Improving the Usability of Mobile Enterprise Applications by Applying Location and Situation Information

Axel Spriestersbach[1], Hartmut Vogler[2], Peter Ebert[2]
[1] SAP AG, Corporate Research, Viencenz Prießnitz Str. 1, 76137 Karlsruhe, Germany
[2] SAP Labs, LLC, 3410 Hillview, Palo Alto, CA, 94304, USA
email: {axel.spriestersbach, hartmut.vogler, peter.ebert}@sap.com

Abstract:
As of today, only simple office applications such as calendar and email applications are widely used on mobile devices - more complex applications still lack acceptance. The limited user interface features and the more difficult data input on mobile devices demand a redesign of complex applications to meet the needs of employees on the road.
Especially in the context of enterprise applications, the complexity of information presented by typical desktop applications has to be reduced to the essential amount needed in a specific mobile usage scenario. This requires that the application is aware of the user context, including the device type and the current location. By using context information, parts of an application can be automated to minimize the required amount of user interaction.
This paper describes the evolution of a typical enterprise application for *Sales Order Entry* towards a location-aware and personalized mobile application. Our implementation bases on a real word scenario for sales force automation and incorporates recorded customer feedback from an earlier prototype implementation.
The major contribution of this paper is the integration of real-time location information with enterprise backend data. We use a *Geographical Information System* (GIS) based mapping service to access customer data depending on a location provided by an *Enterprise Resource Planning* (ERP) system. We will describe the mapping between GIS and ERP data and show how this approach improves the usability of mobile applications.

1 Introduction

Real-time access to enterprise data and applications for the mobile work force, such as sales representatives or consultants, is a crucial factor for enterprises. As of today a lot of workflows, especially for the employees on the road, are still paper-based and therefore lack automated processing and real-time backend integration.
The new small, lightweight and inexpensive mobile devices and increased coverage of wireless data networks make mobile solutions applicable for a broad usage. Although there are still several technical problems (e.g. quality-limited connectivity, slower processing speed and less memory capacities), ergonomics and usability are the most crucial factors in enhancing the acceptance of mobile enterprise applications and services. We see the integration of context information as one of the most promising approaches to improve the usability of mobile applications.
The user interface features of mobile devices are limited and data input is more complicated compared to desktop computers. In particular, the user interfaces of complex enterprise applications need to be redesigned in order to reduce information input and output to the essential amount the users need in certain situations. To achieve the goal of automating and reducing user interaction, the application has to be aware of the context of the user including the current location and type of the device.
Based on a customer scenario in the field of sales force automation, an initial mobile solution and recorded customer feedback, this paper describes the evolution of a *Sales Order Entry* application towards improved usability. We extended an earlier approach described in [SVLZ01] with additional context and explain improvements of the UI of the logon screen, as well as the customer and product selections. In addition, we describe how to replace an IR Beacon based tracking solution by a completely GSM based location system. Furthermore, we describe an approach of combining basic location information with enterprise backend data. Our implementation is completely integrated with an SAP enterprise application suite.
The following part of the paper contains the customer scenario and the description of our initial implementation, a WAP based solution for sales order entries. In section 3 we explain the evolution of the Sales Order Application on a conceptual level. Section 4 describes the changes in the application flow and how we make use of a GSM based location tracking service. The paper concludes with a brief summary and an outlook of our future work.

2 Application scenario and past work

The scenario described in the paper is an example for current industrial sales force automation. Starting point of our work was a request by one of our customers to replace a paper based sales order entry process with an automated solution.
The paper-based process consists of several steps. First, the customer and sales representative browse through a product catalogue and complete a paper form with the sales records. The sales representative collects the forms during the week. The paper forms are sent to a secretary service that turns the forms into an

electronic format (Microsoft Excel). A backend system processes electronic forms as part of a weekly batch job.
One constraint of the scenario is, that sales representatives are assigned to dedicated product lines – such as low-end or high-end products- which causes cross-selling problems since paper forms are prone to false entries.
Overall, the paper-based approach had several drawbacks. The manual data entry was very error-prone, the process caused delays up to one week and there was no immediate feedback to the customer.
Our customer wanted an automated process with direct customer feedback that immediately checks the availability of the ordered goods and their specific delivery time. The automated solution also accelerates the process from over one week to a few seconds and saves the expenses for the secretary service.
The proposed WAP based solution is realized as two applications - a *Sales Order Entry* and a *Availability Check* application that already exist for desktop computers in our enterprise application suite. The decision for a WAP based application was made for several reasons. First: An online connection to the backend system is required to verify product availability. Second: Our customer targeted the European market and GSM is currently the only technology capable of providing wide-area mobile network access in Europe. Third: The WAP phones allow pure browser-based implementations. In addition, WAP technology is available for industrial deployment and the fact that sales representatives are already equipped with a mobile phone reduces hardware investments.

