returns a vibrotactile feedback in two different situations. One is when the user is hovering over the remove marking buttons. This is to take advantage of the natural mapping and simulate the physical feel of touching a button. The other situation where the vibrotactile feedback is given is when a remove marking button goes from the inactive to the active state. The reason for this is to increase the spatial awareness of the user.

2.3 Physical devices

The vibrotactile actuator is a small membrane from a speaker. The speaker was disassembled, and all parts not needed for the membrane to vibrate was removed to decrease the size of it, as seen in Figure 2. The actuator was attached to an ordinary USB mouse and connected to the audio output channel of a sound card. To activate a vibration in the actuator a sound signal was sent to the computers audio output. Earplugs were used to prevent the test subjects from hearing the sound the vibrotactile actuator emitted, so they could only feel the vibration.

Fig. 2. A figure showing two pictures of the vibrotactile actuator used for this experiment.

2.4 Vibrotactile signals

Since the vibrotactile actuator was not a motor but a speaker the use of sound signals was a must. Several different sound signals were recorded and analyzed. The types of sound signal that proved to be the easiest to feel were the ones that resembled the sound of a vibrating motor lying against a hard surface. The two best signals were used in the test application.

2.5 Test subjects

Ten test subjects were used in the experiment. All of them university students. The ages of the test subjects were between 22 and 28. Three of them were women and the rest were males. The test subjects were divided into two groups. One group for every version of the test application.

2.6 Design

The test subjects were all tested individually. A brief introduction to the experiment was given. The introduction was the same for all test subjects but what held individually for each test subject prior to each test. They all had to do three training runs to understand the mechanics of the test application. The training was done with their corresponding version of the test application. For the proper test, all test subjects had to use the test application five times. The points were noted after every test and a mean value for points were calculated for all test subjects. The tests where done in a calm environment where only the test subject and the supervisor were present. After all of the data was collected a statistical analysis was made in MINITab.

2.7 Task

The task was to prevent the bars, as seen at Section 2 in Figure 1, from going down to their minimum size.

3 Results

Table 1 shows the results from the tests. The first column shows the group number while the second column shows the mean value of the performance points for every test subject.

Table 1. A table showing the results from the experiment.

Group	Result
1	56
1	53
1	43
1	37
1	45
2	58
2	54
2	56
2	45
2	46

The mean value for the group that only got visual feedback is 46.8. The standard deviation for that group is 7.694. The mean value for the group that also got vibrotactile feedback is 51.8 and with a standard deviation of 5.933.

The results from the one-way ANOVA test can be seen in Figure 3.

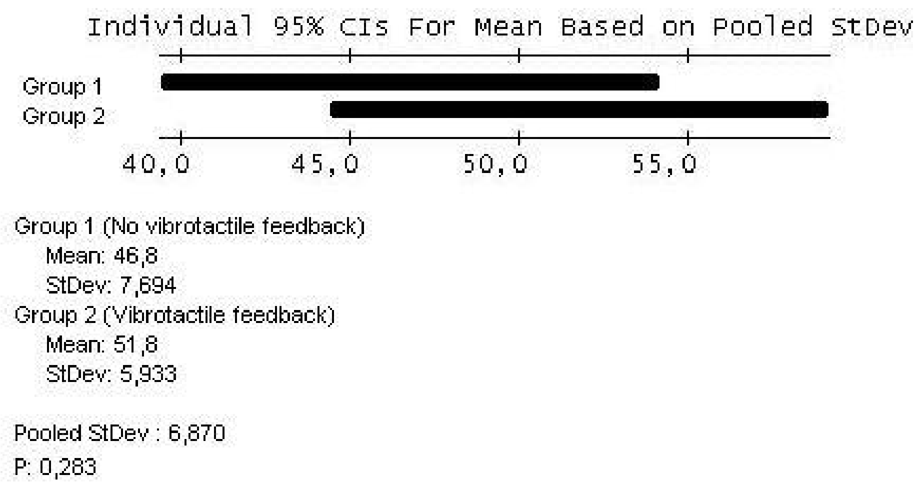

Fig. 3. The results from the one-way ANOVA test done in MINITab.

4 Discussion

4.1 Result data

According to the results there is a difference in the mean value for the two groups. The group that got vibrotactile feedback in addition to the visual feedback has a higher mean value of the performance points. This can be seen as an indication that there the performance increases while getting vibrotactile feedback. But as seen in Figure 3 the one-way ANOVA the results does not show any significant difference in performance between the two test groups because their confidence intervals (CI) are overlapping. There could be several reasons for why this is the case. First and for most, the amount of test subjects was too low to do any proper statistical test. The reason for this is that there was simply impossible to find, and test the amount of participants that was actually needed within the given time frame.

4.2 Software design

Another issue that might have affected the results is the design of the test application. The design was based on discussions with experienced users of computer games, and in which situations they felt a loss of performance due to information overload in the visual channel. But, some test subjects from the group that had vibrotactile feedback, commented that the task was so difficult to perform that they did not feel that the vibrotactile feedback really helped. Another test subject, from the group that only had visual feedback, said that the waiting for feedback from Section 3 in Figure 1 was not his main concern while performing the task.

Some test subjects from both groups also had concerns about the randomness of the decrease of the bars at Section 2 in Figure 1. They felt that they had been unlucky and had gotten a rapid decrease of some bars, combined with unfortunate deployment of markings and had because of that failed earlier than they should have. This happened once in a very early state of the test thus reducing the overall performance for that test subject. This was not an unknown issue and it is mainly because of that, every test subject had to take the test five times and a mean was calculated. The reason for this was to reduce the impact unfortunate incidents on the final results.

4.3 Physical devices

The decision to only use one vibrotactile actuator for both the spatial awareness feedback and the naturally mapped feedback might have been a mistake. Because if a test subject hovered over a button which gave feedback, at the same time as the application returned feedback to indicate that a button has gone from inactive to active, the test subject might have not noticed that two different vibrotactile feedbacks were actually given. An issue with the vibrotactile actuator was that, although its vibrations could be felt, they were quite weak.

Some test subjects also complained about ergonomic issues concerning the size and placement of the vibrotactile actuator. They felt that the vibrotactile actuator prevented them from holding the mouse in a way that they were used to. However, this should not have affected on the final results since the placement of the vibrotactile actuator was the same for all test subjects.

4.4 Future work

For future work, a more in depth study of how to make a test application with a high demand on the visual channel, should be made. It is also important to make a test application in such a way that the high demand on the visual channel is the most important aspect of the application. Another solution would be to implement it in an already existing game application, testing it on some users with prior experience of the game, and after that conducting interviews to investigate whether or not they perceived that their performance increased. Also, the use of more than one vibrotactile actuator would be a great addition so that the test subjects easier can identify the type of vibrotactile feedback that is actually given. Using distinct vibrotactile feedback for each of the buttons when indicating that a button has gone from inactive to active might also increase the performance of the test subjects. This might be the case because then the test subjects would not have to visually identify which button that is active before targeting it with the mouse pointer. Replacing the current vibrotactile actuator with one driven by motors should also be considered. The reason for this would be to ensure that the test subjects does not actually hear the signal that the actuator emits.

5 Conclusions

Even though the mean value of the performance points is higher for the group that got vibrotactile feedback in addition to the visual feedback the results from 3 does not prove that the use of vibrotactile feedback in an environment with a lot of visual feedback improves the performance of the users.

References

1. W.Lindeman, R., Yasuyuki, Y., L. Silbert, J., Lavine, R.: Effective vibrotactile cueing in a visual search task. In: INTERACT'03. (2003) 89–96
2. E. MacLean, K.: Designing with haptic feedback. In: Symposium on Haptic Feedback in the Proceedings of IEEE Robotics and Automation, Palo Alto, USA (2000)
3. Oakley, I., McGee, M.R., Brewster, S., Gray, P.: Putting the feel in look and feel. In: CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2000) 415–422
4. D.Laycock, S., N.Day, A.: Recent developments and applications of haptic devices. Computer Graphics Forum **22** (2003) 117–132

5. W.Lindeman, R., L. Silbert, J., Mendez-Mendez, E., Patil, S., Phifer, D.: Effectiveness of directional vibrotactile cuing on a building-clearing task. In: Procedings of ACM CHI 2005, Washington, DC, USA (2005)
6. Dennerlein, J.T., Martin, D.B., Hasser, C.: Force-feedback improves performance for steering and combined steering-targeting tasks. In: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2000) 423–429
7. Akamatsu, M., MacKenzie, S.: Movement characteristics using a mouse with tactile and force feedback. International Journal of Human-Computer Studies **45** (1996) 483–493
8. W.Lindeman, R., Page, R., L. Silbert, J., N. Templeman, J.: Using vibrotactile cues for virtual contact and data display in tandem. In: Proceedings of 11th Int'l Conferance on Human-Computer Interaction, Washington, DC, USA (2005)

Mobile Music Fitness Coach: Interaction techniques and guidelines

Vivian Esquivias

Department of Computing Science
Umeå University, Sweden
dit03mes@cs.umu.se

Abstract. A mobile music fitness coach is the result of the combination between a mobile music player and physical performance. These devices aims to help and encourage its users with their workout. Mobile devices have other requirements than stationary devices and the problem today is that the development of interfaces and interaction techniques for mobile devices has not reached the same extent as the development for stationary devices. This article introduces important aspects within the mobile music fitness coach area and discusses devices existing today with focus on their interface and interaction techniques. Finally, based on the subjects discussed, a suggestion of guidelines are presented and used for evaluation.

1 Introduction

Music, fitness and mobile devices are things that are important in most peoples lives. Most people around us are proud owners of mp3 players or mobile phones with incorporated music player. These devices are often used when exercising since this is an readily available and quite personal motivator. The fact that people use music when performing physical activities is not a secret, everyone has at some point felt the strange but positive effect that music has when exercising and likes it. The result of the combination between the mobile device, the music player and exercise is the Mobile Music Fitness Coach (MMFC). This paper explores different MMFC devices, some that already exist on the market (Sony S2 sports [1], Nike+iPod sports kit [2] and Adidas polar project fusion [3]) and some that are still in progress (PersonalSoundtrack [4], lifetrak [5] and MPtrain [6]).

The focal point of this paper is to explore how these devices interfaces are designed as well as the different interaction techniques that are used and with this information discuss the important aspects that a designer needs to consider when developing mobile music devices for people on the move.

This paper starts with *Background* where the area of MMFC and the factors that make this area interesting are introduced. The next section *Mobile Music Fitness Coach* describes the main characteristics of a MMFC and emphasize devices (existing and in progress) within this area. The following section *Interaction techniques for mobile devices* introduces interaction techniques used today and also alternative techniques. The section *Discussion* discusses the devices and techniques introduced and suggest several guidelines. These guidelines are finally used to evaluate the devices. *Conclusion*, this section concludes the paper.

2 Background

The combination of fitness, mobile devices and music is a natural development since each of these areas and devices within them are very popular in the world today. Fitness is a well discussed subject, people are engaged to this for different reasons: shape, wellness, devotion, competition etc. No matter the sport or form of exercise they are often performed accompanied with music. There are several reasons for the linkage between music and physical performance, music for example encourages, motivates and alleviates the exhaustion. Most people listens to music everywhere and it is the cheapest and most available motivator. The following are introductions to the different areas that are exploited in a MMFC.

The benefits of physical performance: Many studies have been done to find evidence of the positive effect that physical performance has on health and the prevention of serious health problems [7]. The results have shown very strong evidence for this effect, which means that a person that exercises regularly, will develop protection against potential health threats and thereby will also have a chance of a longer live. According to Biehl [8], physical performance also affects a persons well-being and quality of life in a positive way. For example, a healthy body results often in a healthy mind which can encourage a person to accomplish his/her goals in life and feel happy about it. These are some of the reasons that fitness and exercising are so popular and important for many people.

The effect of music on physical performance: Many researches and surveys have been done within the subject and even though the results are not always consistent many of them show that there is a connection between music and physical performance [9–11]. Some researchers state that the music is used as a way of distraction and that helps the performer to ignore or lose focus on the pain and exhaustion [9, 11], other state that the different features in music such as rhythm, beat and tempo, when used properly, can affect the performers physiology (heart rate, blood pressure etc.) and psychology(self-esteem, motivation, arousal etc.) [9, 11].

The expansion of mobile devices: The growth of mobile devices on the market the past few years has not passed by unnoticed, according to Bryant [12] "the overall portable audio market is expected to take off from 81.7 million units in 2006 to 185.4 million in 2011." The growth of mobile devices indicates that these devices are very popular and becoming part of peoples lives. With this growth come also the challenges that a designer must face when designing interfaces for these mobile devices, such as screen size and the limitation with input and output capacities [13]. To acknowledge this need is one step closer to creating devices with good and satisfying design, the next step is to explore new ways of interaction and even to create guidelines for future designers, which is the main objective with this paper.

3 Mobile Music Fitness Coach

The idea of developing a MMFC has been discovered by many researchers and also well known manufactures. In this section three existing devices and three devices in progress will be introduced. A MMFC is characterized by being a mobile music player with extra features that motivates, helps and encourages its users with their training. Some examples of these are: pulse monitor[1], calorie counter[2], pedometer[3], automatic generated playlist[4] etc. In some devices the music features (sound, beat, rhythm etc.) are combined with the players features in order to enhance the experience for the user. Other devices are made context aware in order to learn the users habits and surroundings and make the usage of the device more automatized.

3.1 Devices on the market

There exist some devices that classifies as MMFC in the market today, in this section three of them will be introduced.

Sony S2 Sports Walkman, NW-S205F (2006): This is a water resistant mp3 player, designed for active users that enjoy music while running or doing other physical activities. The features this device provides its users are calorie counter, pedometer, stopwatch and FM tuner. While on the move the user can choose to turn on the *Music Pacer* that selects the music based on the users pace or the *Shuffle Shake* that is accessed by shaking the player three times and shuffles songs from a playlist [1]. The player has a cylindrical design and can be placed around the users arm with the included straps.

Nike+iPod Sports Kit (2006): To use the result of this cooperation, the user needs a pair of Nike+ shoes, an iPod Nano and the Sports Kit that engage the shoes and the iPod. The kit consists of a sensor that uses an accelerometer and then wirelessly transfers the obtained data to a receiver connected to the iPod. On the iPod display the users can see their time, distance covered and calories burned. As a special feature the user can program a *PowerSong*[5] that can be played anytime by a click. Through the Nike+iPod menu the user can choose between different workouts. The system keeps the user updated by voice feedback, this feedback can be features like the time, distance, pace or calories burned but it can also be congratulations when the user reaches a personal best. After the workout the user can transfer the data stored in the iPod by

[1] Measures the performers heart rate
[2] Measures the performers consumed energy
[3] Counts each step a person takes by detecting motion of the hips
[4] Playlist that selects music based on factors like heart rate or pace
[5] A particular song selected by the user as a motivational song

Fig. 1. Sketch of the Sony S2 Sports Walkman.

synchronizing it to nikeplus.com. At this website the user can see previous results or compete with other Nike+iPod Sports kit users [2].

Adidas Polar Project Fusion (2005): This system is developed and designed for expert and elite runners. A part of the system has textile heart rate sensors incorporated into the fabric, these sensors monitors the users ECG[6] and sends the data gathered to the systems main computer, this data is also presented to the user as visual feedback. A sensor in the special designed shoes, measures the users moves and sends also the data to the main computer. This main computer is the brain in this system, it coordinates the data gathered from the subsystems and provides its users constantly with the latest information about the changes in their body. Apart from that, this computer offers features such as preparation and planning before the run, a fitness test and a guide to optimize the users training. After the run the user can synchronize the gathered data on a PC using the systems software [3].

3.2 Devices under development

This section will introduce three systems in progress within the MMFC area.

PersonalSoudtrack (2006): This is a mobile music player that consists of a context-aware playlist. It selects music in real-time from the users own music library, based on the users pace. According to Elliott and Tomlinson [4] by using this playlist the music player becomes more personal since it adjusts the music to its users movements. Playing an upbeat song can increase the users tempo and in the same way by playing a slower song the users tempo can decrease. The user can skip a song at anytime and by doing this the music device can learn the preferences of music at a specific situation.

Lifetrak (2006): This is a context-aware mobile music device that selects music based on the users location, current time, pace and information about the current environment (weather, traffic, amount of sound etc.). By considering all these factors the device maximizes its probability of selecting the most precise song for the user in a certain situation. A very important aspect in this system

[6] Electrocardiogram, shows the electrical activity of the heart over a time span

is the feedback to the user. When the user feels that the music played is not accurate for the moment it can be changed by using one out of two options. The first option consists of a graphical interface where each contextual factor (weather, traffic etc.) can be adjusted. The second option consists of two simple icons *love it* and *hate it*. These options allow the user to operate with the interface in an easy way.

MPTrain (2006): This is a system incorporated in a mobile phone that offers its users assistance with their training by monitoring their physiology and encouraging the user with accurate selected music. The users decide the pace they want to keep to reach their goals. The system consists of wireless sensors that keeps the system updated with the users physical condition (heart rate and speed). It also uses an algorithm from which the system can learn to combine the right musical features with the users exercise level and physical condition [6]. Based on these factors the system selects the matching song for the user. Before the user goes out for a run he/she specifies the desired rate and effort level on the mobile phone, the interface for this is simple and consists of input boxes. While running the user can see a line graph on the screen, that can be used to compare the actual performance with the one selected before.

To summarize, one thing that the devices in progress have in common is that they are becoming more context aware, they take advantage of factors as heart rate to generate playlists. These automatic generated playlists are used in order to let the user concentrate more on the task than on the player. The devices on the market have made an attempt to minimize the attention costs by offering audio feedback and features that can be selected without any visual attention. These devices also uses websites as additional motivators and with these, the users can keep track on their achievements.

4 Interaction techniques for mobile devices

The development of interaction techniques and guidelines for stationary computers and systems has come a long way but is still an important research area. The research has not reached the same extent when it comes to mobile devices, and especially not for mobile devices used in physical performance, but is nonetheless developing in pace with present technology. The limited research that exists concerning interaction techniques for mobile devices deals with exploring new ways of interaction, where the users limited attention is taken into account as well as the fact that he/she is on the move and needs easy access to the device without affecting their main task [14, 15, 13]. This section introduces some of the techniques used today (based on devices mentioned earlier) and also some new ways of interacting.

4.1 Interaction techniques today

This section will discuss the different interaction techniques used today, this will be done based on the examples mentioned earlier (Sony S2 sports, Nike+iPod and adidas-polar fusion).

The Sony S2 sports player is controlled by using its buttons that are placed on the front of the cylindrical player, this player has also a wheel on one side that can be turned in both directions and also pushed in and pulled out. On the same side as the buttons the player has a small display where for example the current played song can be shown. The use of display in this system is optional, hence the small size (see figure 1).

The nike+ipod system is controlled through the iPod nano, using a display screen and a "click wheel". From this wheel the user can access the menu to reach the player's features, but there are also short cuts on the wheel to access the basic functions of a music player, such as next-previous track and play/pause. This system gives both visual and voice feedback. The display on this system is important in order to control the player, the voice feedback keeps the users updated and reduces the demand of visual attention.

The adidas polar system uses the watch screen for visual feedback and the buttons on the watch to control the system or change the information displayed. This system is visually demanding for the user since it does not provide any audio feedback.

The devices in progress are not discussed here since they have not reached their final design yet. To summarize, most of the interaction techniques today on mobile devices consists of pressing buttons of different kind and displaying information on a screen. Some of this systems offer their users voice or sound feedback in order to reduce the demand of visual attention.

4.2 Alternative interaction techniques

As an attempt to improve the mobile and wearable devices in the world today, Brewster et al. [13] emphasizes alternative interaction techniques. This section will introduce two ways: gestural and 3D audio interactions.

One interaction technique that liberates the users from using their hands when they interact with their device is interaction by 3D head gestures. To test this Brewster et al. [13] introduced the 3D audio radial pie menus, where the idea behind is to keep the user in the middle and the menus around the users head. Each piece of the pie representing a menu option. The user becomes aware of the menus from sounds or speech around him/her and by nodding in the direction of the sounds the user can select an option from the menu. Figure 2 is inspired from Brewster et al. and illustrates the multiple sounds around the users head and also the pie menu.

A system that uses sounds for interaction is the GpsTunes (prototype). With a mobile gps and a mp3 player, the system guides the user from a place to another, by shifting the volume and direction of the sound output, or as Strachan et al. [15] describe it "a handheld system capable of providing a user with the ability to navigate themselves to a desired location, through continuously adapted music, whilst 'on the move'." The system guides the user forward, left or right by altering the sound source. For example if the user is supposed to go left, the system will play the sound from the left. By varying the volume, the system informs the user how far from the target destination he/she is, the closer

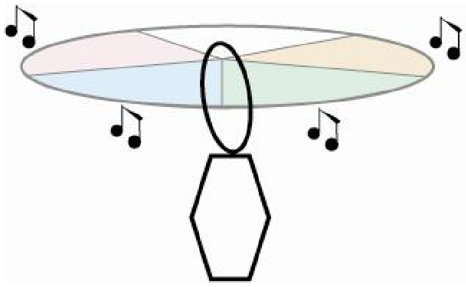

Fig. 2. 3D menu with multiple sound sources.

to the target the higher the volume and in the same way if the user moves away from the target, the volume decreases. When the user finally reaches the goal a special sound is played.

Another technique is interaction through 2D hand gestures. A system that uses this technique is the TouchPlayer (prototype). This is a mobile music player that introduces two new ways of interacting with a mobile device: gestures and feedback through non-speech sounds [14]. The gestures used to control the player consists of simple and logical moves across the touch screen. For example a sweep from the left to the right side of the screen implicates a change to the next track, a single tap means stop and a sweep from the bottom to the top of the screen increases the volume. An important part of the design is the feedback, for this earcons (non-speech sounds) and altering of the sound source is used. Earcons are abstract rhythmic/melodic patterns used to create messages. These messages represents different parts of an interface, for example a piece of the menu pie in figure 2 [16].

5 Discussion

This section discusses the devices and interaction techniques introduced earlier and suggest guidelines to use when designing MMFC interfaces. These guidelines have their base in Nielsen's ten design principles [17] and also in the results of the researches within the MMFC area.

Important factors: According to Brewster et al. [13] there are several factors to consider when designing mobile devices and the first is the fact that users cannot afford the loss of attention that arises when the user is forced to look at the display while interacting. The second is that the users often have their focus on other things when using mobile devices, it could be running, walking or just navigating the environment and the main reason for this is safety [13]. The third factor is the limitations on the devices, the small screens and restricted input/output capacities.

Target user: Another factor to have in mind is the target user. Who are the typical users of MMFC devices? I would say that the target users are in an age range between 13-55. These are people that have music and exercise integrated in their daily lives, but are more regular than elite sports people. It is important to know who the device is aimed for in order to give the user what he/she wants. For a regular user of a MMFC it can be sufficient with a music player and small features like step counter or distance monitor, this user may even be more interested in the appearance and usability of the device. The Nike+iPod system fits this user best, since it has an attractive appearance and is user friendly. An elite runner will request more advanced features such as specific information about his/her condition and will appreciate this more than other aspects. The best system for this user is the Adidas Polar system, it is not as user friendly as the Nike+iPod but is much more technologically advanced.

Context awareness: Many of the systems presented in this paper are context aware [1, 4–6]. That means that they use information about the users environment and condition in order to offer the users more automatize features. The Sony S2 sports player selects music based on the users heart rate, the Lifetrak system takes more factors into account (pace, location, weather etc.). The advantage with devices like the Sony player is that the user does not need to teach the device anything, while the Lifetrak system depends on it. On the other hand the Lifetrak player becomes more personalized.

The Mptrain is also a context aware system that differs from the others since it is incorporated into a mobile phone. The advantages with that is that no additional device is needed, the user can use his/her mobile phone instead of buying yet another device. The disadvantages are that the phone can be a bit cumbersome to carry around while exercising and also that the device is limited to the phones technical capacities (screen size, storage, input/output).

Interaction techniques: The interaction techniques used today (pressing buttons and displaying information on screens) have been used for many years and are known to work well, at least for stationary devices. These techniques work for mobile devices also but not as well, therefore there are many researches trying to design and test new interaction techniques more accurate for mobile devices. The techniques introduced in this paper where 2D gestures and 3D sound. Pirhonen et al. [14] states that the main advantage with these techniques is that the user can control the player without the need to look at the device when he/she is on the move. The result of the tests that Brewster et al. [13] conducted to try 3D sound and 2D gestures as interaction techniques showed that this is a good approach to use when designing an *eyes free* mobile device and that audio feedback was a very important feature to use in order to reinforce the techniques. But all these test are made by simple prototypes that need to be developed and further tested.

5.1 Guidelines

Considering all the factors just discussed, this section will present suggestions on guidelines to use in order to design a good user interface for a MMFC device. The following are Nielsens ten principles [17] adapted for a MMFC device interface:

1. *Visibility of system status* - in order to keep the user informed about the systems current status, feedback is required. The most effective feedback in

this case would be voice or sound feedback [2, 15], since this allows the user to maintain his/her attention on the task rather than on the device.

2. *Match system and the real world* - Nielsen says “speak the users language” and refers to keeping the dialog with the user at a familiar level. This is especially important when providing the user with voice feedback [2].
3. *User control and feedback* - it is important to make the user feel in control of the system, meaning that he/she can cancel and quit any operation at any time. Even when dealing with auto generated playlists the user needs to be able to skip a song or change some feature of the device/playlist [4].
4. *Consistency and standards* - using the same words and such through the entire system keeps the system consistent and the user has no need to be faced with ambiguity.
5. *Help users recognize, diagnose, and recover from errors* - when an error occurs, it is important to give the user enough information about the error and also guide him/her out of it.
6. *Error prevention* - a good and simple error preventer in this case is the lock function, this function prevents the user from accidentally pressing a button or changing something on the player.
7. *Recognition rather than recall* - the designer should not expect the user to remember the systems options, instead the user should be provided with cues and guidance through the system. A good example of this is the arrows that symbolize next or previous track and also the arrows that indicate how the volume can be increased or decreased.
8. *Flexibility and efficiency of use* - the system should support both novice and more experienced users by offering guidance for the novice and short cuts for the expert users.
9. *Aesthetic and minimalist design* - to keep a minimalist design is especially important in this case since the device is often small and cannot hold to much information in order to be easy to understand and quick to enact [5].
10. *Help and documentation* - when needed the user should be provided with guidance.

