Sensor-Based Adaptation of User Interface on Android Phones

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

[1]Norwegian School of Information Technology
Oslo 0185, Norway
[2]Brunel University
Uxbridge, UB8 3PH, United Kingdom
tmg@nith.no, george.ghinea@brunel.ac.uk,
hansjar@jarlehansen.net

Abstract. The notion of context-aware computing is generally the ability for the devices to adapt their behavior to the surrounding environment, ultimately enhancing usability. Sensors are an important source of information input in any real world context and several previous research contributions look into this topic. In our research, we combine sensor-generated context information received both from the phone itself and information retrieved from cloud-based servers. All data is integrated to create a context-aware mobile device, where we implemented a new customized home screen application for Android enabled devices. Thus, we are also able to remotely configure the mobile devices independent of the device types. This creates a new concept of context-awareness and embraces the user in ways previously unavailable.

Keywords: sensor, interface adaptation, Android.

1 Introduction

Context-aware computing builds on a combination of various technologies such as computers, mobile devices, human, sensors, cloud, web, Internet and software services. All such technologies and services are interconnected in order to communicate with each other and exchange of useful information such as location, weather information, traffic conditions, road direction, and health and safety.

1.1 Sensors

Sensors are an important source of information input in any real world context and several previous research contributions look into this topic. For instance, Parviainen et al. (2006) approached this area from a meeting room scenario. They found several uses for a sound localization system, such as: automatic translation to another language, retrieval of specific topics, and summarization of meetings in a human-readable form. In their work, they find sensors a viable source of information, but also

M. Kurosu (Ed.): Human-Computer Interaction, Part V, HCII 2013, LNCS 8008, pp. 56–61, 2013.

acknowledge there is still work to do, like improving integration. Modern smart phones have a number of built in sensors, all of which can usually be accessed as local services through well-defined APIs. In our work, by taking the aspect of sensors and context-awareness and integrating it in a mobile application we reduce the workload for the end users. Thus, in combination with a cloud-based server application we are able to remotely configure the mobile devices independent of the device types, creating a new concept of context-awareness.

1.2 Cloud Computing

Cloud computing has received considerable attention in the software industry. It is a buzzword frequently used for all sorts of services, ranging from hosted virtual machines to simple web based email applications. So what is cloud computing? We have used the definition created by NIST [6] (National Institute of Standards and Technology, United States): "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources …*". Large IT companies like Microsoft, Google and IBM, all have initiatives relating to cloud computing [7] which have spawned a number of emerging research themes, among which we mention: *cloud system design* [6], *benchmarking of the cloud* [5] and *provider response time comparisons*. Mei et al. [5] have pointed out 4 main research areas in cloud computing that they find particularly interesting, namely *Pluggable computing entities, data access transparency, adaptive behaviour of cloud applications and automatic discovery of application quality.*

Our work focuses on *data access transparency*, where clients transparently will push and pull for data from the cloud, and *adaptive behavior of cloud applications*. We adapted the behavior of the Google App Engine server application based on context information sent from the users' devices thus integrating context and cloud on a secure mobile platform [1].

Accordingly, in our research, we combine sensor-generated context information received both from the phone itself and information retrieved from cloud-based servers. All data is integrated to create a context-aware mobile device, where we implemented a new customized home screen application for Android enabled devices. Thus, we are also able to remotely configure the mobile devices independent of the device types.

2 Design

We developed a proof-of-concept client application on an Android device (HTC Nexus One). The HTC manufactured Nexus One represents one of the first worldwide available commercial Android phones. The Nexus One features dynamic voice suppression and has a 3.7-inch AMOLED touch sensitive display supporting 16M colours with a WVGA screen resolution of 800 x 480 pixels. It runs the Google Android operating system on a Qualcomm 1 GHz Snapdragon processor and features 512 MB standard memory, 512 MB internal flash ROM and 4 GB internal storage. Furthermore, the Nexus One also has 5 sensors available: accelerometer, magnetic field, orientation, proximity and light.

The developed application utilized context-aware information from the device in the form of time, location and sensors. Additionally it utilized context-aware information from the cloud-integrated backend to acquire dynamic interface content, contacts and calendar data. At launch, the application would look as illustrated in Fig. 1. This interface would change depending on the user's given context. The applications available would be adapted and customized to match the current computed user context and thereby unobtrusively alter the user experience.

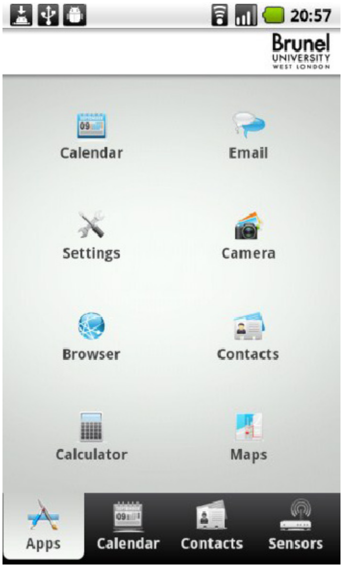

Fig. 1. Home Screen Interface

2.1 Meta-Tagging

To make it possible for users to tag their appointments and contacts with context information we added special meta-tags. By adding a type tag, for example *$[type = work]* or *$[type= leisure],* we were able to know if the user had a business meeting or a leisure activity. We then filtered the contacts based on this information. 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. To add and edit these tags we used the web-interface of Google contacts and calendar.