2.1 Initial mobile application

The initial implementation was realized twofold as a standalone Availability Check application and as a Sales Order Entry application that calls the Availability Check application. The process for recording the sales order was only slightly modified compared to the paper form.
Sales representatives present product catalogs and enter customer orders into an electronic form on their mobile phone. The data is immediately sent to the backend system that performs an availability check for the ordered goods and returns the results. In case of any outstanding orders, the sales representatives uses the Availability Check application and the given order number to check the order status. Figure 1 shows the overall application flow.

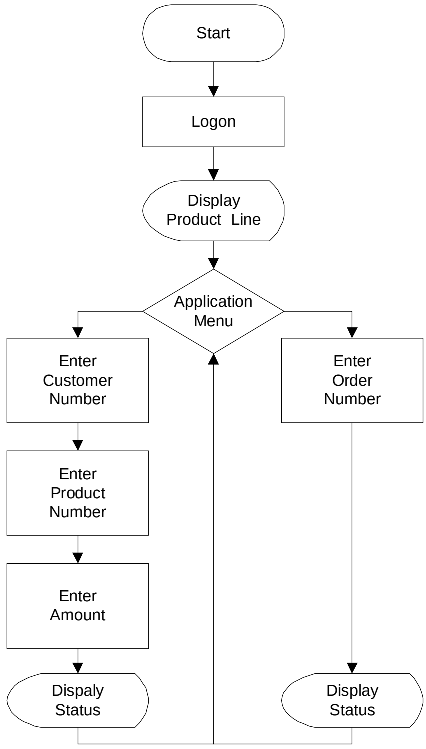

Figure 1 Application flow

The sales representative starts the application by calling a URL or selecting a bookmark. The application responds with the *Logon* screen that asks for a user name and a password. The logon information is sent to the backend system, verified, and if the credentials are valid, access to the application is granted. The backend system also assigns a role to the user. In SAP systems, roles are pre-defined settings for user groups (i.e. all sales representatives or all external employees). In the current scenario, the product restrictions are part of that role. A dialog informs the user about the valid product lines (*Display Product Line*) in order to prevent the user from invalid entries. This dialog is automatically replaced by the *Application Menu* for choosing between the Sales Order Entry and the Availability Check application.

The Availability Check application takes an existing *Order Number*, which is submitted to the backend system. The backend checks the order status and returns the result (*Display Status*) to the user (shown in Figure 2).

Figure 2 Availability check

The other option is the Sales Order Entry application. The user interface of the Sales Order Application (shown in Figure 3) consists of a minimal sales record. A minimal sales record consists of a customer number, a product number, and the quantity of products to be ordered. For an SAP system, a complete sales record contains:

- A 6-digit Customer Number
- A 6-digit Product Number
- The *Quantity* of products to be ordered.

Customer numbers are remembered by heart or taken from printed or electronic customer lists. The customer number is entered once, while the product number and quantity are repeated for every record. Product browsing is done using a product catalog from which the product numbers are taken as well. Figure 3 shows the User Interface for a single sales record. A detailed discussion of the implementation is part of the section 3.

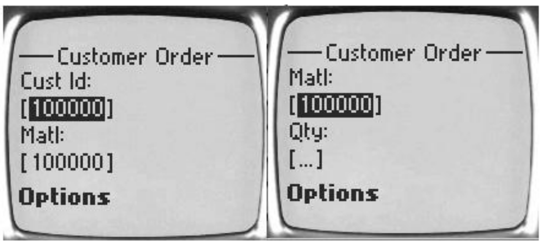

Figure 3 Sales Order Entry on WAP phone

2.2 Lessons learned

The customer rejected the initial implementation mainly for usability reasons. Data entries were too inconvenient for the sales representative and took too long. Our analysis on reasons causing these problems is summarized in the current section.

As described above, the user interface of the Sales Order Application consists of a 6-digit customer number, a 6-digit product number and the quantity that usually takes one or two digits. The average amount of digits is 13.5 to complete a single sales record. Our previous research [FrSc00] indicates that the average time for entering a single number on a WAP phone is 1,48 seconds. Therefore, the overall time for entering a single sales order is about 21 seconds. It is obvious that reducing the amount of data entries would help improving the time needed and therefore the usability of the application.

The initial implementation already used a reduced set of parameters compared with the paper-based approach. The application used only the minimal subset of data entry field necessary for a sales record and replaced the missing parameters with default values. Additional data sources allowed further reducing of the user input thus improving the overall usability.

3 Concept for UI improvements using context information

Based on our experiences and initial implementation, this section discusses improvements to the existing application screens. We analyze the initial application and discuss improvements of the UI using context information for the logon screen, the customer selection, and the product selection.

3.1 System logon

Figure 5 Logon screen

An application server usually requests a logon screen before the user is allowed to access the application - this poses a huge burden on mobile users. Figure 5 shows a typical *Logon Screen* that consists of a user name (for user identification) and a password (for authentication). The burden lies in the amount of data needed to fill in the logon screen. A typical user might for instance use a 6-digit user name and 8-digit password. To complete the logon, this user needs to enter 14 digits. According to [FrSc00], the time for completing such a logon is 23,5 (14*1,69sec = 23,5 sec). Compared to about 21 sec [SVLZ01] for completing a sales record, it is unlikely that a user will accept more than 50% overhead for a logon, only.

