A Multidimensional Approach to Context-Awareness

Kevin GOSLAR, Sven BUCHHOLZ, Alexander SCHILL
Department of Computer Science, Dresden University of Technology,
D-01062 Dresden, Germany

and

Hartmut Vogler
Corporate Research Center, SAP Labs, Inc., 3475 Deer Creek Road,
Palo Alto, CA 94304, USA

Abstract

This paper presents multidimensional context-aware principles for Enterprise Applications.
The usage of information about the environment of an application, device or its user can enhance the functionality of applications. Information about the user as a subset of contextual information enables applications to cultivate a relationship with their human users, which allows individual support of individual users.
This paper introduces a view to different dimensions of context. As contextual dimensions we define different aspects of the environment, which are independent of each other. Examples for contextual dimensions are the personality of the user, her location and the devices she carries.
Another contextual dimension, which this paper presents in detail, is the way in which contextual information is determined. Hard context-awareness is the measuring of contextual information using hardware sensors, while soft context-awareness is the inference of knowledge based on a context-aware reuse of the information base. We present ways to combine data of different applications and infer contextual information out of this data.
We view context-awareness from the perspective of Enterprise Applications and their special requirements. We present special characteristics of Enterprise Applications and how they influence the use of contextual information in enterprises.

Keywords: context-awareness, enterprise applications, information procurement, information reuse

1. Introduction

The utilization of contextual information allows context-aware applications a more detailed look into their environment. This enables them to "understand" the situation, processes and problems in which the user is involved and helps them to determine possibly helpful solutions for those problems on their own. This is an active support of the user by a context-aware application.

Such an interactive behavior is expected from a human partner. It will make us angry if somebody is focused only on her business and doesn't care about our situation. Such a team cannot develop proper solutions. Awareness of the other members and their situation is as a necessity in the interaction with humans. This raises the need for context-aware (CA) applications.

CA observes the physical environment of a device in order to allow that device to behave adequately for its current situation. An example is a mobile phone that knows it is inside an opera house and therefore automatically switches into a silent mode.
There have been several research projects on individual aspects of context-awareness, which have produced useful technologies in their respective domains. We present some of them in chapter 2. In this paper we try to combine some of those developed context-aware technologies to form a more comprehensive picture about the context of an application. We call this multidimensional context-awareness and present it in chapter 3. Our current prototype distinguishes 5 different contextual dimensions, which are the currently used applications, devices, locations, users and roles. We present possibilities to integrate context-aware enhancements into existing application frameworks. For this purpose, we enhanced the mySAP.com enterprise application framework with context-aware functionality and implemented a multidimensional context-aware information service application. We furthermore present the “down to planet earth” opinion of a commercial company for the use of such technologies.

We use soft context-aware principles in our prototype. Soft context-awareness is the reuse of the current information base in a context-aware manner in order to derive statements about the actual and previous physical context. It is a type of context-awareness that works without hardware sensors. We give a more detailed description of our concept of soft context-awareness in chapter 4.
We present our information service application in chapter 5. Chapter 6 contains the cognitions, which the development and use of our application revealed. Chapter 7 concludes the paper.

2. State of the Art

The research community identified many useful concepts for the complex topic of context and context-awareness. We are only going to name a few. Winograd ([1]) distinguishes between the environment of an application (which are all parameters that surround an application) and context as the subset of the environment that is relevant to a specific application. Context is therefore not a static view on the environment; it is rather dynamically seen dependent on the actual application. [1] also distinguishes between different levels of context. Low-level context is data that can be directly measured in the envi-

ronment using sensors; high-level context is inferred knowledge out of low-level contextual data.

There are several approaches for frameworks that allow a convenient access to contextual data. Dey and Abowd ([2]) created a framework for the development and execution of context-aware applications called Context Toolkit. It uses abstraction mechanisms called context widgets to encapsulate the access to environmental data and also allows the refinement of low-level contextual data into high-level knowledge about the context using inference mechanisms.

