Automatic Configuration of Mobile Applications Using Context-Aware Cloud-Based Services

Tor-Morten Grønli[1], Gheorghita Ghinea[2], Muhammad Younas[3], and Jarle Hansen[4]

[1] Norwegian School of IT, 0185 Oslo, Norway
tmg@nith.no
[2] Brunel University, London, UK
george.ghinea@brunel.ac.uk
[3] Oxford Brooks University, Oxford, UK
m.younas@brookes.ac.uk
[4] Systek AS, Oslo, Norway
jarle@jarlehansen.net

Abstract. The area of information technology continues to experience considerable progress and innovation in recent years. Computers have evolved from large and very expensive devices, to mainstream products we take for granted in our everyday lives. Increasingly, cloud-based services have come to the fore. Additionally, many people own multiple computing devices, from normal desktop computers to small mobile devices. We find mobile computing devices of particularly interest and this will be the focus of our study. In this chapter, we investigate a context-aware and cloud-based adaptation of mobile devices and user's experience. Our research displays new and novel contribution to the area of context-awareness in the cloud setup. We propose and demonstrated principles in implemented applications, whereby context-aware information is harvested from several dimensions to build a rich foundation for context-aware computation. Furthermore, we have exploited and combined this with the area of cloud computing technology to create a new user experience and a new way to invoke control over user's mobile phone. Through a developed application suite with the following evaluation, we have shown the feasibility of such an approach. Moreover, we believe our research, incorporating remote, and automatically configuration of Android phone advances the research area of context-aware information.

1 Introduction

We believe that mobile devices, especially in the last few years, have evolved considerably. Not only have they increased performance, they but also provide more features than before, such as the new and improved touch screens. However, in addition to the opportunities with mobile devices, they also present new challenges that are not present in standard desktop computing. Energy consumption, varying network coverage, and the relatively small screen sizes are examples of this. Moreover, in 2011 for

© Springer International Publishing Switzerland 2015
F. Xhafa et al. (eds.), *Modeling and Processing for Next-Generation Big-Data Technologies*,
Modeling and Optimization in Science and Technologies 4, DOI: 10.1007/978-3-319-09177-8_15

the first time smartphones exceeded PCs in terms of devices sold[1], which further highlights the importance of mobile devices and in this case smartphones specifically.

Innovations in hardware capabilities open up new opportunities and challenges when developing systems that run on or integrate with mobile devices. When combined with the wireless capabilities of high-speed Internet through EDGE, 3G, and 4G as well as Bluetooth and WLAN, many new research possibilities appear. Furthermore, with the updated network infrastructure and more affordable payment options from the ISPs (Internet Service Providers), the *always-connected* devices are becoming mainstream.

While smartphones are becoming increasingly powerful, the software they run has also gone through some major evolutionary steps. Particularly the Android and iOS marketplaces have been a very important factor in the platforms' success. In 2011 both Android Market[2] and the App Store[3] reached over 500,000 available applications. The maturity of the platforms and the popularity of apps are giving businesses a new channel to promote products, offer new features, and generally expand their methods of reaching out to potential customers.

Moreover, the ability to purchase and download native applications directly to the smartphones has proven to be a popular service for both consumers and developers. Developers are able to publish their applications quickly and users can navigate through a library consisting of many thousands of applications, providing everything from games and educational software to enterprise solutions. Additionally, the rating systems also provide end users with the ability to directly give feedback on the quality and price of the offered application. These features have certainly made people use their phone for more tasks than before. With the increase in usage and capabilities of the smartphones, new and interesting research opportunities have emerged. These include context-aware solutions and applications that have been around for some time now and have, successfully enriched mobile applications. In this work, we seek to build on these achievements and utilize context as a source of information for user information and interface tailoring [14]. The issue with much of the earlier approaches is that they have only looked at either one source of context-aware information or treated the context separately in the case of multiple sources. We propose a different approach, which combines context-aware information from several dimensions in order to build a rich foundation to base our algorithms on (what algorithms...this is not clear). We exploit cloud computing technology to create a new user experience and a new way to invoke control over user's mobile phone. Our solution is a remote configuration of an Android phone, which uses the context-aware information foundation to constantly adapt to the environment and change in accordance with the user's implicit requirements. Cloud-based service providers and developers are increasingly looking toward the mobile domain, having their expectations focused on

[1] http://www.smartplanet.com/blog/business-brains/milestone-more-smartphones-than-pcs-sold-in-2011/21828

[2] http://www.research2guidance.com/android-market-reaches-half-a-million-successful-submissions/

[3] http://www.apple.com/iphone/built-in-apps/app-store.html

the access and consumption of services from mobile devices [13]. Hence, integrating an application, running on a mobile device, with cloud computing services is becoming an increasingly important factor. Potentially, by utilizing such connectivity to offload computation to the cloud, we could greatly amplify mobile application performance at minimal cost [3]. Our work focuses on *data access transparency* (where clients transparently will push/pull data to/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].

