

# Using GeoFencing as a means to support flexible real time applications for delivery services

Georg Schneider<sup>1</sup>, Björn Dreher<sup>1</sup>, Ole Seidel<sup>2</sup>

<sup>1</sup> University of Applied Sciences Trier, Schneidershof, 54293 Trier, Germany  
{g.schneider, dreherb}@fh-trier.de  
<http://www.fh-trier.de/>

<sup>2</sup> Ole Seidel, alta4 Geoinformatik AG, Frauenstraße 8+9, 54290 Trier, Germany  
{ole.seidel}@alta4.com

**Abstract.** Delivery services are an industry sector that may benefit extremely by the availability of mobile devices and connectivity. Today the support and tight integration of mobile users into the company processes is finally possible using off-the-shelf components. This paper presents the concept and implementation of an application, which is targeted to the domain of delivery services. The complete process, starting from route planning through navigation to delivery confirmation can be supported. The concept of GeoFencing is used to automatically detect different states in the execution of the delivery process in order to trigger context adapted actions like navigation close to target points and delivery confirmation. The system is realized as a GPS-based windows mobile application using a conventional consumer PDA.

## 1 Introduction

The most tremendous changes in computing during the last years have been in the mobile area. Today there is almost everywhere wireless access to information (e.g. [1]) and suitable access devices (e.g. [7]). Furthermore GPS signal receivers start appearing in more and more cell phones, smart phones and PDA's. Consequently the user can locate himself easily and precisely and he can be located if he shares this information. Hence not only applications for leisure activities<sup>1</sup> can use these technologies, work processes that could not be supported up to now, especially processes affecting mobile workers, can finally be integrated in the IT infrastructure of an enterprise and therefore benefit from immediate access to the knowledge sources of the enterprise.

One domain, which can greatly take advantage of these improvements are delivery services. The usual work for delivery services has several steps. First they have to plan their route for the delivery. Once the different targets are fixed they have to find a (possibly short) route, which leads to all the delivery points. After this they can start the tour. Usually they have to confirm the delivery or pickup of items at each point to prove the successful completion of the task.

---

<sup>1</sup> E.g. maps for bikers: [http://www.adfc.de/2807\\_1](http://www.adfc.de/2807_1)

Today all these processes can be supported more or less conveniently using different software systems. The route planning can be furthered e.g. using a map server. The driver could use a navigation system to find the different targets. Furthermore the use of electronic checklists can serve for the confirmation of the delivery. Doing so different problems may arise. There may be changes in the route since additional targets have to be added either from the delivery central since new orders come in when the driver is already on his way or removed since orders are redrawn or the driver cannot fulfill the request for whatever reason. Furthermore there may be targets that are close to each other or even in different stories of the same building. Here the driver has to be informed that he can complete several tasks at the same location. Information about the close vicinity, e.g. changing the navigation to a more detailed view is crucial at this point.

However a comprehensive lightweight solution does not exist that decently deals especially with the challenges mentioned above. Such a system should offer dynamic route planning and navigation but also intelligent close target navigation and delivery confirmation.

In the following we will give a short overview over existing systems. Afterwards we will describe the concept and implementation of our application. The paper ends with a short conclusion and an outlook.

## 2 Related Work

The question how to support fleet route planning and execution using off-the-shelf components like cell phones is discussed in [4]. Here a system is presented based on J2ME [5] where mainly the delivery task is focused. The information about stops and delivery are captured using a cell phone and the information is communicated to the enterprise information system. The navigation task however, which is part of the delivery process is not part of the system.

Concerning navigation systems, there are already many solutions commercially available. The company Garmin [5] for example offers a broad range of navigation systems for different purposes. These systems however are exclusively specialized in navigation. Further tasks like electronic checklists for delivery confirmation etc. can hardly be integrated.

The project REAL [6] on the other hand deals with the mobile user in different contexts, like a user who is driving a car and continues as a pedestrian<sup>2</sup>. This behavior comes much closer to the requirements of a delivery service. Here the system has to guide the user who is mostly driving a van to the target point, where he walks for example to an office inside a building. Obviously there are different requirements for the different contexts: driving versus walking. Ideally the contexts are discovered (semi-) automatically so that a user does not need much explicit interaction with the system.

---

<sup>2</sup> In [2] a comprehensive overview over Map-based Mobile Guides in the academic field is given.

The systems discussed above illustrate particular points that are important for an application for the delivery service domain. Ideally a system would combine the different aspects.