A lot of research has been done in the field of adaptive user interfaces. Eisenstein et al. [3] presents an approach that uses user-interface and task models to describe a dialog using generic components. The actual dialog is rendered out of platform-specific components using these models.

The CyberDesk project ([4]) is an early approach to incorporate contextual data in a desktop application. It provides additional features applicable to application data, e.g. the automatic picking of telephone numbers of people whose name is currently selected or the automatic lookup of selected phrases in web-based search engines.

ICrafter ([5]) is a framework that incorporates different services and devices in a pervasive computing environment. It provides an automatically generated user interface for cooperating services.

The problem with storing contextual information is that the context of a mobile application is subject to structural changes. This implies that the semantical interpretation of environmental data varies and cannot be assumed implicitly. E.g., an incoming call from your wife has to be interpreted different in the evening when watching TV than during a meeting with your boss. Sometimes incoming messages need your immediate attention and sometimes the same message can be read later without any problem. Therefore it is not sufficient to store only the contextual data and take the knowledge about its semantics for granted, as it is done in conventional databases. It is rather necessary to store the contextual data together with its semantic meaning, e.g. using semantic nets.

3. Multidimensional Context

The ideas of multidimensional context awareness presented in this paper are driven by the special requirements of enterprise applications (EA) and with the focus to enterprise applications. For our work we emanate from the following assumptions:

- Users of EA have a different technical background and education. A typical EA suite offers functionality on all levels in a company. Thus the system has to support the CEO of a company as well as a blue-collar worker, even when they are using exactly the same application such as an employee self-service to change the current contact address or entries in the benefits program.
- Since the maintenance of client software consumes a lot of time and money, today's enterprise software is accessible via standard HTML-browsers, which are part of the preinstalled operating system. Thus for the support of mobile applications we also aim at a 'zero-footprint' and just use the technologies, such as WML browsers, available on the device.
- The users access the system and applications with different devices. In the office they are using desktop computers whereas an employee on the road is using a mobile device instead. The interaction schema has to be adapted to the specific design of the device. This becomes even more crucial in case of pure telephony based access using VoiceXML [6].
- An enterprise application suite offers a large variety of applications. A user has to master a bunch of different applications. Some of them are used on a daily base (e.g. a time sheet application to record the working hours) hence the user is very familiar with it. Other applications are used just once a year, such as an employee self-service application for the benefit enrollment.
- Enterprise applications can be very complex. For example, a desktop sales order entry application can end up depending on the configuration with 83 entry fields. Such applications are optimized for power users who are trained.
- Depending on the application we have to support different levels of security. A time recording application for 'clock in/clock out' of work just needs a simple identification. Opposed to that, a notification about the newest sales data send by the business intelligence application to a mobile device has to use high-level security mechanism since these data can cost a company millions of dollars.
- Inside enterprise applications different roles are defined, which enclose for example different access rights. Users are very often assigned to different roles depending on their current tasks.

Having these heterogenous requirements for enterprise applications in mind, we propose to see and use contextual information in a multidimensional way. Let us start with an example where we have just two dimensions, the user's experience with a specific application and with a specific device. Users being familiar with a desktop environment don't need support how to use the mouse or the keyboard (e.g. drag and drop), but the system should assist with a complex application a user has not utilized before. On the other hand, in case a user is familiar with the application but has not used the specific device before, guidance by the application about the interaction schema (e.g. for the voice interface of an application) is essential. Even using an application the first time, the user might be familiar with some of the operations based on the experience with another application and thus doesn't need assistance in that particular area.

In our view, there are an unlimited number of dimensions for contextual information. In our first approach, we started with five dimensions. These are: Devices, Applications, Locations, User data, and Roles. Further dimensions that we haven't addressed and investigated in detail so far are security levels and time. Each dimension has various attributes, which can be seen as sub-dimensions.