The following are additional guidelines with some examples from previously discussed subjects:

1. *Interact without visual attention* - it is important to avoid taking the users attention since it is needed to focus on the task [18]. This can be done by avoiding the users to use their eyes at all or more than necessary, since this causes a distraction from the main task [13, 14]. An interesting feature is the *shuffle shake* on the Sony S2 sports player, by shaking the device three times the player starts to play shuffled songs without taking much of the users attention [1].
2. *Feedback not only visual but also voice or sound* - feedback is very important in any system but is even more in this, when trying to keep the interaction eyes free. Voice and sound feedback are the most effective, according to Norman sound can be used to make things visible [19]. The sound in this case is used partly to confirm a chosen action but also to keep the user

informed [13, 14]. For example the Nike+ipod system uses voice feedback to inform its user about his/her status or as encouragement [2].

3. *Sustain mobility* - the device is intend to be used on the move and needs to be portable and have a proper size [5].
4. *Position of the device* - in order to make it easily accessed it is important to consider where the device is supposed to be carried around by the user (arm, side, pocket etc.) [14].

5.2 Evaluation

This section will use the guidelines presented to evaluate the devices in section 3.

The Sony S2 players minimalistic design, size and position (on the users arm) classifies it as a good mobile and portable device, it also allows the user to interact without visual attention through the features *Music Pacer* and *Shuffle shake*. But when it comes to feedback it is not optimal, since the screen where the information is displayed is very small.

The Nike+iPod system keeps its users informed about their status (time, distance, calories burned etc.) through the iPod screen, this screens size allows the user to see the information without much effort or attention cost. Apart from visual feedback, this system also offers voice feedback. The developers have put some weight on the voice feedback since the device is aimed for mobile users. The iPod has a clean design and can be carried on the arm, this makes the system more portable.

Since the Adidas Polar fusion is carried on the wrist it is very mobile and can be interacted with one hand. But it requires some visual attention and depending on what operations the user needs to do while running, the device is more or less efficient to use. The user obtains feedback through the watch screen, this can be very attention demanding specially since it is a lot of information on a small screen that the user needs to process.

Both Lifetrak and Mptrain are prototypes run from mobile phones which means that they use the phones interface and interaction techniques (touch screen, buttons, sound etc.) to reach the user. Interaction without visual attention can be hard since the prototypes uses the touch screen on the mobile phone. Lifetrak tries to make it easier by simplifying the interface, MPTrain makes it easier by allowing the user to make the selections for the workout (for example, selecting difficulty level) before the workout, this way the user does not need to worry about it during the workout. An advantage is that the mobile phone has a large screen in contrast to other mobile devices and allows the user to take part of the information displayed, without using to much visual attention. A disadvantage with the mobile phone is its size, it can be hard to carry around while exercising and also to find a good position to carry it around in.

6 Conclusion

The development of the MMFC was just a mater of time considering the close connection between music and fitness and also music as a companion in a persons life. Current technology makes it possible to develop more personal devices by making them more context aware, as exemplified with the prototypes presented in this paper. With more research about making mobile devices more user friendly and effective, maybe in the future everyone will have a small but effective personal trainer and workout friend to carry around.

References

1. Sony Electronics Inc.: Product description of S2 Sports Walkman MP3 Player (2008) http://www.sonystyle.com/webapp/wcs/stores/servlet/ProductDisplay?catalogId=10551&storeId=10151&langId=-1&partNumber=NWS205FBLACK, accessed 2008-04-27.
2. Apple Inc.: Product description of Nike + iPod (2008) http://www.apple.com/ipod/nike/, accessed 2008-04-27.
3. adidas AG: Adidas polar project fusion (2005) http://www.adidas-polar.com/phase5/index.html, accessed 2008-04-27.
4. Elliott, G., Tomlinson, B.: Personalsoundtrack: context-aware playlists that adapt to user pace. In: Conference on Human Factors in Computing Systems CHI '06 extended abstracts on Human factors in computing systems, Montréal, Québec, Canada, ACM New York, NY, USA (2006) 736–741
5. Reddy, S., Mascia, J.: Lifetrak: music in tune with your life. In: Proceedings of the 1st ACM international workshop on Human-centered multimedia, Santa Barbara, California, USA, ACM New York, NY, USA (2006) 25–34
6. Oliver, N., Flores-Mangas, F.: Mptrain: a mobile, music and physiology-based personal trainer. In: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services. Volume 159., Helsinki, Finland, ACM York, NY, USA (2006) 25–34
7. Warburton, D., Nicol, C., Bredin, S.: Health benefits of physical activity: the evidence. CMAJ: Canadian Medical Association Journal **174**(6) (Mars 2006) 801–809
8. Biehl, J., Adamczyk, P., Bailey, B.: Djogger: a mobile dynamic music device. In: CHI '06 extended abstracts on Human factors in computing systems, Montréal, Québec, Canada, ACM New York, NY, USA (2006) 556–561
9. Karageorghis, C., Terry, P.: The psychophysical effects of music in sport and exercise: A review. Journal of Sport Behavior **20**(1) (Mars 1997) 54
10. Tenenbaum, G., Lidor, R., Lavyan, N., Morrow, K., Tonnel, S., Gershgoren, A., Meis, J., Johnson, M.: The effect of music type on running perseverance and coping with effort sensations. Psychology of Sport and Exercise **5**(2) (April 2004) 89–109
11. Matesic, B., Cromantie, F.: Effects music has on lap pace, heart rate and perceived exertion rate during a 20 minute self-paced run. The sport journal **5**(1) (2002)
12. Bryant, J.: Portable devices. EDN **51**(5) (February 2006) 8–10
13. Lumsden, J., Brewster, S.: A paradigm shift: alternative interaction techniques for use with mobile and wearable devices. In: Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, ACM New York, NY, USA (2003) 197–210

14. Pirhonen, A., Brewster, S., Holguin, C.: Gestural and audio metaphors as a means of control for mobile devices. In: Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, Minneapolis, Minnesota, USA, ACM New York, NY, USA (2002) 291–298
15. Strachan, S., Eslambolchilar, P., Murray-Smith, R., Hughes, S., O'Modhrain, S.: Gpstunes: controlling navigation via audio feedback. In: Proceedings of the 7th conference on Human-computer interaction with mobile devices and services. Volume 111., Salzburg, Austria, ACM New York, NY, USA (2005) 275–278
16. Brewster, S.: Using nonspeech sounds to provide navigation cues. In: ACM Transactions on Computer-Human Interaction. Volume 5., Glasgow, United Kingdom, ACM New York, NY, USA (1998) 224–259
17. Nielsen, J.: Heuristic evaluation (2005) `http://www.useit.com/papers/heuristic/`, accessed 2008-05-05.
18. Preece, J., Rogers, Y., Sharp, H.: Interaction design: beyond human-computer-interaction. John Wiley and Sons, Inc, New York, United States of America (2002)
19. Norman, D.: The Design of Everyday Things. The MIT Press, London, England (2001)

Challenges with automated fingerprint recognition

Jannike Sehlstedt

Department of Computing Science
Umeå University, Sweden
dit04jst@cs.umu.se

Abstract. Fingerprints are the most common biometric characteristic. Unlike keys, cards and passwords, fingerprints cannot be lost or forgotten. In the mid 1970s, automated fingerprint recognition systems were developed and today the market for the systems is growing. But, the public is reluctant because they perceive the systems as unreliable, a perception with some support. This article reviews a number of challenges that must be met by researchers, manufacturers and vendors to make fingerprint recognition systems more usable and more reliable. User acceptance is one challenge, other examples are fingerprint quality, image quality, hostile attacks and frauds. The article also presents various countermeasures, e.g., feedback, cryptography, liveness detection and cancelable biometrics.

1 Introduction

Fingerprints are distinctive patterns on the tips of our fingers that cannot be lost or forgotten. In the late 19th century law enforcement and forensics began using fingerprints to determine the identity of individuals [1]. Research into automated fingerprint-based identification started in the early 1960s [2], and by the mid 1970s the FBI had developed Automated Fingerprint Identification Systems (AFIS) [3].

Today, the use of fingerprints for identification purposes is no longer limited to law enforcement and forensics only. Airports in several countries are incorporating identification of travelers based on fingerprints [4–6]. Other examples of applications are access to buildings, ATM authentication, and access to computer systems, laptops and cellular phones [1].

Regardless of the amounts of applications, many people have not yet encountered fingerprint recognition systems. The purpose of this article is to identify challenges that have to be met before automated fingerprint recognition systems can become as common as keys, cards and passwords, and also to find out what is being done to overcome them.

The article starts with an introduction to biometric characteristics and biometric systems in general. It continues with a description of fingerprint recognition systems; the most common sensors and algorithms and problems that they encounter. After that, challenges not inherent to the systems are reviewed; those include hostile attacks, frauds and users. All challenges, inherent or not, are then summarized and proposed solutions are presented.

2 Biometric characteristics

Humans have physiological and behavioral characteristics, more or less unique, that can be used to identify an individual. To automatically identify an individual, based on one or more of those characteristics is referred to as biometric recognition or biometrics [1, 7].

According to Jain et al., any physiological or behavioral characteristic that satisfies a number of predefined requirements can be used as a biometric characteristic. The requirements say that the characteristic must be universal, it has to remain unaltered over a period of time, it must be possible to distinguish one person from another based on the characteristic, and it must be possible to make repeated measurements of the characteristic. Examples of biometric characteristics are DNA, ear, face, fingerprint, gait, iris, voice and signature [1].

A fingerprint is a pattern on the fingertip consisting of ridges and valleys. The characteristic is formed during fetal development. Fingerprints are unique, not even identical twins have prints that look exactly the same [1]. Notice the choice of words, the prints do not look exactly the same but they are only slightly different since identical twins have identical DNA and fingerprints are in part genetically determined [8].

Semi-permanent or permanent changes in a fingerprint can occur due to a number of reasons, e.g., scars, manual labor or disease [9]. Elderly persons tend to have poorer fingerprints [9–11]. With age the skin becomes dryer and less firm due to loss of collagen and elastin fibers, which affects the quality of the fingerprint. It is also more likely that an elderly person have incurred damage to the fingers [9]. Gender is another factor when talking about fingerprint quality. An experiment has shown that the prints from men in general have higher quality than prints from women and quality in this case is the probability that the print would yield correct results in a fingerprint recognition process [10].

3 Overview of biometric systems

The process of biometric recognition involves some kind of biometric system and this section gives an overview of biometric systems in general. Since this article is focused on fingerprint recognition there is also a separate section about fingerprint recognition. Jain et al. describe a biometric system as composed of four modules [1]; sensor, feature extractor, matcher and database (see Figure 1). The system can work in two different modes; identification and verification [1].

The sensor is the device where the biometric data is captured. In the feature extractor the acquired data is processed and features of interest during recognition are extracted. The matcher compares the extracted features against stored templates. The templates, which belong to enrolled users, are stored in the system database.

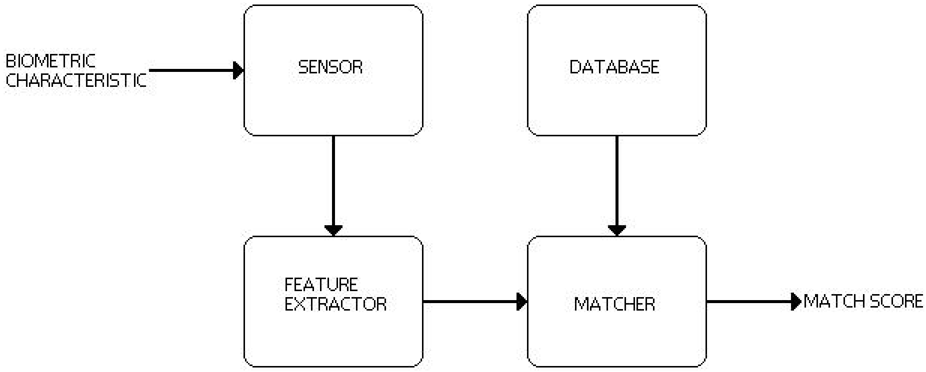

Fig. 1. The four main modules of a biometric system (see [1]).

Identification is the process of searching through the entire template database for a match (one-to-many comparison) while verification only requires a one-to-one comparison since the system only has to decide whether the user is who he/she claims to be. The response from the system is not a simple yes or no, because two samples of the same biometric characteristic are never identical. Instead a matching score is given that indicates how similar the input data and the templates are. A higher score means a stronger possibility that the user is correctly recognized. A threshold is set and if the score is higher, or equal, than the threshold you have a match.

Jain et al. explain two types of errors that should be mentioned when talking about biometric recognition and those are false match and false non-match [1]. False match, also called false accept, is when then system response says match but the two biometric measurements actually comes from two different individuals. False non-match, or false reject, is when two measurements that do come from the same person are assumed not to. Both errors are dependent on the threshold; if the threshold is low the false match rate is high and if instead the threshold is high the false non-match rate is high.

4 Fingerprint recognition systems

A fingerprint recognition system is simply a specialization of a biometric recognition system. As the name implies the biometric characteristic in this case is the fingerprint. Different biometrics require specific sensors capturing the biometric data as well as individual algorithms in the feature extractor and matcher. For fingerprints there are a number of different sensors available, but the two most common in use today are optical and capacitive [8, 12]. The algorithms most frequently used are based on locating minutiae points in the fingerprint image and then comparing them with the minutiae sets stored as templates [2, 12]. Minutiae points are, e.g., where the ridges that form a fingerprint terminate, called ridge endings, and ridge bifurcations where one ridge is divided into two ridges [2, 13] (see Figure 2).

Fig. 2. A section of a fingerprint. The arrow to the left points at a ridge ending and the arrow to the right points at a ridge bifurcation.

4.1 Sensors

Xia and O'Gorman argue that the sensor is a critical factor for the overall performance of any biometric system [11]. They also mention three factors that they find most important when evaluating a fingerprint capture device and those are; cost, size and performance. The cost has to be low in order for fingerprint recognition to be deployed, to a greater extent, for personal authentication. For the sensor to be incorporated into, e.g., laptop computers or PDAs it has to be small but it is not desirable to make it too small since that will impair the performance of the system. Performance depends on resolution, quality and capture area of the fingerprint image as well as durability of the sensor. If the resolution is low it can become difficult to separate ridges from valleys, especially if the spacing between the ridges is narrow. The quality of the captured image may be impaired if the finger is, e.g., dry or wet and lower image quality results in worse recognition performance of the system. Capture area is linked to recognition rate and the rate decreases with smaller area since that means fewer ridges and valleys captured and therefore less distinctive fingerprints. According to Xia and O'Gorman a sensor has to be impermeable to liquids and resistant to scratches, impact, latent fingerprints and electro-static discharge in order to be referred to as durable [11]. Latent fingerprints are the nearly invisible copies of our fingerprints that we leave behind after touching a surface [2].

Optical sensors. A simplified explanation of the principle behind optical fingerprint capture is that the finger is placed on a platen and light is emitted from the inside of the sensor onto the finger. The light that hits the ridges is not reflected in contrast to the light that hits the valleys which is totally reflected. Then, by focusing the reflected light onto a camera an image of the fingerprint is captured.

The development of new cameras and the improvements in optical components have enabled changes in the sensor design; changes that have resulted in smaller capture devices with lower price [11]. Because a certain distance between the platen and the imager is required there is a limitation in how small an optical sensor can become, so size may in some applications still be an issue. Further, instead of the traditional glass platen the new sensors have plastic platens which are more susceptible to scratches. Sensors with either kind of platens have to be cleaned periodically to avoid dirt buildup [11].

Capacitive sensors. A capacitor consists of two plates and the capacitance varies with the distance between them. In a capacitive sensor the finger is placed on a sensing surface. The skin of the fingerprint is considered to be one of the plates of the capacitor and the other plate is not one but a two-dimensional array of capacitors plates located below the sensing surface. The fingerprint is formed by ridges and valleys and because the ridges will be closer to the array of capacitor plates than the valleys there will be a difference in capacitance that enables differentiation between ridges and valleys. By charging the capacitor to a known value and measuring the change in voltage output over time at each capacitor plate in the array, an image of the fingerprint is determined [11].

Capacitive sensors are the most prevalent of capture devices belonging to the group of, so called, solid-state sensors. Other variants are thermal, radio frequency and pressure sensors [14]. The purpose when designing solid-state sensors, which became commercially available later than optical sensors, was to solve some of the problems with the optical devices [11]. But solid-state sensors have problems of their own. The silicon chip is rather sensitive and in other contexts the normal procedure would be not to touch it. When used for fingerprint recognition the chip has to be touched and must therefore sustain scratches and resist damage due to corrosive substances present in finger oils. Two other challenges are electro-static discharge and cost. A larger chip is more expensive than a smaller one and since the size of the chip has to correspond to the size of the fingerprint it is not possible to reduce the chip size in order to accomplish a lower price [11].

4.2 Algorithms

Two out of four parts of a biometric system (see Figure 1) are algorithms; the feature extractor and the matcher. Most of the individuality of fingerprints is captured by the spatial distribution of two basic local ridge structures (see Figure 2); ridge endings and bifurcations [2, 15]. Therefore, most fingerprint recognition systems have algorithms based on the extraction and matching of minutiae [2, 16, 17].

Minutiae extraction. The performance of a fingerprint recognition system depends heavily on the minutiae extraction algorithm [2, 15]. There are different approaches to minutiae extraction but according to Yager and Amin the most prevalent method is extraction form ridge skeletons which are binarized and thinned ridge structures [2].

In minutiae extraction from ridge skeletons they identify three preprocessing steps; orientation field estimation, ridge detection and ridge map thinning [2]. By estimating the orientation field you gain information about the local average directions of the ridges [2, 13]. That information is often used in fingerprint templates where each minutia is associated with the direction of the ridge at that minutia [15]. A ridge map is created by binarizing the fingerprint image and this step is what Yager and Amin call ridge detection [2]. Ridge map thinning is the process of making the representation of ridges one pixel thick without altering the topology and it results in the ridge skeleton. Preprocessing is followed by minutiae extraction. A ridge ending is a black pixel with only one black neighbor and a ridge bifurcation is a black pixel with more than two black neighbors [2, 15].

Noise in the fingerprint image cause undesired spikes and breaks in the ridge skeleton. Algorithms are designed to compensate for that by removing spikes and connecting gaps between ridges, before the minutiae extraction step [15], and by using a trained classifier to validate extracted minutiae [2]. But still many false minutiae may be detected. Two other drawbacks with this approach are that it requires extensive processing and that a lot of information is lost through binarization and thinning [2].

Minutiae matching. The minutiae extracted are commonly represented by their location and orientation. In most matching algorithms no difference is made between ridge endings and ridge bifurcations [2]. Jain et al. divide minutiae matching into two stages; alignment and matching [15]. Since the placing and pressure of the finger against the sensor is likely to be different at different occasions the minutiae set extracted from the input image has to be translated and rotated before it can be compared to the template. Yager and Amin mention four main complications with alignment of minutiae sets [2]. Different regions of the fingerprint may be captured and only those parts of the minutiae sets that overlap should be aligned. True minutiae may be missed and false minutiae may be extracted which may result in overlapping sets with different number of minutiae. Nonlinear deformation of minutiae sets makes perfect alignment impossible, if not explicitly modeled, and instead of perfect alignment the optimal alignment according to some criteria must be found. The algorithm must be able to handle these situations and, according to Yager and Amin, at the same time be very efficient [2].

During the matching stage a match score is calculated. A common implementation is to count the number of corresponding minutiae and normalize with the total number of minutiae [2]. In a fingerprint there are up to 80 minutiae, but since fingerprint sensors have a limited size sometimes only part of the print is captured. To be considered a match the number of corresponding minutiae has to exceed a threshold. In many courts of law twelve minutiae have to correspond in order for a match to be sufficient evidence [18].

The quality of the fingerprint image is critical to the performance of the algorithms [2, 15], and the fingerprint sensor has an indirect impact on the accuracy of the algorithms by partly determining the image quality [19]. There are a number of factors causing low quality in fingerprint images either by adding noise to the image or by distorting it. The imaging system may cause distortions due to imperfect imaging conditions [15]. Some factors are inherent within the individual and difficult to compensate for, e.g. age and gender [10]. Other factors are connected to the contact between the fingertip and the surface of the sensor. Jain et al. identify three such factors [15]. The pressure of the finger against the surface is likely to be different at different occasions as well as the position of the finger and due to that two impressions of the finger will not look the same. Another problem is when dirt, humidity in the air, dryness of the skin etc. are blocking some of the ridges on the finger from being in complete contact with the sensor and at the same time causing valleys to be in contact. The fingerprint can be semi-permanently or permanently damaged by, e.g., manual work or accidents causing what Jain et al. call irreproducible contact [15].

According to Bazen et al. the main drawback with fingerprint recognition systems based on minutiae extraction and matching is error propagation from the extraction stage to the matching stage [13]. With errors they mean missed and spurious minutiae in both the template and the input fingerprints which cause the identification or verification decision to be based on matching of affected minutiae sets.

5 Challenges from the outside

In the previous sections some challenges, inherent to the fingerprints or the fingerprint recognition system itself, were mentioned; varying fingerprint quality, sensor issues and algorithms limitations for example. In this section challenges that come from the outside are reviewed. They have been divided into three categories; hostile attacks, frauds and users.

5.1 Hostile attacks

The four modules of a biometric system, as well as the data being transmitted may be attacked by imposters and hackers. This section contains a number of possible attacks, starting with threats to the sensor module and continuing with threats at each step in the recognition process. The attacks and their locations in the biometric system were identified by Ratha et al. [7].

Imposters may try to spoof the biometric system by presenting fake biometrics at the sensor [7], a review of different kind of frauds specific for fingerprint recognition can be found below under the heading Frauds. Because data is being transmitted between the modules, hackers may attack the system at various locations using different methods. One possibility is to bypass the sensor and resubmit e.g. an old copy of a fingerprint image; another is using a Trojan horse to attack the feature extractor [7]. A Trojan horse is a piece of software containing hidden code, the software performs a useful function while the hidden code performs a usually malicious function [20]. In this case the intruder selects feature sets that the hidden code makes the feature extractor produce. Further, if the feature extractor and the matcher are separated it is possible to direct an attack on the data transmitted from the feature extractor to the matcher by replacing extracted features with fake features. The matcher can also be attacked directly making it produce false match scores. Templates are, as mentioned, stored in a database. An attack directed at the database with the intention of altering the templates could result in legitimate users being denied access or impostors gaining access. The same thing could happen if the templates are altered on their way from the database to the matcher. Finally, the decision made by the matcher could be overridden [7].

5.2 Frauds

A biometric system being monitored by a human is difficult to spoof by presenting a biometric characteristic to the sensor that is not your own. Forcing an authorized person to present his/her characteristic to the system would be impossible and presenting a fake characteristic could easily be detected. But that is not the case with an automated system. Matsumoto et al. have identified a number of approaches that an attacker can use in order to deceive a fingerprint recognition system [8]. One approach is for the imposter to present his/her own finger to the sensor and hope for a high false match rate. By causing errors in the recognition process the chance of gaining access can increase. Some fingerprint systems are prone to errors after being exposed to e.g. high humidity, strong vibrations, heat or flashing light. Another approach, more likely to succeed, is the one mentioned in the beginning of the section; to force an authorized person to press his/her finger against the scanner. Two versions of that approach would be to press the registered finger onto the scanner while the person is unconscious, or to cut the finger off and then use it for deception as some people fear according to [3]. The registered finger or a latent print of it could be used to create an artificial clone of the finger. A more futuristic idea is that a genetic clone of the registered finger could be used.

Matsumoto et al. have focused on the approach regarding artificial clones of registered fingers [8]. They have performed an experiment where seven optical sensors and four capacitive sensors were tested to see if they could be fooled by gummy fingers. Molds were made in two different ways and they were then used to produce artificial fingers from gelatin. One mold was made by pressing a live finger against a plastic material and the other was made after first using a digital microscope to capture a fingerprint image from a residual fingerprint. The results of the tests show that during the verification procedure the artificial fingers made from a live finger mold was accepted from 68-100 percent of the times and the artificial fingers made from residual fingerprint mold was accepted more than 67 percent of the times [8].

A less scientific study was performed by Thalheim et al. [21]. They succeeded in deceiving some capacitive sensors simply by breathing on them, that way reactivating the residual fingerprint left on the sensors surface. The sensors were also deceived by a latent fingerprint reactivated by placing a water-filled plastic bag against their surfaces. Another method, the one most difficult for the sensors to reveal, was dusting the residue of a fingerprint using commercially available graphite powder, securing it with adhesive film, placing it on the scanner and applying gentle pressure on it. The two optical sensors tested could also be deceived with that method, but only when a halogen lamp was shining on the scanners from a distance of about 30 centimeters. The optical sensors were also spoofed by an artificial finger. Thalheim et al. made a mold by pressing the fingertip into warm wax and then filled it with silicon resulting in an artificial finger [21].

5.3 Users

Studies show that people are concerned about their privacy when using biometrics and that they find biometrics invasive [3, 22]. The privacy issue is relevant because biometric characteristics could provide information about a person's background that would not be obtainable if for example a password was used for recognition [1]. Related to that is the concern that information contained in different databases could be coordinated and shared [7]. Another important issue is that people do not trust the security of the systems; in a survey by Moody there was a strong disagreement with the statement/claim "Biometric devices provide more security at an ATM" [3], and in an experiment by Heckle et al. the participants were unable to understand the security aspect of biometrics which made them hesitant to using the technology [22]. During interviews with the participants of the experiment a reappearing opinion was that biometrics would be better accepted if it was more common [22]. An indication of that can be seen in [3]. Among the biometrics in use today the fingerprint is the most widely recognized, and on the question which biometric people would prefer to use instead of passwords for computer logon, office access and ATMs, in all three cases, a majority chose fingerprints [3]. People's perception of biometric systems is also affected by the context of use. In an application context where there are personal benefits for the user there is a greater perception of usability and a higher acceptance rate than in contexts where there is only system or corporate benefits [22].

