Integrated Context-Aware and Cloud-Based Adaptive Home Screens for Android Phones

Tor-Morten Grønli[1], Jarle Hansen[2], and Gheorghita Ghinea[1]

[1] The Norwegian School of Information Technology,
Schweigaardsgt. 14, 0185 Oslo, Norway, and
School of Information Systems, Computing and Mathematics, Brunel University,
Uxbridge UB8 3PH, London, United Kingdom
[2] School of Information Systems, Computing and Mathematics, Brunel University,
Uxbridge UB8 3PH, London, United Kingdom
george.ghinea@brunel.ac.uk, jarle.hansen@brunel.ac.uk,
tmg@nith.no

Abstract. The home screen in Android phones is a highly customizable user interface where the users can add and remove widgets and icons for launching applications. This customization is currently done on the mobile device itself and will only create static content. Our work takes the concept of Android home screen [3] one step further and adds flexibility to the user interface by making it context-aware and integrated with the cloud. Overall results indicated that the users have a strong positive bias towards the application and that the adaptation helped them to tailor the device to their needs by using the different context aware mechanisms.

Keywords: Android, cloud computing, user interface tailoring, context, context-aware, mobile, ubiquitous computing, HCI.

1 Introduction

The home screen in Android phones is a highly customizable user interface where the users can add and remove widgets and icons for launching applications. This customization is currently done on the mobile device itself and will only create static content. The content, buttons and widgets, are shown in the same position with the same icons until the user manually changes it. Our work takes the concept of Android home screen [3] one step further and incorporates this in regular Android applications. We add flexibility to the user interface by making it context-aware and integrated with the Google cloud. Cloud computing is focused on sharing data and computation resources over a scalable network of nodes. It has gained popularity over the last few years and large companies like Microsoft, Google and IBM all have initiatives promoting cloud computing [1]. The configuration of the application home screen is stored in the Google cloud and the server will push events to the registered mobile devices that are executed on the phone. Mark Weiser [2] argued that machines should fit the human environment and not force the humans to enter theirs. We try to follow

J.A. Jacko (Ed.): Human-Computer Interaction, Part II, HCII 2011, LNCS 6762, pp. 427–435, 2011.

this idea by registering the context of the user from many different sources and automatically adapting the home screen based on the predefined user configuration.

The paper is organized as follows: Section 2 describes the research background, followed by application design and implementation in section 3. Section 4 addresses the preliminary results from our work and section 5 concludes the paper.

2 Background

Developers and researchers agree that context is an increasingly important factor when designing new mobile applications. Context-aware tailoring of information sources for end users can greatly increase the perceived use and agility of mobile environments.

Context-awareness in mobile applications has caught the attention of researchers [12, 13] and proves to be more and more an invaluable resource. For example, Ludford et al. [14] looked at the use of context to provide useful information to the user on their mobile phone. In their work, context is based on the location and/or time of the day. Another definition of context is the user's planned activity in combination with the location. This is interesting because it generates quite a lot of information about the user, but information is of reduced interest if we do not combine it with other contextual dimensions or aggregate the data. On an overall basis the use of context in applications is often missing or single dimensional. This focus should be changed, since automated information aggregation from contextual sources would possibly be able to not only support the everyday tasks of the user, but also improve efficiency and ease the work of the user by automatically tailoring information to the user's needs and/or adapting the application to the user's current setting. The widespread use of smart, mobile devices have led researchers and developers to consider context as an important criteria for highly mobile systems. The notion of context-aware computing is generally the ability for the devices to adapt their behavior to the surrounding environment, hence enhancing usability [15].

Towards this goal, Dey and Abowd [2] state that if we understand context fully in a given environment and setting, we would then be able to better choose what context-aware behaviors to sustain in applications. This could lead to more realistic applications and thereby applications more meaningful to users. Edwards [4] exemplifies this when he uses context information to build an application. In his application different layers represent different sources of information and they can be reused in later settings. Edwards argues that context is a major part of our daily life and that computing with support for sharing and using contextual information (context-aware computing) would improve user interaction. Indeed when viewing people rather than systems as consumers of information, a new infrastructure is needed.