2.2 Sensors as Input Data

The sensors on the mobile device were also used as input to the application. We used the API available on the Android platform and through the *SensorManager* base class called were able to access all of the built-in sensors (eg. accelerometer, light) on the mobile device. We started out by just showing the input values from the sensors in our pilot study. When we expanded the application after the initial tests we wanted to use the sensor input to further enhance the user experience. We ended up with using two features directly in the application, the accelerometer and the light sensor. The accelerometer was used to register if the device was shaking, which probably meant that the user was on the move, for example running or walking fast. In these cases, the user interface was automatically changed to a much simpler view which has bigger buttons and is easier to use when on the move.

The second sensor we used in our experiment was the light sensor. By constantly registering the lighting levels in the room we adjusted the background color of the application (Fig. 2). We changed the background color of the application very carefully, as it would be very annoying for the users if color changes were happening often and were drastic. Accordingly, color was gradually faded when the lighting values measured from the environment changed.

Fig. 2. Interface adaptation to ambient light

3 Evaluation Results

The developed prototype was evaluated in two phases. In the first, a pilot test was performed with a total of 12 users. These users were of mixed age, gender and computer expertise. The results from this phase were fed back into the development loop, as well as helped remove some unclear questions in the questionnaire. In the second phase, the main evaluation, another 40 people participated. Out of the 40 participants in the main evaluation, two did not complete the questionnaire afterwards and were therefore removed making the total number of participants 38 in the main evaluation. All 50 participants were aged between 20 and 55 years old, had previous knowledge of mobile phones and mobile communication, but had not previously used the type of application employed in our experiment. None of the pilot test users participated in the main evaluation.

Table 1. Evaluation results

	Statement	**Mean**	**Std. Dev.**
User Interface			
1.	It is easy to see the available functions	3.50	0.51
2.	The features of the application are hard to use	1.89	0.73
3.	The adaptability of the application is a feature I approve	3.45	0.55
Sensor Integration			
4.	The background color in the application changes when the lighting in the room changes	3.55	0.65
5.	When moving around, a simplified user interface is not presented	2.11	1.06
6.	I find sensor integration annoying and would disable it on my device	1.84	0.72
Context Awareness			
7.	The close integration with Google services is an inconvenience, I am not able to use the system without changing my existing or creating a new e-mail account at Google	1.76	0.88
8.	Calendar appointments displayed matched my current user context	3.58	0.55
9.	The contacts displayed did not match my current user context	1.29	0.52
10.	I would like to see integration with other online services such as online editing tools (for example Google Docs) and user messaging applications (like Twitter and Google Buzz)	3.29	0.73

The questionnaire that was employed in the second phase had three different parts, dealing with the user interface, sensor integration, and context-awareness, respectively, and the evaluation results are summarized in Table 1. Edwards [4] argued that such tailoring of data and sharing of contextual information would improve user interaction and eliminate manual tasks. Results from the user evaluation support this. Users find it both attractive as well as have positive attitudes towards automation of tasks such as push updates of information by tailoring the interface.

4 Conclusions and Future Work

Sensors are an important source of information input in any real world context. The work presented in this paper has shown that it is feasible to implement sensor-based context-aware integration through a suitable interplay between on-device context-aware information, such as sensors, and cloud-based context-aware information such as calendar data, contacts and applications, building upon suggestions for further research on adaptive cloud behavior as identified in [2,3].

By taking advantage of the rich hardware available on modern smartphones, the developed application is able to have tighter and more comprehensively integrated sensors in the solution. From user evaluation one can learn that although sensor integration as a source for context-awareness is well received, there is still research to do. In particular this has to do with the fact to what extent what thresholds should be used for sensor activation and deactivation. We have shown that it is feasible to implement sensors and extend their context-aware influence by having them cooperate with cloud-based services. Future interesting work could investigate to what extent what thresholds should be used for sensor activation and deactivation and explore if there are differences in people's perceptions of different sensors.

References

1. Binnig, C., Kossmann, D., Kraska, T., Loesing, S.: How is the weather tomorrow?: towards a benchmark for the cloud. In: Proceedings of the Second International Workshop on Testing Database Systems. ACM, Providence (2009)
2. Christensen, J.H.: Using RESTful web-services and cloud computing to create next generation mobile applications. In: Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications. ACM, Orlando (2009)
3. Dey, A., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: 1st International Symposium on Handheld and Ubiquitous Computing (1999)
4. Edwards, W.K.: Putting computing in context: An infrastructure to support extensible context-enhanced collaborative applications. ACM Transactions on Computer-Human Interaction (TOCHI) 12, 446–474 (2005)
5. 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 (2008)
6. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. National Institute of Standards and Technology, Special Publication 800-145 (2011)
7. Parviainen, M., Pirinen, T., Pertilä, P.: A Speaker Localization System for Lecture Room Environment. In: Renals, S., Bengio, S., Fiscus, J.G. (eds.) MLMI 2006. LNCS, vol. 4299, pp. 225–235. Springer, Heidelberg (2006)