In our model we separate the identification of a dimension, its stored information and the rules how an application makes use of this contextual information. For example, the identification of a location (such as the latitude/longitude coordinates) is independent from the description of a location (such as services or additional devices at a specific location) and the rules how the application should adapt to the current location of a user. Since we want to support multiple devices and applications, the common approaches to store contextual information either on the client or in the backend application is not applicable. Thus, we have to gather and store this information in a layer between the device and the application. Additionally, we need master data of the user and the application from the backend system, e.g. the user's master data. Since the backend system and its data model is not supposed to these kinds of queries (fast and non transactional search) we replicate parts of the master data also in the middle layer based on our own data model.

4. Hard and Soft Context-Awareness

Computer systems were invented to support processes in the real world. Context-awareness was conceived to enhance this support. For this reason, computer systems have been extended with sensors or sensor networks (like infrared beacons mounted inside every room, see [7]) that monitor the physical environment. We call these approaches *Hard Context-Aware*, because they extend existing systems with new hardware and infrastructure. Although hard context-awareness produces remarkable functionality, we doubt that such technology will be available in office environments, public areas or private households. The benefits in relation to the implementation effort are too unfavorable in the foreseeable future. Sensor network infrastructures will probably conquer laboratories, -secured areas, or hospitals where permanent localization of people is an essential issue. The lack of security and exchange standards for such systems makes widespread sensor networks in public areas a rather utopian scenario.

As mentioned before, computer systems were invented to support processes in the real world. This implies that they reflect the real world in their operational data. The evaluation of all available data could enable a soft context-aware agent to infer data about the context. We call the reuse of the information base in a context-aware manner *Soft Context-Awareness*, because it works only with software, no hardware sensors are required. E.g. scanning the email of a user might reveal potential to-do items for her. There are different rules that apply and different strategies to acquire information in both context-sensing technologies. Table 1 lists some characteristics of both types of context-awareness.

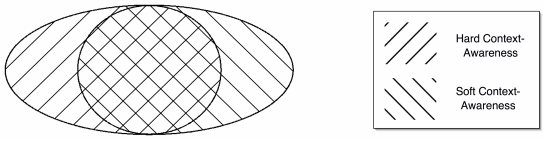

figure 1: A new view to context-awareness

It is obvious that hard and soft context-awareness are both subsets of context-awareness in general and that they complement each other. Furthermore exist many scenarios where both types of context-awareness apply. This situation is depicted in figure 1. Soft context-awareness does not work in all cases, but can produce useful results with little modifications of currently existing applications. It works best in comprehensive environments with plenty of information representing different parts of the physical context that can be reused. Examples for such an environment are Enterprise Application suites. Enterprise Application Suites are frameworks for Enterprise Applications and comprise solutions for common tasks in the administration of companies such as enterprise resource planning, supply chain

Table 1. Comparison of hard and soft context-awareness

	Hard Context-Awareness	**Soft Context-Awareness**
Input sensors	hardware sensors that measure physical parameters	software sensors that monitor the information base of local and remote services and applications
Identification	physical identification using biometrical sensors	logical identification using login data and passwords
Access rights	to enter physical domains (open doors), use devices	to access information (use applications and databases)
Processes of interest	physical activities (e.g. a meeting takes place in room x, user is driving in a car)	software processes (e.g. there is an incoming call from Jeff, the users schedule says she is in a meeting with her boss)
Tools of interest	which devices are available to the user	which applications/information/services are available to the user
Problems of interest	how can devices help the user to achieve her goals	how can applications help the user to achieve her goals
Status messages	showing the status of the user's devices (on/off)	showing the status of the user's applications (activated/deactivated)

management, or customer relationship management. As a matter of principle, such applications contain detailed operational data. Hard context-awareness, on the other hand, can complement the view to the context by making only an aimed measurement of missing data using sensors.

5. Prototype Application

We have implemented some of the described concepts in a context-aware information service called KEVIN. This application demonstrates the combination of different contextual dimensions and the integration of such technologies into an existing Enterprise Application Framework. The current implementation spans multiple devices, applications, users, roles and user interface media.