The implementation of sensors is the second half of our context-aware dimension. By enabling such services we facilitates exploitation of new resources in a mobile environment. Traditionally sensors are an important source of information input in any real world context and several previous research contributions look into this. The work of Parviainen et al [7] approaches this area from a meeting room scenario. Here, we can find a variety of applications in which a sound source localization system may be useful, such as: Automatic translation to another language, retrieval of specific topics,

and summarization of meetings in a human-readable form. The paper describes briefly the source localization system developed for these tasks as well as evaluating the results from preliminary experiments. They find sensors a viable source of information, but also acknowledge there is still work to do, like improving integration.

Another contribution focusing on the information retrieval possibilities is the work of Chatzigiannakis et al. [8]. They study how wireless sensor and actor networks are comprised of a large number of small, fully autonomous computing, communication, sensing and actuation devices, with very restricted energy and computing capabilities. Further they investigate how such devices co-operate to accomplish a large sensing and acting tasks such as information gathering from real world events.

In line with our research Boulis et al [9] look into how sensor usefulness changes in response to the dynamic needs of multiple users. They propose a framework, SensorWare, to define and support lightweight and mobile control scripts that allow the computation, communication, and sensing resources at the sensor nodes to be efficiently harnessed in an application-specific fashion. Their proposal seeks to remedy the limited flexibility problem at the expense of increased responsibility for the programmer.

By taking the aspect of sensors and context-aware information and integrating it in a mobile application we acquire desirable effects for the end users. When this is further combined with remote configuration from cloud-deployed applications the benefits, could be even greater. Cloud computing focuses on sharing data and makes it possible to distribute storage and computations over a scalable network of nodes. Large IT companies like Microsoft, Google and IBM, all have initiatives relating to cloud computing [10]. One of the key features under this paradigm is scalability on demand, where the user pays for the amount of computation and storage that is actually used. Moreover, this scalability is usually completely transparent to the user. Mei et al. [10] have focused on finding interesting topics for future research in the area of cloud computing, and have drawn analogies between cloud computing and service and pervasive computing. They identify four main research areas: 1) *pluggable computing entities*, 2) *data access transparency*, 3) *adaptive behaviour of cloud applications* and 4) *automatic discovery of application quality*. Our work is closely related to *adaptive behaviour of cloud computing applications*. In our work we implement context-awareness in a mobile environment. Further we take advantage of remote configuration of in application settings, making the local application adept to changes saved in the cloud and pushed to individual attached devices.

One important issue with cloud computing is privacy. Data is often stored on external servers that one has no control over. Mowbray and Pearson [11] have looked at how it can be possible to implement a client-based privacy manager in a cloud-computing environment. Their solution reduces this risk by helping users to control their own sensitive information. This is implemented using an obfuscation and de-obfuscation service to reduce the amount of sensitive information held within the cloud. They have also implemented a feature to allow users to express privacy preferences about the treatment of the personal information. Security is also an important issue in our work when dealing with sensitive user data. We address this by using state of the art Open Authentication services implementation in our application and integrating with the Google login process.

To summarize, our research investigates integrated context-aware and cloud-based adaptive application home screens for android phones.

3 Design and Implementation

The user start by logging in to a webpage using his Google account, s/he is then presented with several options that will customize the home screen on his mobile device. One example is that the user can configure that the screen brightness on his mobile should be automatically lowered when the battery charge is below 20%.

Another example is that the user is able to select the applications he wants to be available on the home screen of his mobile device. By selecting for example the *Maps* application and pressing *Save configuration* this will be saved in the cloud and automatically pushed to his phone, figure 1.

Fig. 1. Server overview

The phone will update the home screen when the push message is received from the Android Cloud to Device Messaging system. By configuring these options on the cloud we get several benefits. The configuration values are not tied to one specific device, so there is no need to manually configure each device. By registering the device in the cloud application, any predefined configuration values are pushed to the phone. It is also easier to add more advanced configuration options when the user can take advantage of a bigger screen, mouse and keyboard than those found on small or mobile devices.