6 Solving the challenges

The previous sections give an overview of biometric systems in general and fingerprint recognition systems in particular and reveal many of the challenges that come with those kinds of systems. In this section those challenges are summarized and various solutions proposed in the literature are described.

6.1 Fingerprint image quality and feedback

Fingerprints, which are the most common biometric characteristic [2], are permanent but due to injuries, manual work and age their appearance change [9–11]. Such changes can affect the performance of a fingerprint recognition system by impairing the quality of the fingerprint image and they are difficult to compensate for since they are inherent. According to an empirical estimation about 4 percent of the population may have fingerprints that fingerprint sensors cannot image sufficiently well [1].

The quality of the fingerprint image can also be impaired because the pressure of the finger against the sensor is too low or too high, or the position of the finger is wrong [15]. Such problems can be avoided or at least minimized if the system gives the user appropriate feedback. A study by Theofanos et al. showed that feedback resulted in improved fingerprint image quality [10]. Today, many systems only indicate when the user should start and stop presenting their finger and, in verification mode, if an identity claim is accepted or not [10].

6.2 Identification and classification

A biometric system can work in two different modes; identification or verification [1]. One problem for biometric systems is that the one-to-many comparison being performed when working in the identification mode requires a large amount of computational resources [1]. To reduce the number of comparisons that has to be done, and consequently reducing the computing requirements, it is common to classify fingerprints based on their global appearance [2]. Then, at identification, the fingerprint is only compared to templates belonging to the same class.

6.3 Issues and advances in sensor technology

The two most prevalent fingerprint sensors, optical and capacitive, have many problems in common. They are both susceptible to scratches, they have to be cleaned repeatedly because dirt from the fingertip stick on the sensor surface and cause low quality prints [11]. If the finger is dirty, wet or dry the quality of the fingerprint image captured will be low because both technologies collect their fingerprint information from the outermost layer of skin [23]. Solid-state sensors based on radio frequency signals are supposed to overcome some of those problems by reading the live layer of skin that resides below the surface [23]. Other sensors scanning various skin layers are based on a technique called Multi Spectral Imaging [16]. There is also a fingerprint sensor technology based on ultrasound that is less affected by dirty, dry and wet fingers, but the image capture takes longer to perform and the equipment is usually bulky [24].

Progress in the solid-state fingerprint sensors technology has resulted in a swipe sensor. Xia and O'Gorman identify a number of advantages with swiping the finger across a small, linear, solid-state sensor [11]. By swiping, a longer area of the finger is captured, than for touch sensors, and larger capture are means higher recognition rate. As mentioned, the cost of solid-state sensors depends heavily on the size of the silicon ship and in this case the chip size can be reduced by a factor of ten. The size reduction provides other benefits as well; the sensor is more durable since there is a smaller area to damage and it can easily fit into small mobile devices. Latent fingerprints are not a problem since only a slice of residue is left on the sensor. Finally, there is no dirt build-up because the swiping action cleans the sensor surface.

All of the sensors above require that the fingertip touches the sensor, either by pressing it onto the sensor surface or swiping it across the surface. A new technology, based on a multi-camera system, is completely "touchless" [16]. By avoiding touch the captured fingerprint image is not distorted in any way.

6.4 Algorithms

The most common fingerprint recognition algorithms are based on extraction and matching of minutiae [12, 16]. However, several drawbacks with that approach have been identified. In the case where minutiae are extracted from ridge skeletons the computational complexity is high, information is lost due to binarization and thinning and that is one reason why the results for low quality images can be poor [2], and errors are propagated from the feature extraction stage to the matching stage [13]. Some algorithms extract minutiae direct from grey scale images but according to Yager and Amin those are complex and difficult to implement [2]. Prabhakar et al. propose a combination of the two approaches; minutiae are extracted from ridge skeletons and then verified based on information from the grey level image [17]. To overcome some of the problems with the minutiae-based approach, algorithms that extract and match other features have been developed. One example is the correlation-based fingerprint verification system proposed by Bazen et al. [13]. It does not require much preprocessing and low quality prints and missed and spurious minutiae is not an issue. Tests show that the system is neither better nor worse than other fingerprint verification systems but that it is less applicable for real time applications because it is based on template matching which is a rather computationally expensive procedure [13]. But since minutiae points capture most of the individuality in fingerprints [2, 15], more recent advances are concerned with enhancing and updating algorithms connected to minutiae extraction and minutiae matching [25–30].

6.5 Hostile attacks and various countermeasures

A biometric system can be subject to a number of hostile attacks at different locations and by different means. The four modules can be attacked and also the communication channels between them. There also exist different methods to protect the system. The communication channels can be encrypted, the matcher and the database can be placed at a secure location and cryptography can be used [7]. Ratha et al. put extra effort into how to thwart replay attacks, bypassing the sensor [7]. Replay attacks can be avoided by hiding additional information in the compressed biometric data. That makes it possible to check if submitted data was generated by the sensor, or not. Another proposition is a challenge/response method. Instead of challenging the user, which is often the case with password-based authentication systems, the sensor is challenged. The sensor receives a challenge string, which is different for different presentations, and then computes a response string that depends both on the challenge string and the content of the biometric data. Thanks to that, the method can check both the integrity and the liveness of the submitted signal.

6.6 Frauds and liveness detection

An unsupervised system is more susceptible to frauds than a supervised system. A registered user can be forced to press the finger against the sensor, consciously or unconsciously, the finger can be severed from the user's hand or an artificial or a genetic clone of the finger can be made. Combining fingerprint recognition with another authentication method, for example PINs or passwords, is one way to deter such frauds [8, 31]. Matsumoto et al. also propose a system that requires input from two persons [8]. Countermeasures for severed fingers and artificial fingers, other than those mentioned above, are multimodal biometrics [1, 31], liveness detection [8, 31] and a method where several fingerprints from the same person are enrolled and the system asks the user to present a specific sample [31]. A multimodal biometric system is a combination of multiple biometric modalities [1, 31]. There are many different methods to check whether a finger is alive or not but no method is invincible [31, 32]. Intrinsic properties of the finger may be analyzed; examples are elasticity, capacitance and opacity. Another method is to measure the pulse or the perspiration of the finger, properties that Toth calls involuntary, to detect liveness [32]. The liveness decision can be based either on information already captured by the fingerprint sensor or on life signs acquired from additional hardware. Drawbacks with additional hardware is that life signs could come from an impostor while the biometric data is fake, and it is often expensive and bulky [31, 32].

6.7 Mistrust and openness

There is a lack of confidence in the reliability of biometric systems [3, 22]. Matsumoto et al. point out that manufacturers and vendors of biometric systems should, after examining the security of their systems, make public the results in order for users to better understand the systems security [8]. Vendors do not want to disclose the specifics about their systems [33] and they give unrealistic performance claims [7], that way encouraging people to try to outsmart them. Recently, a man decided to put a fingerprint identification system deployed by SAS at domestic flights in Sweden to the test [34]. The purpose of the system is to confirm that the persons boarding the plane are the same as those checking in [35]. This man checked in manually with his ID card instead of using the automated fingerprint system. At the gate he, nonetheless, put his finger on the reader and out came a boarding card belonging to someone else [34]. Incidents like that have a way of sticking in people's minds degrading their already limited trust in biometric systems. The International Biometric Group (IBG) argues that the biometric industry would benefit from being more open about their technology and also from giving realistic performance claims.

6.8 Privacy concerns and cancelable biometrics

Another concern among users is invasion of privacy. Ratha et al. introduce something called cancelable biometrics [7]. The biometric data is intentionally distorted in the same way at every presentation. If a person uses his/her fingerprint for authentication in different systems the distortion transform should be different. Further, the transform function must be non-invertible to make it impossible to recover the original data from the distorted. This way, if the biometric data is compromised one only has to change the distortion transform and also, it is impossible to coordinate or share data, about an individual, from various databases.

7 Discussion

The purpose of this article was to find out what challenges automated fingerprint recognition systems have to meet and what is being done to overcome them. A number of challenges and possible solutions have been identified.

Fingerprints are not optimal as biometric characteristics. The quality of fingerprints varies not only between individuals, but also fingerprints from the same person can vary in quality depending on, e.g., placement and pressure of the finger against the sensor. Low fingerprint quality causes fingerprint sensors to capture low quality fingerprint images, and low quality fingerprint images causes feature extraction and matching algorithms to produce unreliable results.

Studies show that sensors are far from perfect. They can easily be deceived by various spoofing-attacks unless they deploy liveness detection of some kind. Even if liveness detection is deployed the system may be vulnerable; hostile attacks can be directed at other modules than the sensor or at the data transmitted between them. There are suggested countermeasures, but it is important to remember that no system is perfect and insistent impostors will eventually succeed.

It seems like much effort is put into the development of new sensor technologies and novel algorithms. Therefore the overall performance of fingerprint recognition systems is likely to improve, and the systems will probably become more and more optimized and adapted to the intended sector of application both regarding price, size and performance.

Public perceptions are another matter. Studies show hesitant users, concerned about their privacy and questioning the security of biometric systems. The lack of trust in the systems is partly connected to their deficient performance but improved performance is probably not enough to convert the public. IBG and Matsumoto et al. are right advocating openness amongst manufacturers and vendors of biometric systems. Adequate error rates and an understanding of the underlying technology of the systems could influence contemplated users in a positive direction. The biometric industry is growing. Since we are creatures of habit and afraid of being the odd ones out an increased deployment of fingerprint systems should also have a positive effect. These are only guesses because despite the results from the studies, little seems to be done to gain user acceptance.

Xia and O'Gorman argue that the most important factors restraining fingerprint recognition are cost, size and performance [11]. Considering the amount of research focusing on development of new sensors and improvement of algorithms and then looking at the bad publicity every mistake made by a biometric system causes, I have to disagree. I believe that resistance among intended buyers and users, the real potentates in the market, is more important. Therefore, future research should also include the topic "How to change public perceptions regarding fingerprint recognition?" if automated fingerprint recognition systems ever are to be as common as keys, cards and passwords.

References

1. Jain, A., Ross, A., Prabhakar, S.: An introduction to biometric recognition. Circuits and Systems for Video Technology **14**(1) (2004) 4–20
2. Yager, N., Amin, A.: Fingerprint verification based on minutiae features: a review. Pattern Analysis and Applications **7**(1) (2004) 94–113
3. Moody, J.: Public perceptions of biometric devices: The effect of misinformation on acceptance and use. Issues in Informing Science and Information Technology (2004) 753–761
4. TT: USA tar avtryck på alla fingrar (2008) `http://www.aftonbladet.se/resa/article2126138.ab` accessed 2008-03-26.
5. UK-Airport-News.info: Fingerprinting at Heathrow, Gatwick and Manchester Airport from next month (2008) `http://www.uk-airport-news.info/manchester-airport-news-090308.htm` accessed 2008-05-05.
6. Vidlund, S.: SAS tar fingeravtryck på resenärerna (2008) `http://www.aftonbladet.se/resa/article1718502.ab` accessed 2008-05-05.
7. Ratha, N.K., Connell, J.H., Bolle, R.M.: Enhancing security and privacy in biometrics-based authentication systems. IBM Systems Journal **40**(3) (2001)
8. Matsumoto, T., Matsumoto, H., Yamada, K., Hoshino, S.: Impact of artificial "gummy" fingers on fingerprint systems. In: Proceedings of the Conference on Optical Security and Counterfeit Deterrence Techniques IV. (2002)
9. Elliott, S., Kukula, E., Sickler, N.: The challenges of the environment and the human / biometric device interaction on biometric system performance. In: Proceedings HBDI and Environment. (2004)
10. Theofanos, M., Micheals, R., Scholtz, J., Morse, E., May, P.: Does habituation affect fingerprint quality? Software—Practice and Experience **29**(5) (2006) 457–478
11. Xia, X., O'Gorman, L.: Innovations in fingerprint capture devices. Pattern Recognition **36**(2) (2003) 361–369
12. Maltoni, D.: Most common fingerprint sensors and algorithms (2008) Personal communication 2008-04-17.
13. Bazen, A.M., Verwaaijen, G.T.B., Gerez, S.H., Veelenturf, L.P.J., van der Zwaag, B.J.: A correlation-based fingerprint verification system. In: Proceedings of the ProRISC2000 Workshop on Circuits, Systems and Signal Processing. (2000)

14. Biometrika: A technical evaluation of fingerprint scanners (2007) http://www.biometrika.it/eng/wp_scfing.html accessed 2008-04-14.
15. Jain, A.K., Hong, L., Pankanti, S., Bolle, R.: An identity authentication system using fingerprints. Proceedings of the IEEE **85**(9) (1997) 1365–1388
16. Dass, S.C., Jain, A.K.: Fingerprint-based recognition. Technometrics **49**(3) (2007) 262–276
17. Prabhakar, S., Jain, A.K., Pankanti, S.: Learning fingerprint minutiae location and type. Pattern Recognition **36**(8) (2003) 1847–1857
18. Pankanti, S., Prabhakar, S., Jain, A.K.: On the individuality of fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(8) (2002) 1010–1025
19. Cappelli, R., Ferrara, M., Franco, A., Maltoni, D.: Fingerprint verification competition 2006. Biometric Technology Today **15**(7–8) (2007) 7–9
20. Denning, P.J.: Computer viruses. Technical report, Research Institute for Advanced Computer Science, NASA Ames Research Center (1988)
21. Thalheim, L., Krissler, J., Ziegler, P.M.: Body check: Biometric access protection devices and their programs put to the test (2002) http://www.heise.de/ct/english/02/11/114/ accessed 2008-04-07.
22. Heckle, R., Patrick, A., Ozok, A.: Perception and acceptance of fingerprint biometric technology. In: Proceedings of the 3rd symposium on Usable privacy and Security. (2007) 153–154
23. AuthenTec: Technology (2007) http://www.authentec.com/technology.html accessed 2008-04-14.
24. Cerebrix Technologies: Fingerprint sensor technologies (2008) http://www.cerebrix.in/content/view/3/4/ accessed 2008-04-14.
25. Cavusolgu, A., Gorgunoglu, S.: A fast fingerprint image enhancement algorithm using a parabolic mask. Computers and Electrical Engineering **34**(3) (2008) 250–256
26. Ji, L., Yi, Z.: Fingerprint orientation field estimation using ridge projection. Pattern Recognition **41**(5) (2008) 1491–1503
27. Miao, D., Tang, Q., Fu, W.: Fingerprint minutiae extraction based on principal curves. Pattern Recognition Letters **28**(16) (2007) 2184–2189
28. Lumini, A., Nanni, L.: Advanced methods for two-class pattern recognition problem formulation for minutiae-based fingerprint verification. Pattern Recognition Letters **29**(2) (2008) 142–148
29. Feng, J.: Combining minutiae descriptors for fingerprint matching. Pattern Recognition **41**(1) (2008) 342–352
30. Tong, X., Liu, S., Huang, J., Tang, X.: Local relative location error descriptor-based fingerprint minutiae matching. Pattern Recognition Letters **29**(3) (2008) 286–294
31. Schuckers, S.A.C.: Spoofing and anti-spoofing measures. Information Security Technical Report **7**(4) (2002) 56–62
32. Toth, B.: Biometric Liveness Detection. Information Security Bulletin **10** (2005) 291–297
33. Group, I.B.: Liveness detection in biometric systems http://www.biometricsinfo.org/whitepaper1.htm accessed 2008-04-07.
34. Chaaban, S.: Fingret lurade SAS-systemet (2008) http://www.expressen.se/resor/1.1114579/fingret-lurade-sas-systemet accessed 2008-04-08.
35. Chaaban, S.: Systemet används bara av SAS (2008) http://www.expressen.se/resor/1.1114590/systemet-anvands-bara-av-sas accessed 2008-04-08.

Quality of service with IPv6 for mobile television in mobile networks

Magnus Larsson

Department of Computing Science
Umeå University, Sweden
dv04mln@cs.umu.se

Abstract. Mobile television is a new application that utilizes the mobile network. A big problem in this area is how to maintain a good QoS. Currently, the Internet cannot give any guarantees of QoS, but with IPv4 being replaced with IPv6, this situation will change. This article identifies both global QoS properties and specific QoS properties for mobile television, and looks at how IPv6 can improve these properties compared to IPv4. The final conclusions reveal that by changing Internet Protocol to IPv6, several aspects of QoS for mobile television will be improved. The new flow label field and the fixed header size are some of the properties of IPv6 that will enhance QoS for mobile television, specifically the properties loss, delay, jitter and mobile resources will be improved.

1 Introduction

Both mobile devices and mobile networks have evolved rapidly the last few years. This has lead to many new areas of applications for mobile devices. One new area that is in constant development is multimedia applications for cellular phones, such as mobile television and streaming movies. With all these new technologies there are also several problems.

One major problem with multimedia applications is how to keep the connection between the two devices as loss free and stable as possible, in other words how to maintain the Quality of Service (QoS). This is a problem in regular wired networks, but it is an even bigger problem in wireless and mobile networks since these networks suffer from more loss, higher latency and lower bandwidth capacity. Several different techniques are used to improve QoS, but there is currently no globally implemented support for QoS in the Internet Protocol (version 4), and it is consequently impossible to guarantee a certain level of QoS [1].

Multimedia is a term that is hard to define in a precise way. This also means that it is difficult to specify the properties of a multimedia application. In this article it is defined as an application that receives a multimedia stream from an external node or server over the mobile network. Examples of these kinds of applications are mobile television, radio over the network and pay-per-view movies. Multimedia content is in this context referred to as the stream of data that is transferred. This data stream could basically be any kind of data but is usually some kind of mix between audio and video. According to Kurose *et al.* [2], there are three different classes of networked multimedia applications:

- *Streaming stored audio and video*, such as old television shows and movies. Some delay is accepted before the playback begins, but once it has started it should not have to stop and buffer.
- *Streaming live audio and video*, like live radio or live television. A startup time is accepted, but since the media is live the application cannot ask again for lost data.
- *Real-time interactive audio and video*. This is the most demanding applications since it requires interaction between two parties. A delay of up to 400 milliseconds is acceptable, and specially for these kinds of applications, loss is preferred over delay.

The goal of this article is to discuss in which way the new Internet Protocol version 6 (IPv6) will help to improve QoS in mobile networks, with focus on the multimedia application mobile television.

The next section in the paper discusses the techniques behind mobile television and the different protocols that are used for transferring data. Section 3 talks about QoS in general while QoS for mobile television is the subject of Section 4. A brief overview and the most important parts of IPv4 and IPv6 are presented in Section 5. In Section 6, there is a discussion of how IPv6 will improve QoS for mobile television. Finally, there are some conclusions about this article in Section 7.

2 Mobile television

Mobile television is quite a new application and is still being developed and evaluated, despite this there are several operational mobile television services. In Sweden only, there are at least four big telecom companies that offer this service (Telia, Tele2, TRE and Telenor). The goal with mobile television is to let the consumers watch television shows wherever they go. A common example of usage is when one is waiting for the bus or is in the subway. Figure 1 shows an example of a mobile television application. In some sense, mobile television should be like watching television in one's home.

There are however some differences between mobile television and regular television. The quality of the video and sound is not as good as regular television, and there is a delay of around five seconds when changing channel. Another difference is that one cannot always watch the regular television channels, instead there are loops of programs or stored shows that are viewable, at least in the case of TRE's mobile television service [3]. In an article from Computer Sweden [4], this issue is brought up, stating that if mobile television is going to survive as an application, improvements have to be done on the content.

To transmit the television content to mobile devices two main techniques are used, either via the 3G network or via dedicated broadcast networks. The 3G network uses unicast communication to distribute the television shows, which

Fig. 1. Example of TRE's mobile television service (Copyright ©2006 Hi3G Access AB, used with permission).

means that every user has to establish its own connection to the server to be able to stream the content. This also allows the user to watch shows that are stored on the server, not only live shows [5]. This property makes the mobile television service to be a mix of two network multimedia classes, both streaming stored audio and video as well as streaming live audio and video.

On the other hand, the broadcast networks such as DVB-H [6] and DMB [7], distribute the content on the whole network. This resembles a standard television broadcast in the sense that one only can watch what is currently being broadcasted.

Currently, most mobile television services use the 3G network and unicast techniques. The main reason for using unicast instead of broadcast, is that broadcast is expensive to use since it requires the special networks mentioned above. However, there is a protocol being developed, called *Multimedia Broadcast Multicast Service* (MBMS), which will make it possible to use broadcast in the 3G network. This will be a good complement to the unicast technique since areas with high usage may utilize broadcast instead of unicast and thereby save capacity on the network [3].

This article will focus mainly on the applications that use unicast over 3G networks since this is currently the dominating standard for transmitting mobile television.

2.1 Multimedia protocols

The protocols being used with mobile television can vary depending on which company that provides the service. But the basic protocol every multimedia service uses is UDP. The alternative to UDP is TCP, but TCP has too much overhead and uses congestion control which makes the latency too high for multimedia applications [1, 8, 9].

Commonly, middle layer protocols are utilized to simplify the transmission of data for multimedia applications. Some of the protocols that has this purpose are; Real-time control protocol (RTCP), Real-time Transport Protocol (RTP) and Real Time Streaming Protocol (RTSP) [1].

As an example, we can look at TRE's mobile television service. It uses UDP in conjunction with RTSP. RTSP enables an interface for controlling the multimedia stream from the server and provides, among other things, a play and a pause command to start and stop the stream from the server. TRE does not use RTSP together with some other middle layer protocol, they use RTSP directly on top of regular UDP. But a common combination is to use RTSP together with RTP. The purpose of RTP is to simplify the actual data transfer, since RTSP is only a control protocol and does not handle any data streams [3, 10, 11].

3 QoS in general

Delay, loss, jitter and throughput are some of the terms that are used to define QoS. Common for all these terms is that it is possible to measure them in applications and thus get a reading on QoS. They are also what can be improved when changing protocol on the network layer. The rest of this section identifies and defines the most important QoS properties that are used later on in this article. The definitions are adapted from Dixit et al [12] and Gulliksson and Lindström [1].

Loss is the phenomenon when a packet is dropped somewhere along the way to the receiver and never arrives. This can only happen when sending data over UDP since TCP has a system for recovering from packet loss. Most multimedia applications uses UDP, but these applications has to be able to handle loss themselves, as they get no support from UDP with this task. Loss could for instance occur if a buffer in a router is full when the packet arrives. But, when using wireless mobile networks, packets can also be lost due to bad signals because the wireless signals are not as reliable and tolerant to interference as wired networks.

Delay, also known as End-to-end delay, is the total time it takes for a packet to be processed in the source node until it has been transported to, and processed in, the destination node. Depending on which application that is used, different maximum delay times are accepted.

Jitter is basically the variation in delay time. For various reasons the delay can differ between different packets depending among other things on the work load in the routers. This time variation can have a big impact on multimedia applications since they demand continuous playback. However, if the jitter is not too big, it can be removed in the application itself using buffers.

Throughput is the amount of data that can be transported over the network per second. High throughput can solve many problems regarding QoS, but not all of them.

Error rate is the measurement that shows the number of bit errors. Checksums may be used to check for bit errors but it is often hard to recover from bit errors without retransmitting the whole packet. Except for the five terms mentioned above Gulliksson and Lindström. [1] also mention time of the connection and reliability of the network as other metrics for QoS. These are maybe not as obvious and important compared to the five above, but they may play a role in the overall QoS.

Other more vague terms also exists that can be used to determine QoS, but these are often harder to measure. One example is the overall satisfaction of the service, from the users point of view. It is impossible to measure this in an objective way, it has to be done with interviews and such. There are technical things like delay, loss and jitter that plays a big role here. But other things like user expectations, previous experiences and which hardware is used to display the content are also very important to how the QoS is experienced. This phenomenon is called user perceived QoS and is discussed for instance in an article by P.M. Ruiz [13]. The user perceived QoS properties will not be considered in this article.

4 QoS for mobile television

The same QoS properties that are mentioned in Section 3 also applies to mobile networks. But, since mobile networks are wireless and the mobile devices have different capabilities compared to regular computers, there are even more things that affects the QoS for mobile television. Chalmers and Sloman in refernce [14] talk about QoS in mobile networks, and although their work is quite old, many of the QoS-properties are still valid in current networks. The properties they come up with are:

- Since mobile devices often have less CPU-power, smaller battery size and smaller memories there are hard constraints on the applications. They should not be to big (in terms of memory) and not to computationally heavy (take lot of energy and demands high CPU performance).
- The quality and speed of the link can vary very much in one session. Therefore is the link quality of big importance to QoS.
- Since mobile devices move around a lot it is important for the reliability of the service to be able to maintain the connection to the device.

Most other QoS aspects for mobile television are the same as for stored and live multimedia applications for mobile devices. Specific for mobile television is that it demands a higher bandwidth than other mobile streaming services [3].

One last thing that may be looked at as a QoS property is that if mobile television wants to resemble real television, it should be fast to surf between different channels. When changing channel, the mobile device has to buffer some of the program before it starts to display the content. This is also something that will be looked at in Section 6.

5 The Internet Protocol

The Internet Protocol (IP) is a major part of the network layer in the protocol stack of the Internet (Figure 2). The network layer is responsible for ensuring that the packets, also known as datagrams, reach the destination node. To accomplish this there are routing protocols in the network layer that determine which path the datagrams should take, in order to arrive at the intended destination [2].