There are more user-friendly ways for user identification. Compared to a desktop PC that is used by many different users, a mobile device is a personal device, which is typically only used by a single user. Therefore, the identification of a device, combined with the mappings of devices and users, enables the application to identify a user by identifying his personalized device.

One way to identify a device in a GSM network, is using the callers telephone number (so called MSISDN number). The MSISDN number is part of the handshake for establishing a telephone call, which is also transmitted by the GSM network when establishing a WAP connection with a modem pool. Modem pools use the MSISDN number to verify incoming calls against a database of valid caller numbers. For valid caller numbers, an IP number is assigned and network access is granted. Unknown numbers are rejected. Modem pools maintain mappings of MSISDN numbers to IP addresses in a database. To identify a user on the application level, the web-application queries the modem database for a phone number based on the request-IP. To finally identify a user, the application queries a profile database to get the user ID for the given MSISDN number. The application inserts the user ID into the user name field of the logon dialog and returns it to the user.

An alternative approach is, to encode the user ID as a parameter in the requesting URL (e.g. http://www.sap.com?userID=d000000). The application retrieves the user ID from the request and modifies the logon. The advantage of this approach is that there is only minimal modification to the infrastructure (i.e. web servers, modem pools, mobile phones) and to the application. The disadvantage is that the user needs to complete the URL with his user ID. An option is a set of pre-configured user ID variables on the mobile device that is accessible with scripting functionality (i.e. WMLScript). However, this requires modifications to the phone or SIM card that can only be implemented by the phone manufactures (modification to the phone) or the network provider (modification to the SIM).

The user identification is verified by a password that is supposed to be complicated and therefore even harder to enter via the phone keyboard. To get rid of the password, the GSM infrastructure - as the "*largest deployed security infrastructure*" [KeGa02] - could provide authentication. Unfortunately, authentication services based on GSM technologies are not available to the public and interfaces for application-based authentication do not exist. However, if those interfaces existed and applications were allowed to directly use the infrastructure, the amount of user interaction would be reduced by a few clicks. As of today only prototypes for GSM-based authentication exist. However, there are several industrial and research projects such as the Witness Project (www.wireless-trust.org), working on this topic.

In addition, a company's security policy should be considered and adapted to the mobile environment. Some logon processes that are inconvenient for desktop browsers are appropriate for mobile environments. For example, security tokens (i.e. the RSA SecureID card) that generate one-time pads (usually short numbers such as a 6 digit number) that replace the password are inconvenient for desktops, but since numbers are easy to enter on a phone, and some devices (as the Ericsson R380) have build in security tokens, these one-time pads are very useful for phone-based applications.

Using the technologies described, the effort for a logon can be reduced from about 23,5 sec to a single click. Even though the technical effort for such an architecture is significant, especially the authentication via GSM, we expect the barriers to decrease due to open standards and standard service interfaces.

3.2 Customer selection

Another big part of required user interaction lies in completing a sales record. As described earlier, the absolute minimum data set for a *Sales Order Record* consists of a customer number; a product number and the quantity of products.

The customer number identifies the customer and enables the system to assign information (i.e. the shipping address or billing information) to a sales record. For our phone-based solution, we chose the customer number instead of a customer name, in order to simplify entering this data using a numeric keyboard. However, a 6-digit number is still inconvenient to enter. Instead of entering the customer number, a sales representative should be allowed to select his customer from a list. Unfortunately, a company's customer base is often too large to display even for desktop browsers. Here, current context information of a user allows narrowing down the number of records significantly.

A first step in narrowing down options is based on a sales person's profile. Usually, only a subset of all customers of a company is assigned to a single sales person. In our scenario, the customer list is limited by the product line. A sales representative's customer base – stored in the profile or role – might be sufficient to narrow down the options (e.g. down to 10 options - dependent on the device). However, in most cases the number of options is still too large for mobile phone displays.

We propose a solution using the device's location. Location tracking technologies such as GSM or GPS allow locating a mobile device with a precision of up to 100 meters. The relation of the device's position to customer addresses allows searching for customers within a specific area. The combination of customers in an area and a sales representative's customer base is sufficient for simplifying the selection of a customer on the mobile device. We will describe methods for device tracking in GSM networks and searching within a company's customer base later in the "*Extraction of location data from GSM networks*" section.

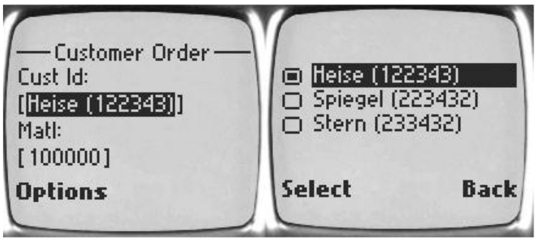

Figure 6 Customer Selections

Figure 6 shows a resulting dialog where the *Customer Number* input is replaced by a selection list. The new method requires a single selection (two interactions with the phone) and saves, compared to the original 6-digit customer number, 4 interaction steps and about 6 seconds (= 4 *1,48seconds [FrSc00]).

