Combining GPS and GSM Cell-ID positioning for Proactive Location-based Services

Nico Deblauwe  
Department of Fundamental Electricity and Instrumentation  
Faculty of Engineering  
Vrije Universiteit Brussel, Belgium  
Email: nico.deblauwe@vub.ac.be

Peter Ruppel  
Mobile and Distributed Systems Group  
Institute for Informatics  
Ludwig-Maximilians-Universität Munich, Germany  
Email: peter.ruppel@ifi.lmu.de

Abstract—Mobile terminals with built-in GPS receivers are becoming more and more available, thus the public deployment of location-based services (LBS) becomes feasible. Upcoming LBS are no longer only reactive but getting more and more proactive, enabling the users to subscribe for certain events and get notified when e.g. a friend approaches or a point of interest comes within proximity. However, power consumption for continuous tracking is still a major issue with mobile terminals. In this paper we define this problem and propose solutions for an energy-efficient combination of GPS and GSM Cell-ID positioning for mobile terminals. We introduce several strategies for extending the lifetime of the battery and show how these strategies can be integrated into existing middleware solutions. Simulations based on a realistic proactive multi-user context confirm the approach.

Index Terms—Proactive Location-based Services, GPS, GSM Cell-ID, Position Management

I. INTRODUCTION

Location-based services (LBSs) take into account the position of one or several mobile targets in order to detect, process and communicate spatial events. LBSs can be classified into reactive or proactive services [1]. While the former are simply invoked in a request-response manner (e.g. “Where is the next ATM from here?”), the latter automatically detect spatial events a user has subscribed to beforehand, which means that the targets need to be continuously tracked. Generally speaking proactive LBSs detect when a mobile target enters or leaves a certain geographical zone. Such a zone can be e.g. a polygon-shaped area comprising a big shopping mall and the LBS informs customers automatically about special discount offers when they enter the site. Or an offender tracking LBS generates an alarm when a person released on parole departs by more than a certain distance from his home address, leading to a circle-shaped zone with the center at the residence. A third and particular challenging scenario is the correlation of the positions of multiple simultaneously moving targets, e.g. a buddy tracker, which alerts a user automatically when one of her friends approaches. Such LBSs are also based on geographical zones, however, these zones are no more static but have to be changed over time. The underlying base mechanism is called proximity and separation detection and will be described below.

All kinds of LBSs require mechanisms to determine the current position of a mobile target, which can be done either in a terminal-based, terminal-assisted or network-based fashion. No matter what procedure is chosen, there will always remain the problem that positioning consumes power on the mobile target (except for the scenarios where electromagnetic induction is used) and that mobile devices have limited battery capacity.

For proactive LBSs it is not necessary to have a constant spatio-temporal accuracy all the time in order to meet the requirements. E.g. a buddy tracker does not need to know the exact positions of other users when they are definitely far away. Only if two users approach at an area within they might be in proximity, it is necessary to determine the exact position in order to decide whether they really are close-by or not.

Power consumption on the mobile device can be reduced if several positioning methods are combined in an efficient manner. Looking at positioning methods that are available with nowadays mobile phones or mobile network providers respectively, there are many alternatives like e.g. Cell-ID, EoTD, U-TDoA, OTDoA, GPS or A-GPS (refer to [2], [3], [4] for more detailed explanations). Since most of the mobile phones have not yet a built-in GPS receiver, the most widely available method today is Cell-ID. It is referred to as a method to derive the position of a terminal based on the coordinates of the serving base transceiver station. Cell-ID positioning offers only very low accuracy but area-wide coverage and very low power consumption. GPS is favorable because of its high accuracy, but it is the most power-consuming positioning method and current mobile phone batteries last only a few hours when the GPS receiver is turned on.

The goal of this paper is to provide a mechanism for proactive LBS that efficiently combines GPS with GSM Cell-ID positioning in order to reduce the power consumption on a mobile terminal.

A central position management server is assumed for monitoring and tracking the mobile targets’ positions. Instead of collecting positions periodically, each terminal is configured by the server dynamically with a so-called position update request (PUreq). The messages are exchanged over the data channel of a mobile network like GSM or UMTS. A PUreq represents a certain geographical area, requesting the target to report back with its position to the server when that area
has been entered or left. That way the number of messages transmitted over the mobile network is significantly reduced.

