

# Capricorn - An Intelligent User Interface for Mobile Widgets

Fredrik Boström, Petteri Nurmi,  
Patrik Floréen, Tianyan Liu,  
Tiina-Kaisa Oikarinen  
Helsinki Institute for Information Technology HIIT  
Department of Computer Science P.O. Box 68,  
FI-00014 University of Helsinki, Finland  
firstname.lastname@cs.helsinki.fi

Akos Vetek, Péter Boda  
Nokia Research Center  
P.O. Box 407, FI-00045 Nokia Group  
firstname.lastname@nokia.com

## ABSTRACT

Widgets are embeddable objects that provide easy and ubiquitous access to dynamic information sources, e.g., weather, news or TV program information. Interactions with widgets take place through a so-called widget engine, which is a specialized client-side runtime component that also provides functionalities for managing widgets. As the number of supported widgets increases, managing widgets becomes increasingly complex. For example, finding relevant or interesting widgets becomes difficult and the user interface easily gets cluttered with irrelevant widgets. In addition, interacting with information sources can be cumbersome, especially on mobile platforms. In order to facilitate widget management and interactions, we have developed Capricorn, an intelligent user interface that integrates adaptive navigation techniques into a widget engine. This paper describes the main functionalities of Capricorn and presents the results of a usability evaluation that measured user satisfaction and compared how user satisfaction varies between desktop and mobile platforms.

## Categories and Subject Descriptors

H.5.2 [Information interfaces and presentation]: User interfaces; H.3.5 [Information storage and retrieval]: On-line information systems; H.3.4 [Information storage and retrieval]: Systems and software

## General Terms

Human Factors, Design

## Keywords

Adaptive user interfaces, Mobile Widgets, Mobile Usability

## 1. INTRODUCTION

Widgets are autonomous objects that can be embedded into an operating environment. Each widget encapsulates

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*MobileHCI 2008* September 2-5, 2008, Amsterdam, the Netherlands  
Copyright 2008 ACM 978-1-59593-952-4/08/09 ...\$5.00.

a specific information source and provides easy, ubiquitous access to the information provided by the source. Examples of popular widget categories include news, weather sites, travel planners, dictionaries, games and comics [1, 7].

Widgets are typically used through a so-called widget engine. The widget engine is a specialized runtime component that allows users to interact with widgets and to manage them. Current widget engines suffer from some usability problems. First of all, as the number of supported widgets increases, finding relevant and interesting widgets becomes increasingly difficult. As an example, both Apple Dashboard [1] and Nokia Widgets [7] support more than 1,000 widgets. Another challenge is that the widget engine's user interface easily becomes cluttered with widgets that are no longer relevant. On a mobile phone, clutter causes unwanted scrolling, which decreases usability. Finally, interactions with information sources can be cumbersome, e.g., using the mobile phone to search within a highly-structured website.

In order to facilitate widget management and interactions on a mobile phone, we have developed Capricorn, an intelligent interface for mobile widgets. Capricorn integrates adaptive navigation techniques into a widget engine, and it is highly customizable (adaptable). Our target environment is a mobile phone, though Capricorn is web-based and hence it can be used also on desktop computers. Capricorn supports both local and web-based information sources. An example of a local information source that we support is the mobile phone's calendar. The system supports nearly 100 widgets, some of which are customizable. This paper describes the main functionalities of Capricorn and presents results from a usability evaluation that measured overall user satisfaction and compared how the level of satisfaction varies across different platforms (desktop and mobile). The reason to consider also the desktop platform in the usability evaluation is that it allows us to evaluate the overall usability of the interface irrespective of potential problems caused by the limited interaction capabilities of mobile phones. The results of the usability study indicate that users are satisfied with Capricorn on both platforms. Though resource limitations of present day mobile devices caused some problems with the mobile platform. For example, the browser crashed a couple of times, and occasionally the screen rendering was slow (see Sec. 4).

The rest of the paper is organized as follows: Sec. 2 discusses related work. Sec. 3 introduces the Capricorn interface and describes the adaptive navigation techniques that we use. Sec. 4 presents the results of the usability evaluation. Finally, Sec. 5 concludes the paper.