The context-aware part of the system handles multiple input sources. The simplest form is the context taken from the phone, where battery level and screen brightness is two of the input options used in our application. We also integrated the system with Google Calendar. We added special tags, *${type=work}* or *${type=leisure},* in the

description field of the scheduled meetings to describe the context. If the tag *${type=work}* was added, this lets the application know that the user is in a work setting and it will automatically adapt the contacts based on this input. In a work context only work related contacts would be shown.

The user interface in our solution consisted of two main components: 1) The home screen on the mobile device and 2) the webpage to add the configuration options. The home screen is of course the most important part. This screen handles most of the user interaction with the phone and all other applications can be launched from this screen. An overview of the system is shown in figure 2.

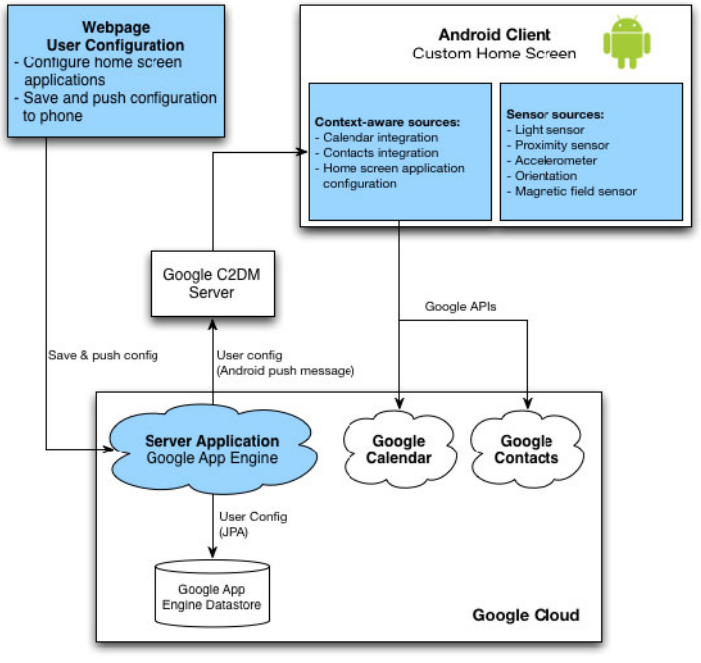

Fig. 2. System architecture

The blue boxes in the diagram represents the parts of the system we created. The white boxes, like Google calendar and Google contacts, are external systems we communicated with. The server application was deployed on Google App Engine and data stored in the Google cloud. We also integrated with the Google login service, making it possible for all users with a Google account to log on to the system.

When a new user logs in to the system with a Google account we automatically store this information in the Google App Engine data store. By using JPA (Java Persistence API) in combination with objectify we were able to store and retrieve data with minimal boilerplate code. The push functionality was implemented using the C2DM (Cloud to Device Messaging) framework. This is described in more detail in

the next section. Each C2DM enabled device would register with Google and receive a registration id. This id was sent to our cloud server through a normal HTTP GET call. We then proceeded to store this registration id in the data store. When the user pushed the save button on the webpage (see figure 1), to update the settings on the Android home screen, this would trigger a push message to be sent to the previously persisted registration id.

In addition to this we added support for input from several sensors available on the mobile device. In our experiment we look at input from 5 sensor sources: accelerometer, magnetic field, orientation, proximity and light. For light we added the functionality of changing the background color based on the value received from the sensor. If the room was dark the screen shows a white background. When the user moves to a room with more light the screen will gradually get darker.

4 Application Walkthrough

The user interface was evaluated with a sample of users who expressed their opinions in respect of statements targeting ease-of-use, functionality and ease of adaptation. Overall results indicated that the users have a strong positive bias towards the application and that the adaptation helped them to tailor the device to their needs by using the different context aware mechanisms. In this section we will provide an overview of the mobile Android client in light of user evaluation feedback. Figure 3, below, shows the configurable home screen as displayed on the device.