In the following, the mobile terminals are assumed to possess a GPS receiver (either built-in or externally via a Bluetooth connection) and the capability to determine the ID of the GSM cell they are currently located within. For deriving the Cell-ID, different implementations are possible: one is by using the Hayes command set (also called AT-commands, a specific programming language originally developed for communication with modems), which is supported by most phones. At the terminal two different positions can be known: a high resolution GPS fix and/or the GSM Cell-ID. By detecting when it is possible to use the low resolution Cell-ID, and when there is the need to switch to GPS, it is possible to minimize the overall positioning cost for the terminal, being expressed here as the amount of consumed battery power. For this, efficient server-side strategies have been developed to correlate the positions of several targets and issue PUreqs to the terminals telling them when to switch from Cell-ID positioning to GPS and vice versa.

This paper is structured as follows: section II defines proximity and separation detection and presents new strategies for combing GPS and Cell-ID positioning. Section III evaluates the approach based on a simulation and section IV shows how the approach can be integrated into an existing LBS middleware solution. Section V concludes the paper and describes future work.

II. DEVELOPED STRATEGIES

Reducing the GPS usage can be reached through detecting the moments that Cell-ID is providing sufficient accuracy for the envisaged application. Two different approaches can be pursued: either the GPS receiver is switched off as much as possible; or switching on the GPS receiver is postponed as much as possible. Both approaches are worked out, the first being referred to as circle-based and the latter as cell-based strategies. First, proximity and separation are defined formally, and the used reference strategy is explained.

A. Proximity and separation

For detecting the spatial relation between every pair of terminals \(t_i\) and \(t_j\), the distance \(dist(t_i, t_j)\) is mapped to one out of two states: proximity, when the two terminals are located nearer to each other than a proximity distance \(d_p\), or separation, when those two terminals have moved further away from each other again than a separation distance \(d_s\). To deal with the inaccuracies of positioning methods and to avoid excessive position reporting in the neighborhood of \(d_p\) and \(d_s\) respectively, a third parameter is needed, being the borderline tolerance \(b\). Then proximity detection is defined as follows [5]:

- If \(dist(t_i, t_j) < d_p\), proximity must be detected.
- If \(d_p \leq dist(t_i, t_j) \leq d_p + b\), proximity may be detected.
- If \(dist(t_i, t_j) > d_p + b\), proximity must not be detected.

The separation conditions are formulated analogous:

- If \(dist(t_i, t_j) > d_s + b\), separation must be detected.
- If \(d_s \leq dist(t_i, t_j) \leq d_s + b\), separation may be detected.
- If \(dist(t_i, t_j) < d_s\), separation must not be detected.

Another way for managing these relations could be by using the \(n\)-body constraint as defined in [6], which prescribes that \(n\) moving objects are embraceable by a sphere of diameter \(d\) at the same time. By choosing \(d_s = d_p = d\), \(n = 2\), and \(b = 0\), the two formalisms yield the same result. Due to its broader scope, the first formalism will be used in the remainder of this work.

There is not much knowledge available on how to choose \(d_p\), \(d_s\) and \(b\) in an optimal way. In order to avoid unstable behavior, it is a good practice to choose \(d_s - d_p > 2b\) to avoid repetitive state switching at the borderline of proximity/separation [5]. Furthermore will the exact choice of the values depend on the envisaged application and environment. For a staff tracker in an indoor environment (e.g. with WLAN fingerprinting as positioning technology), it makes sense to choose values in the order of 50m. For an outdoor buddy tracker application, values in the range of 100–500m are more logical. Such an outdoor scenario is assumed in the remainder of this paper.

B. Reference strategy

There exists some research on managing proximity and separation among mobile targets. For example, any of the strategies described in [5] or [7] could have been used as our reference strategy. They have in common that every terminal gets assigned a free movement area. As long as a terminal stays inside its assigned area, per definition no changes in the proximity/separation relations are possible. Upon reception of a position update message (PUmsg) from a terminal \(t_i\), which happens when the free movement area is left, the positioning server performs three actions:

1) Poll the terminals for which proximity (or separation) is possible, because the minimal (maximal) distance between the updated position of \(t_i\) and their free movement areas is smaller than \(d_p\) (or bigger than \(d_s\)). Polling forces a terminal to transmit its current position to the server.

2) Verify the proximity and separation relations for all the involved terminals; inform the application if changes are detected.