3.3 Product selection

From a usability point of view, the task of entering a product number is similar to the task of entering the customer number. Selecting products from a list decreases a user's cognitive load. Unfortunately, there

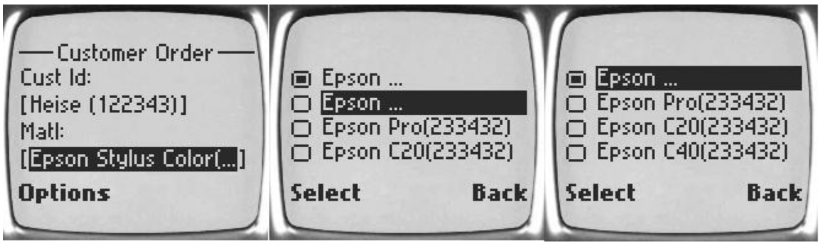

Figure 7 Adapted Product List

are usually too many products available for a selection, as is true for the customer selection. Our approach for narrowing down the options for the product numbers is similar to the approach for customer numbers: we tried to incorporate contextual information to narrow down the set of product numbers.

Adaptation to the estimated customer's demand narrows down the selection list. Customer demand may be available directly from the customer or estimated based on a customer's sales history. Explicit demand could be a list of products a customer needs to order. Examples for explicit demand are products that are out of stock or part of the client's orders. This kind of a product list may be available from the customer's backend system and accessible to the Sales Order Application. The Sales Order Application accesses this product list and completes the product list on the mobile phone accordingly. Other sources for customer demand are customer profiles that are stored in sales histories of CRM systems. Demand-estimation engines are able to calculate and adapt the product lists based on this data.

Other valuable information comes from the interaction with the Sales Order Application itself. While sales representatives and customers interact with the application, the product list is further adapted. Products already been sold during a session are deleted and replaced with products from the demand-estimation list. Figure 7 shows an example of an adapted product selection. The left screenshot shows the application with a selection list for both, the customers and products. The screenshot in the middle shows the selection of a product (*Epson…*) while recording a sales order. The right screenshot shows the same selection for a later sales order were the first occurrence of the product (*Epson…*) in the list is removed and completed with a new product (*Epson C40 (233432)*).

Overall, the improved product selection requires only a single selection step and therefore saves four interaction steps, compared to a 6-digit product number dialog in the original application.

3.4 Analysis of the approach

The improved scenario minimizes the needed interactions of more than 28 steps (logon: 14 + sales record: 6+6+2) down to a minimum of seven steps (logon: 1 + sales record: 2+2+2) for the optimized solution. It is worth mentioning that context related work is generally domain and application dependent. Therefore, the concepts explained here might not apply to all customers and all situations. However, the methods and principles for improving the usability of mobile applications are easily re-usable in other scenarios. For example, consultants that are using an activity-recording application must enter a customer and project number for the project they are working on. Generally, the list of customer or project numbers is too large for immediate display on a mobile phone. Based on the location and/or profiles, it is possible to reduce those lists as well.

4 Improved implementation

Considering context information is a sound way to improve the interaction design for mobile applications. However, including context also means modifications to the application itself, its process flow and the supporting architecture. Context must be sensed, algorithms for combining basic data with context need to be developed and architectures need to be modified or extended. This section discusses the technical details of our prototype, the location tracking in GSM networks, the coupling of location with backend data, as well as the use of user profiles and customer data.

4.1 Detailed application flow

The interaction diagram in Figure 9 outlines the modified application flow for the *Sales Order Application*. The figure contains five major processing units. The *User/Phone* represents the user and the device the user is interacting with (a phone in our case). A WAP browser installed on the phone provides the UI and is responsible for the communication between the phone and an application server that executes the *Application*. We will abstract from the application server and use the term *Application* for rest of the paper - the *Application* contains a classical SAP R/3 system and the SAP Web Application Server [SAP02b], a fully J2EE-compliant infrastructure with a Java package [SAP02c] to connect to the R/3 system. The *Profile DB* stores user profiles and customer data. The information stored in the *Profile DB* may have been collected by other systems (e.g. CRM) that are not part of Figure 9. The *GSM Location Service* is a service that returns the location (latitude, longitude & altitude) of a phone for a given MSISDN number. The figure abstracts from the real process of extracting the location from the GSM network as explained in section