In the next chapter we will present the GeoFencing idea, which supports this (semi-) automatic context acquisition process and we will discuss in more detail the requirements of the delivery process and the user interface issues.

## 2 Design concepts for the delivery support service

The design of the user interface is a crucial point for mobile applications. On one hand, the user has to interact with a device that can only offer a limited screen size in order to be used as a mobile device on the other hand the user is not completely concentrated on this device since he usually has to drive, walk, etc.. The GeoFencing concept is a means to support the context acquisition of the user's context in order to reduce the explicit interaction with the mobile system.

First of all we will have a closer look at the work process in our domain.

### 2.1 The delivery service process

In general the process for delivery services consist of the following steps:

#### □ Route Planning

The route has to be defined in terms of targets or stops for the delivery agent. This task is usually done on a PC in order to set up the initial route for the delivery agent. However there is also a possibility to dynamically introduce new stops during the delivery process on the device. We suggest storing this information in a file using the XML format. The belonging DTD looks like the following:

```
<!Element route (point*)>
<!Element point (name,street,number,zip,city,
longitude,latitude,delivery,pickup,radius,completed)>
<!Element name (#PCDATA)>
<!Element street (#PCDATA)>
<!Element number (#PCDATA)>
<!Element zip (#PCDATA)>
<!Element longitude (#PCDATA)>
<!Element latitude (#PCDATA)>
<!Element delivery (parcel*)>
<!Element pickup (parcel*)>
<!Element radius (#PCDATA)>
<!Element completed ('true' || 'false')>
<!Element parcel (ID)>
```

In this case radius refers to the GeoFencing idea, which will be described in Sect. 2.2. This XML file can be efficiently transferred between the mobile device and the enterprise.

After having defined a route the ordering of the stops has to be determined. This

can be accomplished manually or using tools that automatically generate efficient route plans.

□ **Navigation**

The navigation task starts once the route is defined and the delivery agent starts his trip. Usually the stops and the current position of the agent are displayed on a map using a GPS enabled client. However our scenario distinguishes between two different modes. The “driving mode” and the “delivery mode”. In the driving mode, the user drives a car to the next stop. Once he arrives in the close vicinity of the target, he parks the car and walks to the target position. The system shall detect these two modes automatically (see Sect. 2.2) and present situation adapted information. For the first part conventional car navigation systems can be used. When the user leaves the car a different situation occurs. GPS-based navigation may be impossible since build in GPS receivers often are not precise enough for accurate pedestrian navigation. If the delivery agent enters a building with his car, he cannot receive a GPS signal at all. In this case written descriptions, sketches or even photographs are more adjuvant.

□ **Delivery**

The delivery is the final step. The confirmation of the delivery or the pickup of goods can be easily supported using a checklist or a form.

## 2.2 Context detection and GeoFencing

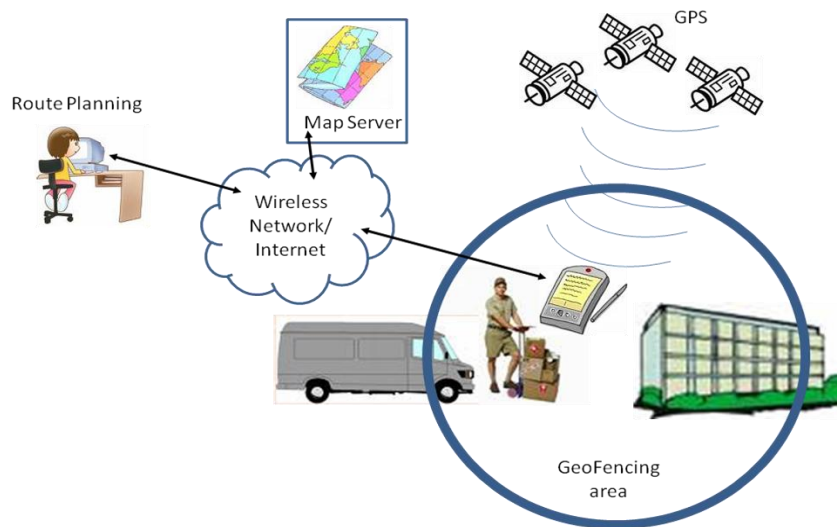
One of the goals of the research area “ambient intelligence” [8] is that the system nicely adapts to the user and his context so that it offers the right information at the right time. In our case the system has to detect two different contexts. It has to decide if a delivery agent is driving the car or if he is about to deliver or pickup parcels. A first approach would be to find out if a user is standing (velocity is zero) or driving (velocity greater than zero). This information can be gathered from the GPS receiver. Unfortunately the situation is not that easy. The driver may stand in front of a red traffic light whereas the speed would also be zero but the user is not in the “delivery mode”. Moreover experiments show that the GPS signal sometimes shows a certain velocity even if the receiver is immobile.