The main contribution of our work in this chapter describes a cross-source integration of cloud-based, context-aware information. This solution incorporates remote, web-based configuration of smartphones and advances the research area of context-aware information and web applications. By expanding and innovating our existing work we propose a new, novel solution to multidimensional harvesting of contextual information and allow for automatic web application execution and tailoring.

2 Concepts and Background

Mobile devices are integrated with an increasing number of tasks in our day-to-day activities. Not only is society very reliant on the devices themselves, but also the infrastructure. One relevant example in this context is an application developed by British Airways for handling airplane tickets. This application replaces, for many destinations, the check-in process and paper boarding passes completely[4]. A screenshot of the application is presented in Figure 1.

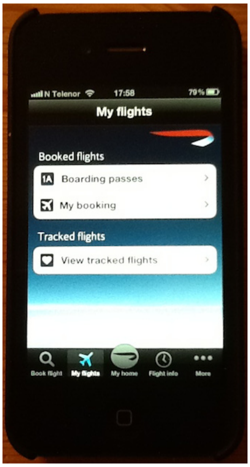

Fig. 1. British Airways iPhone app

[4] `http://www.britishairways.com/travel/iphone-app/public/en_gb`

Because mobile devices are usually carried around everywhere and have the capability to communicate with external resources, they provide an ideal match for these kinds of applications. It replaces other items and tasks, like boarding passes and check-in procedures, with a simple and well-integrated application that is more practical. Moreover, one does not need to print out a boarding pass or stand in queue at the check-in counter.

These features of the mobile devices are a result of many different components cooperating. Most applications communicate with various resources; these can be local to the phone, like sensors, or backend services that provide the wanted information. There are several research areas that have made significant contributions toward these technological advances we are able to use today. These areas are presented in the next section of this chapter. We will concentrate on four concepts that are particularly important when it comes to mobile devices and the integration of network communication, namely 1) *Distributed Computing,* 2) *Mobile Computing,* 3) *Pervasive Computing,* and 4) *Internet of Things.*

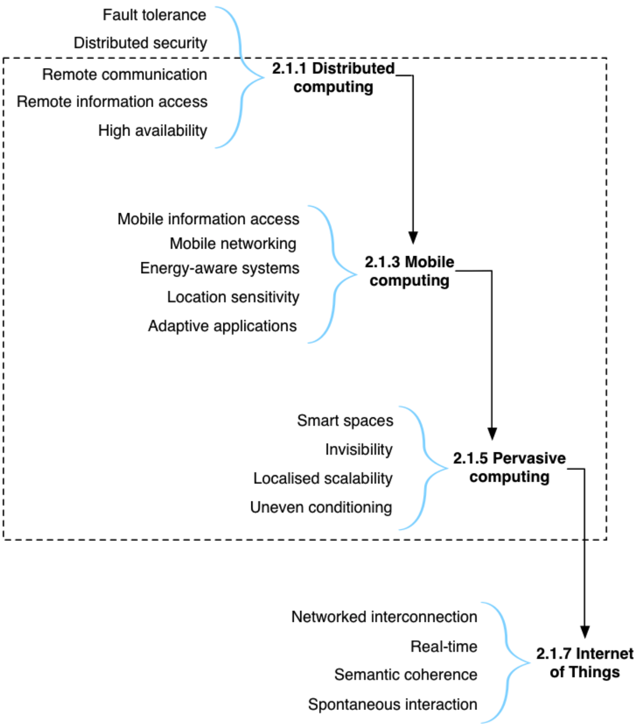

Fig. 2. Distributed Computing, Mobile Computing, Pervasive Computing, and Internet of Things overview. Adapted from Satyanarayanan [17] and Zhang et al.

As presented in Figure 2, these majors steps in mobility and computing are related both in regards to technology and research challenges. When moving toward the right of the figure, there is a tendency to either add new problems or make existing ones more challenging [17]. On the figure, we have marked (with a stapled box) the most important issues that are investigated in this chapter.