KEVIN is a service that provides personalized information to the user on the most appropriate medium that is available in the user's current situation. In many situations there are multiple devices available in the user's environment, for example a fax machine or network printer. Some of them could be used in the communication process of the application with the user. This allows receiving large messages without being limited to the constraints of a single mobile device and is especially useful with multimedia-enriched messages that contain, for example, high-resolution images. An example is shown in figure 2.

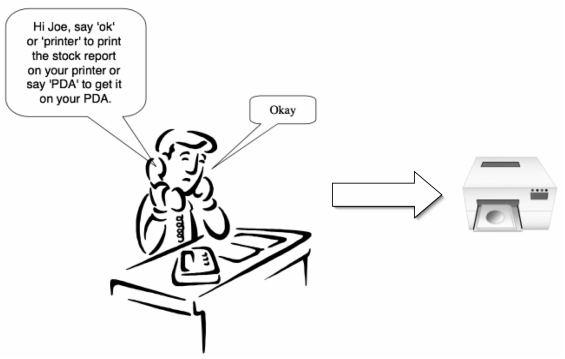

figure 2: Example usage scenario of KEVIN

Thanks to our multidimensional context-aware approach, not only the content and representation of the message to the user is adapted. It is furthermore possible to consider available mobile and stationary devices, several multimodal interfaces, other available applications and external services, knowledge about the preferences of the user as well as workflow and task information from the enterprise's backend system in the message passing application. This is another step towards the integration of information processing technology into the user's activities. The global perspective to many contextual parameters allows all involved components to behave consistently according to the current situation.

KEVIN works as depicted in figure 3. The user (1) requests information using one out of three possible input media on an arbitrary client device (2), which can be speech (by calling the application using a normal telephone), a WAP session (by using a WAP-capable mobile phone), or a web page (using a web-connected PDA or PC and a web browser). The information service KEVIN (3) identifies the user by checking her voiceprint [8] or password and retrieves her user data from the user profiles database and her role from the backend system (4). It obtains the requested data in the backend system or using other Internet based information services (6, e.g. the current exchange rates of the user's stock options). Depending on the user's role the application provides only permitted and appropriately detailed information. A context information service (5, CIS) determines the devices available in the user's environment and their addresses (8). The CIS is explained in the next paragraph. Before returning the requested information using a push service (7), KEVIN sends the user a list with available output devices. With a sufficient user history it is possible to select the mostly preferred device of the user in that particular situation automatically. The output of the requested data can be made using 7 different output media, which are a voice message, a WAP page, a web page, an email to one of the user's email accounts, an SMS message (short textual message) to the users mobile phone, a fax to a fax machine in the user's environment or to the user's office, or a printed letter to one of the user's postal addresses.

The CIS uses soft context-aware principles for the recognition of the user's current location and the available devices at that location. It works as follows. Devices like telephones and computers transmit their own network address (their telephone number or IP address) when calling other devices. This enables the soft context-aware server to estimate the current location of the user when she calls the server using a stationary device. If she uses a mobile device, it is sometimes possible to recognize this fact by evaluating the provided network address. In Germany, for example, is it possible to distinguish be-