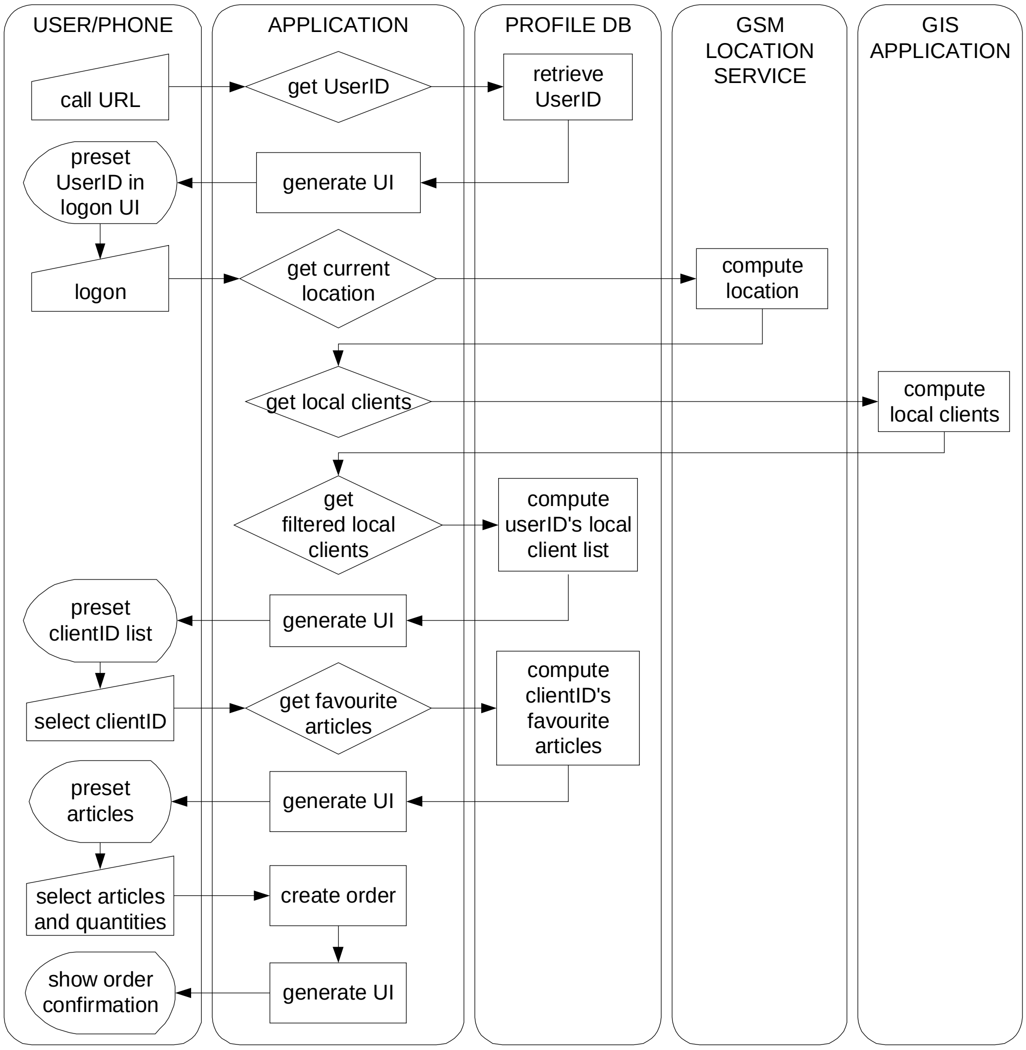

Figure 9 Interaction Diagram

4.2. Finally, the *GIS Application* is a service for mapping a location to a list of customers.

Upon entering the *Sales Order Application* URL in the WAP browser (*User/Phone*), the browser sends a WAP request to the *Application*. The WAP browser submits the user identification (UserID) as part of the request. The User ID may either be encoded indirectly as a MSISDN number, or directly as a URL parameter. In case the UserID is not submitted as a parameter, the *Application* will try to retrieve it using the submitted MSISDN number. The mappings of MSISDN numbers to UserIDs are stored in the *Profile DB*. The *Profile DB* retrieves the UserID using the submitted MSISDN and returns the UserID to the *Application*. The *Application* generates a logon dialogue containing the preset UserID and returns the dialogue to the *User/Phone*. After the *User* has completed the logon dialogue with a proper password (or using other technologies as explained in "*System log*" section), the *Application* starts collecting contextual information. First, the *Application* queries the *GSM Location Service* for the current position of the *User/Phone* using the *User/Phone*'s MSISDN number. The *GSM Location Service* returns the requested location as a point expressed in a geospatial format that can be processed by the *GIS Application*. The *Application* forwards the received point location to the *GIS Application* in order to receive a list of customers that are located within a defined radius of the *User/Phone*, sorted ascending by distance from the *User/Phone*. We used Orion described in section "*Orion: GIS application on top of SAP data*" as *GIS Application*. The *Application* passes the resulting customer list to the *Profile DB* in order to further reduce the number of records in this list. The *Profile DB* filters the received list and returns only records of customers that are assigned to the sales representative. In the next step, the *Application* generates a client selection dialogue based on the filtered list and returns this dialogue to the user. Upon selection of a client ID by the user, the *Application* queries the *Profile DB* for a list of favorite articles that the selected client has ordered before.

After generating an article selection dialogue based on the client's favorites and receiving a selection including quantities from the user, the *Application* creates an order. If the order was successfully created, the *Application* generates and returns an order confirmation to the *Phone/User*.

4.2 Extraction of location data from GSM networks

The following section describes solutions for location tracking that are based on the GSM network infrastructure. We want to replace the beacon-based approach we used before to archive zero installation on the mobile device.