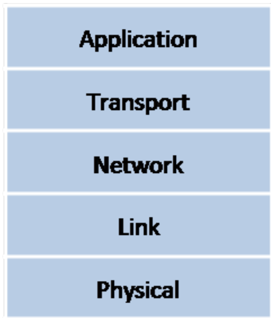

Fig. 2. The layers of the Internet protocol stack

The role of the IP is to define both the actual datagrams that is sent from the network layer, as well as how the datagrams should be handled at each node it passes. Currently, IP version 4 (IPv4) is used and it has been in service for the last decades. However, there are some major issues with IPv4 that has lead to the development of a new IP standard, IPv6. Currently, IPv6 is being gradually deployed around the world [2]. According to Lebers's site [15], there are currently 3.3% of the autonomous systems in the world that are using IPv6.

In the next subsections, IPv4, IPv6 and IP mobility are discussed in some detail. This sets the foundation of the whole article and from these basics, some conclusions are made in Section 6 that answers the main question of the article. Most of the technical details are from the RFC's of the protocols, i.e. the Internet Engineering Task Force (IETF) standards. The RFC is mentioned for every protocol that has one, and references to other sources are made where it is needed.

5.1 IPv4

The datagram format of IPv4 [16] is shown in figure 3. Each row is 32 bit (4 bytes) long except for the data field which has a size that is calculated from the *Datagram length (bytes)* and *Header length* fields. The header is 20 bytes large in total, 24 bytes if any options are included.

The field *Type of service* (*TOS*) [17] defines if a packet needs a certain kind of treatment, for example low delay or high throughput. This field is not implemented by all routers and is sometimes used for either Differentiated services (DiffServ) or Explicit Congestion Notification (ECN) [18]. According to the RFC for *TOS*, the IETF now mandates that this field should be u sed by the hosts.

Time-to-live (*TTL*) is a field that specifies how many seconds a packet may live before it is destroyed. The field has to be decreased by every router the packet passes, even though it has been less than one second between two routers. Therefore the *TTL* specified is a theoretical upper limit, not the actual time it will live.

IPv4 is a best effort protocol[2], which means that it does its best to deliver every packet as quick as possible, but no guarantees of QoS are given. This makes it impossible for the other upper layers in the protocol stack to give any guarantees about QoS. However, there are protocols that try to tackle the QoS issue. Two of the most common approaches are Integrated Services (IntServ) and DiffServ.

IntServ operates at a low level and allocates resources for every connection that is created, before a data packet is sent. This technique is however rarely used since it scales very bad, but it might be used in small local networks [12].

DiffServ [19] on the other hand is more popular. Instead of allocating resources for every connection, it gives promises to different classes of traffic in the network. Currently, the class of a packet is often determined by looking at the *TOS* field. The *TOS* field is divided into three sections, where the first three bits specifies the importance of the packet (0-8). The following three bits decides if low delay is required, if high throughput is needed and if high reliability is demanded. The last two bits are undefined in the standard IP protocol, but is sometimes used for ECN.

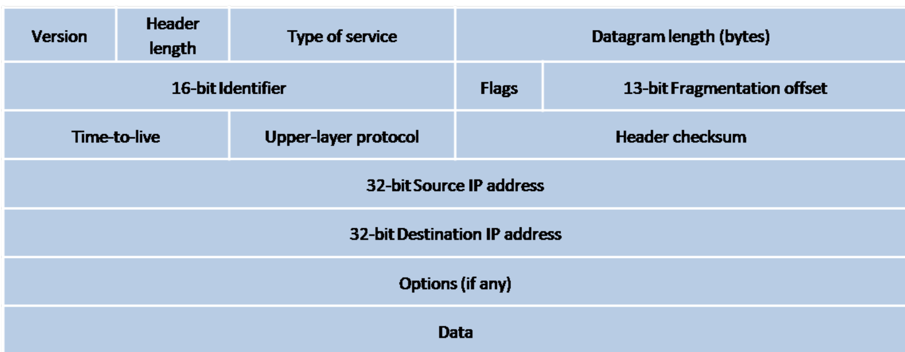

Fig. 3. The datagram format for IPv4.

5.2 IPv6

The main reason to why IPv6 [20] is being developed and deployed is due to that the addresses space in IPv4 is too small and soon there will be no addresses left to use for new devices [21]. However, this is not the only thing that is being changed in the protocol. The rest of this section explains the most important parts of IPv6 and compares them with IPv4 when possible to.

The header of IPv6 (Figure 4) is 320 bit (40 bytes) long. It has no optional fields and the size are therefore constant. Instead of optional fields such as in IPv4, IPv6 uses so called extension headers which are placed between the standard IPv6 header and the header of the upper layer protocol. It is generally only the end node that is allowed to look at the extension headers, which reduces complexity and time consumption in every router along the path. The only exception to this is if the extension header *Hop-by-Hop Options* are used, then every router along the path has to parse it and take appropriate actions. The *Next hdr* field is used to specify if there are any extension headers.

Traffic class specifies a priority of the datagram being sent, it has the same function as the *TOS* field in IPv4. But, since IETF now recommends that this field shall be implemented, it means that it probably will be easier to implement some kind of QoS-control.

If the *flow label* [22] is used it indicates that packets with the same flow label and with the same sender and receiver belongs to the same flow. When a router discovers a flow it could treat it in a special way. It is however not yet specified in what way.

The *Hop limit* field specifies how many routers the packet may pass until it should be ignored. When a packet passes a router, the router decreases the *Hop limit* with one. When the field reaches zero the packet is destroyed. This field is has the same function as the *TTL* field in IPv4, but is specified in number of hops instead of seconds as in IPv4.

The *Fragment extension header* may also be interesting regarding QoS. It is used when the packet is too big to be sent in one Maximum Transmission Unit (MTU). If a packet is too big it is divided into several pieces. Each piece of the fragmented packet is small enough to fit into one MTU and the task for the *Fragment header* is to keep track of the fragmentation. The new aspect of fragmentation in IPv6 is that the splitting is only performed in the source node, and the merging of parts is done in the receiving node. In IPv4 every router along the path to the destination may fragment the packets if they do not fit into one MTU. This could be very ineffective since parts of packets could be fragmented several times along the way if routers have smaller and smaller MTU's.

5.3 IP in mobile networks

A mobile device is per definition a device that is carried around in different places, and may change its location very often. Whenever a mobile device is used to connect to the Internet, one problem is that it might have to change its connection point during a session. Gulliksson and Lindström [1] addresses two

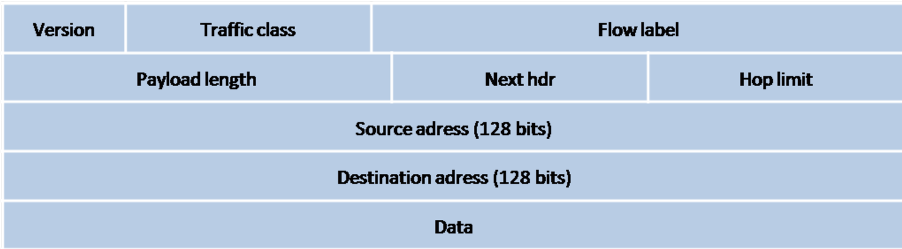

Fig. 4. The datagram format for IPv6.

different existing methods to solve these problems. Their examples are Mobile IP [23, 24] and Cellular IP [25]. However, when having to use these protocols, additional delays and possible points of failure are introduced that makes it even harder to maintain a good QoS. But, it is essential to use a protocol that takes care of this, otherwise the user will lose the connection every time she gets too far from the initial connection point.

6 QoS improvements using IPv6

This section looks at the different details of IPv6 and shows how they will improve QoS for mobile television. It is divided into subsections where each subsection discusses one property of IPv6, mentioned in Section 5.2.

The QoS properties that are looked at are stated in Section 3 and 4 and are listed here again; Loss, Delay, Jitter, Throughput, Error rate, Mobile resources, Link Quality, Mobility and Buffering. In the end of each subsection there is a list with all QoS measurement that is improved by the IPv6 property of that section.

Not all of the QoS properties will be affected when changing to IPv6. Properties that are not affected; link quality and error rate.

6.1 Traffic class

The traffic class field in IPv6 will in itself not improve QoS compared to IPv4 since IPv4 has the TOS-field, which is practically the same as the traffic class field. The big improvements in this area are found when and if, there will be agreements on which standards to use. With standards it is meant both the classification of the type of data, and also which service that will maintain the QoS, e.g. DiffServ or IntServ.

Affected QoS properties: None at the moment.

6.2 Flow label

The flow label will, if implemented correctly by routers, be useful for many different aspects of QoS. Since it defines a certain flow of packets, the routers may handle each flow in a separate way and thereby prioritize multimedia flows. Still, the routers cannot allow multimedia flows to have the highest priority all of the time, then the other packets never get through.

Loss will decrease due to the fact that multimedia flows gets prioritized in the queues and the packets more rarely are loosed. Delay and jitter decreases because of the same reason and the packets will get through faster. The other QoS properties will generally not be affected when introducing the flow label.

According to the RFC of the flow label [22], the lack of flow label in IPv4 makes it hard to determine existence of flows. A flow is in IPv4 defined using source address, source port, host address, host port and transport protocol.

Affected QoS properties: Loss, Delay, Jitter and Throughput.

6.3 Fixed size of header

Because of the fact that the header of IPv6 has a fixed size and has less number of fields, it requires less computational power to process, compared to IPv4. This is especially good in mobile systems, since mobile devices have many constraints regarding their resources. Also delay decreases and throughput increases since it takes less time to process the headers in each router.

Worth noticing is that the header of IPv6 is larger than the one of IPv4 which is a disadvantage of IPv6 since it introduces more overhead on the network. On the other hand, this extra weight is impossible to remove because the new longer addresses are needed. However, research in the area of large headers has been performed and a solution with header compression has been presented. With this method it will be possible to decrease the header down to 4-5 bytes for UDP/IP headers [26].

Affected QoS properties: Mobile resources, Delay and Throughput.

6.4 Removal of header fields

Disregarding the addresses that are bigger in IPv6, the header size is decreased with 4 bytes compared to IPv4. Also the number of fields is decreased from ten in IPv4 to only six in IPv6. The advantage is that it is easier for the routers and hosts to parse the packets, leading to saved resources for the mobile devices. Reduced header size also increases the throughput and decreases the delay in the network since there are less overhead in each packet.

Affected QoS properties: Mobile resources, Delay and Throughput.

6.5 Fragmentation

The new fragmentation technique in IPv6 will help to improve throughput since it makes an optimal fragmentation of the packets already at the source node. The number of fragmented parts will therefore not be more than necessary. It is important to keep the number of packages to a minimal since every part of the fragmented packet needs an extra header. Extra headers will introduce more overhead.

Also the total delay for each packet will be decreased since the routers along the path between the source and the destination will not need to fragment packets, this produces less overhead and requires less processing. The same reason will decrease jitter and buffering times as well.

Whether or not fragmentation is needed in mobile television is however not known since there are little information about packet sizes of mobile television and MTU's in mobile networks. Palmkvist [3] at TRE did not know the required packet size for mobile television.

Affected QoS properties: Throughout, Delay, Jitter and Buffering.

7 Conclusions

This article has identified QoS properties for mobile television from the network-layer point of view, and then investigated how IPv6 will improve these QoS properties as compared to IPv4. It was shown in the article that several aspects of QoS will be improved, but there are still many things that have to be implemented in IPv6 to get all these benefits (see Section 6).

As it turned out, most QoS properties for mobile television are also valid for almost every mobile multimedia streaming service (see Section 4). From one point of view this is good, since it gives a hint of what might be expected from IPv6 for the general multimedia streaming service. However, the downside with this is that is not possible to see how IPv6 will improve QoS for mobile television in specific, and thus the uniqueness of the article is somewhat lost.

To sum up, there are many properties regarding QoS for mobile multimedia services that will improve when switching to IPv6. But one question that arises is if IPv6, by itself, will be enough to handle the evermore increasing traffic on the mobile networks, since more and more multimedia applications are being developed and used on the network. Maybe MBMS (see Section 2) will be able to help with this problem, but that is still to be discovered.

8 Acknowledgments

I would like to acknowledge Jonas Palmqvist at TRE for answering some of my questions regarding mobile television.

References

1. Gulliksson, H., Lindström, J.: Ljud och bild över nätverk. Studentlitteratur, Lund, Sweden (2004)
2. Kurose, J.F., Ross, K.W.: Computer Networking. Addison-Wesley, Boston, Massachusetts (2005)
3. Palmkvist, J.: Mail conversation (2008) Personal communication with Jonas who is in charge of technical issues for mobile television at TRE.
4. Computer Sweden: Tv 2.0 räddning för mobil-tv (2008) `http://computersweden.idg.se/2.2683/1.149082/`, accessed 2008-04-29.

5. Knoche, H., McCarthy, J.D.: Design requirements for mobile tv. In: ACM International Conference Proceeding Series; Vol. 111. (2005) 69–76
6. DVB Project Office: This is the official mobile tv website maintained by the dvb project office (2008) `http://www.dvb-h.org/`, accessed 2008-04-29.
7. RAPA: dmb—digital multimedia broadcasting (2008) `http://eng.t-dmb.org/`, accessed 2008-04-29.
8. Postel, J.: User Datagram Protocol. RFC 768 (Standard) (August 1980)
9. Postel, J.: Transmission Control Protocol. RFC 793 (Standard) (September 1981) Updated by RFC 3168.
10. Schulzrinne, H., Rao, A., Lanphier, R.: Real Time Streaming Protocol (RTSP). RFC 2326 (Proposed Standard) (April 1998)
11. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard) (July 2003)
12. Dixit, S., Wu, T.: Content networking in the mobile Internet. John Wiley and Sons, Hoboken, New Jersey (2004)
13. Ruiz, P.M., Garcia, E.: Improving user-perceived qos in mobile and wireless ip networks using real-time adaptive multimedia applications. Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on **3** (2002) 1467–1471
14. Chalmers, D., Sloman, M.: Survey of quality of service in mobile computing environments. Technical report, Department of Computing, Imperial College, London, SW7 2BZ (1999)
15. Leber, M.: Global ipv6 deployment progress report (2008) `http://bgp.he.net/ipv6-progress-report.cgi/`, accessed 2008-05-08.
16. Postel, J.: Internet Protocol. RFC 791 (Standard) (September 1981) Updated by RFC 1349.
17. Almquist, P.: Type of Service in the Internet Protocol Suite. RFC 1349 (Proposed Standard) (July 1992) Obsoleted by RFC 2474.
18. Ramakrishnan, K., Floyd, S., Black, D.: The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (Proposed Standard) (September 2001)
19. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Service. RFC 2475 (Informational) (December 1998) Updated by RFC 3260.
20. Deering, S., Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard) (December 1998) Updated by RFC 5095.
21. Hinden, R.M.: Ip next generation overview. Communications of the ACM **39**(6) (1996) 61–71
22. Rajahalme, J., Conta, A., Carpenter, B., Deering, S.: IPv6 Flow Label Specification. RFC 3697 (Proposed Standard) (March 2004)
23. Perkins, C.: IP Mobility Support for IPv4. RFC 3344 (Proposed Standard) (August 2002) Updated by RFC 4721.
24. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. RFC 3775 (Proposed Standard) (June 2004)
25. Comet Group in Columbia University: Cellular ip (2008) `http://comet.columbia.edu/cellularip/publications.htm`, accessed 2008-04-29.
26. Mikael Degermark, Mathias Engan, B.N., Pink, S.: Low-loss tcp/ip header compression for wireless networks. In: International Conference on Mobile Computing and Networking. (1996) 1–14

Application development on portable devices with seamless network handover

David Jonsson

Department of Computing Science
Umeå University, Sweden
dit03djn@cs.umu.se

Abstract. This paper gives an introduction into different wireless networking techniques and the workings of a seamless handover technique. Guidelines and suggestions regarding application development and design of portable device applications are addressed. Different aspects of designing applications for portable devices, such as video streaming applications and login procedures are also discussed. The paper highlights the importance of usability and ease of use when implementing applications supporting seamless roaming on portable devices. It should be seen as a source of inspiration for designing mobile device applications with roaming in mind.

1 Introduction

With an increasing number of wireless networks and mobile devices equipped with a central processing unit, memory and storage capacity equal to or better than computer hardware comes a higher definition of portability. This change occurring from the early 1990's, in terms of being able to move the device from different places, has created the ability to use it in different environments where parameters such as connectivity might change in time. Keshav [1] claims that the number of mobile devices will increase in the future and will become a more dominant source for Internet usage. With an increasing number of mobile devices using the Internet, the future will require a higher demand for Internet sites and applications capable of handling mobile code and mobile agents [2].

There are obvious problems when users maintain mobility in a wireless environment. Users, whilst mobile, want to be able to stay connected when switching between different networks with their portable devices. Switching between networks manually gives the user a disconnection time of around nine seconds. This will interrupt the connection thus creating applications to be interrupted as well. Designers creating applications for mobile devices need to increase their knowledge of such events to be able to design applications with seamless handover in mind. The handover technique makes it possible for the link layer of the device to switch between networks with a low latency interruption.

Researchers have developed support for mobile devices so they can switch between different networks, i.e., WiFi and WiMAX without high latency interruptions [3]. This type of seamless handover has been shown successful and some basics behind the technology will be discussed and introduced. Many researchers have focused on WiFi and WiMAX, which are among the more popular used network standards. Therefore, this paper will begin with a short description of the two and how they function. An introduction to the media independent handover technique will be discussed and how to use this technique in combination with the mentioned network standards. In order to determine how application developments for portable devices are made, some basics behind design and articles about the design of graphics user interfaces will also be addressed. Existing research within improved mobile Internet usability and the usability from different types of current networks will be compared. How improvements have been made in mobile usability in forms of performance and satisfaction will be shown. Using seamless handover technique with wireless networking poses obstacles regarding buffering/quality of service when using e.g. video streaming applications over networks. It also poses difficulties handling user verifications and login procedures when switching between different networks in being able to verify the actual user and device. How an application developer can solve these two issues will be highlighted and appraised. Further suggestions regarding design guidelines to develop applications for portable devices with seamless handover in mind will also be discussed.

2 Wireless Networking

2.1 Quality of service

Quality of service (QoS) is a measurement of what the end-user is experiencing from the specific service. Instead of defining QoS as a set of different service qualities stitched together it needs to be divided into QoS for every specific layer in which a system is built up from. The result is what the end-user will see and that is a collaboration of all the different layers QoS. QoS on the application level is based on the end-users expectations, demands and experience from a specific service. An example where an end-user might feel the need for a pause during a video stream where the video will not actually stop until many seconds have passed could be a service with low QoS, since users will most likely experience the time elapsing from action taken to action occurring is below an acceptable level. Application developers can decide to implement different application level QoS depending on what type of quality the end-user is willing to pay. Depending on the quality of a video stream the throughput will differ. QoS on the system level is based upon actual hardware and the collaboration between them. A high QoS on the system level will lead to a good collaboration between hardware parts such as cpu, memory and graphic processing unit etc. According to RFC 2216 in Gulliksson et al. [4] the definition of QoS on the IP layer is: "The nature of the packet delivery service provided, as described by parameters such as achieved bandwidth, packet delay and packet loss rates".

For the application developer designing a video streaming application service the need for basic knowledge about QoS on different layers is significant. Streaming video is very demanding since it can be based on either a real live feed or a delayed data stream with a buffer from a sender to the receiver. Buffering ensures quality of service, if fluctuations over the network take place. With a buffer the sender can even loose contact with the receiver during a short amount of time without the end-user even noticing. Video streaming is also very hard to copy, as no actual file is being saved on the receiving end which makes the copying process more difficult [4].

2.2 WiMAX

Wireless Internet connections can usually be separated into two different types of wireless connections. The first one is called fixed wireless broadband. It can be seen as the major competitor to other fixed wired broadband solutions such as xDSL among others. The other type is called mobile broadband and this type has the basic fundamentals: portability, nomadicity and mobility. In other words the connection has to be able to work in different locations and therefore not only be able to work as a fixed wireless solution.

WiMAX (worldwide interoperability for microwave access) technology is a wireless networking type which tries to be both a mobile and a fixed wireless solution. According to Fourty et al. [5] WiMAX has a theoretical data rate of up to 70 Mb/s and a coverage area of up to 50 kilometres.

One of the basic strenghts from WiMAX is the possibility to use different frequency allocations. This in return gives different data transfers, coverage among others depending on the frequency used. When speaking of WiMAX it is consequently easy to divide it into Fixed and Mobile WiMAX.

Two of the greater differences between them lie in the frequency used and the ability to be mobile. When comparing the coverage area with solutions such as 802.11 standards it is obvious to understand the strength of WiMAX. Since 802.11 standards are able to cover an area of only up to 100 feet inside buildings and less than 1000 feet outdoors.

Fixed WiMAX was not designed as a mobile solution, but mobile WiMAX on the other hand supports mobility to an acceptable degree. Compared with other solutions such as 3G, it lacks a lot of mobility since 3G is a technology developed from the beginning to work as a wireless mobile solution [6].

2.3 802.11 standards

People often talk about WiFi when they actually mean a 802.11 connection. At this time of the paper there are many different 802.11 standards developed and in use, such as 802.11a, b, g and just recently the n standard. 802.11a was the first of them to be standardized. Since then all the others have evolved and just like the wired ethernet standards they have reached higher connection speeds with time. What most of these different 802.11 standards have in common is that they work in the frequency allocation of 2.4 GHz.

When comparing the different standards and their capacity to handle speed the difference between them is significant. For further insights in different frequency allocations and theoretical data rates see Table 1, which shows a more detailed comparison of the different wireless standards.

Table 1. Comparison of different wireless network standards [5].

Name	Standard	Data rates	Max range	Frequency
RFID	ISO 10536 and ISO 14443	106 Kb/s	3 m	several
Bluetooth	IEEE 802.15 1	2 Mb/s	100 m	2,4 GHz
UWB	IEEE 802.15.3	Up to 50 Mb/s	10 m	
Zigbee	IEEE 802.15.4	250 Kb/s	10 m	2,4 GHz
Zigbee	IEEE 802.15.4a	20 Kb/s	75 m	0,9 GHz
Wi-Fi	IEEE 802.11b	11 Mb/s	100 m	2,4 GHz
Wi-Fi	IEEE 802.11a	54 Mb/s	30 m	5,5 GHz
Wi-Fi	IEEE 802.11g	54 Mb/s	100 m	2,4 GHz
Wi-Fi	IEEE 802.11n	320 Mb/s	30 m	2,4-5,5 GHz
WiMAX	IEEE 802.16a	70 Mb/s	50 km	2,5-3,5-5,8 GHz
MBWA	IEEE 802.20	1 Mb/s	100 m	$<$3,5 GHz

What users need to consider when using wireless connections of these types is that the more users using a single access point the less capacity they get since they have to share the total amount of capacity from the access point. Therefore the connection speed is often far less than the given maximum capacity an access point can deliver.

An important feature with the new 802.11 standards such as g and n is that they are backward compatible with the earlier ones such as a and b. It is therefore possible to use a newer standard client and be able to compromise on speed capacity but still be able to be flexible and use several different access points. If a slow client with the a standard connects to an access point with higher capacity all clients will be affected and drops down to the lowest standard in the network. This has a disadvantage but has proven to be solved where newer wireless access points possess the ability to run in a some what of a mixed mode where they are able to handle a large amount of different types of standards without dropping down to the lowest one. Thus the higher capacity clients receive better speed performance.

Local area networks of different 802.11 standards are either built up used in ad-hoc or access point mode. An ad-hoc mode router works as in a user kind of perspective. Receiving data only passing it on forward, whereas an access point router instead handles all the traffic from every single user and makes sure it reaches the right receiver.

Application developers designing applications with the intent of sending the data wirelessly needs to know the major differences between wired and wireless networking, making it possible to ensure a quality of service i.e. QoS for the application. When using the handover technique with wireless networks such as 802.11 it is important to know the differences and assuring the possibility to roam seamlessly between available 802.11 networks. The QoS of a wireless network is harder to implement and therefore needs to act differently than wired connections. By implementing the CSMA/CA method where CA stands for collision avoidance the QoS of a wireless network is increased making sure wireless devices does not interrupt each other. When the CSMA/CA method is implemented the clients and access point listens to traffic and the clients will try to send wireless information when no other traffic is being sent. If a collision is detected all connected clients wait on a random amount of time and tries to send again. If it is still being used by others it continues waiting a random amount of time and keeps repeating this procedure until it can start sending without any other clients interrupting [7]. With many users connected to the same access point router using applications/services such as video-on-demand, it will be sensitive to these interruptions possibly making the video stream to have data losses which at the end will make the video to reduce in quality for the end-user.

For an application developer designing applications using the handover technique it is useful to know the difference in parameters whereas data rates and max range might influence the choices of application network compatibility. Ensuring QoS for the whole system and thus ensuring it to roam with less problems occurring. According to Gulliksson et al. [4] implementing an application/service with the intent of video-on-demand in Digital TV standard mode will demand a throughput of 5 Mbps between the service provider and end-user. Inspection of the table above will show that some of these standards are unable to offer this type of service since the data rate is not enough, an application developer therefore needs to have this in consideration when designing.

2.4 802.21 media independent handover technique

The potential problems with handover techniques as mentioned earlier in the introduction needs to be solved. The largest problem being the actual latency time disconnecting the device. How it further makes it harder for application developers to develop applications switching between networks without any data loss during the switch procedure. Choong et al. [3] introduce a proof of concept that media independent handover techniques make seamless networking possible. When roaming seamlessly between different wired and/or wireless networks it is important to have an intelligent connection-monitor layer being able to monitor changes in a network such as status, availability and signal strength parameters. These parameters ensuring the connection being kept alive. The connection-monitor should, when changing, be able to notify the application layer of a device that changes are going to be made. By notifying applications the connection-monitor layer will be able to reduce the actual handover time latency and ensure that a user can roam seamlessly. The IEEE 802.21 is trying to standardize a set of methods and procedures that enables media independent handover across what is often referred to as mixed networks. They have specified three media independent services - event service, command service and information service.