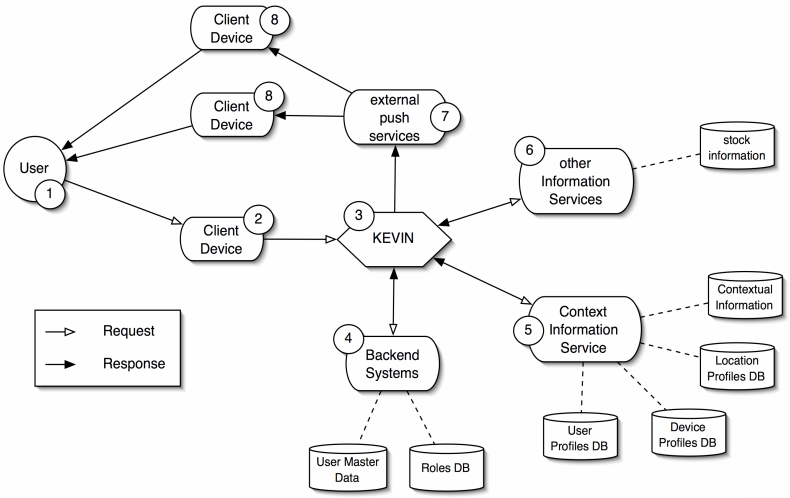

figure 3: Workflow of KEVIN

tween mobile and wired phones by the area code of the telephone number. Sometimes it is also possible to infer useful data from the IP address of the user's device, for example the Internet provider the user uses to log onto the net. A route-tracing program can evaluate, from where (from which city) the user is connected. If the user is in another city we know she makes a trip, which might trigger actions like forwarding all her incoming calls and messages to a place where she is able to access them from her remote location.

An evaluation of the user's history can reveal facts like devices she previously used in a location. Thereby Location Profiles are growing, which list all available devices in specific locations. These lists can be completed by information from device management services (like JINI [9] or UPNP [10]) that list other possible devices in those locations. In the end, we get a context-enriched phonebook that lists all devices using which we can reach the user.

If the user calls the server using a mobile phone, we can infer the fact that the user is on the move and that she has her mobile phone activated. Now we know that we can send her SMS (short text messages) or maybe also EMS (texts and pictures).

Using soft context-aware principles, a context-aware information server can determine facts about the location of the user and some of the capabilities of her devices without additional hardware sensor technology and without software enhancements on the mobile devices. This type of context-awareness can be implemented right away, with today's technology. It just needs some more configuration effort in the beginning with the benefit that this configuration can be reproduced every time the user experiences the same situation (e.g. being somewhere and getting a message) again.

We used the following technologies. The whole application is running inside a mySAP.com Enterprise Application Framework and uses its event and routing mechanisms. We use a voice gateway of Nuance [11] for vocal access to our application. It uses VoiceXML to describe the vocal user interface (VUI). The user and location profiles are stored on a Netscape Directory Server [12]. We use LDAP [13] for the directory access; because it is a very lightweight and powerful web-accessible database technology. The EA-framework also provides email functionality.

Table 2 shows the user data that we store in the contextual database. The first part stores general information about the user. The second part contains location profiles of the user, describing user specific addresses in locations where the user resides frequently. Devices considered insecure by a user can be excluded. E.g. a user may consider a display as insecure because it is broadly visible in a public place and therefore not suitable to present private or confidential data. The third part of the user profile consists of service profiles containing login data for external services. In the fourth part, the user application history, we log data about previous user actions to infer the user's expertise level for different contextual dimensions. In the beginning, we estimate the user's expertise with devices and applications from the number of times she already used an application or device.

Table 2. User profile structure

Master data	- user id - name - address - gender - birthday - mobile phone nr
User Location profile	- location name and description - email addresses at the location - tel.nr at the location - fax.nr at the location
User Service Profile	- users login name for the service - users password for the service
User Application History	- current expertise level for this application - nr-of-visits of this application at all - nr-of-visits of this application since the expertise level changed - nr. of visits of this application when the expertise level is due to change - time of the last login

6. Lessons Learned

During the work with our application we learned the following things. Thanks to multidimensional context-awareness it is possible to create applications that consist of several devices and programs and all components adapt consistently to the current situation of the user.

By using soft context-aware technology it is possible to derive statements about the context of the user without any hardware extensions such as sensors. However, soft context-awareness is sufficiently working only in highly interconnected environments, such as enterprise application frameworks. In isolated standalone applications it cannot develop its strengths.

In the prototype application, we use soft context-aware principles to determine the expert level of the user for different contextual dimensions. Thanks to the adaptation of the user interface regarding these expert levels, both beginners and experts can use the application more effectively. While beginners get an easy interface with lots of guiding explanations, experts are given an enhanced but hardly commented interface.