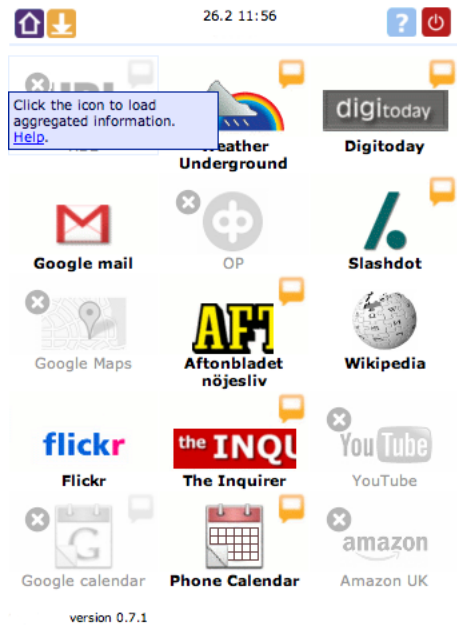


Figure 1: The Capricorn Interface

## 2. RELATED WORK

Widgets are related to mashups, which are services that combine two or more information sources to create a new service. Tuchinda *et al.* specify four tasks in the construction of mashups: data retrieval, source modeling, data cleaning and data integration [8]. A widget, on the other hand, encapsulates a single information source and is responsible for the three first tasks. Mashups are currently widely researched, and much research is being carried out, especially in order to facilitate the creation of mashups; see, e.g., [9].

Also intelligent user interfaces for managing complex software are related to our work. Debevc *et al.* introduce an adaptive toolbar for Microsoft Word that suggests adding or removing icons [3]. McGrenere *et al.* also focus on Microsoft Word and provide the user with two interfaces [6]. One of the interfaces has all functionalities, whereas the other one the user can customize. The interfaces include a toggle button that allows the user to switch between the two interfaces. User evaluations suggest that the combination of adaptive techniques and customization can facilitate management of complex software and improve user experience. These results have also motivated our work.

## 3. CAPRICORN USER INTERFACE

Capricorn is built as a web application and it can be used with any standard web browser – both on desktop computers and on mobile phones. The interface makes extensive use of JavaScript and AJAX<sup>1</sup> technologies to provide a smooth user interface, as well as to reduce load times and network traffic. The default width of the interface is sufficiently small for high-resolution mobile phones, but it scales down automatically on devices with lower resolution. This prevents the interface from expanding over the screen width, which would require the user to scroll the screen sideways.

<sup>1</sup>Asynchronous JavaScript and XML

## 3.1 Main Interface

The main screen of the Capricorn interface is shown in Fig. 1. The buttons in the toolbar correspond to (left-to-right): Home, Menu, Help, and Logout. The Home button provides quick access to the main screen from any other page, the Help button displays the help page, and the Logout button allows the user to log out from the system. In addition to having a help page, hovering over an icon provides the user with context-sensitive help information (i.e., a tooltip).

The Menu button is a drop down menu that contains links to the following pages:

- **Services and settings:** allows selecting what widgets are shown on the main screen and to customize the order of the widgets.
- **View your context:** shows what information about the user the system can currently access.
- **Give feedback:** allows the user to give feedback to the developers.

Having a menu button with several functionalities was necessary as otherwise the toolbar would wrap on two lines on low-resolution devices.

Below the toolbar is a canvas, where widgets are displayed. Each widget is represented using a icon. The user can interact with the icon by clicking on the corresponding icon. Each icon can also have one or more badges attached to it. Badges attached to the right-hand side of widget icons (e.g., Weather Underground or Google Calendar in Fig. 1) indicate that the widget is a quick-access service that provides aggregated or otherwise visually manipulated information from the original information source. Widgets that do not have the quick-access badge take the user to the corresponding website (e.g., Amazon or YouTube in Fig. 1).

The contents of a quick-access service are shown using a separate information layer that extends over the widget canvas. This reduces the widget's load time and makes it easier to return to the widget canvas. Normal services are loaded in a separate frame below the toolbar. In certain cases, the toolbar's functionalities are extended so that the toolbar also provides quick access for searching the website of the information provider. For example, Wikipedia and Wikitravel support the search functionality.

Badges on the left-hand side of the widget icons are quick-removal buttons that allow the user to remove infrequently used widgets (see Sec. 3.3). The quick-removal buttons are only shown for infrequently used widgets. Other widgets can be removed through the **Services and settings** page, which is described next.

## 3.2 Managing Widgets

Widget management takes place through the **Services and settings** page, which allows users to select and deselect widgets, as well as to rearrange the order of widgets on the widget canvas. A screenshot of the widget management interface is shown in Fig. 2.

