

Capricorn – An Intelligent Interface for Mobile Widgets

Fredrik Boström, Patrik Floréen, Tianyan Liu,
Petteri Nurmi, Tiina-Kaisa Oikarinen
Helsinki Inst. for Information Technology HIIT
P.O. Box 68, FI-00014 University of Helsinki
{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, for example weather, news or TV program information. Widgets are typically rather static – they provide the information regardless of whether the information is relevant to the user’s current information needs. In this paper we introduce Capricorn, which is an intelligent interface for mobile widgets. The interface uses various adaptive web techniques for facilitating navigation. For example, we use collaborative filtering to recommend suitable widgets and we dim infrequently used widgets. The demonstration presents the Capricorn interface focusing on the adaptive parts of the interface. The user interface is web-based, and as such platform independent. However, our target environment is mobile phones, and thus the interface has been optimized for mobile phones.

Author Keywords

Adaptive user interfaces, Information aggregation, Context awareness.

ACM Classification Keywords

H5.2 Information interfaces and presentation: User interfaces. H3.5 Information storage and retrieval: On-line information systems. H3.4 Information storage and retrieval: Systems and software.

INTRODUCTION

Capricorn is an adaptive tool that provides fast and easy access to online information sources, such as favorite news providers, travel planners, weather services, TV-guides, personal calendar events etc.

Capricorn operates both on PCs and on mobile phones, as it is used through a normal web browser. However, our target environment is mobile phones, so the user interface has been optimized for such use. Capricorn receives information about the user’s situation using modules installed on

Permission to make digital or hard copies of part or all 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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUI’08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain.
Copyright 2008 ACM 978-1-59593-987-6/ 08/ 0001 \$5.00

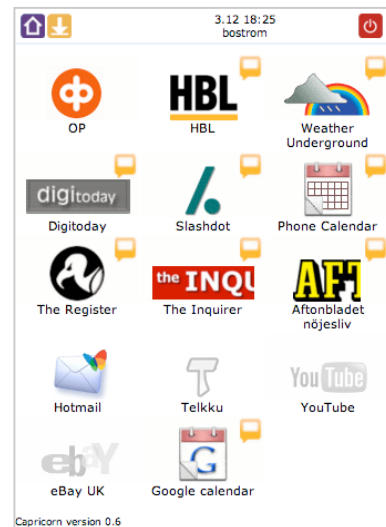


Figure 1. The Capricorn interface.

the mobile phone. These modules provide information regarding for example the location of the user, the current phone profile and events on the phone’s calendar. The widgets can use the context information and user’s usage history to adapt their contents. We briefly describe the interface and architecture of Capricorn, after which we describe what kind of user modeling techniques are used and how. We finalize the paper by discussing our current research directions.

Interface

The main screen of the Capricorn interface is shown in Figure 1. The buttons in the toolbar correspond (from left-to-right) to Home, Menu, and Logout. The Home button provides quick access to the main screen from other pages and the Logout button allows the user to log out from the system. The Menu button is a drop down menu containing the following items:

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

- *Questionnaire*: a user survey. This functionality is used for system evaluation and is not part of the core system.

System Architecture

Capricorn is built using a modular architecture. Each module has its own functionalities and communication protocol. A blackboard module handles the communication between modules centrally. The blackboard supports different data formats, multiple users, different widgets and events.

Part of the modules resides on the mobile phone, and part on a server. The modules on the mobile phone communicate with the server over 3G or WLAN. As the foundation for the whole system lies a server-side database containing user data, available services, context data etc. Currently, the following modules reside on the mobile phone: PhoneInfo (provides basic information about the phone status, such as cell id), PhoneCalendar (entries in the phone's calendar), and a module BeTelGeuse (see [1]) for gathering data from available sensors over Bluetooth.

The following modules residing on the server are implemented: Weather (providing local weather information for the city given by the user), GoogleCalendar (providing information available in the user's Google Calendar), and RSS. The RSS module gathers news headlines at predefined intervals (currently 10 minutes). The user can choose which news providers she is interested in. The RSS module uses a 'feed definition file' written in XML that defines each feed's source and how the feed should be interpreted. This allows the system to customize each feed, for example by allowing certain tags normally not suited for mobile viewing, such as images.

User interface

The user interface module is responsible for conveying information to the user and gathering user input. It has been optimized for mobile use with regards to both input methods and scalability. The possibility to use Capricorn on touch screen equipped devices has also been taken into consideration when designing the user interface.

The interface default width is sufficiently small to fit on a high resolution mobile phone, but 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.

The user interface is built using PHP and is usable in any standard web browser, both on PCs and mobile phones. It makes extensive use of JavaScript and AJAX (Asynchronous JavaScript and XML) technologies to provide a smooth user interface, as well as diminishing load times and network traffic.

Adaptivity

The current implementation of the interface supports two types of adaptation: dimming infrequently used items and recommending new services. In addition, we are currently implementing content adaptation that takes into account situation information.

Dimming

In Capricorn, infrequently used widgets are dimmed and we provide a quick link for removing them. In Figure 1, the Telkku, YouTube and eBay widgets have been dimmed out, since they have not been used frequently enough. The algorithm that we use for deciding which items to dim out is based on stochastic approximation. More precisely, we maintain for each selected service a weight that lies between 0.0 and 1.0. Whenever the user interacts with a widget the weights are updated so that the weight of the widget that was used is increased towards one whereas the weights of the other widgets are decreased towards zero. When the weight reaches 0 the item is dimmed.

Collaborative Filtering

Most widget sites provide information about the most popular widgets. In addition to providing information about new and popular widgets, we use collaborative filtering based on an association rule algorithm. This lets us recommend also frequently seen combinations of widgets. The algorithm that we use is described in [2].

CURRENT WORK

We are currently building the first adaptive widgets, which will consider mainly location information. For example, we are going to use GSM cell identifiers to determine the country in where the user is. This can, among other things, be used to automatically change the time zone for the clock as well as localize news feeds. We are also going to build a weather widget that fetches the weather information for the current location without the user having to manually input the city. We are mainly working with custom semantics that users associate with GSM cell identifiers. However, it would also be possible to use a geo-coding service for mapping GPS information into postal addresses, and use the postal addresses to determine the appropriate location.

In addition to location information, we are also going to consider user preferences. When users read news items, we can record the topic (domestic, foreign, economy, sports, etc.) and learn high-level preferences of the user. The topic can furthermore be associated with context information in order to learn context-specific preferences.

We are currently in the process of starting a user study regarding the usability and usefulness of the system. A usability expert has evaluated the system, and the resulting suggestions have been taken into account in the current version of Capricorn.

REFERENCES

1. Nurmi, P., Kukkonen, J., Lagerspetz, E., Suomela, J., and Floréen, P. BeTelGeuse – a tool for Bluetooth data gathering. In *Proc. BodyNets 2007*.
2. Han, J., Pei, J., Yin, Y., and Mao, R. Mining frequent patterns without candidate generation. *Data Mining and Knowledge Discovery* 8, 1 (1992), 53-87.