

# Vs Golf — Developing Location-based Multi-Player Games

Georg Treu<sup>1</sup> and Johannes Martens<sup>1</sup> and Matthias Schicker<sup>2</sup> and Marc Breisinger<sup>3</sup> and Axel Küpper<sup>1</sup>  
Mobile & Distributed Systems Group, Department of Computer Science  
Ludwig-Maximilians-Universität München, Germany  
[georg.treu|johannes.martens|axel.kuepper]@ifi.lmu.de<sup>1</sup>, mail@marcbreisinger.de<sup>2</sup>, K\_o\_M@gmx.de<sup>3</sup>

## Abstract

*Location-based multi-player games are currently being pushed by the industry, one main factor being that highly-accurate positioning technologies like GPS as well as broadband mobile networks like UMTS are finally getting widespread. Nevertheless, generic frameworks for efficiently managing user locations are still missing, which strongly complicates game development. This paper presents TraX, a device-centric framework for Location-based Services, which especially supports proactive and multi-player services and thus perfectly matches the challenges mobile game development faces today. The suitability of TraX is evaluated by means of VsGolf, an actual location-based multi-player game, which has been developed and tested by students in only four days.*

## 1 Introduction

Mobile games have been around at least since the introduction of the Gameboy [2] in 1989. Today most cellular phones are equipped with a diversified assortment of such games, typically representing adapted versions of former non-mobile video games popular in the 1980s.

However, much like the process PC games underwent in the 1990s, currently a new trend for mobile games can be observed, which leads from standalone and single-user applications towards Internet-hosted gaming services addressing a world-wide community simultaneously wandering through one virtual scenery. In fact, mobile games offer a great second possibility going beyond that concept: they can build a bridge between the virtual world that may be shared within a community and the real, physical surroundings of each individual player.

One way for achieving this is to incorporate the geographic positions of the players into the game, which classifies it as a *Location-based Service (LBS)*. The steady growth of LBSs in the recent time can be attributed mainly to the fact that highly-accurate positioning technologies like the

*Global Positioning System (GPS)* as well as broadband mobile networks like UMTS are finally getting available for the mass market. Also, GPS can be integrated with mobile phones, which suddenly allows their holders to share their position with others over the Internet.

The first generation of LBSs was reactive, that is, a user explicitly requests the service, for instance, for finding out about restaurants or ATMs located in her surroundings, see also [10]. In contrast to that, mobile gaming as sketched above poses more challenging demands on managing users' positions. For example, spatial events, like a player entering or leaving a certain geographic area, have to be detected expeditiously and in a proactive fashion. Also, the positions of multiple target persons have to be interrelated together.

Although the market for location-aware gaming is rapidly growing, the new requirements of supporting both, location-awareness and mobile networking at the same time, make their development a lot more complicated than that of classical online games. As a possible solution, this paper presents TraX, an LBS platform which undertakes the task of obtaining, transferring and processing mobile targets' position data in order to facilitate mobile game development. For evaluating the suitability of TraX, a location-based multi-player game called VsGolf has been developed by students in four days.

The remainder of this paper is structured as follows: the next section presents a detailed classification of mobile games as well as resultant challenges. Section 3 then presents the TraX framework for LBS position management, which especially supports proactive and multi-user services. Section 4 introduces the developed mobile game VsGolf, the associated course of development, the software architecture, as well as experiences with the prototype. Finally, Section 5 concludes the paper and sketches future work.

## 2 Mobile Gaming

Mobile gaming represents an emerging market in the domain of video games. Especially location awareness en-

ables game developers to realize new kinds of gaming concepts and interaction types.

One of the first games using location as input was *Pirates!* [7]. In this combat and item collecting game, palm-tops were equipped with Wireless LAN, while proximity sensors in the environment constituted the location service. *Botfighters* [1], a search and destroy combat game based on cell-id positioning, is regarded as the first location-based cellular phone game. The communication between the players was solely based on SMS, which carried the location of a player or a certain action, for instance, shooting another player's robot. An enhanced version, *Botfighters2*, uses a J2ME client and GPRS in order to incorporate collaborative gameplay, like trading weapons, updating characters and bonus material. Another example of a game is *Skeeters* [6], first published 2005 by Nokia, which uses picture processing from the inbuilt camera to determine the users' movement of the phone. Games that exploit GPS on mobile phones are e.g. 'Capture the flag' [8], *GPS::Tron* [3], an adaption of the classic arcade game *Tron*, *Mogi* [4], a treasure hunt game, which is based on a J2ME application deployed by the Japanese network operator KDDI and which uses GPS, and *REXplorer* [5], a game that besides the entertaining component is a guide through the city of Regensburg. A more detailed overview of location-based games on cellular phones is given in [17].

As one can see, mobile games are experiencing the same evolution as classical video games: from stand-alone, solitary gameplay to so called Massively Multiplayer Online Role Playing Game (MMORPG) with many kinds of social interactions. This development is supported by the integrated network capabilities of mobile phones and the personal relationship between the user and her mobile phone.

The development of location-based multi-player games comes with a couple of challenges. Some of them also exist in the classical video game domain, some of them are new due to the location awareness, different mobile phone platforms and usage patterns.