There are two ways to add new services to the widget canvas. First of all, the user can select suitable widgets from a list of available widgets. The options in the list can be narrowed down to a specific widget category. Currently we have 16 categories and there are 2 to 20 widgets in each category. In addition to the list of available widgets, the

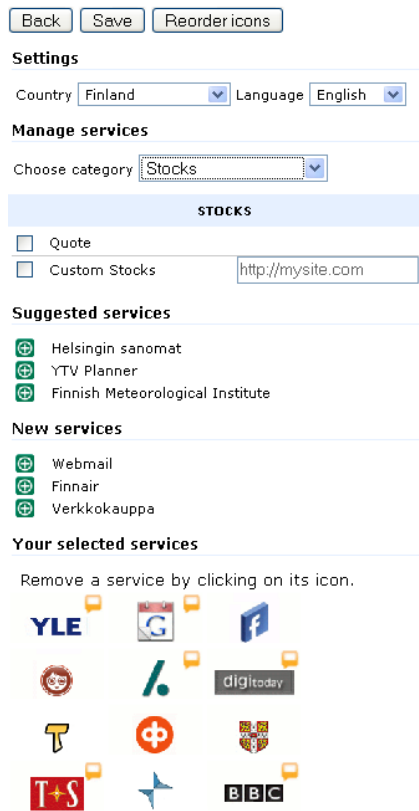


Figure 2: The Services and Settings Page

management interface provides shortcuts for adding certain services. More specifically, the user is shown the most recent widgets and we use collaborative filtering to recommend potentially interesting widgets (see Sec. 3.4).

In order to provide visual feedback to the user, the **Ser-****VICES** and **SETTINGS** page provides an automatically updated preview of the widget canvas. The preview can also be used to remove unwanted widgets by clicking on the corresponding icon in the preview.

Active widgets are automatically laid out on the widget canvas so that newly added widgets are inserted at the bottom of the canvas. However, the user has the option of (manually) rearranging the widgets on the canvas. As drag-and-drop interaction is difficult on mobile phones (except touchscreen enabled mobile devices), the reordering interface uses click-based interaction instead. When the user is in the reordering view, clicking on an icon selects the icon and arrows appear on the sides of the icon that was clicked. Clicking on one of the arrows swaps the widget with the widget pointed to by the arrow; see Fig. 3.

### 3.3 Information Dimming

The user interface of a widget engine easily becomes cluttered when the user does not bother to remove widgets that are no longer relevant. Capricorn automatically identifies infrequently used widgets and it uses dimming to present the findings to the user. In addition, the widget canvas provides a quick-removal badge for each dimmed widget. In Fig. 1, examples of dimmed widgets include YouTube, Amazon UK and Google Maps. The quick-removal badge is shown on



Figure 3: The Reordering Interface

the left-hand side of these widgets. The algorithm that is used to determine infrequently used widgets is described in [2]. We have chosen not to automatically remove unused widgets, since we believe that a spontaneously disappearing widgets would only confuse the user. Additionally, an unused widget is not necessarily unwanted, and removing it would work against the user's intentions.

### 3.4 Collaborative Filtering

Commercial widget platforms have demonstrated that the number of supported widgets easily becomes unmanageable. For example, Apple Dashboard supports more than 1,000 widgets, and most widget categories contain several hundreds of widgets [1]. Our preliminary analysis on widget usage has indicated that users tend to use (i) relatively few widgets per category, (ii) certain popular widgets will dominate the selection of widgets, and (iii) user choices are often uncorrelated unless the user selects one of the less-popular widgets. Motivated by these observations, we are currently using association rules to generate recommendations about suitable widgets. Association rules are able to capture the most frequently used items, and they are also able to provide recommendations about commonly occurring combinations of items. The algorithm we use is described in [4].

## 4. USABILITY EVALUATION

### Participants

Twenty-six users participated in the study. The study was conducted as an intercept study at the public areas (e.g., cafeteria, library) of a Faculty of Science. Each participant was given a movie ticket for participating in the study.

The majority of participants were males (19). With the exception of one user, the age of the participants was between 19 and 30. More specifically, 18 participants were in the age range 19-25, and 7 participants in the age range 26-30. Six users had previous experience with widget platforms, but only two users said they were using widgets regularly. The other users with previous experience were using widgets once a month or less frequently.

### Procedure

The study was carried out as a crossed-design where the used platform was considered as the experiment condition. We had two experiment conditions: desktop computer and mobile phone. The platform that was used first was varied. Eleven users started with the mobile phone and fifteen users

#	Question
Q1	Overall, I am satisfied with the ease of completing the tasks.
Q2	Overall, I am satisfied with the amount of time it took to complete the tasks.
Q3	Overall, I am satisfied with the support information (help messages, icons, other visual cues).