The GSM network infrastructure is an earth bound infrastructure that consists of fixed *Base Transceiver Stations* (BTS) that communicate with mobile GSM terminals - the mobile station/phone (MS) - via a radio link. The BTSes are the basis for location-based services in GSM. A phone's location is calculated relatively to those BTSes for which the operator knows the location.

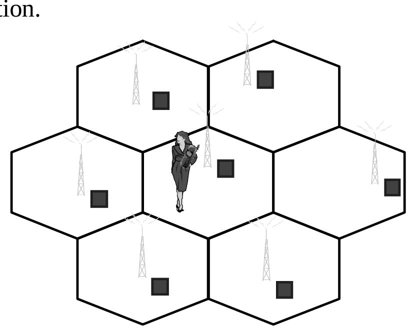

Figure 10 GSM Cell Structure

The BTSes divide the network into cells identified by Cell-IDs as shown in Figure 10. While there may be several BTSes in range, a phone is registered only to a single BTS with the best signal.
The easiest way for locating a phone is using Cell-IDs. However, the accuracy of this method depends on the cell size and is usually very poor (between 2km to 20km in diameter).

To track a location more accurately, triangulation is necessary. The approaches for triangulation are classified into network based/handset assisted, or handset based/network assisted solutions. The network-based technologies usually measure the signal delay to triangulate the device's position. The network measures the arrival times of GSM packets by at least three different BTSes. The BTSes receive the packets with different signal delays. The delays allow triangulating the relative positioning of the device with respect to the BTS. The relative position combined with the absolute BTS location allows computing the global position of a phone. Examples for such technologies are for instance the *Time of Arrival* (TOA) or *Enhanced Observed Time Difference* (E-OTD) methods. The network-based technologies heavily rely on network-based technologies and network infrastructure, so network based location-tracking services need to be operated and offered by the network providers.

We focused on a handset based/network assisted approaches for the location tracking, due to the lack of available network based tracking services. Handset-based solutions rely on the power level measures to triangulate the device's location. GSM devices measure the signal strength from all nearby BTSes. Whenever the signal quality of the BTS a device is registered to drops below a certain level, the device registers to the BTS with the best signal quality. A list of surrounding BTSes and their signal strength are stored in the phone's SIM card.

The *Subscriber Identity Module* (SIM) is a smart card primarily used for user authentication. SIM cards reside insight the mobile terminals and communicate via *Short Messaging Service* (SMS) with the GSM network. SIM based applications can access the cell ID lists including the field strength and return the list to the server upon request. The communication is tunneled through SMS. The combination of the cell ID, their field strength, the absolute BTS's positions, and the topology of these surrounding area allows the tracking of devices. The precession of the tracking depends on the density of the base stations.

The major advantage of this approach is that it allows locating mobile terminals with a decent precision at low cost. Mobile terminals and the GSM network need not be modified and the communication via SMS is very cost efficient. Merely mapping services of cell-id and field strength need to be added for the location services. However, currently only a few providers offer such mapping services. Another drawback of the handset-based solution lies in the need of an application to be installed on the SIM cards, which is usually not allowed by the network providers for security reasons.

The challenge of tracking a device or user locations is not a purely technical one anymore. The challenge for an application provider is using the existing technologies in an efficient way. As of today, all solutions, including the GSM tracking described before, are project solutions that depend on unique devices or network-based technologies. There is a lack of standard interfaces, the network providers are using incompatible technologies, and it is unclear whether location services will be publicly available in the future. However, there is a growing momentum in the market and location based services are recognized as the "killer app" for mobile computing. As of today, some location services are available to the public and the standardization work started as well (i.e. http://www.locationforum.org).

4.3 Orion: GIS application on top of SAP data

Orion is the code name of a Business GIS research prototype based on ESRI (Environmental Systems Research Institute) software [ESRI02]. Orion has been developed by SAP and is currently being used by the SAP sales force.

Even a basic geospatial business query like "Show me all customers in a radius of x meters of location y" as shown in Figure 9 (compute local clients) requires so called Geographical Information System (GIS) functionality on top of a database or application server. In order to enable such a GIS system, customer addresses have to be geocoded. This means, each customer address has to be mapped to a standardized latitude, longitude (, altitude) data set, which is typically stored as part of a customer record in a database. A variety of geocoding software tools and services are currently available. We used ESRI ArcView 3.2 to geocode the customer data.