3) Determine the new free movement areas for the set of involved terminals, and inform them by sending out the PUreqs.

Because of its efficiency and fairly simple implementation, we have opted to use the Dynamic Centered Circles (DCC) strategy, which was first proposed in [5], as our reference strategy. Figure 1.a shows a snapshot of free movement areas that were calculated with DCC. Each terminal \(t_i\) is assigned a circular area (called distance job). The center point of the circles is the last reported position, and the radii are chosen in such a way that the mutual distances between a pair of targets can never fall below (above) the proximity distance \(d_p\) (separation distance \(d_s\)) without either one of the two terminals leaving its circle and thus invoking a PUmsg. This allows
the server to effectively monitor spatial relations without needing to track the terminals continuously. The following two sections present novel strategies that manage multiple positioning technologies efficiently at the mobile terminals.

C. Circle-based strategies

The following strategies are an extension of the GPS-based approaches, like DCC (hence the name). The server will calculate the circular zones as before. By matching these distance jobs to the cell grid of the GSM network, each of them can be associated with two sets of cells: the safe cells, which are the cells that are fully contained by the circular zone, and the border cells, which are the cells that coincide with the circular zone. There is no need for having the GPS receiver switched on while being in a safe cell, which leads to a first class of energy optimization strategies: when possible, both the calculated distance job and a zone job with the list of the safe cells will be sent to terminal. Figure 1.b shows a snapshot of the same situation; terminals $t_1, t_2$ did receive a PUreq containing two jobs. Since there were no safe cells found for terminals $t_3, t_4$, their PUreq contains only a distance job.

When available, the terminal will opt for monitoring the zone job, since it is more energy efficient. There are now three possible ways to respond to the event of leaving a safe cell:

**Strategy 1**: switch on the GPS receiver and start monitoring the, less stringent, distance job. The server will only be notified if the circle is left.

**Strategy 2a**: switch on the GPS receiver and transmit a PUmsg containing the GPS-determined position.

**Strategy 2b**: transmit a PUmsg containing the Cell-ID of the border cell that was entered.

Refer to section III-B for a discussion on the intrinsic differences between these strategies.

D. Cell-based strategy

Xu and Jacobson presented in [6] an algorithm for managing spatial queries on a database more efficiently. In one of their indexing methods, the available space is subdivided into a grid of equal-sized cells. The current cell of each target is always known at the location server, and thus this index is only altered when a target performs a partition update, that is when a cell change is reported. Based on the index, it is possible to divide up all ongoing queries (n-body constraints in their terms) issued to the location system into three classes: Class A refers to queries which are certainly satisfied; class B queries on the other hand can safely be assumed to be not satisfied. Finally, queries that cannot be answered based on the cell information alone are categorized as class C.

**Strategy 3**: Instead of defining an own grid, we decided to use the cells of the GSM network, with the Cell-ID being equivalent to their index. Now it is possible to break up the list of involved terminals in two groups:

- **Cell-ID list**: terminals for which the accuracy of the cell grid is sufficient to monitor proximity (when $\text{mindist}(c_i, c_j) > d_p$, with $c_i = \text{cell}(t_i)$) or separation (when $\text{maxdist}(c_i, c_j) < d_s$) relations. These are equivalent to class A and B queries.
- **GPS-list**: terminals for which no statement can be made about proximity/separation based on information of the cells they reside in. This list is equivalent to class C queries.

The rightmost snapshot of figure 1 illustrates this: terminals $t_1, t_4$ are part of the Cell-ID-list, and terminals $t_2, t_3$ are on the GPS-list (because $\text{mindist}(c_2, c_3) = 0$). All the terminals that are part of the Cell-ID list will now receive a zone job that contains their current cell. This forces a terminal to send a new PUmsg when its current cell is left. For the terminals that are on the GPS-list, the DCC algorithm is used to calculate distance jobs. Looking for safe cells for the terminals on the GPS-list would not make sense, since the circles of the distance jobs are too small to contain safe cells. Actually, if the circle would contain a cell, in most cases a zone job would have been issued already.

III. Evaluation

For testing the applicability of the proposed strategies, a simulator was constructed to model the behavior of an
LBS community. In a proactive fashion, a member of the community is informed as soon as another member approaches (comes into proximity) or leaves again (separation detection). The goal of the simulation is to determine the percentage of time the GPS receiver can be switched off at the mobile terminal. First some details on the design are provided, after which the simulation results are discussed in detail.