Figure 1 shows the media independent handover and its services inserted between the IP and Link Layer. The event service is retrieving information regarding parameters from the link layer i.e. link status and delivers the dynamic changes to the upper layers which then perform the appropriate actions based

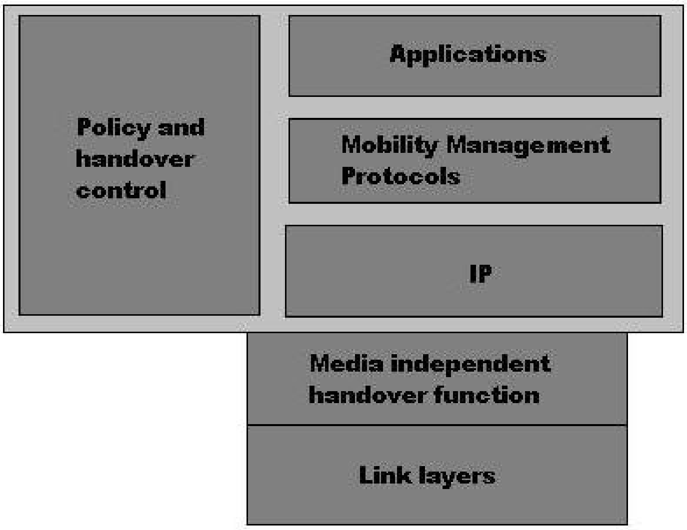

Fig. 1. Shows how the media independent handover layer is inserted [8].

on the information retrieved. The command service sends commands given from the upper to the lower layers. Connection decisions is an example where the upper layers decide and command service gives the command information to the link layer which then takes the appropriate action. The Information service holds network information and stores parameters regarding the different available networks. It handles the information which is crucial for being able to make intelligent handover decisions [2].

During the use of the media independent handover technique it is very useful to use a fast, efficient and continuous implemented handover protocol. With a primary purpose of being responsible for the actual connection being kept alive when switching between access points and base stations. Chai-Keong et al. mention the design issues that need to be considered when designing a handover protocol [9]. Exploiting the technique of radio hinting is one of several methods for retrieving seamless handover that has been proposed by Keeton et al. [10] and discussed by Chai-Keong et al. [9]. The idea of the method is to use the signal strength as an indicator for network connectivity. Depending on the changes in strength the handover can be able to act intelligent and trigger a handover when i.e. a degradation of signal strength is shown, thus be able to sustain connected without any high latency handovers occurring.

When switching between different networks Choong et al. [3] describes various models and go in-depth on two interesting techniques. Interesting since they manage to keep a short amount in milliseconds of loss in connectivity. Break-before-make and make-before-break are two proven models that are efficient in different ways. Break-before-make disconnects from the current network before it connects to the new. The key benefit from using it instead of the make-before-break is the ease of implementation. Make-before-break has the advantage of being able to connect to the new network before tearing down the old which gives a potential shorter amount of loss in connectivity but will be a bit harder to implement.

3 Usability, feedback and understandability

When creating network based applications on portable devices it is important for the application developer to be able to develop user friendly applications. A well implemented application in terms of efficiency and performance is not enough, it also needs a user friendly design in order to achieve popularity among end-users. Knowledge about usability, feedback and understandability is therefore essential for developers.

Norman [11] highlights two fundamental principles when designing for people. The first principle is to provide a user with a good conceptual model. The designer has his own design model or conceptual model and the user makes up his/her own user model when interacting with the system. If the user's model does not match the designer's conceptual model problems using the system might arise. The conceptual model does not have to be complicated as he states, a pair of scissors has a good conceptual model and most users know how to use them. A good conceptual model has to be able to give the user a good clue what the effects might be of using a certain thing, it also has to give information regarding errors. If a user makes errors due to misinterpretation, the user should be able to determine what will happen if doing so. The second principle is to make things visible. Combining these to principles and the designer has a good principle of design for understandability and usability [11].

Norman continues with the seven stages of action as design aids. By using seven questions and trying to answer them he comes up with four different points which all can be included in one or more of the seven questions [11]. Fig. 2 below shows the seven questions as a valuable aid for designers since it provides a good basic checklist of questions to ask for ensuring that the Gulfs of Execution and Evaluation are bridged.

How easily can one:

- **Determine the function of the device?**
- **Tell what actions are possible?**
- **Determine mapping from intention to physical movement?**
- **Perform the action?**
- **Tell if system is in desired state?**
- **Determine mapping from system state to interpretation?**
- **Tell what state the system is in?**

Fig. 2. A figure illustrating the seven questions which can be seen as an aid for designers [11].

- **Visibility.** A user should be able to tell just by looking at a device in which state it is in and also be able to determine the alternative actions that can be made.

- **A good conceptual image.** The designer should provide the user with a good conceptual model where consistency and coherency are keywords.
- **Good mappings.** The system should provide good mappings in forms of relationships between actions and their results. Between different states and what is visible.
- **Feedback.** The user should always receive full and continuous feedback during the usage of a device [11].

Nielsen [12] has a slightly different definition when it comes to usability. He speaks of usability and its five attributes:

- **Learnability.** A system should be easy to learn and use so the user quickly can start working with it.
- **Efficiency.** The system should also be efficient, so the user in a fast pace can use the system and be effective.
- **Memorability.** Users which have taken a break from using a system should be able to easily remember how to use it again whenever they want to start using it again. With a high memorability the user therefore does not have to start all over again, but can restart using a system where it ended last time.
- **Errors.** If a user makes an error it should be easy to fix them and major errors should not be able to occur.
- **Satisfactory.** Finally the user should feel satisfied by using the system. They should be able to enjoy and like to use it [12].

4 Mobile device application design

4.1 User interface design

The user interface enables interaction between the end-user and the system. Satisfactory according to Nielsen [12] is essential and a user interface developed without user satisfactory might lead to user unsatisfactory which can be devastating.

A company called UIQ Technology have a document containing design guidelines for application developers and designers who are developing applications for them. UIQ Technology is developing their own user interface called UIQ to some of the worlds leading mobile phone manufacturers. Their user interface is designed and tested with the Symbian OS, which makes it interesting since they are designing their own user interfaces for mobile devices. Many of their suggestions and recommendations can be implemented outside of their own development, in more general terms of mobile design and application development on mobile devices. They start by discussing the importance of performance, how it should always be insured. Applications and complexity can grow with time and that is the reason why performance needs to be very good at all times.

Modal dialogs are good to use in an application since they prevent users from doing anything else until the user has taken an action regarding the actual dialog. An example of a useful dialog is the “confirm” when a user is trying to delete a file in a Windows environment. By using this type of dialog the user will reduce the risk of doing something unintended, such as deleting a file by mistake by hitting the wrong button on the keyboard. Modal dialogs prevents errors from occurring and by combining with good mapping and feedback this type of dialog gives the end-user a rich user experience.

Notification messages are a one way asynchronous communication from the device to the user. When a notification message is shown they override the current action based on the urgency of the notification. An incoming call is a good example of a notification message with high urgency. It will therefore appear in the foreground with an accept/decline call action and this type of notification message overrides most of the other. When the call has been accepted and the conversation is over the user should be redirected to the last action. Without any memorability the system will be unable to redirect the user to the last action, making it more difficult for the user to return. By redirecting the user automatically it can further prevent errors from occurring unlike when a user manually needs to return by interacting with the user interface.

Battery power notifier is another high urgency notifier which can be implemented as a notification dialog with an "OK"-button or a more easy one with only a visual notifier telling the user by blinking or in a combination with sound that the current battery power is low. A battery notifier needs to give feedback in an efficient way preventing the user from getting the battery to run out. Eventually causing data loss due to failure in saving the present work for instance.

Busy messages should be used whenever it is necessary. It disables the user to do any inputs on the device while processing. The busy message should always give a reason: "Loading..." etc. rather than "Busy" or "Please wait". If possible the message should display how long time the process will take. A progress dialog is therefore useful to provide the "cancel"- and "stop"-button so the user can regret an action whenever he or she feels it is necessary. Busy messages needs to be implemented with feedback such as the cancel or stop button, by doing so it reaches a higher level of efficiency since the user can abort something he or she is considering to be slow. By chance the user might have pushed a wrong button and error prevention is therefore also inserted by the cancel and/or stop button assuring the user redirection to the last action when pressed.

Errors may occur and because of that it is of utmost importance to inform the user in a meaningful and precise way. The application or user should never have to feel or appear as unintelligent. In many cases a well described error gives the user a good clue to what the real problem might be. By using meaningful information in the error message confusion and making the user anxious will be reduced [13].

4.2 Platform independent user interface

Eisenstein et al. [14] discusses the importance of user interfaces and how they should be able to work on several different computing platforms. The basic idea is that a user interface should be able to work in many different contexts and still be able to preserve its consistency and usability. They also talk about how a designer has to specify how a user interface should be able to adjust itself whenever the context changes. They claim that without any abstract description of how a user interface should be developed it is more likely that the development of the interface will take a lot of time, undergo and discover errors and might even become a failure.

MIMIC is a well rounded user interface modeling language which Eisenstein et al. [14] choses for their design technique because of its three relevant model components: platform model, presentation model and task model.

The platform model describes the many different platforms and devices which are executable on the system. It handles the information regarding every different platform available for the user interface as an element and every element has attributes which describes the different features and constraints for the specific element. By using this model at a run-time level it may be sensitive to sudden changes that may occur in the use. The platform could recognize a sudden change of a factor in the user interface and act in an appropriate way.

A presentation model describes all the visual information regarding the user interface. It handles the hierarchy of the different visual aids such as windows, buttons widgets, sliders, boxes, their selection and placement. The different widgets have a classic dividing between the AIOs and CIOs. The AIO: an abstract interaction object, which is platform-neutral. Each AIO has different CIOs: concrete interaction objects which can only be run on a specific platform. The AIO/CIO dividing is done so that an easy way of running the user interface on many different platforms is possible.

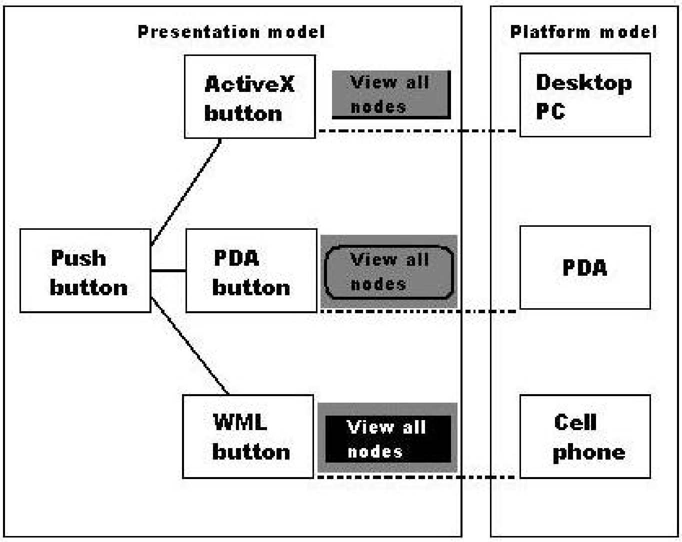

Fig. 3. Shows how the different CIOs are built up by a single AIO and mapped to the different platform models [14].

Figure 3 below visualizes how MIMIC uses the "Push Button" acting as an AIO and divides the CIOs into three different types depending on which device the user interface is running on. Based from the linkage between the platform and the presentation model MIMIC tries to identify which to use in a specific situation.

The task model is a structured representation of the different tasks a user may be able to do on the application. The model contains information regarding the different tasks, subtasks and and what will happen if they are used [14].

4.3 Device and platform independent user interface

Applications able to work on several independent devices and platforms has potentially a larger user market than device and platform specific applications. Enabling independence leads to a higher demand for compatibility which can result in more design considerations for the developers. MANNA is a hypothetical software application that handles many of the posed difficulties when designing portable user interfaces. MANNA is a multimedia application that has the ability and must be able to run on many different platforms, it also has to be able to utilize and be used simultaneously over the Internet. It was developed for geologists, engineers and the military. When users are using MANNA the application tries to utilize its different strengths depending on which platform it is running on. The workstation is an example where the user interface does not have any significant constraints. Nevertheless it does not have the ability to be mobile and is therefore only used when a user is not on the move. When the user uses a mobile device for the application many different constraints arises such as display size, resolution and batter power. It must be able to take advantage of the mobility of the device and still be designed in a good manner so that the user interface does not pose any difficulties when used. When switching between the different platforms such as a cellular phone, palmtop computer and the workstation it also switches between different user interfaces based on their specific strengths and weaknesses [14].

Since the devices have different capabilities, the selection of an appropriate presentation structure is very difficult. To solve this Eisenstein et al. [14] suggests that the creating of an intelligent mediator agent will solve this difficulty. The mediator agent should be able to determine which screen resolution a specific device have and then choose an appropriate presentation structure for the device. The mediator agent should also be able to handle the changing of a resolution of a device at run-time and make it more dynamic to changes, which leads to an easier integration of new devices and their platform models in future design. By implementing the mediator agent the user only needs to specify the device actual screen resolution and the agent then finds the correct mapping to the presentation structure [14].

4.4 Mobile devices as a source for information

Buchanan et al. [15] carried out a HCI-based survey of over 100 people where they wanted to report on a large scale WAP based user centered survey, as a result of the lack of information from earlier studies based on mobile computing and usage of WAP as an Internet source. User centered where information where gathered from how users instead of using a traditional workstation had to use a small screen device where information might not always be implemented for the small screen. They were able to identify three main usability problems when using WAP technology on the Internet: screen size, navigation and site structure, and input methods.

The most criticized of the three problems is the screen size. How bigger screen sizes can lead to better usability than small ones and small screens can lead to a lower enthusiasm when used. A parallel with the very popular Post-it notes are discussed, how a Post-it note is very common when small amounts of information will be used. Whereas larger information is not satisfactory to use on a Post-it note cause of the limited space available. The small screen device could easily be compared to this example and therefore be very effective when a short amount of information is gathered. But when a large amount has to be visualized a larger screen size is more preferable.

Navigation and structure is another setback where WAP has received lots of criticism. In early stages many sites were not designed for smaller screens, the use of a small screen was harder due to the fact that there were too many selections available and too many moves to be made for a user to reach its goal. A user study showed that many users got tired only after one click and gave up reaching their goal.

The input method is also a problem when using WAP Internet usage since the user has to make more inputs in forms of clicks, scrolls etc. when searching through the pages.

By enhancing the search when using WAP they managed to present three alternative interfaces where the user does not have to make as many efforts in navigation and inputs as earlier. The first one called 'Horizontal' scrolling and the key feature lies in a set of headings and only the words that fit across the screen will be shown for every heading. When scrolling over a specific heading the truncated characters of the heading will show the complete sentence. Vertical scroll method is the second alternative where it is showing all the headlines in the same way as the first with the exception that no text is truncated and therefore the user has scroll a bit more to read the whole heading. Lastly 'Paged method' is where the headings are broken down into different segments of pages and three headlines are shown at each page. Users scroll pages when searching through headlines instead of scrolling through headlines. All three methods have their own advantages and disadvantages. The first method shows small fragments of all headings and leaves the user with less scrolling than the others. The second makes the user read each heading in an easier way but then it demands a more extensive scrolling. The third changes the ordinary scrolling to scroll through pages, but can on the other hand add more complexity to the interaction.

From their analysis of WAP problems, their surveys and other work they have identified a set of design guidelines for WAP usability [15]:

1. Well designed pages that can handle the issues when using WAP as an Internet source device is essential. The information should have focus on valuable content and the use of keystrokes or text entry should be minimized. Simply converting an already existing web page to try and adjust it phone platforms will most likely fail.
2. Try and use a simple hierarchy in the navigation such as the phones own menu structure so the user will feel a similar way of navigating.
3. Try and reduce the vertical scrolling by simplifying text that needs to be displayed.
4. By simplifying navigation and by changing text input with other types of interaction the number of keystrokes might be reduced which you can expect the users will prefer.
5. By combining theoretical and empirical evaluation further insights might be provided [15].

5 Discussion

The application developer designing an application with seamless handover needs to implement a good solution to the login issues that might arise with the handover technology. An end-user should firstly, when using a portable device, get feedback from available networks and possible login procedures from them. When selecting a specific available network the user interface should show a modal dialog of the current network with possible user name and password. “Connect” and “Cancel” buttons should further ease the user experience when connecting. When connected to a network it is important to show the user with high visibility that he/she is connected to a network. A notification message or icon could therefore be implemented visualizing the status of the connection, making sure the user knows what kind of state the device is in. The end-user should be able to save current settings from the network in such a way the user does not have to retype the user name and password every time when connecting. When roaming between two networks information regarding new user name and password must be visualized. The end-user needs feedback from the device telling it to type in a potential new user name and password for the network it is trying to connect to. If both networks have saved login settings a good way of visualizing the actual roam between them could be implemented simply by informing the user with an icon notifying the end-user about the handover. When switching between the networks during load a busy message showing the end-user “Loading” or “time left” can be useful. If errors occur during the handover an explanatory reason why it failed should notify the user what went wrong. The error having high urgency should appear above all other actions and redirect the user to last action taken when informed. Visualizing for the end-user what the device is currently connected to is significant, since QoS for different wireless networks might differ.

To be able to construct a video stream application able to cope with seamless handovers, the need for adaptation is of great importance. Depending on the data rate from the specific network the application needs to adjust the video quality e.g. resolution, frame rate etc. This adaptation would preferably be automatized, not visualized for the end-user making it useful to while visualizing the change of network also is able to show if quality in video is reduced/raised. This adaptation needs to be implemented at run-time level since video streaming is very delicate and used in real time environments. The reduce/raise will ensure the application having a QoS of which it is always trying to optimize the video quality depending on the achievable data rates from networks. The end-user should always be able to reduce the quality manually in personal settings in order to change the current allocation of data rates from the device. This personalized setting should be implemented in a settings menu overriding the optimization if possible even when switching to a new network. With a low latency handover regarded in milliseconds, the end-user is able to switch without the possibility of even noticing an interruption during the actual change in video playback mode. Depending on the quality of the video stream the throughput will differ, making the implementation of the buffer size to adopt depending on network capacity as well. A high video quality with high throughput must have a larger buffer size than one with low video quality, in order to guarantee some sort of acceptable degree of QoS for the end-user.

Satyanarayanan [16] characterizes the fundamental challenges when dealing with portable computing into four constraints. One of the constraints is the disability to be as resource-rich in comparison to a static device. Depending on the size, weight and power elements the cost for processing speed, memory and storage will be set. Security and vulnerability of loss or damage are constraints, where a portable device runs a potential larger risk of being stolen or damaged due to the fact that it is mobile and therefore taken from different locations a numerous amount of times. Wireless connections are more sensitive to changes in the environment which will lead to variable performance and reliability. A portable device relies on charged batteries which will reduce in performance with time and these concerns will influence the hardware and software. Satyanarayanan highlights the importance of a portable device being able to adapt and react dynamically to different external changes. He claims the key to mobility lies in the adaptation.

A designer developing applications with the intention of using it with portable devices needs to start by defining what the purpose of the specific application is. What type of application that is going to be done, the customer and developer needs to have the same mental model over how the system needs to be made. A need for understanding how the potential end-users might use the system/application. Need for understanding how the users will use the system/application will further ease the development. A reasonable assessment needs to be made on the basis of how they will use the mobile device with the application.

How shall data be sent wirelessly? Since wireless data transfers are more easy to sniff and hijack, securing the data could be necessary? Will the application use the TCP or UDP-standard for data transportation? The difference in these two might cause a slight change in development since they are using different features when being sent. A TCP-packet is more secure since it is able to resend itself if data packets are being lost on the way where UDP is unable to use this feature but on the other hand gives the application a higher performance in speed when implemented. VoIP is a common type of application using the UDP-standard for data transportations. If packets are being lost on the way this type of application will more sensitive to the losses and lead to what might be a noticeable performance reductions in quality of visual and audible information.

How the users will use the device in combination with the application will further simplify the development. Will they use the device and always or often be on the move or will they use it more stationary? If the user's goal consists in having a large portion of movement while in use the application needs to have an intelligent handover technique, which makes it possible to roam seamlessly without any high latency setbacks in a way that it can assure the quality of the connection in parameters such as performance. If the users have special needs or demands of what kind of network which is preferable the application needs to act intelligent and be able to adapt based on the users own preferable settings. An example where the user has an agreement with a specific Internet Service Provider and therefore wants to use it as often as possible despite the fact those other available networks might appear to be accessible. The application might need to dynamically change due to the facts that a network of preference is being used despite better ones available with higher speed performance. With a lower performance in speed the application might need to adjust and optimize itself based on the preferences given by the chosen network. This adaptation needs to be done without the user noticing any significant changes in quality of the service.

By informing the end-user when adaptations are made due to the changes in the environment in an easy and user-friendly way is an essential part. This user-friendly and understandable information needs to be implemented in the user interface with a large portion of consideration which can become a challenge for the application developer. Depending on the type of the change the end-user needs to be given the appropriate feedback and suggestions on how to eventually take actions which will lead to a better solution.

6 Summary and conclusion

802.11 and WiMAX are two commonly used wireless networking techniques. In combination with a seamless handover it is possible to develop applications able to roam seamlessly between wireless networks. Design guidelines regarding video streaming applications and login procedures with portable devices using wireless networking with seamless handover has been addressed. Other guidelines are also discussed and proposed in more general terms of application development for portable devices using handover techniques. To be able to further state any of these guidelines and suggestions mock-ups, prototypes and eventually some sort of survey would prove invaluable. Surveys proved invaluable in order to be able to further investigate this process which can be seen as future work.

References

1. Keshav, S.: Why cell phones will dominate the future internet. SIGCOMM Comput. Commun. Rev. **35**(2) (2005) 83–86
2. Kotz, D., Gray, R.S.: Mobile agents and the future of the internet. SIGOPS Oper. Syst. Rev. **33**(3) (1999) 7–13
3. Choong, K.N., S., K.V., de., C.F., Y., L.A.L., C., M.: Sip-based ieee802.21 media independent handover—a bt intel collaboration. BT Technology Journal **25**(2) (2007) 219–230
4. Håkan, G., Jakob, L.: Ljud och bild över nätverk. Studentlitteratur, Lund, Sweden (2002)
5. Fourty, N., Val, T., Fraisse, P., Mercier, J.J.: Comparative analysis of new high data rate wireless communication technologies "From Wi-Fi to WiMAX" (2005) http://ieeexplore.ieee.org/servlet/opac?punumber=10434.
6. Andrews, J.G.: Fundamentals of WiMAX. Prentice Hall, Indianapolis, USA (2007)
7. Roshan, P.: 802.11 Wireless LAN Fundamentals. Cisco Press, Indianapolis, USA (2004)
8. Mussabbir, Q.B., Yao, W.: Optimized fmipv6 handover using ieee802.21 mih services. In: MobiArch'06: Proceedings of first ACM/IEEE international workshop on Mobility in the evolving internet architecture. (2006) 43–48
9. Toh, C.K.: The design & implementation of a hybrid handover protocol for multimedia wireless lans. In: MobiCom'95: Proceedings of the 1st annual international conference on Mobile computing and networking. (1995) 49–61
10. Keeton, K., Mah, B.A., Seshan, S., Katz, R.H., Ferrari, D.: Providing connection-oriented network services to mobile hosts. In: MLCS: Mobile & Location-Independent Computing Symposium on Mobile & Location-Independent Computing Symposium. (1993) 8–8
11. A., N.D.: The Design of Everyday Things. First MIT Press edition (2001)
12. J., N.: Usability engineering. Academic press (1993)
13. UIQ Technology: Uiq style guide (2002) www.uiq.com.
14. Eisenstein, J., Vanderdonckt, J., Puerta, A.: Applying model-based techniques to the development of uis for mobile computers. In: IUI'01: Proceedings of the 6th international conference on Intelligent user interfaces. (2001) 69–76
15. Buchanan, G., Farrant, S., Jones, M., Thimbleby, H., Marsden, G., Pazzani, M.: Improving mobile internet usability. In: WWW'01: Proceedings of the 10th international conference on World Wide Web. (2001) 673–680
16. Satyanarayanan, M.: Fundamental challenges in mobile computing. In: PODC'96: Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing. (1996) 1–7

Design of GUIs and GUI animations with markup-based documents

Robert Johansson

Department of Computing Science
Umeå University, Sweden
dit03rjn@cs.umu.se

Abstract. With today's rapid development of competitive products for multiple devices it is important to offer a GUI which the users favor in usability, look and feel. This article describes possibilities with markup-based documents defining GUIs and GUI animations. Four different paradigms of GUI definition are described. The paradigm of focus is the Document-based Approach, using separate files to define GUI and program logic. A clear example compares an animation implemented in both the Code-based paradigm realized in C#-code and the Document-based Approach realized in a separate XAML markup-based document. The separation of the GUI and code enables the GUI designer and programmer to work in parallel and can thereby save time. The other two paradigms, GUI-oriented Documents and the Document-oriented Approach, are described but not further studied. GUI development tools to create the different parts in GUIs, e.g., graphics, GUI components and animations, are listed and compared. The use of animations in GUIs is motivated and the principles needed when designing them explained. The separate markup-based documents seem easier to understand, allowing more people create GUIs and GUI animations.

1 Introduction

The first idea of an interface using a screen to visualize information came from Engelbart [1] in 1962. He describes a building architect working in something very similar to current CAD-programs. The architect uses a display to see the objects he is designing; the input is achieved through a keyboard and some other devices. Since Engelbart introduced his idea of a Graphical User Interface (GUI), GUIs have taken form in many shapes and devices. From Smalltalk development environment UI, Apple Lisa GUI, Windows 1.0 and Amiga Workbench to more recent versions of Windows and Mac OS [2]. We see GUIs almost everywhere in mobile devices such as PDAs and mobile phones, medical equipment, ATMs etc. With the formation and popularization of Internet and WWW during the 90s many people without programming knowledge became familiar building GUIs using the markup-based HTML. Further on DHTML enabled direct manipulation on the web and Style sheets helped separating content and style of the presentation [3].