Although new problems appear with the paradigms on the right, it is important to build on previous research findings. Distributed computing has a considerable knowledge base that helps both mobile and pervasive computing moving forward. Internet of Things is also building on knowledge learned from the previous paradigms. We will not go into detail on the Internet of Things, as this is outside the scope of our work, but we include a general description to complete the overall picture of the four research areas.

2.1 Distributed Computing

Up to the late 1970s / early 1980s computing systems were usually centralized mainframe installations and access to these was done through a text terminal [19]. When the Personal Computer (PC) started to enter the mainstream market it provided local workstations. However, these devices usually provided limited communication options. Since the 1990s, when the Internet became a commercial success, the computers have been connected in a large network. With the communication options and availability of the Internet, it was possible to share resources, like servers and printers, and enable a high degree of information sharing between users. With the Internet becoming popular, these local computers started to connect to external resources, and the initial distributed systems appeared.

A distributed system is (Tanenbaum & Van Steen [21]):

"… a collection of independent computers that appears to its users as a single coherent system".

This definition consists of two parts, which are the hardware and software. The hardware must cooperate to complete specific tasks, while the software should try to unify these hardware resources into one coherent system. Distributed computing thus takes advantage of networked resources, trying to share information, or even have different resources join forces to be able to achieve complex tasks that might take too long for just one standalone computer. Compared to implementing a single-machine system, distributed systems have their own challenges and issues.

One important concept within distributed computing is the remote network communication between devices. Different components of the system need to communicate, this is at the core of the distributed computing area. There are several attempts at making the development and overall design of such systems easier and more reliable from RPC to peer-to-peer communication and remote method invocation.

2.2 Mobile Computing

Mobile computing started with the appearance of the laptop computers and WLANs (Wireless LANs) in the early 1990s [17]. It emerged from the integration of cellular technology with the Web [20]. Mobile computing, as the name suggests, is the process of computation on a mobile device.

Network communication is a very important part of mobile computing. Usually, communication between the device and external resources is involved, where a device can request a specific set of services from back-end servers or other devices. One of the main goals of mobile computing is to offer mobility and computing power. This can be offered by providing decentralized and distributed resources on diversified mobile devices, systems, and networks that are interconnected via mobile communication standards and protocols [22]. This aspect is very important and we will address this in more detail in several of our experiments.

Creating a distributed system for mobile devices is usually even more challenging than the standard client-server architecture, where the client is situated on a desktop computer. It is difficult to create systems where there is no fixed infrastructure available and devices can enter or leave the system at any time. Also, devices can be disconnected due to issues such as an empty battery or poor network coverage. Satyanarayanan [23] identified four constraints that characterise mobile computing:

1) Mobile elements are resource-poor relative to static elements. Desktop computers and servers will usually have more hardware resources than laptop computers, smartphones, and especially lower-end mobile phones. With less hardware resources, certain considerations have to be taken when developing the systems, both on the server and client side. There are constraints specific to the applications that are created to run on mobile devices, which would not be a consideration for normal desktop clients. Energy-consumption and varying network coverage are two examples.

2) Mobility is inherently hazardous. Security in mobile computing is considerably more difficult than in environments that have a fixed infrastructure. In mobile computing the devices are *on the move* and used in many different locations and settings. This increases the possibility of theft or to simply misplace the devices. Another challenge is privacy. Smartphones often record various data about the user and the device, for instance the location. It is crucial that this kind of sensitive information is not accessible by anyone other than the authorized users.

3) Mobile connectivity is highly variable in performance and reliability. Moving from an environment with good network coverage and bandwidth to environments that offer only low speed connections or no network connection at all is an important factor the systems must handle. There are also differences in the hardware capabilities of the devices. Some devices might only offer slow wireless connections, like EDGE, while others have the capability to connect to faster networks, with, for example, 4G. If the system does not recognize and adapt to these differences, it can impact the user experience. For example, if a system sends high-quality video to a device with a very limited wireless connection, the result is long loading times and a poor user experience.

4) Mobile elements rely on a finite energy source. Battery technology has improved over time, but it still remains one of the main challenges for mobile computing. Although hardware components are created to be vey energy efficient, the smartphones are often equipped with large touchscreens. These large screens use a considerable amount of power. The concern for power consumption must span many levels of hardware and software to be fully effective. Applications should thus be able to adapt the content and settings on the device according to the battery level.

One common example of a feature that is implemented to minimize the battery usage is the light sensor, which registers the amount of light in the room and adjusts the screen brightness accordingly.