A. Simulator design

The simulator moves a configurable number of targets on a field of 7.5km by 5.5km and executes the proposed strategies. The simulated time is close to 1.5h (5000s). Since our goal is comparing the proposed strategies to each other, we have opted to study them in a rather agile environment of continuously moving targets, that is, rest periods have been explicitly excluded. We adopted a constant velocity of $v = 50\text{km/h}$. It has to be stressed that this is a worst case scenario because during the day users normally remain stationary quite often. The routes of the targets were calculated with a simple mobility model: each terminal moves with a constant velocity into a constant direction, until the borders of the test area are reached. There, a new direction is randomly chosen. In [5] they showed that the choice of mobility model has no effect for the simulation results, when the goal is comparing different strategies mutually. Finally, we did select a proximity distance $d_p$ of 150m, a separation distance $d_s$ of 300m and a borderline tolerance $b$ of 50m.

B. Simulation results

The first batch of simulations compares the different strategies mutually. The reference strategy used is DCC (refer to section II-B). For the mobile phone network, cells with an average size of 0.25km$^2$ were chosen, which corresponds to a sub-urban situation. The ideal measurement for evaluating the proposed strategies would be the actual battery usage (or life time). However, it is rather difficult to model the exact energy consumption of all the different parts of a mobile terminal, so we chose to monitor the time that the GPS receiver is switched on. This provides a reasonable good indicator for the amount of battery power that can be saved. The global trend in figure 2 is that the percentage of time that the GPS is used is proportional to the number of terminals. This is insurmountable, and due to the decrease in free movement space per terminal. When comparing the different strategies mutually, one can notice that the server-side strategy outperforms the others for $N < 10$, but shifts then to be the worst performing one. Strategies 2a and 2b always outperform strategy 1 (about 10% extra saving of GPS time). Strategy 2b slightly outperforms strategy 2a for $N = 10$, afterwards this gain mitigates. The location of the turn-over point (lying here around $N = 10$) where the server-side strategy becomes less efficient, is related to the size of the cells of the underlying network.

This was verified by repeating the simulations for different cell sizes. Table I shows how the efficiency increases for decreasing cell sizes; the first column reflects a suburban situation, and the last column could be a dense urban situation. Though 20 terminals might seem few for a service that will be mass deployed, it is actually a high number of users to track simultaneously. Having thousands of users in a network does not mean that all of them need to be tracked mutually: only the ones that are related need to be checked. Additionally, it’s not because a contact list in a buddy tracker contains 100 names, that all will be online at the same time.

Besides stretching the time that the GPS is switched off as much as possible, it is important to limit the number of times that the GPS needs to be started. For a warm start, when the ephemeris and almanac data are still present and valid ($\approx$ not older than four hours), it takes $7 - 15s$ before a position fix is obtained. For a hot start, that is when the time is known as well, this reduces to about 5s till the first position fix. For the presented simulations, the latter case can be assumed relevant most of the time. Though the simulations do not take into account these startup delays explicitly, it is still possible to get an idea of this effect by looking at a histogram (refer to figure 3) with the interval durations when the GPS is switched on. Strategy 3 outperforms the other strategies; in less than 5% of the cases it needs to switch on the GPS for a short while (10s or less). Also strategy 2b performs rather well, limiting the short GPS periods to about 10%. In a situation with 15 terminals (figure 3, right graph) one can see that in almost 50% of the cases where the GPS is switched on, it will

---

Fig. 2. Percentage of time during which the GPS receiver is switched on in dependence on number of terminals

<table>
<thead>
<tr>
<th>Cell surface (cell radius)</th>
<th>0.22km$^2$ (≈ 500m)</th>
<th>0.14km$^2$ (≈ 400m)</th>
<th>0.08km$^2$ (≈ 300m)</th>
<th>0.03km$^2$ (≈ 200m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 terminals</td>
<td>40%</td>
<td>35%</td>
<td>30%</td>
<td>20%</td>
</tr>
<tr>
<td>10 terminals</td>
<td>67%</td>
<td>58%</td>
<td>40%</td>
<td>33%</td>
</tr>
<tr>
<td>15 terminals</td>
<td>75%</td>
<td>68%</td>
<td>56%</td>
<td>39%</td>
</tr>
<tr>
<td>20 terminals</td>
<td>87%</td>
<td>75%</td>
<td>60%</td>
<td>43%</td>
</tr>
</tbody>
</table>