However, not all possible adaptations make sense. A user interface that changes too much tends to irritate the user because she does not recognize the functionality she already knows. Other adaptations might simply not be recognized by the user and can be omitted. Experienced users tend to focus on the content of the work. They do not recognize the user guidance of the application anymore. Therefore no sophisticated adaptation of the user

guidance is needed for power users. Simply omitting such elements should be a sufficient adaptation for them. A consistent user interface and application behavior is necessary for an intuitive use of applications. Separately developed adaptations can easily lead to fragmented user interfaces and behaviors, which do not correspond to adaptations of other application parts. The global view to the user's environment by multidimensional context-awareness can provide benefits here. For consistent adaptations it is also necessary to plan the adaptation process as a whole. We found that it is nice if the application looks and behaves similar on all the different platforms. A coherent application handling reduces the learning effort for using the application in different situations on different devices.

7. Conclusions and Future Work

In this paper, we presented a multidimensional context-aware approach. Contextual dimensions are separate aspects of the current context that we handle separately. As a starting point we identified the five most important contextual dimensions for our enterprise application frameworks, which are: devices, applications, locations, users, and roles. Further interesting dimensions we haven't addressed and investigated in detail so far are security and time.

Several contextual dimensions are determined by the evaluation of the existing information base, which is reused in a context-aware manner for this purpose. We call this type of information procurement soft context-awareness. It allows a lightweight context-aware functionality without many of the drawbacks that hardware-based context-awareness has.

We based our work on the requirements of enterprise applications. We determined the principles of context-aware computing in enterprise environments, which are different because enterprise applications are usually more complex and have more different users than normal desktop applications.

We implemented a prototype that demonstrates some of the principles we presented in this paper, especially multidimensional and soft context-awareness.

In the future we will develop a context-aware middleware for Enterprise Application Frameworks, which provides context-aware services transparently to all Enterprise Applications. We also want to improve the dynamic handling of contextual information. So far, our prototype captures the context only at the application startup but does not really support changes and events at runtime.

8. References

[1]: T. Winograd: "Architectures for Context", *Human-Computer-Interaction (16)*, 2001

[2]: A.K. Dey and G.D. Abowd: "The Context Toolkit: Aiding the Development of Context-Aware Applications", *Workshop on Software Engineering for Wearable and Pervasive Computing*, Limerick, Ireland, 2000

[3]: J. Eisenstein, J. Vanderdonckt and A. Puerta: "Adapting to Mobile contexts with User-Interface Modeling", *3rd IEEE Workshop on Mobile computing Systems and Applications* 2000 *(WMCSA'00)*, 83-92

[4]: A.K. Dey and G.D. Abowd: CyberDesk: The Use of Perception in Context-Aware Computing, http://www.cc.gatech.edu/fce/cyberdesk/pubs/PUI97/pui.html, 1997

[5]: S.R. Ponnekanti, B. Lee, A. Fox, P. Hanrahan and T. Winograd: "ICrafter: A Service Framework for Ubiquitous Computing Environments," Ubicomp 2001

[6]: Voice eXtensible Markup Language Specification Ver. 1.00, VoiceXML Forum, http://www.voicexml.org/specs/VoiceXML-100.pdf

[7]: R. Want, B. Schilit, N. Adams, R. Gold, K. Petersen, J. Ellis, D. Goldberg and M. Weiser: The PARCTAB Ubiquitous Computing Experiment, Xerox Palo Alto Research Center, Palo Alto, 1995

[8]: Voice Authentication Demo. Nuance Communications, http://www.nuance.com/demos/demo-shoppingnetwork.html

[9]: About Jini Network Technology, http://www.jini.org/about/technology.html

[10]: About Universal Plug and Play Technology, http://www.upnp.org/about/default.asp#technology

[11]: Nuance, http://www.nuance.com

[12]: Netscape Directory Server, http://wp.netscape.com/directory/v4.0

[13]: LDAP Documentation, University of Michigan, http://www.umich.edu/~dirsvcs/ldap/doc