Based on these characteristics of mobile computing, Satyanarayanan [17] identified five main research challenges. These are *Mobile Information Access, Mobile Networking, Energy-Aware Systems, Location Sensitivity,* and *Adaptive Applications*, all of which form the bulk of research interest in the area today.

2.3 Pervasive Computing

The initial move towards pervasive computing started in the 1970s, when the PC brought computers closer to people [20]. However, it was not until the early 1990s the idea of truly ubiquitous/pervasive computing started to take shape. In 1991, Mark Weiser wrote about what he envisioned the *computer for the twenty-first century* would be like. The work was based on research done at Xerox, where staff had working prototypes of what they called *ubiquitous computing*. The terms *ubiquitous* and *pervasive* computing are used interchangeably throughout this chapter. The idea was that computers blend into the environment, to the point where people would no longer notice their presence. Pervasive computing thus consists of a new class of devices that make information access and processing available for everyone from everywhere at any time.

Mark Weiser [24] stated that: "Prototype tabs, pads and boards are just the beginning of ubiquitous computing. The real power of the concept comes not from any one of these devices – it emerges from the interaction of all of them." When looking at the current state of mobile computing it is quite impressive how this vision is getting increasingly closer to reality. Mobile phones, tablets, and e-readers are now very much a part of our everyday lives. Pervasive computing provides a rich diversity of applications and can connect to worldwide networks, thereby, providing access to a wealth of information and services. Pervasive computing builds on mobile computing, but adds characteristics such as transparency, application-aware adaptation, and an environment sensing ability [25].

In many situations, mobile devices require the ability to communicate with each other. This leads to a close connection of both distributed and mobile computing with pervasive computing. Pervasive systems require support for *interoperability*, *scalability*, *smartness*, and *invisibility* to ensure that users have seamless access to computing [20]. All of these features and advantages create new challenges that did not exist before [25]. In some cases, research done in distributed and mobile computing can be applied directly to the pervasive computing area. For others, the demands of pervasive computing are sufficiently different that new solutions have to be sought [17].

One important challenge in this research area is privacy. Privacy in pervasive computing environments can be a very difficult problem to solve [26]. According to Satyanarayanan [17], privacy in pervasive systems is greatly complicated when moving from distributed and mobile computing. Tracking location and other possibly sensitive data resources can cause considerable challenges. Not only can this cause a problem technically, but also from a user acceptance perspective.

Another challenging aspect is in the hardware domain. Satyanarayanan [17] and West [26] mention the issue that mobile devices are becoming increasingly smaller and placing severe restrictions on battery capacity. One example of research done in this field is a system developed by Parkkila and Porras [28]. They use a method called *cyber foraging*, where the idea is to use nearby computers in the same local network to handle the heavy tasks.

3 Internet of Things and Cloud

Internet of Things (IoT) is a networked interconnection of everyday objects and is rapidly gaining popularity, thanks in part to the increased adoption of smartphones and sensing devices [29]. Zhang et al. [18] categorises the Internet of Things with four main challenges:

1. **Networked interconnection** - all physical objects must be mapped into the Internet.
2. **Real-time** - requires real-time searching techniques.
3. **Semantic coherence** - recognize objects accurately and support a light-weight representation to accommodate semantics across smart spaces.
4. **Spontaneous interaction** - handle each interaction in an efficient manner such that the entire system is scalable and real-time.

It is evident that Internet of Things builds on pervasive computing, where words like *smart spaces, scalability,* and *interconnection* are all described in the previous sections. In other words, IoT further enhances pervasive computing through the communication among physical objects (or things) which have computing capabilities as well as physical attributes [16]. However, like the other research directions, Internet of Things adds its own challenges and opportunities.

One of these issues is scalability. As reported by BBC[5], the number of devices connected to the Internet is expected to reach 15 billion by the year 2015. This increase in network connected devices causes problems for the infrastructure, with issues like IPv4 addresses quickly running out. Even though the new IPv6, that was approved as early as 1998, is currently being pushed out to companies, it is a slow process. IPv6 is a network layer protocol, which was designed to increase the address space for nodes within the Internet [30].