Furthermore the location information gathered from a mobile and moving GPS receiver also has certain fuzziness of 10 to 50 meters depending on the situation. This leads to the idea to explicitly model an aspect of uncertainty into the system. In order to trigger the delivery mode context a threshold for the velocity is introduced, which has to be under-run for a period of time when the user is close enough to a target point.

GeoFencing is the idea to build a virtual fence around the geographic position of a user, e.g. a circle which represents the user rather than a point. Once the target point is within the fence we assume that the user is at the given position. The radius of the fence may vary depending on the landscape (city versus countryside), velocity of the user (highway versus street in a residential area) or signal quality of the GPS signal.

### 3 Implementation

Figure 1 illustrates the architecture of our system. The mobile device is a HTC P 3600 with a 400 MHz Samsung processor, 64 MB RAM, 128 MB ROM and a built in GPS receiver running Windows Mobile 5.0. The mobile client has been developed using C#. As map server the ArcGIS server from ESRI and the mobile SDK<sup>3</sup> are used. Currently the map server is only queried once at the beginning of the tour for defining the order of the stops and the belonging route. Afterwards the data is copied to the mobile device together with the XML file specifying the route. This solution has been realized for the first version of the system because each query is billed and the system does not need to transfer huge amounts of data over the wireless network.



**Fig. 1.** The system architecture

The threshold for velocity, duration for detecting that the vehicle is parked can be parameterized as well as the GeoFencing radius.

Figure 2 shows a screen shot of the mobile client. The blue dot within the yellow ring shows the user, whereas the yellow ring depicts the GeoFence. The black circle shows the current target. The number “5” close to the blue dot signifies that the GPS receiver sees 5 satellites. The color yellow stands for a mediocre signal quality. The color changes as soon as the signal quality improves. An automatic adaptation of the GeoFencing area based on the signal quality is not yet realized.

<sup>3</sup> <http://edn.esri.com/index.cfm?fa=mobile.gateway>



**Fig. 2.** A screen shot of the mobile User Interface.

## 4 Conclusion and Outlook

In this paper, we have presented an application to support the work of delivery services in a holistic manner. The system will be deployed and used by the first customers in the near future. Since the mobile device used is an off-the-shelf consumer device, the costs are relatively low whereas the possibilities to extend the system is high, which is promising regarding the market launch.

In the future we will make a field test with the first customer and try to identify, which of our ideas from Sect. 2 can be realized and which further services are needed.

## References

1. 3rd Generation Partnership Project, <http://www.3gpp.org/>, download March 11. 2008
2. Baus, J., Cheverst, K., Kray, C., A Survey of Map-based Mobile Guides. In: Map-based mobile services - Theories, Methods and Implementations Meng/Zipf (Eds.), Springer Heidelberg, P. 197-213, 2005, ISBN: 978-3-540-23055-7
3. Garmin <http://www.garmin.com/garmin/cms/site/us>, download March 11. 2008
4. Ghosh, S., Jani, N.G., Das, S., Enabling real-time fleet route planning and execution in a pervasive transportation environment. In: 2nd International Symposium on Wireless Pervasive Computing, 2007. ISWPC '07, ISBN: 1-4244-0523-8
5. Java 2 Platform, Micro Edition (J2ME), <http://java.sun.com/j2me/>, download March 11. 2008
6. Krüger, A., Butz, A., Müller, C., Stahl, C., Wasinger, R., Steinberg, K.E., Dirschl, A., The Connected User Interface: Realizing a Personal Situated Navigation System. In: Proceed-

- ings of the 9th international conference on Intelligent user interfaces, Funchal, Madeira, Portugal, P. 161-168, 2004, ISBN:1-58113-815-6
7. PDAPhoneHome, <http://pdaphonehome.com/>, download March 11. 2008
  8. Weber, W., Rabaey, J.M., Aarts, E. (Eds.), Ambient Intelligence, Springer Heidelberg, 2005, ISBN: 978-3-540-23867-6