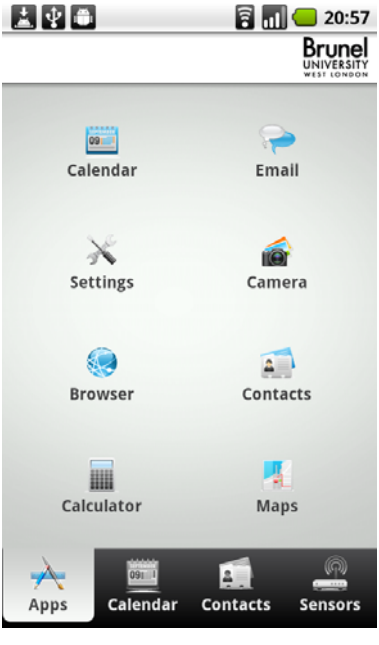

Fig. 3. Application home screen

The information presented on the home screen is quick launch icons for different applications installed. The specific information on the screen is tailored in accordance with the chosen user settings in the cloud based web interface (shown in figure 2). By taking advantage of the brand new Cloud 2 Device Messaging (C2DM) API provided by Google for Android 2.2 we were able to push messages to the devices without the application running. Not only is this a much simpler solution for the users it also has advantages like providing better battery life since we do not have to rely on a polling mechanism. The Android Cloud to Device messaging service is currently in beta and is not available to the general public. Feedback from user testing very positive to this feature and one commented on the possibility for this home screen view to replace the original Android phone home screen. We see this as a viable and interesting idea worthy future pursuit.

Also displayed in figure 3, in the bottom part of the picture, is one of the navigation menus from in the application. By selecting the "Calendar" tab, the application switches to calendar mode and presents upcoming appointments to the user. The upcoming appointments presented are only relevant to the users context, meaning that information are filtered based on a user-context algorithm. The special tags, *${type=work}* and *${type=leisure}*, provide one out of several foundations for the application to compute the users context. Other factors influencing the decision is context-aware dimensions such as GPS tracked location and time. Once a user-context is defined, this will also influence the selection of contacts displayed if selecting the next tab in the bottom part menu. The features of custom selected contacts and calendar items generated a lot positive feedback and response in our user test group. One user commented: "The tailoring and filtering of information eases my job of searching for information". Another user pointed out that this feature could also be applied to other cloud stored data such as documents and notes.

In our application we have, as previously described, input from five different sensor sources: accelerometer, magnetic field, orientation, proximity and light. This makes for another novel dimension to use in our user context computation. We actively use sensor input for other tailoring of the application as well. Figure 4 and figure 5 displays an example of the light sensors in action. By integrating with the different sensors, such as light, we are able two automatically adjust brightness of the display for the end user. By continuously measuring the amount of light around the devices we are capable of changing the background color based on the value received from the sensor. This shows in the application as a darker background color in the application the brighter the room is. This adaptation is especially useful for the user when using the application in outdoor scenarios because of the improved readability of the user interface.

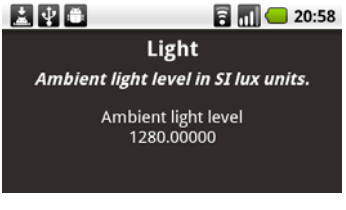

Fig. 4. Light sensor in action 1

Fig. 5. Light sensor in action 2

Figure 4 and figure 5 shows a sub screen of the implementation of the light sensor. The figures show how the background can change according to the measured lux value, also displayed in these screenshots for information purposes. From user testing the implementation of the light sensor as a context-aware dimension variable for user context computation where very welcomed. The ease of use of the application under very different lighting conditions due to the background color adaptation where highlighted as the major positive factor. As figure 6 illustrates we are currently looking into making even more sensor data available and include them into the context-aware equation.