Another major issue is the resource scarcity of things in IoT, as unlike classical/standard computing systems, things (e.g.,) may not have high processing and computing capacities. However, there is a growing trend of exploiting cloud computing capabilities in order to cater for resource scarcity in IoT. The idea of cloud is built around an economy of scale and the provision of more resources, better scalability and flexibility of service provision [1] . Cloud computing tends toward computing as a service [9] that deals with the utilization of reusable fine-grained components across a vendor's network, and cloud-based services are usually billed using a pay-per-use model [15]. Large IT companies like Microsoft, Google, and IBM, all have initiatives relating to cloud computing which have spawned a number of emerging research

[5] `http://www.bbc.co.uk/news/technology-13613536`

themes, among which we mention: *cloud system design* (Mell and Grance, 2011), *benchmarking of the cloud* [10], and *provider response time comparisons*. Mei et al. [11] have pointed out four main research areas in cloud computing that they find particularly interesting is the *Pluggable computing entities, data access transparency, adaptive behavior of cloud applications and automatic discovery of application quality.*

The Internet of Things is an up and coming area, which will undoubtedly attract more research interest in the near future. The popularity is increasing, as shown by Google trend graph searches.

4 Design and Implementation

This research looks into the utilization of web resources for tailoring of the user experience. Earlier approaches have typically looked at one source of context-aware information or, in case of more than one source, the information is utilized separately. Based on earlier work of ours [7],[8], we propose a different approach, where we combine context-aware information from several dimensions to build a rich foundation to base our algorithms on. This will allow for cloud computing to be used to create new user experiences and new ways to invoke control over a smartphone. With this a basis, it is possible to have an always adapting user interface.

As a part of our solution we chose the cloud computing platform in order to have a feature rich, scalable, and service-oriented server framework. Traditional REST framework services were considered, but found to be insufficient in terms of scalability and extensibility, i.e., to add and remove context-aware sources in an ad hoc manner. The cloud-based approach also has the advantage of being run as a platform as a service instance in the separate hosting instance of Google App Engine.

For our user experiment we implemented an application suite, a fully functional demonstration of the system. One of the main technical goals of our system is to make the interaction between the cloud and the mobile device as seamless as possible for the user.

The system was designed with three major components: an Android client, a cloud server application, and the remote Google services. Figure 1 gives an overview of the implementation of the system. The blue (or shaded) boxes in the diagram represent the parts of the system we created). The white boxes, like Google calendar and contacts, are external systems the system communicates with. The server application was deployed remotely in the cloud on the Google App Engine, while data was also stored remotely in Google cloud services.

After the Android client was instaled on the mobile device, the device will register itself to the Google. The users would start by logging in to the webpage. This webpage is part of the server application hosted on the Google App Engine. The login process uses the Google username/password. By leveraging the possibilities with Open Authorization (OAuth) the system provides the user with facility of sharing their private calendar appointments and contacts stored in their Google cloud account without having to locally store their credentials. OAuth allowed us to use tokens as means of authentication and enabled the system to act as a third-party granted access by the user.

After a successful authentication the user is presented with a webpage showing all configuration options. Because the configuration for each user is stored in the cloud, the system avoided tying it directly to a mobile device. One of the major benefits of this feature is that the user did not need to manually update each device; users have a "master configuration" stored externally that can be directly pushed to their phone or tablet. It is also easier to add more advanced configuration options when the user can take advantage of the bigger screen, mouse, and keyboard on a desktop/laptop PC for entering configuration values than those found on mobile devices. On the webpage, by selecting the applications user wants to store on the mobile device and pressing the "save configuration"-button, a push message is sent to the client application.

4.1 Cloud to Device Messaging

The system exploits the push feature of Android 2.2 in order to send messages from cloud to devices, i.e., the C2DM (Cloud to Device Messaging). The C2DM feature requires the Android clients to query a registration server to get an ID that represents the device. This ID is then sent to our server application and stored in the Google App Engine data store. When a message needs to be sent, the "save configuration"-button is pushed. We composed the message according to the C2DM format and sent it with the registration ID as the recipient. These messages are then received by the Google C2DM servers and finally transferred to the correct mobile device.

The C2DM process is visualized in Figure 2. This technology has a few very appealing benefits: messages can be received by the device even if the application is not running; saves battery life by avoiding a custom polling mechanism; and takes advantage of the Google authentication process to provide security.

Our experience with C2DM was mixed. It is a great feature when you get it to work, but the API is not very developer friendly. This will most likely change in the future since the product is currently in an experimental state, but it requires the developer to work with details like device registration and registration ID synchronization. Although C2DM does not provide any guarantees when it comes to the delivery or order of messages, we found the performance to be quite good in most of the cases. It is worth mentioning that we did see some very high spikes in response time for a few requests, but in the majority of cases the clients received the responses within about half a second. Performance measurements (we recorded while doing the user experiments) reported an average response value of 663 milliseconds. It is also important to note that issues like network latency will affect the performance results.