TABLE I PERCENTAGE OF TIME THAT THE GPS IS SWITCHED ON, IN DEPENDENCE OF THE CELL SIZE FOR 5, 10, 15 AND 20 MUTUALLY TRACKED TERMINALS (USING STRATEGY 2B)
stay on for more than 150s. This means that strategies 3 and 2a will tend to use the GPS when there is a long-term need for a high resolution (e.g. when terminals are close to each other), and need it less for quick in-between high-resolution fixes. That in almost 65% of the cases the duration is less than 10s for strategy 2a is inherent to the algorithm and can be seen as it’s biggest shortcoming. Because the studied algorithms have correct performance (timely and faultless detection of proximity and separation) as a requirement, it is impossible to avoid the need for high-resolution GPS fixes every so often. Further experiments have shown that this need aggravates for an increasing number of terminals, but relaxes for smaller cell sizes.

Finally the communication needs of the strategies needs to be verified. For mobile terminals, communication with the server passes over the air interface, for which cellular bearer services like GPRS, EDGE or UMTS can be used. Because bandwidth is a scarce resource and because these services are usually charged (either per used time unit, or per data volume), it is sensible to reduce the total amount of exchanged messages. Figure 4 shows that strategy 1 does not yield a significant extra message load (less than 5%), compared to DCC, which is used as reference. The increase in number of messages is caused by server-initiated pollings: a terminal will always respond with its active technology, being Cell-ID when monitoring the zone job. If this is insufficient for the server, a second polling is performed, requesting a GPS-position from the terminal. This leads to a doubling of the polling messages compared to the GPS-only approach. Because the circle-based strategies 2a and 2b transmit a PUmsg as soon as their most stringent monitoring condition is violated, we did expect the increase, which lies around 20%. That strategy 2b performs marginally worse than strategy 2a is because some of the PUmsg with Cell-ID will cause the server to initiate a polling for a GPS-position. For the cell-based strategy, the main reason for the message load doubling lies in the less efficient zone job calculation: in contrast to the circle-based strategies, this is limited to a single cell. Further optimization is possible, by building a more hybrid algorithm.

To put these figures in perspective for $N = 5$ terminals: a terminal on the move, in a hostile environment of continuously and fast moving terminals, receives and transmits a message on average every 30s (DCC) to 20s (cell-based strategy). By relaxing the assumptions to the situation of a pedestrian user, this reduces to the exchange of two messages (up and down link) every 3 to 5 minutes. Additionally, in reality, a person will stand still at certain places for long periods (e.g. at home, at work) and short periods (e.g. in a shop or restaurant), so the message count will be even smaller. Since accurate predictions would require a lot of assumptions on the user’s behavior pattern, we have opted not to do so.

C. Conclusion

We looked at different strategies for reducing the time that a GPS receiver is needed. From the evaluation, where the most dynamic proactive scenario was simulated, we can conclude that the third strategy performs best in most cases, since it combines a very good to acceptable reduction in GPS time with the attractive property that it does almost not need quick high-resolution fixes from time to time. This comes at the cost, however, of an increased message load, to be communicated over the air interface. Furthermore we can conclude that that strategy 2b and 2a have about the same performance when it comes to reducing the total GPS time. Strategy 2b should be given preference for its better interval duration properties. Finally, it can be seen that the first, most simple, strategy allows already a reasonable GPS time reduction.

These conclusions are valid for moving and stationary targets. However, the situation of two targets staying in each others direct vicinity for a longer time (e.g. office or school situation), is not dealt with efficiently by the presented algorithms. Then the GPS receiver will stay switched on if no
additional motion detection mechanism is used at the terminal.

IV. ARCHITECTURE

A. Existing work

Küpper et. al. [1] developed a layered architecture for proactive LBS that can be used to easily integrate the strategies described above. The goal is to hide the gathering and correlation of the targets’ positions from the LBS application. Figure 5 shows the layers of the model:

- The **positioning** layer deals with the positioning methods that are present at the terminal.
- The **low-level position management** layer (LLPM) manages the different possibilities for controlling the terminals: by sending either a PUreq (containing a distance job and/or a zone job) or by polling the terminal directly.
- The **high-level position management** layer (HLPM) monitors the positions of the terminals. Its main function is managing the proximity and separation relations among the different pairs of mobile targets. Functions for detecting spatial relations between more than two terminals, e.g. k-nearest neighbors or clique detection, are possible too.
- The **application layer** enables LBS applications to subscribe for certain location-based events among mobile targets of interest and to get notified automatically. For proactive single user services like a city guide, an application can skip the HLPM layer and place directly a PUreq or perform a polling.