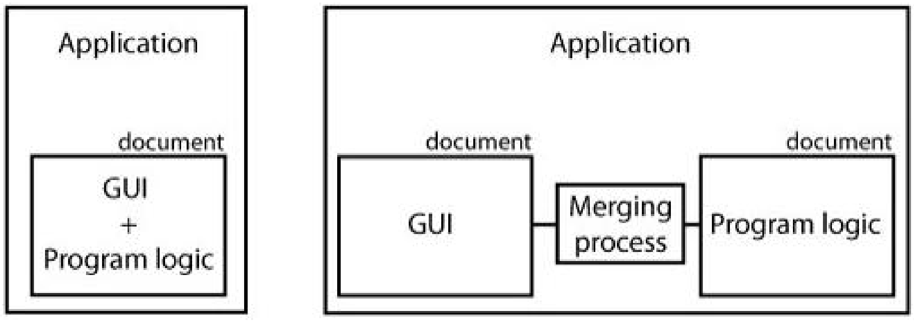

Fig. 1. The GUI and the program logic in the same document or in two separate documents. The merging process is needed when using two separate documents to merge together the GUI and program logic.

The last ten years, it has been increasingly popular to build GUIs defined by a markup language, most often XML. The GUI components are written in a separate markup-based document and connected to another document containing the program logic (see Figure 1) [4]. Although there existed markup-based GUIs, Draheim et al. [4] claim that none of these were generally accepted when they wrote their article in 2006. Since Windows Vista was released during late 2006 and early 2007 this may have changed due to the fact that Vista uses markup-based documents to construct the GUI and a separate code file for the program logic [5]. The approach using a separate markup-based document to define the GUI is the focus for this article.

Another trend when designing GUIs is the use of animations to give the user a richer experience. A richer GUI is less static and monotonous and thereby more exciting to use. Most research concerning animations was performed in the first half of the 90s. Back then smooth transitions between visual states in software were visualized as discrete steps and smooth transitions not an issue [6]. In current software, animations are common elements and seen in most desktop operating systems, cell phones and other handheld devices. Animations aiming to get user attention are common in web advertisement.

The goal for this article is to describe the possibilities with markup-based GUIs and further GUI animations. Markup-based techniques for GUI construction will be compared to other, pros and cons will be discussed. This is interesting from both a human computer interaction and an economical perspective. Developments of products with GUIs are in a rapid speed today. Creating GUIs with good usability and rich experience for these products are important to get many and satisfied users. If GUIs defined in a separate markup-based document is a technique that can help gain these benefits will be better explained here.

2 GUI Paradigms

Draheim et al. [4] identifies four paradigms for building GUIs.

Code-based GUIs The "traditional" way to build a GUI where program code is written to define, link and add properties to the GUI components. The code defining the GUI can in some cases be mixed with code for other purposes which make it harder to create, maintain and understand. For the ability to create a code-based GUI one must have to know the program language or at least have enough programming skills to be able to understand and learn it. Draheim et al. [4] continue saying that there is no way to describe the GUI in an acceptable complex level only by looking at the code, making it very hard to analyze.

GUI-oriented Documents. The second paradigm identified by Draheim et al. [4] is GUI-oriented Documents. They are GUIs defined in documents containing static elements with basic possibilities for user interaction. The static elements do not allow the same level of interaction as real GUIs (for example code-based) but the documents are easier to understand and analyze. One example is HTML-documents where navigation is possible through hyperlinks. The documents can also contain images and mathematical formulas, and in later versions forms with buttons and other GUI controls.

The Document-based Approach. Draheim et al.'s [4] third paradigm is the one this article looks further into because it is usually realized with markup-based documents. Even though code-based GUIs have full-blown ability for interaction and user experience, they have shortcomings. They require a higher level of knowledge for understanding and possess the risk to mix GUI definition with program logic. Also the GUI-oriented documents above are constrained because they are static. One solution to these problems is the Document-based Approach. The document represents the GUI and its components while a separate file contains code for program logic. The program logic makes it possible to add more advanced functionality. When a component in the GUI is activated by a user interaction an event occurs which executes the desired program code.

The Document-oriented Approach. Draheim et al. [4] propose in their fourth paradigm the idea to integrate final appearance with development. The editing and display of GUIs should look exactly the same and are accomplished with the same tool. The difference between display and possibility to edit the GUI is set by the user access rights. A visual GUI editor will not generate any hidden definition of the GUI that can cause problems when running the application. Tailorability is easily made possible by setting access rights on components that shall be tailorable in size or similar. An implementation of the idea in this paradigm is not presented by the authors [4].

3 GUI development tools

A general problem in software development is that programmers produce poor graphics while graphical designers are poor programmers for the desired functionality[7]. An application needs both. The programmer is superior in creation of the functionality behind the GUI, while the graphical designer is superior in creation of the GUI presentation; layout, graphics and animations. Coming from different environments, used to different tools and having different focus can cause problems when the graphical designer and programmer shall work together. GUI development tools should support their cooperation.

When designing the GUI of an application the following must be taken into account [7]:

- The physical aspects of the device, for example screen resolution and display size.
- The information that will be shown on the screen.
- The context; by who, where and when will the application be used?
- The efficiency of use by practicing guidelines and benchmarks of usability, but also end-user knowledge and characteristics.

There are lots of visual tools for editing GUIs. Through a WYSIWIG-interface (What-You-See-Is-What-You-Get-interface) the designer can compose the layout and the GUI code or document is automatically generated. However the generated code or documents are often poorly structured from the programmer's point of view. For example the layout can be non-dynamic, the code or document badly structured and not supporting understanding [8]. In fact tools that automatically but unpredictably generates GUI definitions have in history not been received well by programmers [9].

The possibilities to create GUI graphics are often limited in software development tools. In contrary are tools for creation of graphics not supporting the creation of functionality. Graphical designers prefer tools best suited for creation of graphics. But when the graphical designers work is moved and connected with the program logic the programmer might have to reproduce the graphics if they do not match the programming environment. This does not support an iterative development process and causes duplication of work. The application might also need to be re-compiled when a change is made in the GUI [7]. Chatty et al. [7] implemented two tools to avoid these problems and gain higher functionality for graphics in software development. TkZinc, a graphical engine which better allows programmers and graphical designers to work together and IntuiKit, which practice the Document-based Approach and helps separating the GUI from the program code by among other using XML-based SVG vector graphics. The use of these two tools reduced the programming effort, by not having to re-code graphics and overall project length when the programmer and GUI designer could work in parallel [7].

Since the GUI animation is a part of the presentation, the knowledge of a graphical designer are best suited to create them. Tools for creation of animation is often better suited for the graphical designer, e.g., Adobe Flash [10], but not as powerful for a programmer when creating functionality. Other tools, using the code-based paradigm, defines the animation in code which requires the not best suited programmer to do it. But there is tools, e.g., Microsoft Expression Studio [11] supporting the creation of GUIs, graphics and animations defined in a markup-based document. This document can be connected to program logic in several different programming languages. Tools like this, practicing the Document-based Approach, can hopefully make the programmer and graphical designer understand each other better, resulting in better GUIs.

4 Markup-based GUIs

The separate markup-based GUIs are all realizations of the Document-based Approach. Different techniques have different focus and capabilities. Some techniques, e.g., Mozilla XUL removes the GUI-limitations of HTML web documents and delivers full GUIs rendered from the web [4]. Other techniques, e.g., Microsoft XAML, enables GUI creation for desktop applications and supplies the creation of GUI components, graphics and animations defined in a markup-based document. A third possibility when using separate markup-based documents for GUIs is device independence, an important aspect with all different devices today. A device independent user interface defined in a markup-based document is presented by Phanouriou [12] and called UIML.

4.1 XML

The use of XML can be found in a wide variety of applications. Techniques for markup-based documents defining GUIs most often use XML as foundation. The Extensible Markup Language (XML) is a restricted form of the Standard Generalized Markup Language (SGML) developed by a working group formed 1996 [13]. A XML document contains entities which contains data. The XML document itself has to be valid towards the XML specification [13]. The document can then contain any markup tags, defining elements and attributes the user wishes. The design goals for XML are among others to support a wide variety of applications, to have ideally zero optional features, be human legible and reasonably clear; the design shall be formal and concise and make it easy to write programs that access it [13]. If these goals can be fulfilled they fit a markup-based GUI that is easy to understand and works with various applications.

4.2 Advantages of markup-based GUIs

The advantages of markup-based GUIs are many. First of all the user interface and program logic are separated, the main idea of the Document-based Approach. This enables the GUI-designer and programmer to work on their material without interfering with each other [4, 8]. It is hopefully also a benefit when their work shall be merged together.

The ability to only describe the GUI can avoid a lot of problems. It enables different types of users to have different user interfaces [4, 3]. The same program logic can be used for multiple interfaces, for example one wizard for beginners and an advanced mode for experts [3]. It is also easier to fit the application in different hardware, for example screen sizes [4, 3]. A stand alone GUI document, which can fit many devices, would avoid the risk developing for a product that suddenly goes off the market and losing all the money spent [3]. Another thing is that the program logic can be replaced with another programming language that fit another platform but the GUI document can still be the same [4, 8]. Of course this requires compatibility between the two parts. It is also possible to isolate the GUI into an already installed part, for example the Mozilla Amazon Browser, which runs in a web browser tab. No need for higher user access rights are required to install the GUI and program code is loaded only when needed. Code at the moment not needed is accessible online.

The documents are also very easy to edit because they can be opened in an ordinary text editor [4]. This gives the opportunity for the graphical designer and programmer to open the same document in an application best suited for their work. Further there is necessarily no need for compilation for changes in the GUI to give effect, as there are for code-based GUIs [4]. A compiled program is unusable when there is a single-bit error, but with a single-bit error in a markup-based document the damage is small and the application still usable [12].

5 Animations in GUIs

This article looks deeper into how animations can be defined in markup-based documents. The knowledge of a graphical designer is better suited to create GUI animations for presentation than the knowledge of a programmer. In some cases this is not possible because the animations are defined in a code-based document. Good is, that the animations as well can be defined in a separate markup-based document (practicing the Document-based Approach), making it easier for a graphical designer to accomplish GUI animations.

5.1 Why animations?

Animations can be used to reduce the users cognitive load by making on-screen objects movement and change more realistic. An animated change can be easier to perceive because it retains an objects shape during the change [14, 15]. It is easier to enjoy a GUI with objects behaving more smoothly and realistic [14, 15]. Mistrust for objects behaviors makes it harder to accept when shown on screen [16]. An animation can also in a preferable way change the users focus to recently created areas of importance [14]. Through a smooth and gradual change of the screens appearance the animation can help avoiding sudden changes [16]. Cartoon animation, adding movement or similar between the changes of two static states, is an inspirational technique to gain a more realistic appearance of GUI objects, e.g., growing menus, objects blurring when moved and bouncing when reaching the end of a region [6]. A benefit when using animations during manipulation of objects is that they look more realistic than just a virtual object. For example minimizing a window looks more realistic if it is animated when shrinking than if it just disappears. Chang and Ungar [6] claim that the animation occurring during an object operation gives the object solidity by adding the feeling of solidity, not only the look of it. Finally, Thomas and Calder [16] conclude that while the use of animations in GUIs can give users directions and catch their attention, the designer should reconsider when using animations in situations where it is distracting by dragging more attention to the actual animation than the actual task.

5.2 Principles of animations

This section explain the content necessary to create animations to show what are needed to define them in either a code-based or markup-based document. Naps et al. [17] identifies some graphical primitives required for drawing such as point, line, arc, ellipse, circle, rectangle, triangle and text. With these primitives and combinations of them pretty much anything can be drawn. Attributes of the primitives or groups of them can be modified during animation by e. g. rotating, scaling and moving them (see Figure 2) [16]. Also attributes as color, transparency and shape can be changed during the animation [18]. These changes from one state to another are sometimes called transitions. The transitions do not necessary happen at once but gradually [19]. Key times control different values of attributes over time in an animation. These key times are often named key frames [20]. The key frames are sometimes not enough, for example if the change from one value to another is supposed to start slowly, go fast in the middle and end slowly [6]. To manage this, splined interpolation is a good technique [20] (see Figure 3). This can be an advantage for more realistic effects [19]. Splined interpolation can be suitable when animating a ball bouncing; the movement slows down when the ball is hitting ground and accelerates when the ball goes towards the sky. To obtain full experience when animating in a GUI it is important that GUI components as buttons, lists and images can be animated as well. This is possible in the XAML separate vector and markup-based document

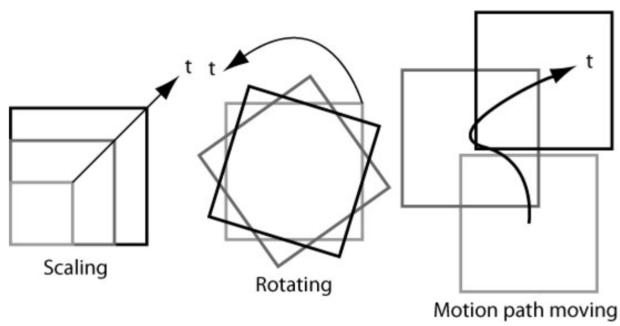

Fig. 2. Rotation, scaling and motion path movement carried out on the graphical primitive rectangle. Time(t) proceeds from the weakest shade of gray to black.

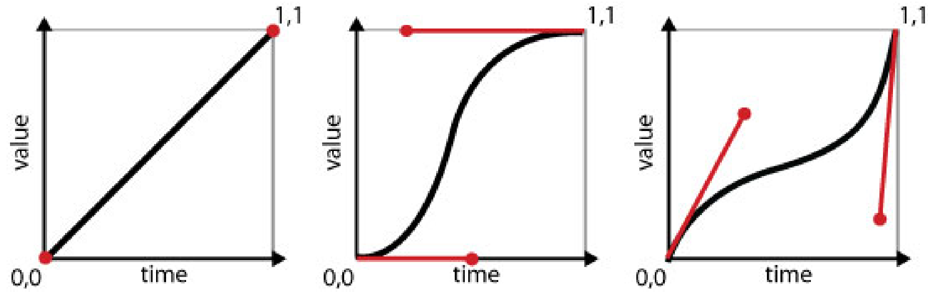

Fig. 3. Splined interpolation: The speed of the animation is set by the slope of the curve. The left graph shows uniform pace where the speed is the same during the whole animation. The middle shows a slow-in, slow-out animation and the right a fast-in, fast-out. The effect is controlled by the parameters represented by the dots [20, 19].

(practicing the Document-based Approach) [5]. Another technique using vector based graphics is SVG. SVG defines two-dimensional vector graphics in XML (XAML can also handle 3D-graphics [5]). Graphical primitives define the look of the graphics, for example a circle with radius 30 units and fill color blue, in contrast to raster-based graphics where a matrix of pixels with different colors create the blue circle. The biggest benefit of vector graphics is that it is scalable, meaning that it can change size without losing any quality or information (compared to raster-based graphics). This enables the graphics to easier fit different screen sizes and be magnified for visually impaired. Notice that current displays are raster-based, which means that vector graphics have to be pixelized before they are viewed [21]. The scalability of vector graphics, both within their own elements and in the whole without losing any information, fits animations very well. With a markup-based document, e.g., XAML it is possible to trigger animation either in code or in the markup-based document [18]. This enables a designer to run his/hers animation without any knowledge about the program code, making it easier for the best suited graphical designer to create and test the animation.

Separate markup-based documents is well suited to define vector based graphics and animations. Transitions, key times, splined interpolation values and attributes of objects is easy to set and organize. The example in next section will show this.

6 Comparison of animation techniques

To illustrate the difference between an animation defined in code and markup, an animation (see Figure 4) has been created using Microsoft .NET. Techniques used are the markup-based document XAML and C# code. XAML is an example of the Document-based Approach and C# code is an example of the traditional Code-based paradigm. In the animation a window moves from the

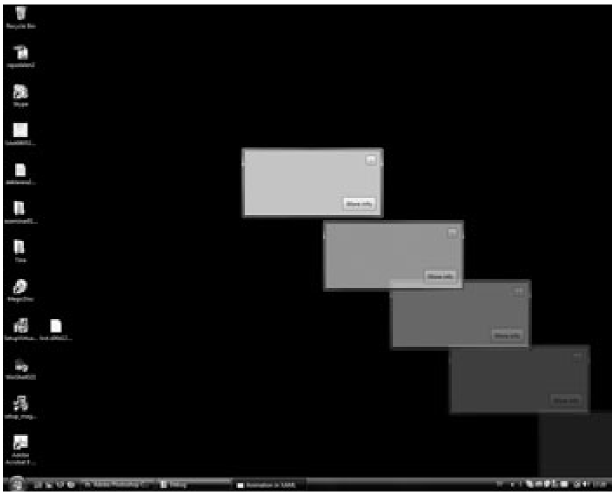

Fig. 4. The animation made to compare code and markup. A GUI Window moves from the bottom right to the middle of the screen.

bottom right of the screen to the middle of it. Notice that the GUI elements are created in other parts of the document/code. Their attributes are set in markup and code before the animation can run. Figure 5 and 6 show a part of the animation defined in the compared techniques. First of all we see the `Window_Loaded` method and `Window.Loaded` event which handles the event triggering the animation. After that we see a `DoubleAnimationUsingKeyFrames` with attributes for the value the animation is changing into (300), how long time the animation should appear (3 seconds) and which spline values (0.7, 0.0, 0.2, 1.0) to use. The animation `DoubleAnimationUsingKeyFrames` is in its turns placed in a `Storyboard`. The `Storyboard` controls the attributes for which GUI element that should be animated (the Window) and which property that should change (the Top property or distance from display top). There is also a similar animation changing the Left property of the window to make it move to the vertical center of the screen. This code shows the basic principles of an animation. Other animations work in a similar way. An event triggers the animation and a number of attributes values are changing over a specific period of time with possible spline values setting the pace. Comparing the C# code in Figure 5 and XAML markup in Figure 6 we see that the hierarchical structure of the document is easier to see in the markup. It is clearer that the `Storyboard` contains the `DoubleAnimationUsingKeyFrames`. Imagine having knowledge of HTML and web development and not in programming. Without collecting any specific measures, which one is easier to understand? Also imagine creating a bigger animation with many attributes and elements changing. Which document structure could handle this best? The XAML markup-based document is just describing GUI components, animations and possible events triggering the

```
private void Window_Loaded(object sender, RoutedEventArgs e) {
        DoubleAnimationUsingKeyFrames animationTop
        = new DoubleAnimationUsingKeyFrames();
         animationTop.KeyFrames.Add(
                new SplineDoubleKeyFrame(300,
                KeyTime.FromTimeSpan(
                        TimeSpan.FromSeconds(3)),
                new KeySpline(0.7, 0.0, 0.2, 1.0)));
        storyboard = new Storyboard();
        storyboard.Children.Add(animationTop);
        Storyboard.SetTargetName(animationTop, this.Name);
        Storyboard.SetTargetProperty(animationTop,
        new PropertyPath(Window.TopProperty));;
        storyboard.Begin(this);
        ...
```

Fig. 5. The animation changing the Top property in code. After the dots in the end are code to animate Left property as well.

```
<EventTrigger RoutedEvent="Window.Loaded">
        <BeginStoryboard>
        <Storyboard>
                <DoubleAnimationUsingKeyFrames
                  Storyboard.TargetName="animatedWindow"
                  Storyboard.TargetProperty="Top">
                        <SplineDoubleKeyFrame Value="300"
                           KeyTime="0:0:3"
                           KeySpline="0.7,0.0 0.2,1.0" />
                </DoubleAnimationUsingKeyFrames>
                ...
```

Fig. 6. The animation changing the Top property in XAML. After the dots in the end are similar tags for the Left property and then closing tags.

animations. There is no risk mixing these with the program logic as when defining everything in C# code. Also take a look at the way of assigning values. The XAML uses for example `Storyboard.TargetProperty=‘‘Top’’`, assigning the value Top with an equals sign. In the code a method is called to set the value to Top, `new PropertyPath(Window.TopProperty)`.

The implemented animation is simple but explains the basic principles when animating in a GUI and the difference between defining it in code and markup. The intention is to visualize the different techniques and raise thoughts about them.

7 Discussion

Developing GUIs and applications for multiple devices is made easier with markup-based GUIs. The scalability of vector graphics and separate GUI document are strong factors achieving this. HTML enabled more people to create their first GUI, markup-based GUIs might enable more people to create full blown desktop GUIs including animations. It seems like markup-based GUIs have many benefits compared to the traditional code-based GUIs. The ability for the GUI designer and programmer to work in parallel can shorten development time and save money. If the graphical designer, superior in creating good looking GUIs with high usability, and the programmer creating advanced functionality, can focus on the part they do best, HCI factors should reach higher levels.

The paradigms of GUI design [4] state four ways to define the GUI. The traditional and code-based have full capability but is harder to analyze and interpret. The GUI-oriented documents, e.g., HTML, are easier to analyze but have limited capability. The most important aspect of the third paradigm, the Document-based Approach, is that it separates the GUI from other parts of the application. This article has discussed how this can be realized with markup-based documents, but other ways are of course possible. A separate code document or another defining language can be used. The fourth paradigm is another way to define the GUI where the editing and usage are separated with user access rights. How this can be realized is a question not discussed in this article.

It is important that the connection between the document defining the GUI and the GUI components with the program code is easily achieved. It requires a GUI document, e.g., a markup-based one, which is easy to understand for both the GUI designers and programmers in order to enable them create and deliver good substance in between them. A, from both parts understandable, document would make it easier for a programmer to understand and create graphics, and easier for the graphical designer to understand the material it creates that the programmer use. A project not affording both a programmer and a graphical designer can hopefully get better even if a programmer makes the design. It is also easier for an application with high requirements on the GUI and less on the functionality to be built by a graphical designer with enough programming knowledge to create required functionality.

A good GUI document will avoid unnecessary work and save time. Saving time is also an achievement gained by WYSIWIG-editors. But also these have to define the GUI somewhere. Whether markup-based documents can make the automatically generated definitions less unpredictable can be further examined. A guess is that they are less unpredictable since they are easier to understand and analyze [4]. GUI development software enabling creation of GUIs, graphics and animations having a clear connection to the programming environment which the programmer use is important. Good tools exist, for example the Adobe suite for graphics and Visual Studio for programming, but not as many good combinations of them. Adobe has Flash to create vector graphics, animations and applications, but not in markup-based documents. Microsoft offers Expression Blend Studio for this, not fully as well-used as Flash but generates markup-based documents. Adobe Flash as GUI editor fits well for prototyping GUIs and animations, but is for the one who have tested harder to use when creating applications with more advanced functionality. Adobe might solve this with Adobe Flex [22] and MXML [23], a markup-based document to easier define advanced GUIs in flash. A question the future choice of developers will answer.

Markup-based documents can contain definitions for vector graphics and animations [5, 21]. Whether this is good or not can be further studied. But they enable developers having knowledge about markup-based documents to create and understand the definitions for GUIs, graphics and animations just by learning the individual elements defining them. A document containing all of these elements gets big quite fast in an application with large functionality. As a suggestion the GUI can also be separated in GUI components, graphics and animations. By placing these parts in separate documents even more work can be carried out simultaneously. It might also be easier for a specific animation or graphical component to be found and edited if the files are structured in a good way.

The use of animations can make GUIs behave more realistic and by this receive higher acceptance from the user [16, 6, 14, 15]. The animations can also catch user focus for important tasks [14]. Markup-based documents can define animations and vector graphics [21]. The scalability of vector graphics works well in animations. Vector graphics also fits well when creating tailorable GUIs and GUIs for multiple devices. Changes can be made in the GUI without having to recompile the program. This enables users to easier modify or tailor the GUI and animations as they want. When comparing techniques for animations one see that it is easier to see the hierarchical structure of elements in the markup-based document. Elements containing other elements are placed inside them. These parts are in code added to another one with an add-procedure or similar. The result is only as clear as one see it; further research interviewing programmers and graphical designers about the different techniques can be made. The possibility to trigger animation events from the markup-based document can be used by the designer in an early stage of the development process when building prototypes. The programmer does not need to trigger the animation, just add the advanced functionality that appears together with it.

8 Summary and Conclusions

This article describes possibilities with markup-based documents defining GUIs and GUI animations. We see that the possibilities are many. The separate document avoids mixing up the GUI definition with other parts of the application allowing the graphical designer and programmer to work in parallel which saves time and thereby money. Markup-based documents have a lower learning threshold than program code which should make the sympathy in between the graphical designer and programmer reaching a higher level. This understanding should improve HCI factors in general and make the products obtain more and satisfied users. It should also allow WYSIWIG-editors to generate less unpredictable GUI definitions and allow more people, e.g., with knowledge in HTML, to create full-blown GUIs.

Markup-based documents can define all of GUI components, scalable vector graphics and animations. All these factors are important when attempting to grant the user a rich experience. When comparing techniques we realize that it is easier to interpret the hierarchical structure of elements in the markup-based document. Further research comparing code and markup would give more clear answers. Clear is it that regular GUI animations can be defined in a markup-based document without specific programming knowledge, allowing more graphical designers to accomplish sophisticated GUI animations. Whether all these benefits come to be applied or not will be seen in the years to come.