Table 1: The Usability Questionnaire

Desktop		Mobile Phone	
Median	IQR	Question	IQR
2	1	Q1	2
2	2	Q2	2
2	2	Q3	2

Table 2: Summary of the responses. IQR corresponds to interquartile range, which can be used to characterize spread (variance) in ordinal data.

started with the desktop.

In the experiment, users were given a set of tasks that they needed to perform using Capricorn. The nature of the tasks included information retrieval and widget management. After completing the tasks, the user was given a short questionnaire with three questions. The same procedure was repeated for both experiment conditions. After completing both experiment conditions, the user was given a post-study questionnaire with 26 questions. In this paper we focus only on the short questionnaires. The questionnaire that was used is based on the IBM computer usability satisfaction questionnaire [5]. The questions that were used are shown in Table 1. All questions were presented on a seven point Likert scale with the value one (1) corresponding to strong agreement and value seven (7) to strong disagreement.

## Analysis

We analyzed separately the responses that were given after completing the tasks on a desktop computer and on a mobile phone. As the first step of analysis, we performed factor analysis on the responses and interpreted the varimax rotations. The factor analysis indicated that the answers to questions Q1 and Q2 (see Table 1) depend on the platform that was used. We also verified that these differences were significant using a Wilcoxon rank sum test. Both Q1 ( $z = 3.37, p < .001$ ) and Q2 ( $z = 3.42, p < .001$ ) were found significant. Finally, we performed multinomial logistic regression on the responses using age, gender, previous experience and first experiment condition<sup>2</sup> as the control variables. None of the regression coefficients were significant, which suggests that the variability is fully explained by the platform that was used.

The responses to the questionnaire are summarized in Table 2. As expected, the responses indicate that people are more satisfied with using Capricorn on a desktop than on a mobile phone. However, both platforms receive positive scores for all questions. Moreover, most of the users that gave significantly lower scores to the mobile phone had external problems with the mobile phone. For example, the mobile phone's browser crashed for some users because

of memory running out. In addition, several users complained about the phone's browser being slow and difficult to use. These issues are related to the inherent resource limitations of present day mobile devices and the heavy use of AJAX/Javascript in Capricorn. Support for Web 2.0 technologies has only recently been introduced in mobile browsers, and running AJAX applications requires more CPU and memory from the device than running traditional, non-AJAX based web applications.

## 5. SUMMARY AND DISCUSSION

We have presented Capricorn, a web-based widget engine that uses adaptive navigation techniques to facilitate widget management. Results from a usability study indicate that the usability of Capricorn is good both on desktop and mobile platforms. This furthermore suggests that the interface techniques that we have adopted can be used to build usable, web-based mobile applications. Though, our results also indicate that, even though capabilities of mobile browsers are getting closer to their desktop counterparts, there are still some challenges that require careful consideration from the application developers.

## 6. REFERENCES

- [1] Apple Inc. Apple Mac OS X Dashboard. <http://www.apple.com/macosx/features/dashboard/>. [Retrieved 26-02-2008].
- [2] F. Boström, P. Floréen, T. Liu, P. Nurmi, T.-K. Oikarinen, A. Vetek, and P. Boda. Capricorn - an intelligent interface for mobile widgets. In *12th International ACM Conference on Intelligent User Interface (IUI)*, 2008. Demonstration.
- [3] M. Debevc, B. Meyer, D. Donlagic, and R. Svecko. Design and evaluation of an adaptive icon toolbar. *User Modeling and User Adapted Interaction*, 6(1):1-21, 1996.
- [4] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8:53 - 87, 2004.
- [5] J. R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions. *International Journal of Human-Computer Interactions*, 7(1):57-78, 1995.
- [6] J. McGrenere, R. M. Baecker, and K. S. Booth. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, 2002.
- [7] Nokia Inc. Nokia. s60 goes beyond web browsing with web run-time and widgets. <http://sw.nokia.com/id/e67c057a-7549-4edb-b9f1-193ab27cce71/S60WebRunTimeandWidgets.pdf>. [Retrieved 26-02-2008].
- [8] R. Tuchinda, P. Szekely, and C. A. Knoblock. Building mashups by example. In *Proceedings of the 2008 International Conference on Intelligent User Interfaces (IUI)*, pages 139 - 148. ACM, 2008.
- [9] J. Wong and J. I. Hong. Making mashups with Marmite: Re-purposing web content through end-user programming. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, pages 1435-1444. ACM, 2007.

<sup>2</sup>I.e., whether mobile phone or desktop was used first