B. GSM Cell-ID server

We assume that it is not possible to use the GSM network for positioning purposes without the agreement and cooperation of the mobile network operators. Since it is their network (and investment), it is not likely that they will allow an external organization to make money out of it. For protecting their investment, operators have both legal and technical tools at their disposal. The first will mainly be used against companies, since it allows recuperating (a part of the) lost incomes; where the latter tool can serve to block community efforts. For example, the results of mass war-driving campaigns can be easily undone by switching the network settings (e.g. channel allocations) at a semi-regular basis. As a consequence it is safe to assume that all the needed network information is available.

For the practical implementation, we suggest to use an external server for managing this GSM Cell-ID information (refer to figure 6), which allows a clear separation of the service and data provider. This way a service provider can easily serve clients of different networks operators, without that the data needs to be local. The latter is rather important for the mobile network providers, who are not very keen on sharing information on their network in a structured way. When this data is managed by a separate GSM Cell-ID server (basically a GIS server) with access control, possibly even linked to billing schemes, for the queries. This approach makes even more sense if the approach gets extended to UMTS networks. Because of cell breathing, the shape of the cells will change, and to deal with this a link to the current network configuration is needed.

C. Implications on server architecture

In contrast to the single-technology case, the accuracy of the position information can vary for incoming PUmsgs. To deal with this we used two different classes for mapping a terminal’s position:
impact on the lifetime of the terminal’s battery can be clearly improve to 57% and 80% respectively. The major positive of the time. In a dense urban environment, these numbers 13% (20 users simultaneously tracked) and 60% (5 users)

server using multiple positioning technologies. In an suburban environment, the GPS receiver can be switched off between

criteria; and 3) an onset on how to implement a tracking

network and provide the actual data to the HLPM.

The connection to the GSM Cell-ID server could be made either at the LLPM or HLPM level. Considering the functional description of the layers from Küpper et. al. [1], the best place to provide an interface would be the LLPM. The HLPM needs to be aware off the possibility to use Cell-ID positioning. The LLPM takes care of managing which terminal uses which network and provide the actual data to the HLPM.

V. CONCLUSIONS AND FUTURE WORK

We have demonstrated in this paper that it is possible to combine GPS and GSM Cell-ID positioning in a sensible way. Our main contributions are 1) a set of concrete strategies usable for proactive LBS in a multi-user environment (≈ tracking); 2) an evaluation of these strategies using multiple criteria; and 3) an onset on how to implement a tracking server using multiple positioning technologies. In an suburban environment, the GPS receiver can be switched off between 13% (20 users simultaneously tracked) and 60% (5 users) of the time. In a dense urban environment, these numbers improve to 57% and 80% respectively. The major positive impact on the lifetime of the terminal’s battery can be clearly seen.

We would suggest three directions for the further work:

- Further development of the strategies. There is still a potential to improve the presented strategies: e.g. reducing the message count for the cell-based strategy, or taking into account the border cells for the circle-based strategies. Furthermore, the development of a prototype has started. We expect to have to adjust the strategies slightly in order to assure performance in real life, for which we have found two mobile network providers willing to provide the needed data.

- Multi-technology positioning servers. This paper describes the combination of GPS and GSM Cell-ID positioning. Most of the used concepts are actually technology independent, but the way they are implemented is not general yet. For the development of a positioning server that is truly independent of the specific underlying technologies, more research about the LLPM layer needs to be done, especially how it will interact with the underlying positioning layer and above lying HLPM layer.

- Though our work focuses on providing the needed functionality for proximity and separation detection in a multi-user environment, it stays valid in a larger context. For instance, the presented building blocks suffice to develop an electronic city-landmark guide that is energy-efficient (single user, proactive LBS). In a next step, this could be extended to a public transport navigation LBS [8] by connecting the positioning server to the database with vehicle positions, modeling each of them as moving landmark.

ACKNOWLEDGMENTS

A part of this research is supported by a grant of the IWT, the Flemish Institute for the improvement of the scientific-technological research in industry.

REFERENCES