References

1. Engelbart, D.: Augmenting Human Intellect: A Conceptual Framework. Stanford Research Institute (1962)
2. Reimer, J.: A history of the gui (2005) `http://arstechnica.com/articles/paedia/gui.ars`, accessed 2008-04-09.
3. Abrams, M., Phanouriou, C., Batongbacal, A.L., Williams, S.M., Shuster, J.E.: Uiml: an appliance-independent xml user interface language. In: WWW'99: Proceedings of the eighth international conference on World Wide Web. (1999) 1695–1708
4. Draheim, D., Lutteroth, C., Weber, G.: Graphical user interfaces as documents. In: CHINZ'06: Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction. (2006) 67–74
5. MSDN: Introduction to windows presentation foundation `http://msdn2.microsoft.com/en-us/library/aa970268.aspx` accessed 2008-04-14.
6. Chang, B.W., Ungar, D.: Animation: from cartoons to the user interface. In: UIST'93: Proceedings of the 6th annual ACM symposium on User interface software and technology. (1993) 45–55
7. Chatty, S., Sire, S., Vinot, J., Lemort, A., Mertz, C.: Revisiting visual interface programming: creating gui tools for designers and programmers. In: Proceedings of the 17th annual ACM symposium on User interface software and technology. (2004) 267–276
8. Bishop, J., Horspool, N.: Developing principles of gui programming using views. SIGCSE Bull. **36**(1) (2004) 373–377
9. Myers, B., Hudson, S.E., Pausch, R.: Past, present, and future of user interface software tools. ACM Trans. Comput.-Hum. Interact. **7**(1) (2000) 3–28
10. Adobe: Adobe flash `http://www.adobe.com/products/flash` accessed 2008-04-23.
11. Microsoft: Microsof expression `http://www.microsoft.com/expression/default.aspx` accessed 2008-04-23.
12. Phanouriou, C.: UIML: A Device-Independent User Interface Markup Language. PhD thesis, Virginia Polytechnic Institute and State University (2000)
13. Bray, T., Paoli, J., Sperberg-McQueen, C., et al.: Extensible Markup Language (XML) 1.0. W3C Recommendation **6** (2000)
14. Baecker, R., Small, I.: Animation at the interface. The Art of Human-Computer Interface Design (1990) 251–267
15. Donskoy, M., Kaptelinin, V.: Window navigation with and without animation: a comparison of scroll bars, zoom, and fisheye view. In: Conference on Human Factors in Computing Systems. (1997) 279–280
16. Thomas, B.H., Calder, P.: Applying cartoon animation techniques to graphical user interfaces. ACM Trans. Comput.-Hum. Interact. **8**(3) (2001) 198–222

17. Naps, T., Rössling, G., Brusilovsky, P., English, J., Jarc, D., Karavirta, V., Leska, C., McNally, M., Moreno, A., Ross, R.J., Urquiza-Fuentes, J.: Development of xml-based tools to support user interaction with algorithm visualization. SIGCSE Bull. **37**(4) (2005) 123–138
18. MSDN: Animation overview `http://msdn2.microsoft.com/en-us/library/ms752312.aspx` accessed 2008-04-14.
19. Hudson, S.E., Stasko, J.T.: Animation support in a user interface toolkit: flexible, robust, and reusable abstractions. In: UIST'93: Proceedings of the 6th annual ACM symposium on User interface software and technology. (1993) 57–67
20. MSDN: Key-frame animations overview `http://msdn2.microsoft.com/en-us/library/ms742524.aspx` accessed 2008-04-17.
21. Ferraiolo, J., et al.: Scalable Vector Graphics (SVG) 1.0 Specification. W3C Candidate Recommendations are available at `http://www.w3.org/TR` **2** (2000)
22. Adobe: Adobe flex 3 `http://www.adobe.com/products/flex` accessed 2008-04-23.
23. Wikipedia: Mxml `http://en.wikipedia.org/wiki/MXML` accessed 2008-04-23.

Using Macromedia Flash as a Requirements Engineering Tool

Emil Eriksson

Department of Computing Science
Umeå University, Sweden
dit03jen@cs.umu.se

Abstract. This paper presents the foundations of requirements engineering based on a literature study within the field of requirements engineering. In particular requirements elicitation is discussed and different techniques for requirements elicitation are presented. Prototyping is probably the most important tool for validating requirements collected in the elicitation process and should be made iterative throughout the project. Flash is discussed in this paper as a prototyping tool and this paper claims that Flash fulfills some of the aspects of modelling prototypes in requirements engineering. Flash is a dynamic and quickly-implemented prototyping tool with flexible capabilities. Flash can therefore be used throughout a project and serve as an elicitation technique, a specification document and for user-testing. However, to confirm these claims, case studies and documentation on a prototype in use, should be preformed.

1 Introduction

Failures in communication within a project can result in economical effects and this is not due to technical problems, but the result the involved actors cannot understand each other [1]. One of most important tasks when creating a software system is that the software should meet the demands of the end-users. It may be easy to declare, but how do we find out what the end-users, the customers and all people who benefit from the system actually wants? As Cooper [2] states, to describe what the needs of a system is, or what to build to meet the requirements of all the stakeholders is the hardest single part of conceptual work. The term system stakeholders are people that will be affected by the system and therefore have influence on the requirements [3].

Nuseibeh [4] describes the measurement of success in a software application to be how the product meets the needs that were intended in the first place. Furthermore, Nuseibeh states that to create or modify a software product by the requirements is a process. The purpose of the requirements engineering process is to identify the stakeholders and document their needs for analysis, create communication in the design process and support implementation.

Requirements engineering (RE) techniques are becoming more and more well-known and companies today are realizing the importance of documentation.

Standardizations such as ISO 9000 have helped to elucidate the importance of process documentation and standardization [5]. The international standardization organization has declared this standard to secure and satisfy the quality of requirements in the process but they are vague on how to establish the system requirements.

This paper presents the foundations of RE and the common techniques and methods in the application of RE, based in a system or a software process. The paper discusses Macromedia Flash and explores its usage as a RE tool. Section 2 contains the background on RE. Section 3 concerns requirement elicitation and techniques. Section 4 presents prototyping in RE and general prototyping techniques. Section 5 explains Macromedia Flash and provides information on related work in the terms of prototyping with Flash. Section 5 also contains a discussions on how Flash can be used as a RE tool. Section 6 concludes the material presented in this paper.

2 Requirements engineering

Bray [6] states that the usage of RE when writing a software system is of the highest importance. When creating a software system it is in the beginning we lay down the foundation on which the rest of the project rests on. The bigger the project is, the more important is the usage of RE. Bray makes the example of building a house. If a project were to create a small house to contain some lumber in our backyard, we start off with defining the requirements. The house is to be a certain size and position and suddenly we realize that we have made an error in creating the specification. But the problem can be fixed quite easily and cheap since the project is of a smaller size. If we were to say that we are building a big sky scraper and starting with the foundation, creating every floor and then found out that we had made a error in defining the requirements in the beginning. In fact the sky scraper where to withstand the pressure of sixty floors instead of forty, it is not easy to fix and going to effect to total cost of the project. Hence, the importance of RE increases as the project size increases.

Bray additionally declares the *problem domain* as being the focus of attention when finding the requirements, see Figure 1. The *solution system* is the system that will be created from the requirements process and therefore it is of great importance, especially in the later stage of the process. As seen in Figure 1, these two systems lay on top of each other and that represents the interface. The interface, of which explains how these two systems will interact with each other. Figure 1 also shows Bray's three principal areas of concern.

- **Analysis**, the task of analyzing the problem domain and the problems which exist in this domain.
- **Specification**, is a task that concerns the interaction between the systems.
- **Design**, which in it self is not part of the requirement engineering but concerns the internal mechanism of the solution system.

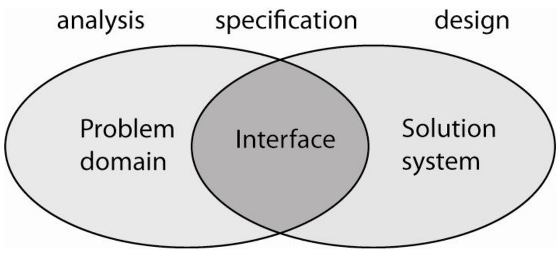

Fig. 1. The three principal areas of concern in the RE process. A reconstructed figure from Bray [6].

Determining the requirements is one of the first stages in a system development process. According to Sommerville and Sawyer [3], requirements are describing system behavior and systems properties or attributes. Bray [6] defines requirements as "the effects the client wishes to be brought about in the problem domain". These clients can be, as Nuseibeh [4] explains, paying customers, users and developers and the difficulty is that they can be numerous and distributed. Their goals cannot agree and thus stand in conflict of each other. It is recommended in the early stage in an RE process to setup a list of stakeholders, since they are needed to determine the requirements [3]. It is of importance that the concerned people in the process are involved at the right time and right expectations are addressed. If not it will lead to indirect and ineffective communication between the customer and the developer [7].

Bray [6] claims that requirements exist in different levels, one is defining requirements for the user another is towards general system properties. It can also be constraints on the system and on how the development should be executed. Bray also separates requirements in different types; Functional or behavioural requirements, performance requirements, design constraints, commercial constraints and other types of requirements. The functional requirements is basically the ordinary needs of a system, they describe how the system should behave. Performance requirements assemble the needs of how quickly, capacity, usability or how reliable the system must be. Design constrains are the truly non-functional requirements. They set the limitations on how the system or software is built, not how it works. In the ideal process there is no design constrains. Requirements can be separated in many types and it differs between many specialists but if the developer can separate these requirements, it can help with the structure and documentation of the process [6].

When speaking about RE, it refers to all the activities in the process of discovering, documenting and maintaining requirements for a computer based system [3] [7]. The usage of the term engineering in this context means that systematic and repeatable methods should be used to make sure the final requirements are fulfilled. Bray [6] declares: "Investigating and describing the problem domain and requirements and designing and documenting the characteristics for a solution system that will meet those requirements". Although the RE process is often described as an ordered series of activities ongoing that are executed, in real life situations it is performed iterative and in parallel.

3 Requirements elicitation

So how do we find the requirements, which the system should deliver to the stakeholders? Elicitation describes the activities performed for investigating requirements for the end-user and everyone else that has a stake in the system [3]. The activities of "collecting" the requirements are called elicitation, because many times the needs of a user are hidden knowledge and it is not as trivial as just gathering the requirements. The hidden knowledge can be explained by being the information that the analyst wants from a user. The information explains to be hidden since the user really does not know what they want, or they cannot explain it in sentences. Hence Bray [6] talks about latent knowledge as being the hidden knowledge that an analyst need to use requirements elicitation techniques to assemble. Elicitation may very often be the first step in RE [4]. According to Sommerville et al. it is a very important part in RE if the product is going to be accepted by the end-users. They also suggest that the following problems should be analyzed:

- Often system stakeholders do not know what they want of a computer system, except the basic and general demands. If they know what they want, they might have trouble explaining their requirements by themselves.
- Consumers of the product express themselves in terms of that they know; it is a so called implicit knowledge. It is the responsibility of the analyst to understand the full extent of the consumer requirements, and also to ensure that this information is delivered to all personnel involved within the project.
- Since stakeholders present different requirements and expressing themselves differently it is the responsibility of the analyst to separate the many sources and see which requirements are alike and which are in conflict.
- Political factors and differences in the organization can have effect on the requirements of the system.

Bray declares that it is of importance to check with the description of the project when deciding what information to collect. Further the analyst should specify which sources the information should come from, clients, pre-existing solutions and users, potential users of the new system and competitor products. All the work with requirements elicitation pays off in many forms. One clear result with a good performed elicitation is that it creates a common goal and vision, and conceptual software solutions between the user and the developer. The user benefits from the elicitation because they acquire a better understanding of their needs and constraints. With this enhanced knowledge they can efficiently evaluate solutions and understand the implications of their decisions. From a successfully elicitation the developers benefit because they create a high-level specification of the problem at hand [7].

Bray [6] proposes that there are as many techniques for elicitation as it is potential sources of information. Here are some of the techniques as Bray gives as an example of what the analyst can use to elicitate:

- **Background reading**. Not so often mentioned but this is an *everywhere techniques* used to present the engineer with preface to the problem domain.
- **Interviewing**. It is a technique to withdraw information from the users head and Bray states that this is the predominant elicitation technique. Interviewing has strengths as "broad band communication (verbal and body language which is readily supplemented by sketched diagrams, etc. As necessary)".
- **Questionnaires**. As all the techniques presented by Bray should questionnaires also be seen as a attachment to the technique of interviewing. To define the ultimate questions the analyst must pre-plan every question and cautiously phrase them to maximise understanding and minimize ambiguity.
- **Task observation**. It is a technique often used as a planned observation when presenting the test person for a task, maybe a prototype, by observing the users reactions the analyst can pickup clues on how good the prototype works. Task observation or task analysis is widely used in observing the interaction between human and machine interactions.
- **Ethnography**. This is also an observation technique. The analyst becomes intimately involved within the workplace situations and for this reason the analyst can achieve a deeper insight in the practices, problems and concerns for the specified project.
- **Use-cases and scenarios**. This technique describes a particular interaction between a projected solution system and the potential user. Use-cases are often involved in the later stage of the RE process.
- **Brainstorming**. In a free-thinking session with no room for constrains or criticisms a group of people generate new ideas.
- **Requirements stripping**. If an existing requirement specification document exists for the user extraction, or stripping, can be done to extract the individual requirements to a new document. This is done because the existing document has generally low quality.

There are always analysts who insist on using the very same technique in all situations, but it may not be sufficient in all aspects [8]. Hickey et al. presents that an analyst selects an elicitation technique in any combination of these four reasons; that it is the one and only technique they know; the analyst uses their favourite; a explicit methodology is used which prescribes a particular method at a particular time; and finally the analyst realizes which technique to used based on experience and intuition to be the effective one in the current circumstance [8].

Unified model of requirements elicitation

Hickey and Davis [8] present a unified model which highlights the nature of iteration in the elicitation process while the method transforms the current state of requirements and its specific situation to an increased comprehension, and a likely modified situation. They emphasise that all analysts who use this method will increase the understanding in the requirement elicitation process, a knowledge which is needed when performing elicitation. As stated before, throughout this paper it is important to know the problem and choose the appropriate technique for requirements elicitation in which Hickey and Davis model covers this aspect as well.

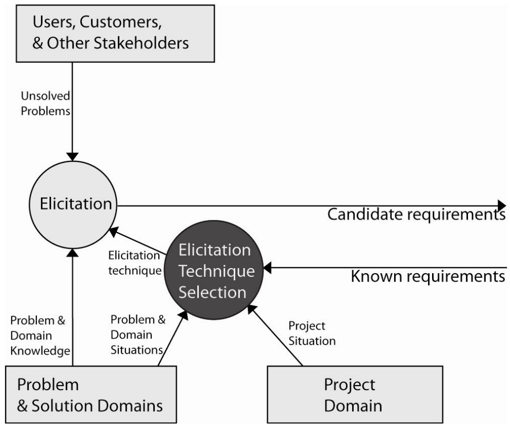

Fig. 2. Details of elicitation activities. A reconstructed figure of the model presented by Hickey and Davis [8].

As seen in Figure 2 is the selection of the elicitation technique implicated on problem, solution and project domain characteristics and also the current state on the requirements. Hickey and Davis further states that we have to know what type of requirements we are looking for, because different states of the requirement has consequence on the choice of the technique. The unified model of requirements elicitation is well documented and can be used as a tool for the in the requirements process.

Once the problem domain is clear and the requirements are collected for the system, a specification is needed. A specification describes how a system behaves, or as how Bray states it "meeting the requirements with designed functionality" [6]. In a system developing process this specification is documented and works as a foundation for design decisions, hence Bray suggests that this is of great importance and should function as a link through all the development process. The document should help all the stakeholders to validate the specifications before the construction phase. But Jakob Nielsen, a well known usability expert, states that it is of no use to involve a user in the design process by presenting abstract specification documents, the user will have a much better chance to understand concrete prototypes [9].

4 Prototyping

Hsia et. al. define prototyping, in the context of requirements engineering, as being the construction of an executable system model to "enhance understanding of the problem and identify appropriate and feasible external behaviours for possible solutions" [10]. According to Bray prototyping is one of the best ways to make sure the information from the elicitation process is valid. A prototype is created to describe the behaviour of a system described in the specification documents. The first prototypes are usually story-boards and paper-prototypes. It has been shown that presenting a system in form of images, instead of text, is convincing when controlling the behaviour of the system with all the stakeholders [6]. For a user-interface software system prototyping has been identified as a viable or maybe even necessary tool for defining and developing, a tool which reduces the risk by early focus on feasibility analysis, the identification of real requirements and removal of unnecessary requirements [7]. The usage of Flash as a prototyping tool for describing system behavior will be further discussed in Section 5.3 ("Flash as a RE tool").

The technique for creating models of prototypes in RE is separated by many authors into different subcategories. Bray [6] bring up the `representational modelling` and `behavioural modelling`. Where `representational modelling` models the appearance of a system by showing the systems output to a user and represents the system in a way that describes the exterior form. `Behavioural models` explains the function of the system rather than the appearance. The prototypes give details to the functionality according to different outputs that respond on users inputs. Nielsen defines a concept called vertical prototyping as being a prototype with less features and focuses on functionality [9]. The prototype becomes very narrow but can be used to test specific tasks on the depth to a test-user. The role of prototyping in RE is to model user-interfaces. This is of great importance especially if the interface is complex, critical and new [6].

4.1 Story-bords and paper-mockups

These techniques is used instead of using complex computer system prototypes, simple printouts of the system is presented in paper-form, or digitally, in a way that gives the stakeholders an understanding of the appearance of the system [6, 9]. The person who shows these paper-prototypes must have a good understanding of the system to show them in the correct order and to explain the system [9].

4.2 Dynamic representational modeling

One of the benefits of dynamic prototyping is that the developers can test certain aspects of the system in the prototype and only choose to implement these aspects in the prototype [6]. Bray explains dynamic representational modelling to be quite a new phenomenon in prototyping for RE. It is a rapid adaptation of software system prototypes which is used to illustrate the appearance and part of the behavioural of a system [6]. These prototypes can be classified in many categories and Bray summons them into four areas: `Exploratory`, when a prototype can serve as an aid to elicitation and refining requirement. An exploratory prototype is often called throwaway prototypes. `Definitive` prototype, when a prototype becomes a part of the specification. `Structural prototype`, when the prototype is used to estimate feasible idesign solutions. Structural prototypes contain many subclasses, one important is performance prototyping. The fourth class of Brays is `Evolutionary prototyping`. The prototype becomes the product by evolution by ongoing refinement during the software development project. Nielsen mentions a form of prototyping called interactive prototyping where the prototype is created as the project goes on. A certain test-user is involved to comment on the prototypes weak-spots. Nielsen continuous with stating that this is a good technique if an excellent programmer is used and the appropriate flexible interface construction system is in use [9].

4.3 Use-cases and scenarios

A use-case is often described as being the definition of the interaction between a potential user and the proposed solution system [6]. Uses-cases are often linear and a less complete technique for documenting the interaction. They are usually applied later on in the RE process to test the acceptance on user and designer to find the last requirements. The case is set up as a scenario and guides a user through certain situation in the solution system and records the user reactions. Use-cases fall in to the definition of `behavioural model`, by Bray, since use-cases does not show the appearance of the system. A problem with use-case is that the user may not understand the whole system, when not presented by the appearance. But if the case is designed correctly the user can carry out the task at hand. Hence use-cases is a technique which is more solution than problem oriented [6]. Nielsen describes scenarios to be a good way to narrow down functionally and appearance and focus on the usability. The scenario prototype is guiding a user to follow a certain path throughout the system and observation can occur by the developers [9].

4.4 Conceptual models

Norman argues that the operation of any device, i.e. a computer software; is learned further eagerly, the problems are easier and more accurately found, if the user has a good conceptual model of the problem at hand [11]. The idea of a conceptual model is that the model will present the user with a clear overview of the system. When we, as users, see a new product we create our own conceptual model and draw conclusions of our experience and the environment of the product. Hence, when speaking about a software development process, it is the designer's task to make a conceptual model understandable by the user and ideally it is the very same as the users. Conceptual models are a part of the design of mental models and Norman separates three different aspects when speaking of mental models; the design model, the user's model and the system image.

The design model is the designer's conceptualization of the task at hand. The user model is the model of how the user sees the assignment and develops explanations of the operation to control the system. The mental model is described by Norman to be the system image. The system image is the image which the developer tries to apprehend. Norman further states that if the developer fails to come up with a good conceptual model for those involved in a project the effect can be that conclusions are mistakenly drawn. In such case it is hard to predict the functions of the application and additionally to expect the unexpected [11].

5 Flash—Creating rich interactive user experiences

At the time of this paper many versions of the program Flash exist and Flash has gone from being a product deliverable from Macromedia as earlier versions as Flash 5, Flash MX, Flash 8 etc. and today is presented by Adobe in the CS3 package. Flash is described as being the environment that can offer a developer the opportunity to create rich, interactive user experience, on the web and mobile for applications [12, 13]. Moock [12] states that Flash can be used to create high-level abstraction with interactive content and applications. The internal properties of Flash are easy-to-use but for more advanced models Moock suggests the usage of scripting in Flash with Flash on scripting language called ActionScript. Flash is constructed of frames displayed on a horizontal line, the timeline. By using ActionScript the developer can control a Flash movie and play, pause, loop and easily jump between different frames to control the animations. Movieclips is an object that can be created within the main movie and each MovieClip contains its own timeline. Which can be used to create separate animation sequences [12]. Flash supports interactivity. Together with ActionScript and Flash build in GUI elements like buttons, textboxes, dropdown menus, checkboxes the developer can easily create interactivity [13].

5.1 Features of Flash

Flash have numerous of features. This paper present some of the key features that highlights a portion of what Flash is capable of.

- **Frame-based timeline**. Flash is constructed of frames displayed on a horizontal line, called the timeline. By using this timeline the developer easily can put motion to the creations. As many other animation programs the animation is controlled by key-frames and tweening [13].
- **Shape primitives**. Flash supports easy-drawing tools for creating elements. Create pie wedges, round off rectangle corners and much more [13].
- **Independent**. Holzinger et.al. present Flash as independent in the meaning of that to run a Flash program on the web, only a browser and a free download plug-in is needed [14].
- **Scalable**. Since Flash is a vector graphics animation program it supports scaling. This has the effect that the quality on an element, when zooming in and out, is the very same as the original element [14].

- **Consistent**. Flash give the impression of looking the same on all operative systems. This because it is run in the browser, on the client-side, and use the very same plug-in [14].
- **Supports sound**. Flash is described to be easy to integrate MP3 format sounds [14]. In the newer version of Flash (Flash CS3) a program called Adobe Soundboth is integrated to make it easier to edit the files [13].
- **Small files**. Because parts of Flash are run on the client side the Flash file can be kept small in size, compared to other animation programs [14].
- **Adobe compatible**. Newest version of Flash CS3 supports the importation of other Adobe program i.e. Adobe Photoshop format and Adobe Illustrator format. The imported files keep the structure in the previous program which can be of great assistance for the developer in Flash [13].

5.2 Prototyping with Flash

Jonathan Kaye and David Castillo wrote a book on Flash for interactive simulations [15]. Kaye et. al answer the question on "Why should Flash be used to simulations?" . They state that Flash is a great tool for allowing designer's and programmers to work together and they further state that Flash is a more than sufficient tool for sophisticated simulations. In doing so they highlight in particular that Flash is not only a graphical design tool for making Web applications and very simple animations. In their book they separate the terms simulation and interactive animation, but in the eyes of the user it is only important how the product meets the objectives of the presented material. They conclude their book by presenting several high-quality examples of how flash is used to create high-fidelity models, advanced simulations and the usage of Flash as a prototype.

Holzinger and Ebner present a paper on "Interaction and Usability of Simulations & Animations: A case study of the Flash technology" [14]. Their paper was concentrated on using simulation and animation towards e-learning purposes. They used Flash, among other tools, to create prototypes with interactive content. In conclusion they stated that a dynamic simulation can explain complex connections in a program, of which is superior to words or static pictures.

Cleotilde Gonzales performed a laboratory study with animated prototypes in the paper ŞDoes Animation in User Interfaces Improve Decision MakingŤ [16]. Gonzales did an empirical study with sixteen created prototypes with different attributes; Realistic or abstract images; smooth or not transitions; and different types of interactivity. The prototypes were created with another program from Macromedia called Macromedia Director. It is similar to Flash which contains a scripting language to control animations and an appearance of timeline. Since the Macromedia Director 8 release it is capable to import Flash files. Gonzales conclude the paper with stating that for an animation to be a great decision support tool the animation should complete these guidelines:

1. The designer should construct the animation on realistic graphical representations instead of abstract images.

2. Smoother pictures and animation transitions creates a more effective decision tool. However it is of importance to ensure the graphical objects conquers with the defined visual characteristics. Transitions should agree with the task structure.
3. The animation should allow the user to interact with different parts in a parallel order.
4. To create a animation which supports quicker decisions the developer ought to be thinking of the users past experience with computers. The developer should also consider the users insight in the task and previous use of animations and developing situations.
5. When developers create an animated interface they not only bear in mind the characteristics of task domain but also the structure on the task the user performs. By designing the animation, for the characteristics of the task domain, the developer may define the best representation, decay and reclamation for the task. Gonzales states that if the developer defines the structure of the task, the animation could be more understandable. Hence, the designer can divide the task in animation segments and therefore create a more logical design which is appealing to support the users awareness proceedings.

5.3 Flash as an RE tool

Can Flash be used as a prototyping tool in RE? As this paper stated in previous section 4.1 is without doubt prototyping in RE one of the most important components for confirming the elicitation process. Here is an example of a design project which this paper will use to explain the use of as a RE prototyping tool: The project is to design a user-interface for an automated-teller machine (ATM). To gather the requirements we used the "Unified Model of requirements elicitation" to select a proper elicitation technique for the ATM-project. The elicitation techniques used were background information and brainstorming. In this early stage of the project we want to take our requirements and test them in a prototype. Prototyping is created to confirm that the requirements that were elicitated are correct but also because elicitation is an iterative process were new requirements are to be found as the project develops. The prototype will now be created with the program Macromedia Flash.

When creating a Flash prototype, i.e for the ATM project, the developer can make the prototype respond to input and generate a satisfactory output by using ActionScript to control certain animations. The developer can also use Flash only to show some of the desirable outputs. Hence, Flash should be able to fulfill some of the aspects of the `Representational modelling` and `Behavioral modelling` for the RE process, presented in chapter 4. As a `representational model` Flash can be used to show the appearance of the system, it can be achieved by simply slide-shows to show a scenario over the desirable problem at hand. To use the ATM-project the scenario could be that the user is guided throughout one complete transaction with images that shows the appearance of the ATM, from inserting the debit-card to collecting the cash. The `representational model` of the ATM only allow the user to follow the flow, the scenario, presented by the designer in the prototype and cannot interact with the prototype in this stage. One great strength with Flash is that it uses vector based graphics and therefore can present images in good quality and also be used in up scaled presentations. The `representational modelling` presented on the ATM could therefore be used to test on a specific user but also to show the scenario on a big-screen presentation for many people.