The calendar and contacts integration was also an important part of the Android application. We decided to allow the Android client to directly send requests to the Google APIs instead of going the route through the server. The main reason for this is that we did not think the additional cost of the extra network call was justified in this case. The interaction is so simple and there is very little business logic involved in this part so we gave the clients the responsibility for handling it directly. The implementation worked by simply querying the calendar and contact API and then using XML parsers to extract the content.

4.2 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.

5 Prototype and Evaluation

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.

The questionnaire that was employed in the second phase had three different parts, dealing with the *web application*, *context-awareness*, and *cloud computing,* respectively, in which participants indicated their opinions on a 4-point Likert scale anchored with *strongly disagree(SD)/ disagree (D)/ agree(A)/ strongly agree(SA)*. Evaluation results are summarized in Table 1.

Table 1. User evaluation questionnaire and results

Statement	Domain	Mean	Std. Dev.
Web application			
S1	I was able to register my device application configuration in the web application	3.61	0.59
S2	I was not able to store and push my configuration to my mobile device from the web page	1.47	0.80
S3	We would like to configure my phone from a cloud service on a daily basis, (webpage user config and Google services like mail/calendar/contacts)	3.18	0.69

Table 1. (*continued*)

Context-awareness			
S4	The close integration with Google services is an inconvenience. We are not able to use the system without changing my existing or creating a new e-mail account at Google	1.76	0.88
S5	Calendar appointments displayed matched my current user context	3.58	0.55
S6	The contacts displayed did not match my current user context	1.29	0.52
S7	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 Plus)	3.29	0.73
Cloud computing			
S8	I do not mind Cloud server downtime	2.08	0.78
S9	I do not like sharing my personal information (like my name and e-mail address) to a service that stores the information in the cloud	2.16	0.79
S10	Storing data in the Google Cloud and combining this with personal information on the device is a useful feature.	3.26	0.60
S11	I find the cloud-to device application useful	3.53	0.51

5.1 Web Application

The statements dealing with the web application at Google App Engine (Figure 3) show that the web application performed as expected, by letting participants register their devices as well as pushing performed configurations to the devices. Also answers from S3 are quite interesting, highlighting a positive attitude toward cloud-based services.

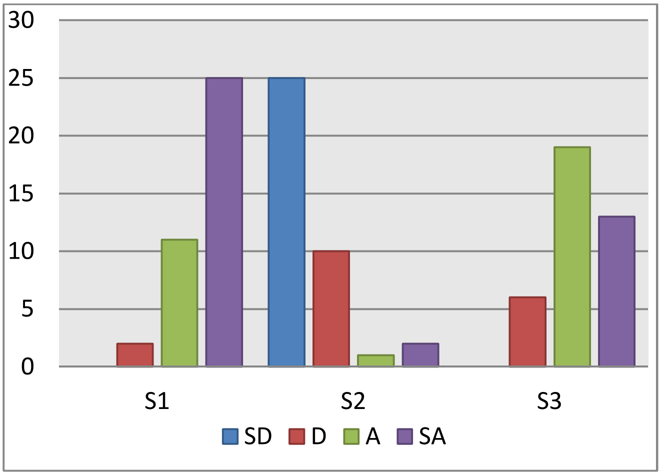

Fig. 3. Web application statements

5.2 Context-Awareness

In terms of context-aware information, participants were asked to take a stand in respect of four statements, with results shown below (Figure 4). For the first statement (S4), although a clear majority supported this assertion, opinions were somewhat spread and this answer was not statistically significant. For the next two statements a very positive bias was registered, indicating correctly computed context-awareness and correct presentation to the users. Again for S7, users are eager to see more cloud-based services and integration.