In the present scenario, the *Application* sends the current user location to the *GIS Application* (Orion) that has access to a geocoded customer database. Based on the user location, the *GIS Application* (Orion) is able to select only those customers that are within a given radius of the user location and returns the selected records to the *Application.* The size of such a selection radius should be at least as big as the absolute minimal precision of the user location data plus the absolute minimal precision of the customer location data in order to make sure that all local customers are selected. For example, if the minimal precision of a given user location ULmin is ±3 kilometers and the minimal precision of a geocoded customer address CLmin is ±1 kilometers, the selection radius should be at least ABS(Ulmin)+ABS(Clmin) = 4km kilometers. In order to minimize the number of selected customers and thus of selectable entries for the user of the mobile device, it is advisable to dynamically adapt the size of the selection radius based on current precision data. Fortunately, one can expect a correlation between the local density of customer locations and the precision of GSM-based location services. For example, in a city with many small GSM cells, the precision of a GSM-based location service as well as the density of customer locations is typically higher than in a rural area. This allows for smaller selection radii in cities and bigger selection radii in rural areas both resulting in a high accuracy of selected customer locations while minimizing the number of selectable records for the user. Another way to optimize selected customer lists is to have the *GIS Application* sort the returned lists by increasing distance from the user location, so that the probability to find the current customer on top of the resulting list is as high as possible.

In addition to the presently disclosed usability advantages of using GIS functionality on top of business data for mobile applications, there is a variety of other benefits, as for example described in [Ebe01].

4.4 Combing GSM location information with Orion

Figure 11 shows our architecture that combines the GSM location service and Orion described in the past two sections. The figure skips the logon and application selection process and focuses on the location dependent part for selecting customers. When the user invokes the Sales Order Application in the *WAP browser,* the application server receives a WAP request (1&2) through the *WAP Gateway* (i.e. Nokia Active Server [Nok02]). For extracting the cell lists, an application stored in the *SIM card* is necessary. As described before, this communication with the SIM based application relies on a set of SMS messages. Therefore, the application requests the *SMS Gateway/Center* (3) to send an appropriate SMS to the mobile terminal. The SMS Center translates the request and sends an SMS to the mobile terminal (4). Upon arrival, the mobile terminal forwards the message to the SIM that started the SIM application. The SIM application accesses the Cell-ID and field strengths list and returns this list via the mobile terminal to the SMS Center that finally returns the list to the application server (5&6).

Up to now the location is only available as a set of BTS IDs and field strengths. The GIS mapping service translates the cell/field strength information into GIS coordinates. Step 7 and 8 queries the GIS mapping service and returns the coordinates to the application.

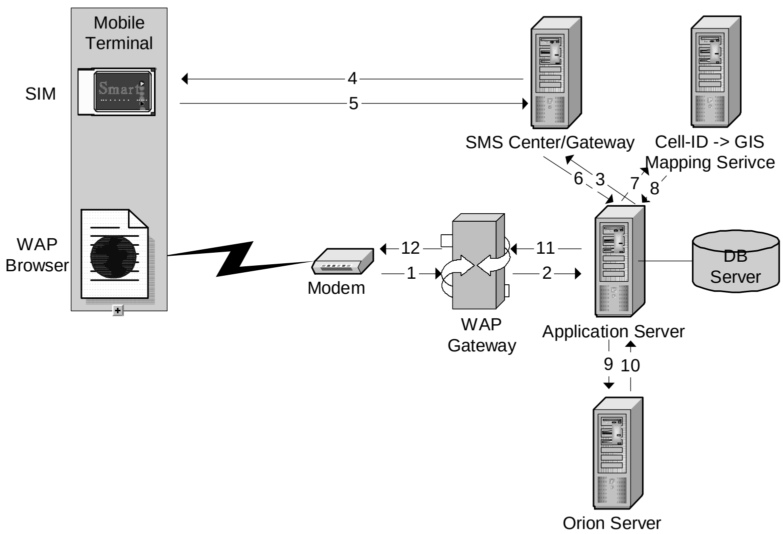

Figure 11 Mapping GSM Location & Application Data

In our sales order entry scenario the application calls the Orion server for customers within a certain range. The Orion server accesses its database and returns a list of nearby customers (10) based on which a WML is created and returned to the mobile device (11&12).

4.5 User profiles and customer data

A major part of our ongoing, and not yet completely integrated, work is the usage of user profiles and customer data in combination with location information. For our implementation we plan to make use of more complex work described in [Gos01] which addresses the specific needs to access enterprise applications and adapt the user interface with different devices as well as with different interaction schemas such as for desktops, WAP phones and pure voice interaction via phones based on VoiceXML [Voi01].

Besides the different interaction schemas, this multi-dimensional approach supports the adoption of applications based on the experience of the user with a specific application as well as with a specific device or interaction schema. For example, a user not familiar with a desktop application needs support by the system to explain what to put in the different fields but no help how to use the screen, keyboard and mouse. In contrast, at a fully VoiceXML-based application a user familiar with the application needs more support with the interaction schema (e.g. the system advises the user "you can say the number or type into the keyboard"). Another dimension of the approach is the incorporation of sensor information or location information. For example, in case a system knows the location (home, office or mobile phone) based on a MSISDN, the system can offer to send further information or a confirmation to the email account or to the fax machine next to the phone.

Usually, information about users and customers are stored in a database server of an enterprise application and are accessible in a transactional way to ensure the consistency of the data. Since web-based applications typically read static data, transactional access is not necessary. Furthermore, the data structure inside the database server is suited for the special application needs and does not address context information and adoptions. Thus, we extract the data out of the database server, map it to our data model, and store the data on a *Lightweight Directory Access Protocol* (LDAP) enabled directory server. For our implementation we make use of the SAP LDAP interface [SAP02a], which enables the exchange of data between an a SAP database server and a directory server using LDAP, as well as an application defined mapping between the different data structures. The synchronization of these data storages takes place in a batch process in defined intervals.