As a `behavioural model` Flash can be used to show the functionality of the software system. Kaye et.al [15] examples of Flash models in their book "Flash MX for interactive simulation" gives an understanding that Flash can be used in an excellent way to show the functionality in a system. With a `behavioural model` it would allow our prototype created in the ATM example to interact with the user. Bray states that animated prototypes have a great advantage, thus the system has been defined in terms of the model and the dynamic representation in the terms of the behaviours; therefore it can be guaranteed that the specifications model reflects in the prototype [6]. Flash could be used as a dynamic representational modelling technique to show the behavioural and a part of the functionally of the system. Since Flash is known to be a fairly quick implementing tool, the Flash prototype can evolve during the project as Norman suggest with a high-quality programmer and designer working together with a user to evolve the prototype. The benefits of using Flash as a prototyping tool is many. One of the most important benefits, except for the knowledge that Flash is a quick tool for implementing prototypes. The benefit is the flexibility on which the designer is allowed to create, and easily change, the behaviours and functions in the prototype.

Our design team has created the interface for the ATM-project and tested the prototype on a specific group of users. New requirements has been identified, from the user-testing with the Flash prototype. Flash flexibility allow changes to be made quickly in the prototype. The Flash prototype has now been updated and can be used to testing on more user-groups or as a tool to inform all the stakeholders on which changes that has been made. Hence, Flash as a requirements engineering tool could serve as a specification document as well as a prototype for the user-testings.

By using Flash as a prototyping tool in RE the designer's is allowed to create a prototype early in the project and use it in the elicitation process; use it for direct user-testing; for in-house presentations to keep all stakeholders informed throughout the project; and the benefit of working with a prototype that is dynamic to changes.

But one of the most known problems in prototyping is to realize when to stop developing the prototype [7]. Many developers state that when the prototype reaches 90 percent of user-pleasure but a more common stop in the development is when the budgeted money for the project runs out. One other vulnerability with showing realistic prototypes early in the RE process is that the prototype can lead to an inaccurate belief that the project itself is far more advanced than it really is. It can lead to frustration when it is shown that it is not [6]. Another danger when using Flash as a prototyping tool can be that it promote unnecessary usage of animation and increase the risk of poor design [14].

Future work
To understand the effect Flash has as a prototyping tool in RE it would be of interest to perform analyses and evaluate prototypes on test-users and all the involved stakeholders in a greater RE process. This could be done by constructing an early prototype, with the guidelines presented by Gonzales in subsection "Prototyping with Flash", and maintaining it within the frame of the `dynamic representational modelling` and document the usage of the prototype as the project evolves.

6 Summary and Conclusion

This paper has composed the parts of requirements engineering which many authors consider to be the most important to emphasize. The paper has explained how we identify the `problem domain` to be able to start the elicitation process. A thorough background on the elicitation process and techniques to perform elicitation has been shown. This paper also has given an insight into a model called "Unified Model of Requirements Elicitation" which enable the developer to get an enhanced understanding of requirements engineering and the elicitation techniques. Prototyping has been stated, several times throughout this paper, to be one of the most important parts in the process of requirements engineering. As a result of this a number of techniques on modelling prototypes have been presented and elucidated. Finally the paper has inspected Flash as a prototyping tool. One could argue that Flash may possibly be used as a requirements prototyping technique. However it is not said that Flash should replace any prototyping techniques, especially not the early use of paper-prototypes, story-boards and use-cases for documentation. Flash could rather assist in the RE process as a requirement elicitation tool, as a specification document and as a dynamic representational model.

References

1. Grudin, J.: Groupware and social dynamics: eight challenges for developers. Commun. ACM **37**(1) (1994) 92–105
2. Cooper, K., Abraham, S., Unnithan, R., L.Chung, Courtney, S.: Integrating visual goal models into the rational unified process. In: ICSE '00: Proceedings of the Conference on The Future of Software Engineering, The University of Texas at Dallas 2601, North Floyd, Richardson, TX 75083 USA (2004)
3. Sommerville, I., Sawyer, P.: Requirements Engineering a good practice guide. John Wiley and Sons Ltd, Chichester, West Sussex, UK (1997)
4. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: ICSE '00: Proceedings of the Conference on The Future of Software Engineering, New York, NY, USA, ACM (2000) 35–46

5. ISO: Interantional organization for standardization. `http://www.iso.org/iso/home.htm`, accessed 2008-05-05 (2008)
6. Bray, I.K.: An Introduction To Requirements Engineering. Pearson Education Limited, Edinburgh Gate, Harlow, Essex CM20 2JE, United Kingdom (1997)
7. Saiedian, H., Dale, R.: Requirements engineering: making the connection between the software developer and customer. Information and Software Technology **42**(6) (2000) 419–428
8. Hickey, A.M., Davis, A.M.: A unified model of requirements elicitation. Journal of Management Information Systems **20**(4) (2004) p65 – p84
9. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers, San Francisco, USA (1994)
10. P. Hsia, A.M Davis, D.K.: Staus report: requirements engineering. Software, IEEE **10**(6) (1993) 75–79
11. Norman, D.A.: The design of everyday things. Basic Books, USA (1988)
12. Moock, C.: ActionScript for Flash MX: The Definitive Guide, second edition. Oreilly media, Sebastopol, CA, USA (2003)
13. Adobe: Homepage. `Http://www.adobe.com/`, accessed 2008-05-05 (2008)
14. Holzinger A., E.M.: Interaction and usability of simulations and animations: A case study of the flash technology. In: Publication (Psychological Sciences - University of Manchester) (2003)
15. Jonathan Kaye, Ph D, .D.C.: Flash MX for interactive simulation. Delmar Cengage Learning, Executive Woods, 5 Maxwell Dr., Clifton Park, NY 12065-2919, USA (2003)
16. Gonzalez, C.: Does animation in user interfaces improve decision making? In: CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (1996) 27–34

Project Manager Leadership Styles in Software Development Projects

Johan Ziprus

Department of Computing Science
Umeå University, Sweden
dit04jzs@cs.umu.se

Abstract. This literature study investigates how different project manager leadership styles affect the outcome of software development projects. Two categories of leaders were identified: the transformational and the transactional. The transformational leader is supportive and gives personal attention to every member of his staff. He/she has new ideas and wants to try new ways of doing things. The transactional leader is more directive and gives rewards only when goals are met. The investigation showed that both a transformational and a transactional leadership style could affect the outcome in a positive way. The comparison did not show though, which of the two leadership styles is the more successful one.

1 Introduction

Companies that carry out software development projects need to be as effective as possible. To be effective a company will have to know how to create effective project organizations. In a project organization, the project leader can be seen as the chief executive officer for that certain project. He/she coordinates the work, the schedule and the budget, amongst other things. Tonnquist [1], Parzinger [2] and Rose [3] argue that it can be hard for a orderer of a project and the project leader to fully understand each other if the orderer has little knowledge about software and programming. This points out that the role of the project leader and what commitment he puts in to get a good communication with the orderer is significant for project outcome. The goal of this paper is to study how the role of the project manager, that is the leadership style he/she is using, is affecting the outcome of software development projects.

Books and articles about project managing and leadership styles were used in this paper. Some of the articles were used only for background information, while three articles, that studied leadership styles in the context of software development, were used as case studies. These three had all made evaluations with a significant number of people working in software development projects and analyzed the data from the evaluations quantitatively. Based on their research questions, methods and findings they were interesting and representative for the aims of this paper.

The paper starts out with a section considering some of the key concepts in project management. An overview of the concepts of working in a project is followed by an investigation of different leadership styles. A discussion about how to define outcome is followed by a summary of the process improvement model Capability Maturity Model. Section 3 starts with a discussion about related work, then the three case studies conducted by Jiang et al. [4], Parzinger et al. [2] and Sonnentag et al. [5] are explained. The section ends with an explanation of Rose et al.'s work [3]. Section 4 presents the results of the case studies and an analysis of the results. Section 5 makes a comparison between the three cases and presents the results of the comparison. The last section draws conclusions from both the Analysis section and the Results section and discusses future work.

2 Background

A project is a temporary organization within the normal organization of a corporation. People from different divisions are put together in a group to do a specific job that is specified through a detailed project goal and project plan [1].

A project leader has the following responsibilities [1, 3]:

- Make sure that the project goal is fulfilled
- Put together the project group
- Plan and organize the project
- Supervise the work
- Motivate the people in the project group
- Make sure technical issues that go wrong are corrected
- Solve conflicts and human-related problems within the project group or with other stakeholders
- Prepare for the project's business opportunities/risks

2.1 Leadership styles

There are probably as many different leadership styles as there are leaders out there but to be able to compare them they have to be categorized. Amongst other ways to classify, Turner [6] identifies four different leadership styles: directive, supportive, participative and achievement-oriented. Directive leaders are controlling leaders that give orders while supportive leaders act more as a resource for his personnel. Participative leaders help out with the actual work and achievement-oriented leaders acts much like the directive leaders but in a more open and instructive way: he tells his personnel not only what to do but why he thinks so.

A more commonly used idea is to put leaders into two categories: the transactional and the transformational leader [2, 6, 7]. The transactional leader is the more conservative one that is directive but not very empathic. This type of leader inspires the staff with rewards when goals are met, opposite to continuously during the project. He/she manages by exception, action is taken mainly when things do not go as planned. The transformational leader is supportive, empathic and gives consideration to the individual. He/she wants to change the way about how people think by constantly challenging his staff with new ideas and different approaches. Transformational leaders can be seen as supportive and participative, while transactional leaders can be seen as directive and achievement-oriented.

Den Hartog et al. [7] examine the widely used Multifactor Leadership Questionnaire (MLQ) developed by Bass to distinguish between transformational, transactional and laissez-faire leadership styles. The laissez-faire leader is a passive leader that avoids both making decisions and taking supervisory responsibility. It is not a suitable leadership style under almost any circumstances [7]. Den Hartog et al. want to investigate if these three leadership styles can be defined quantitatively. The statistical analysis that follows their evaluations support two different models. The first model suggests two different leadership styles, active and passive. The active style contains the transformational and the transactional leader, and the passive style contains the laissez-faire leader. The second model suggests three different leadership styles: transformational, transactional and laissez-faire. This model conforms to Bass' theoretical framework. The statistical analysis did not show any support for more than three leadership styles.

2.2 How to define outcome?

Outcome can be measured in several ways. It can be measured in terms of quality of the final product, in terms of how well the budget is followed or in terms of how well the time plan of the project is followed [4]. Quality, in turn, can be measured in terms of customer satisfaction, or in how well the product or organization conforms to standards like ISO 9000-3 or TQM. TQM (Total Quality Management) is a framework designed to make the work structure in an organization lead to improved quality. It defines quality, foremost, in terms of customer satisfaction. TQM is not specifically adjusted for software development organizations [2].

Outcome also differs according to who measures it. A manager might define outcome in terms of how well a project conforms to standards or company policies. A user on the other hand might define outcome in terms of how the product changes their job or workload. Seen with the eyes of the developers, a project that might be considered a failure by management can be considered successful by the staff. This can be the case if the project is not able to meet deadlines and budget, but is a good technical learning experience for the staff and the eventual delivery has high quality [4].

The goal of this paper is to consider all the aspects of outcome when deploying a software project. Customers might be very satisfied with a certain product even though that certain product might have been twice as expensive to develop as planned. This example shows that quality alone is not sufficient to measure outcome. If the customers are not satisfied with the product, following budget and time schedule is neither sufficient to measure good outcome.

2.3 Process improvement models

There are several models aiming at improving project organizations. One that is commonly used is Capability Maturity Model (CMM). In CMM, there are five levels of software process maturity, used to define how well structured the project organization is [2]:

- Initial – most solutions are specifically created
- Repeatable – generalized cost and schedule plans has been implemented and these can be used by different projects
- Defined – generalized engineering plans have also been implemented
- Managed – data from the project work and the final product are quantitatively collected and analyzed
- Optimizing – the quantitative data is used for improving future projects

3 Reviewed case studies

In their literature review, Turner and Müller [6] discusses the same issues that are the concerns of this paper. They are not concerned with leadership styles in the software development arena though. The authors investigated whether the personality and leadership style of the project manager is a success factor for projects and if different leadership styles are appropriate for different project types. The investigation method was reviewing articles and books about project managing written from 1938 to 2005 and reviewing theories going back as far as 500 B.C.. Some of their findings were presented in section 2.1. The results from their findings were that the literature regarding project outcomes most often ignores the role of the project managers and which leadership style they are using.

3.1 Jiang et al.'s work

Jiang et al. [4] formulate the following hypothesis: Strong project manager performance improves project outcomes. To investigate this hypothesis the authors perform a quantitative case study involving 186 persons, most of them project leaders, department managers or information systems professionals. The respondents fill in a questionnare where the questions are based on a 5-point scale. Twelve questions were used to measure project manager performance (see Table 1). Seven questions were used to measure project outcomes, example of these were: "Able to meet project goals", "Adherence to budget" and "Adherence to schedule". The data was analyzed by means of path analysis.

3.2 Parzinger et al.'s work

Parzinger et al. [2] perform a quantitative study where they formulate two hypotheses, H1: The leadership style utilized during TQM implementation in the software development arena will affect the quality of software produced, and H2: The degree of transformational characteristics displayed by the leader will directly affect the level of software quality during TQM implementation.

To answer the hypotheses they send out 1990 questionnaires, of the ones returned 145 is used. The questionnaire has 10 questions designed to investigate whether the project leader is transformational or transactional. Six of the 10 questions corresponds to the traits associated with a transformational leader and 4 of the 10 to the traits associated with a transactional leader. The questions, which can be seen in Appendix A, were answered on a 5-point scale and then the mean of the 6 was compared to the mean of the 4 to decide whether the leader was transformational or transactional. Other questions, also to be measured with a 5-point scale, were used to measure the quality of the outcome.

3.3 Sonnentag et al.'s work

In this work two hypotheses are formulated. The second one, which is interesting in this context, is: Team leaders' goal orientation is positively related to team effectiveness, in addition to the use of design methods. To test the hypotheses a quantitative, questionnaire-based study of 29 German and Swiss software projects from 19 different companies was conducted. The mean project size was 10 members and the study resulted in 200 answers from project members where 62% were programmers and 26% project leaders. Respondents had on average 5.7 years of experience from software development. The questionnaire measured team leaders' goal orientation on a 5-point scale, as well as the context conditions for the project and the extent of design methods used.

Six to twelve months after a follow-up study was conducted where 135 of the 200 respondents from the first study participated. The follow-up study evaluated the results of the projects in terms of changeability of the product, team effectiveness, maintaining schedule and budget, and success of project. The values of these parameters were collected through questionnaires with standardized scales. A regression analysis, where the measures from the first study were used as independent variables, and the measures from the second study were used as dependent variables, was conducted.

3.4 Rose et al.'s work

Rose et al.'s work [3] contains both a qualitative analysis and a literature review. The aims of their article were to investigate whether project managers' work and approaches can be compared with the approaches to software project management described in the software engineering literature.

The study was conducted at the global company WM-data in Denmark and the qualitative analysis was performed through interviews with experienced project managers. Interviewees were chosen to reflect the variety of types of projects that the company runs. Questions were designed to investigate what the characteristics of the project situations were, what skills and competences the managers used in running their projects, and how those skills and competences could lead to successful project outcomes. Interviews were transcribed and analyzed mainly through frequency counts which helped identify much-discussed topics.

4 Analysis

4.1 Jiang et al.'s work

The results of the path analysis in Jiang et al.'s study [4] showed a relationship between the dependent variable, project outcomes, and the independent variable, project manager performance with path coefficient = 0,28. This shows that strong project manager performance improves project outcomes. It is interesting to discuss the 12 questions asked to define project manager performance in Jiang et al.'s study. Which question can be mapped to which type of leader trait? In Table 1 the questions asked are mapped to the corresponding leadership trait according to the definition of these in section 2.1. Seven out of the twelve questions correspond to a transformational leader and five out of the twelve to a transactional leader. The mean of the seven questions is 3.426 and the mean of the five is 3.356. This means that the average respondent showed almost exactly the same amount of transformational traits as transactional traits. One can conclude that the authors do not directly measure different leadership styles. On the other hand, the seven questions used to define outcome show that Jiang et al. has a holistic view regarding outcome. They take into consideration all the parts of outcome discussed in section 2.2.

4.2 Parzinger et al.'s work

The result of Parzinger et al.'s study [2] was that the overall quality had a mean of 3.11 with a transformational leader and a mean of 2.34 with a transactional leader. These results show that a transformational leader has a significant positive effect on the quality of the outcome. There are several ways to look at this fact. Since, according to this view, there are only two leadership styles, these styles have to contain a lot of traits. It is interesting to try to consider which traits that belong to the transformational leader that makes him the better one in this case. One reason for the transformational leader being the more successful might be that people in general work better under circumstances where they get more support than direct orders. Another reason might be that the transformational leader is more of a "new-thinker". Software development projects, in many ways, revolve around dealing with unique problems and the ability to create new solutions. A leader that wants people to think outside of the box, rather than sticking to old solutions that he knows work, might work better in this kind of working culture. In this context, it is important to take notice of the fact that Parzinger et al.'s study measured the outcome of software projects in organizations, that at the time of the study were implementing TQM. The

Table 1. Questions used by Jiang et al. to measure project manager performance and their corresponding leadership style deduced by this paper.

Question	**Leadership Style**
Project manager understood the various barriers to team development and built a work environment conducive to the team's motivational needs	Transformational
Project manager continuously updated and involved management and users to refuel their interest and commitment to the project	Transformational
Project leadership positions were carefully defined and staffed at the beginning of a new project	Transactional
Project manager conducted effective planning early in the project life cycle	Transactional
Project manager successfully involved key personnel at all organizational levels	Transformational
Project manager communicated individually with each prospective team member about specific tasks, the outcomes, timing, responsibilities, report relations, potential rewards, and importance of the project to the company	Transactional
Project manager defined the basic team structure and operating concepts early during the project formation phase. The project plan, task matrix, project charter, and policy are principal tools	Transactional
The team building sessions were conducted by the project manager throughout the project life cycle	Transformational
Project manager determined lack of team member commitment early in the life of the project and attempted to change possible negative views toward the project	Transformational
Project manager sought senior management support to provide a proper environment for the project team to function effectively	Transformational
Project manager watched for changes in performance on an ongoing basis	Transactional
Project manager focused his efforts on problem (conflicts) avoidance	Transformational

fact that these organizations were undergoing changes might also make it desirable having a manager potent of creating new solutions. This might explain why the transformational leader got better results than the transactional. It does not, though, prove that a transformational leader works better under “normal” working conditions.

The probably most important thing to keep in mind is that Parzinger et al.’s study measured outcome only in terms of quality, not in terms of time and money spent. The results of the study could have been different if these parameters were taken into consideration.

4.3 Sonnentag et al.’s work

The results of the regression analysis in Sonnentag et al.’s study [5] showed that goal orientation in the project leader raised the changeability of the product by 12%, the team effectiveness by 15%, the success of the project by 7% and the schedule and budget was maintained 18% better. These results showed that projects run by project leaders that put up clearly defined long-term goals and stick with these even through hard times has the highest outcome. This suggests that the achievement-oriented leaders are successful in obtaining most of the aspects of the complex concept of outcome. But what kind of leaders have Sonnentag et al. compared the achievement-oriented ones with? The problem is that they have not compared the achievement-oriented leader with other kinds of leaders. They investigate the role of the project leader in a broader sense and show that the more goal-oriented the leader the better the outcome. It is interesting that the dependent variable in Sonnentag et al.’s investigation is very well measured. All of the aspects of outcome are taken into account. But, the independent variable goal orientation is probably not alone sufficient to match the leader into a certain style, even though it can be seen more as a transactional trait, than a transformational. It would have been very interesting to see more independent variables, describing more traits of the project leader, included in this research. Variables like enthusiastic, supportive and directive could have been included to reflect the nature of the transformational leader.

From Sonnentag et al.’s results it can be interesting to point out that maintaining schedule and budget were the outcome factor that were most heavily affected by a goal-oriented leader. This can indicate that a transactional leader is the more successful one in maintaining schedule and budget.

4.4 Rose et al.’s work

Rose et al. identified seven types of managing that project managers must deal with: technical-, process-, team-, customer-, business-, personal (self)- and uncertainty management. Technical management is about having technical competence and either contribute to the technical work or supervise the technical work. Process management concerns the choice of proper software development models and traditional management skills such as planning, delegating and supervising. Team management is about putting together the project group or team, making sure that both the necessary technical and social skills are covered. Customer management concerns involving and building relationships with the orderer of the project. Business management is about adherence to budget and deadlines. Personal (self) management concerns the personal skills of the project manager, skills that are used regardless of the type of management that the project manager is doing at the moment. This is the management type concerned with leadership styles. Uncertainty management is about preparing for the project's business opportunities/risks.

5 Results

It is hard to generalize the results from the three case studies since none of them measured exactly the same thing. Jiang et al. and Sonnentag et al. measured all aspects of outcome, but they measured leader traits in a more generalized way. Leadership styles were not that clearly defined which is important from my perspective. Parzinger et al., on the other hand, measured leader traits pretty thoroughly but did not measure all of the aspects of outcome. Still, I have tried to conclude a generalized result to be seen in Table 2. Jiang et al. showed that a strong project manager performance improves project outcomes, but, as mentioned earlier, they did not measure different leadership styles. Therefore, it is concluded that both transformational and transactional leaders could get good outcomes. Parzinger et al. showed that a transformational leader got better outcome than a transactional, but as discussed earlier, they did not measure all parts of outcome. Sonnentag et al. measured only goal orientation in the leader, which can be seen as a transactional trait, but showed that this trait improves the outcome.

Table 2. The 3 case studies summarized.

	Outcome	
	Transformational	Transactional
Jiang	GOOD	GOOD
Parzinger	BETTER	WORSE
Sonnentag	-	GOOD

6 Summary & Conclusion

It might be that a manager that has a mix of leadership styles is the most successful one. One can probably conclude that both transformational and transactional leaders are needed but under different circumstances. Circumstances to consider: What aspects of outcome are important? Is the project organization undergoing change? If the project organization is under a lot of change, as in Parzinger et al.'s case study, the transformational leader seem to work out better. The companies Parzinger et al. evaluated were all undergoing changes in their organization due to implementation of a new organizational framework (TQM).

Another aspect to consider is where on the Capability Maturity Model ladder the project organization is. In an immature organization many solutions are worked out ad hoc [2], while the more mature the organization is, the more of the work is automatized. This might be a factor affecting which leadership style that is more successful. It might be more successful to have a transformational leader, that is open to new suggestions, in an immature organization where a lot of the planning and structuring is made up along the course of the project. On the other hand, one can argue that a transactional leader might work out better because it is important to have structure and discipline in a "chaotic" project. Independent of one's viewpoint, it might be that the maturity of the organization affects how significant it is which leadership style the project manager is using. In this context it is interesting to look at Rose et al.'s [3] findings discussed in section 4.4. Of the seven types of managing identified only personal management can be derived from the nature of the project manager to a larger extent, the other 6 are probably mostly defined through organizational structure. This is an argument for that leadership style does not influence outcome to a larger extent in a mature organization.

This paper started out with the intention to analyze four types of leaders: directive, supportive, participative, and achievement-oriented. Due to lack of research in this field two other types were used: transformational and transactional. As a final note it would be interesting to conduct a case study with well defined variables to measure outcome and leader traits. This should be done with respondents from a broad line of projects, that is both immature and mature projects, and project organizations going through changes and not.

References

1. Tonnquist, B.: Projektledning. Bonnier Utbildning AB, Stockholm, Sweden (2006)
2. Parzinger, Nath, Lemons: Examining the effect of the transformational leader on software quality. Software Quality Journal **9** (2001) 253–267
3. Rose, J., Pedersen, K., Hosbond, J.H., Kraemmergaard, P.: Management competences, not tools and techniques: A grounded examination of software project management at wm-data. Information and Software Technology **49** (2007) 605–624
4. Jiang, J., Klein, G., Discenza, R.: Pre-project partnering impact on an information system project, project team and project manager. European Journal of Information Systems **11** (2002) 86–97
5. Sonnentag, S., Frese, M., Brodbeck, F.C., Heinbokel, T.: Use of design methods, team leaders' goal orientation, and team effectiveness: A follow-up study in software development projects. International Journal of Human-Computer Interaction **9**(4) (1997) 443–454
6. Turner, J.R., Müller, R.: The project manager's leadership style as a success factor on projects: A literature review. Project Management Journal **36**(1) (2005) 49–61
7. Hartog, D.N.D., Muijen, J.J.V., Koopman, P.L.: Transactional versus transformational leadership: An analysis of the mlq. Journal of Occupational and Organizational Psychology **70** (1997) 19–34

A Parzinger et al.'s research questions

Respondents were to determine, on a 5-point scale, how well these statements, that define *transformational leadership*, corresponded to their project manager:

- Makes everyone around him/her enthusiastic about assignments.
- I have complete faith in him/her.
- Enables me to think about old problems in new ways.
- Gives personal attention to members who seem neglected.
- Finds out what I want and tries to help me get it.
- His/her ideas have forced me to rethink some of my own ideas which I had never questioned before.

In the same way, respondents were to determine how well these statements, that define *transactional leadership*, corresponded to their project manager:

- Tells me what to do if I want to be rewarded for my efforts.
- There is a close agreement between what I am expected to put into the group effort and what I can get out of it.
- As long as the old ways work, he/she is satisfied with my performance.
- He/she is content to let me continue doing my job in the same way as always.