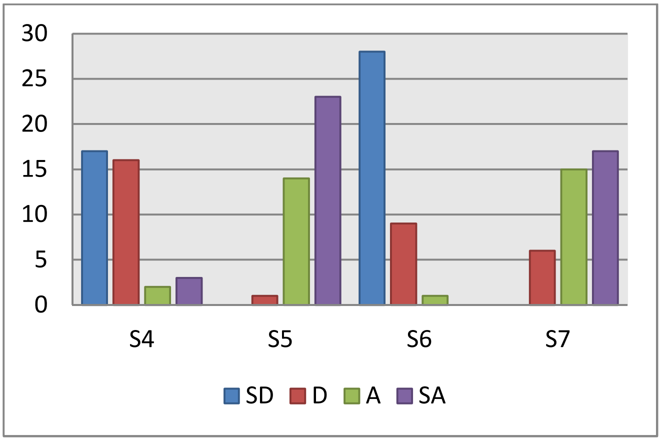

Fig. 4. Context-awareness statements

5.3 Cloud Computing

When inspecting results from the cloud-computing section (Figure 5), results are mixed and differences in opinions do occur. For S8 and S9 the results are not statistically significant, but they indicate a mixed attitude toward cloud vulnerability and cloud data storage. The two statements with statistically significant results, S10 and S11, participants find storage of data in the cloud and using this as part of the data foundation for the application a useful feature and are positive toward it. Their answers also suggest a fondness for push-based application configuration.

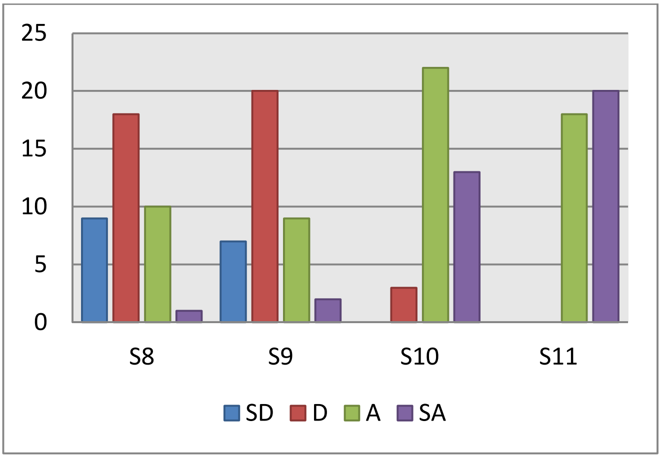

Fig. 5. Cloud computing statements

6 Analysis and Discussion

From the literature we point at the ability for modern applications to adapt to their environment as a central feature [4]. Edwards [5] 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. The users find it both attractive as well as have positive attitudes toward automation of tasks such as push updates of information by tailoring the interface. This work has further elaborated on context-aware integration and has shown how it is possible to arrange interplay between 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 by Christensen [2] and Mei et al. [10][11].

To register the tags the standard Google Calendar and Contacts web interface were used. Such a tight integration with the Google services and exposure of private information was not regarded as a negative issue. As shown in the results, most of the users surveyed disagreed that this was an inconvenience. This perception makes room for further integration with Google services in future research, where, among them, the Google+ platform will be particularly interesting as this may bring opportunities for integrating the social aspect and possibly merge context-awareness with social networks.

Sensors are an important source of information input in any real-world context and several previous research contributions have looked into this topic. The work presented in this chapter follows in the footsteps of research such as that of Parviainen et al. [31], and extends sensor integration to a new level. 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. Although sensor integration as a source for context-awareness is well received, it still needs to be father enhanced. In particular it would be useful to find out appropriate extent and thresholds that 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 in a cross-source web application scenario. Further research includes investigating sensor thresholds and the management of different sources by different people in a web scenario.

In this chapter we investigated into context-aware and cloud-based adaptation of mobile devices and user's experience. Our research has added a new and novel contribution to the area of context-awareness in the cloud setup. We have proposed and demonstrated principles in implemented applications, whereby context-aware information is harvested from several dimensions to build a rich foundation on which to base our algorithms for context-aware computation. Furthermore, we have exploited and combined this with the area of cloud computing technology to create a new user experience and a new way to invoke control over user's mobile phone. Through a developed application suite, we have shown the feasibility of such an approach, reinforced by a generally positive user evaluation. Moreover, we believe our solution, incorporating remote and automatically configuration of Android phone advances the research area of context-aware information.

7 Future Research

It can be very expensive to write separate applications in the native language of the platforms. Therefore, when developing multi-platform systems there is also the possibility to use HTML 5 technology to create a shared application. While we acknowledge this fact, we also see that in today's environment there are scenarios where HTML 5 does not provide the wanted performance requirements or lacks specific features. In the topic of heterogeneity, we see the potential for future research providing more detail on this idea of a common platform, such as a closer investigation of HTML 5 compared to native applications. Looking at large cloud-computing providers, they have access to an enormous amount of data, and the lack of transparent knowledge on how this information is used has provoked concerns. This would certainly be an interesting topic for future research on the topic of cloud computing acceptance in regard to personal information sharing.