Having stored the user data on a LDAP directory server allows the system to retrieve the user ID based on the MSISDN (phone numbers are part of the master data of an user) in a very fast way with a simple query. Additional the customer data such as last ten orders, preferred quantity etc. are in the same way quickly and easily accessible for the application.

5 Conclusion and future work

The paper described the evolution of a paper-based process towards an improved process that uses a context-aware and personalized mobile application. Starting with a customer scenario and an initial implementation, we extended our first location-depended approach [SVLZ01] and improved the user experience. We introduced a GSM based location tracking system to achieve a zero footprint for WAP devices. In our solution, we combined basic context data – the GSM location - with information from the backend using a GIS mapping service. Finally, we explained the changes in the application flow caused by including the context information, user profiles and customer data.

A result of our analysis is that adding context may improve the usability of many applications, but also results in additional efforts for infrastructure, services and modifications to the application. We were able to identify some components that can be reused in other scenarios, while other development depends very much on the application and the scenario and might be hard to reuse.

In addition, there may be other valuable context sources for the sales force automation scenario or additional scenarios for location dependent services.

Also, a lot of effort lies in combining basic context data and backend information to make use of the context on an application level.

Our future work will address several directions. We will investigate additional context sources, for instance sensors, to improve our existing scenario even further. We will also address additional enterprise scenarios where the use of context information might improve the user experience, such as the activity tracking for consultants or health care scenarios that support doctors and nurses.

In addition, we plan to address the combination of basic context and backend data in more detail, as we did for determining the customer list. We already sketched some areas where combinations of basic context and enterprise data might be very interesting. Further investigations of these topics as well as research into a more general way of using the above-mentioned combination will follow.

Last but not least, software methods to support the development of context aware applications as well as the reuse of context dependent parts for various applications are still an open topic. Software methods will decrease the effort for including context into an application and therefore enhance the acceptance of context-awareness in application development.

Reference

[Ebe01] P. Ebert: *Holistic Business GIS*, SAP TechEd, Las Vegas, USA 2000

[ESRI02] ESRI: A Full Range of GIS Solutions, http://www.esri.com/software/index.html, 2002

[FrSc00] O. Frick, A. Schmidt: *WAP - Designing for Small User Interfaces*, http://www.sapdesignguild.org/index.asp?b=/innovation/artikel/wap-design1.html&t=/innovation/innovation_topnav.html, 2000

[Gos01] K. Goslar: *Gewinnung, Darstellung und Verwaltung von Kontextinformationen – Personalisierung* (in German, done at SAP Lab, Inc. Palo Alto), Master Thesis at the Department of Computer Science, Dresden University of Technology, September 2001

[GSM11.14] European Telecommunications Standards Institute (ETSI): *Specification of the SIM application toolkit for the Subscriber Identity Module*, European Telecommunications Standards Institute (ETSI), 1998

[HLC97] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins: *Global Positioning System: Theory and Practice*, Springer-Verlag, 1997

[KeGa02] D. Kennedy and A. Gavras, *m-Commerce – an Immediate Business Opportunity*, http://www.wmrc.com/businessbriefing/pdf/wireless2002/reference/10.pdf , 2002

[MLS00] Mobile Location Services, MLS, http://www.gsmworld.com/technology/mls_presentations.html, 2000

[Nok02] Nokia: *Nokia Active Server*, http://www.nokia.com/corporate/wap/pdf/NACS21_DataSheet_0302.pdf, 2002

[Palm01] Palm, Inc.: Palm OS, http://www.palmos.com/, 2001

[Paegas02] Radiomobil: Paegas Locator, http://www.paegas.cz/rdmnet/english/services/services_locator.htm, 2002

[SAP02a] SAP AG: *mySAP Technology -- Directory Interface for User Management via LDAP*, http://www.sap.com/partners/software/integration-opportunities/interface-certification/technology/bc-ldap/, 2002

[SAP02b] SAP AG: *SAP Web Application Server: Web Application Technology and Web Dynpro*, http://www.sap.com/solutions/technology/pdf/web.pdf, 2002

[SAP02c] SAP AG: *SAP Java Support*, http://www.sap.com/solutions/technology/keycapabilities/java.asp, 2002

[SVLZ01] A. Spriestersbach, H. Vogler, F. Lehmann and T. Ziegert: *Integrating Context Information into Enterprise Applications for the Mobile Workforce – Case Study*, ACM Workshop on Mobile Commerce at MOBICOM'01, Rome, Italy, June, 2001

[TEA01] TEA. Technology for Enabling Awareness, http://tea.starlab.net/tea_vis.html, 2001

[Voi01] VoiceXML Forum: Voice eXtensible Markup Language Specification ver. 1.00, http://www.voicexml.org/specs/VoiceXML-100.pdf, 2001