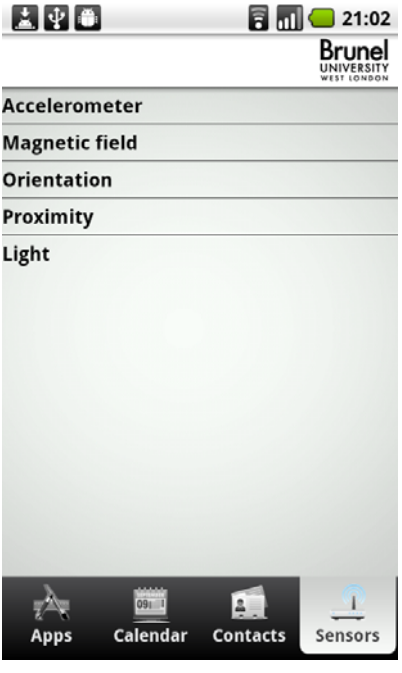

Fig. 6. Sensor panel

5 Concluding Remarks

The most important aspect of our work is how a flexible and extensible solution can be integrated with the Google cloud to provide value to the users. We can take advantage of many different context sources and adapt the home screen to what the user wants. Our experiment gives a few examples, but there are many other possibilities that can be added to the system. Another important part of the system is the integration with sensors on the mobile device. Android gives the developer full access to read values from the sensors on the phone, this is an exciting feature that we would like to extend in future versions of the system.

Overall results indicated that the users have a strong positive bias towards the application and that the adaptation helped them to tailor the device to their needs by using the different context aware mechanisms.

References

1. Mei, L., Chan, W.K., Tse, T.H.: A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues. In: Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference (APSCC 2008), pp. 464–469 (2008)
2. Weiser, M.: The computer for the 21st century. In: Human-computer interaction: toward the year 2000, pp. 933–940. Morgan Kaufmann Publishers Inc., San Francisco (1995)
3. Google Mobile: Android basics: Getting to know the Home screen (2010), http://www.google.com/support/mobile/bin/answer.py?answer=168445#1149468 (last visited October 5, 2010)
4. Kawaguchi, K.: Hudson Extensible continuous integration server (2010), http://wiki.hudson-ci.org/display/HUDSON/Meet+Hudson (last visited October 10, 2010)
5. Göker, A., Watt, S., Myrhaug, H.I., Whitehead, N., Yakici, M., Bierig, R., Nuti, S.K., Cumming, H.: An ambient, personalised, and context-sensitive information system for mobile users. In: Proceedings of the 2nd European Union symposium on Ambient intelligence, pp. 19–24. ACM, Eindhoven (2004)
6. Dey, A.K.: Understanding and using context. Journal of Personal and Ubiquitous Computing 5(1), 4–7 (2001)
7. Parviainen, M., Pirinen, T., Pertilä, P.: A Speaker Localization System for Lecture Room Environment. Machine Learning for Multimodal Interaction, 225–235 (2006)
8. Chatzigiannakis, I., Kinalis, A., Nikoletseas, S.: Priority Based Adaptive Coordination of Wireless Sensors and Actors. In: Proceedings of the 2nd ACM International Workshop on Quality of Service & Security for Wireless and Mobile Networks (2008)
9. Boulis, A., Han, C., Srivastava, M.: Design and Implementation of a Framework for Efficient and Programmable Sensor Networks. In: Proceedings of the 1st International Conference on Mobile Systems (2003)
10. Mei, L., Chan, W.K., Tse, T.H.: A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues. In: Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference, pp. 464–469. IEEE Computer Society, Los Alamitos (2008)
11. Mowbray, M., Pearson, S.: A client-based privacy manager for cloud computing. In: Proceedings of the Fourth International ICST Conference on Communication System Software and Middleware, pp. 1–8. ACM, Dublin (2009)
12. Bilandzic, M., Foth, M., Luca, A.: CityFlocks: Designing Social Navigation for Urban Mobile Information Systems. In: Proceedings ACM Designing Interactive Systems (2008)
13. Rodden, T., Cheverest, K., Davies, K., Dix, A.: Exploiting context in HCI design for mobile systems. In: Workshop on Human Computer Interaction with Mobile Devices (1998), http://www.dcs.gla.ac.uk/~johnson/papers/mobile/HCIMD1.html
14. Ludford, P., Rankowski, D., Reily, K., Wilms, K., Terveen, L.: Because I carry my cell phone anyway: functional location-based reminder applications. In: Proceedings of Conference on Human Factors in Computing Systems, April 2006, pp. 889–898 (2006)
15. Dey, A.K., Abowd, G.: Towards a Better Understanding of Context and Context-Awareness. In: Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (1999)