From several scenarios described in this chapter, we see that future research should continue to innovate and expand the notion of context-awareness enabling further automatic adaptation and behavior altering in accordance with implicit user's needs.

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] Cidon, A., et al.: MARS: adaptive remote execution for multi-threaded mobile devices. In: Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, MobiHeld 2011, pp. 1:1–1:6. ACM, New York (2011)
[4] Abowd, G.D., Dey, A.K.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999)
[5] 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)
[6] Elsenpeter, R.C., Velte, T., Velte, A.: Cloud Computing, A Practical Approach, 1st edn. McGraw-Hill Osborne Media (2009)
[7] Grønli, T.-M., Hansen, J., Ghinea, G., Younas, M.: Context-Aware and Cloud Based Adaptation of the User Experience. In: Proceedings of the 2013 Advances in Networking and Applications (AINA), pp. 885–891. IEEE Computer Society (2013)
[8] Grønli, T.-M., Ghinea, G., Younas, M.: Context-aware and Automatic Configuration of Mobile Devices in Cloud-enabled Ubiquitous Computing. Journal of Personal and Ubiquitous Computing (2013)
[9] Khajeh-Hosseini, A., et al.: The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise. Software: Practice and Experience, Software: Practice and Experience 42(4, 4), 447–465 (2012)
[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 (2008)
[11] Mei, L., Zhang, Z., Chan, W.K.: More Tales of Clouds: Software Engineering Research Issues from the Cloud Application Perspective. In: Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference (2009)
[12] Mell, P., Grance, T.: The NIST Definition of Cloud Computing (2011)
[13] Paniagua, C., Srirama, S.N., Flores, H.: Bakabs: managing load of cloud-based web applications from mobiles. In: Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services, iiWAS 2011, pp. 485–490. ACM, New York (2011)
[14] Strobbe, M., Van Laere, O., Ongenae, F., Dauwe, S., Dhoedt, B., De Turck, F., Demeester, P., Luyten, K.: Integrating Location and Context Information for Novel Personalised Applications. IEEE Pervasive Computing, 1 (2011)

[15] Vaquero, L.M., et al.: A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. 39(1), 50–55 (2008)
[16] Vermesan, O., et al.: Internet of Things Strategic Research Roadmap. European Research Cluster on the Internet of Things, Cluster Strategic Research Agenda (2009)
[17] Satyanarayanan, M.: Pervasive computing: vision and challenges. IEEE Personal Communications 8(4), 10–17 (2001)
[18] Zhang, D., Yang, L.T., Huang, H.: Searching in Internet of Things: Vision and Challenges. In: 2011 IEEE 9th International Symposium on Parallel and Distributed Processing with Applications (ISPA), pp. 201–206 (2011)
[19] Boger, M.: Java in Distributed Systems: Concurrency, Distribution and Persistence, 1st edn. Wiley (2001)
[20] Saha, D., Mukherjee, A.: Pervasive Computing: A Paradigm for the 21st Century. Computer 36(3), 25–31 (2003)
[21] Tanenbaum, M., Van Steen, A.: Distributed Systems: Principles and Paradigms. Prentice Hall (2002)
[22] Kamal, R.: Mobile Computing. Oxford University Press, USA (2008)
[23] Satyanarayanan, M.: Fundamental challenges in mobile computing. In: Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, PODC 1996, pp. 1–7. ACM, New York (1996)
[24] Weiser, M.: The computer for the 21st century. Scientific American 3(3), 3–11 (1991)
[25] Hansmann, U., et al.: Pervasive Computing: The Mobile World, 2nd edn. Springer (2000)
[26] West, M.T.: Ubiquitous computing. In: Proceedings of the 39th ACM Annual Conference on SIGUCCS, SIGUCCS 2011, pp. 175–182. ACM, New York (2011)
[27] West, M.T.: Ubiquitous computing. In: Proceedings of the 39th ACM Annual Conference on User Services Conference, SIGUCCS 2011, pp. 175–182. ACM, New York (2011)
[28] Parkkila, J., Porras, J.: Improving battery life and performance of mobile devices with cyber foraging. In: 2011 IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), pp. 91–95 (2011)
[29] Patel, P., et al.: Towards application development for the internet of things. In: Proceedings of the 8th Middleware Doctoral Symposium, MDS 2011, pp. 5:1–5:6. ACM, New York (2011)
[30] Perkins, C.E.: Mobile networking in the Internet. Mob. Netw. Appl. 3(4), 319–334 (1998)
[31] 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)