**Proactivity.** There are two kinds of game plays a mobile game can be built upon. The first one, so-called *active game play*, features a continuous interaction with the user over the whole period of a game session. In contrast, the second one, which is often referred to as *ambient game play*, describes a discontinuous participation of the player. That is, the game continuously proceeds in the background and provides the possibility to the player to take active part in the game only on occasion. A well known existing example is the nurturing of a Tamagotchi, where a virtual pet lives in an egg shaped device permanently, while the user turns attention to it from time to time. In many cases, ambient games interact with their user in an event-based, also called *proactive*, manner, that is, the game informs the player about certain events in the game play based on a change of the current

location or an action of another user (compare Section 3). On a technical level, this feature requires an asynchronous communication channel between all users, which enables the immediate distribution of event messages in the system.

**Multiple Users.** In the mobile environment the realization of a multi-player game faces additional challenges: Not only the position updates and actions of all players have to be distributed among them, but also the unreliability of mobile networks has to be taken into account. Therefore, the game must sustain the sudden loss and a later reconnection of a mobile player sometimes in combination with the restoring of her previous session. This feature requires a synchronization mechanism, which keeps the devices up-to-date and enables the users to join or disjoin running sessions of a game.

**High level of interaction.** Due to the multi-player support and the characteristics of ambient games, communication often occur in bursts. Therefore, the used platform must be able to handle events in a very efficient way in order to allow for direct feedback to a player's action. Smooth and fluent interactions with the game are a key factor for a great user experience.

**Location Awareness.** The estimation of the player's position, its transformation into an appropriate format and the correlation with other players' position data are an essential part of a location-based game. These steps have to be carried out very quickly and efficiently in order to enable a high degree of interaction via ensuring short response time for the player and a low amount of messages transferred over the air interface. Also, heterogeneous positioning technologies, like GPS, cell-id, or WLAN Fingerprinting, have to be integrated in the game application to serve different platforms and end user devices.

These challenges in developing a mobile location-based game advise the usage of a framework or system, which separates technical issues like synchronization, data transmission and positioning, from the game play related tasks, like the user interface. The following section presents a framework coping with these challenges.

### 3 TraX Middleware

TraX [13] is an LBSs middleware platform, which especially supports *proactive* and *multi-user* LBSs. In contrast to *reactive* services, which are initialized on request by the LBS user, proactive LBSs are automatically triggered as soon as a certain spatial condition becomes true, for instance, when the target person enters or leaves a pre-defined geographic zone. The term "multi-user" is attributed to the fact that TraX can observe pre-defined spatial relations between multiple targets in an efficient way. One example is proximity detection, which triggers an alert as soon as a pair of targets approaches each other closer than a pre-

defined proximity distance. Typical application areas comprise friend finder services, child and pet trackers, and of course mobile gaming.

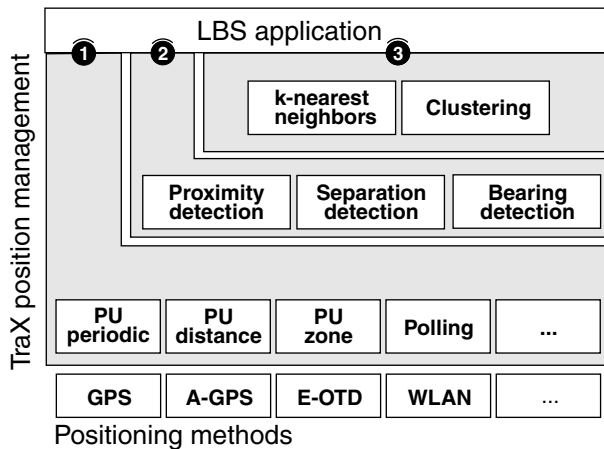


Figure 1. TraX framework

TraX follows a device-centric approach, that is, the targets' positions are derived by carried devices coupled with a terminal-based positioning method like GPS. The device incorporates the *TraX daemon*, which constantly checks whether derived position fixes are to be further processed. For obtaining the positions, it relies on existing middleware platforms developed for LBSs which focus on the control of positioning and hiding this process from the application, for example, the Location API of the Java 2 Micro Edition (J2ME) [9] or the PlaceLab platform [14].

The daemon is dynamically configured by so-called *position update methods*, which can be done either by an application located at the mobile device or by the TraX Server, which is connected with the daemon by means of GPRS/UMTS bearer services. The update methods define conditions upon whose arrival the respective application is called back with the current position of the device, denoted as *position update*. Available update methods comprise the following, see also [15] and [11]:

- **Periodic position update.** A position update is triggered if a pre-defined time interval, the so-called *update interval*, has elapsed since the last position update.
- **Distance-based position update.** This generates a position update if the line-of-sight distance between the last reported position and the current position exceeds a pre-defined threshold, the *update distance*.
- **Zone-based position update.** A position update is initialized when the target enters or leaves a pre-defined

geographic *update zone*, which can be fixed by a circle with a well-defined center and radius or by a polygon.

As depicted in Figure 1, these methods represent the bottommost layer of TraX accessible by an application (1). The layer is also denoted as *low-level position management (LLPM)*, because it is dedicated to managing the position of single targets only. As it turns out, the different update methods constitute a highly bandwidth-efficient and energy-saving mechanism for realizing proactive single-target LBSs, because position fixes are only reported over the air-interface when desired by the service.

What makes TraX unique is its support for multi-user LBSs, reflected in the so-called *high-level position management (HLPM)*. The HLPM is divided up into one layer that provides functions for monitoring the spatial relations between pairs of targets (2) and one that interrelates the positions of more than two targets (3). An example for the former is proximity detection as mentioned before. Other functions are also conceivable, for example, the monitoring of the bearing between a pair of mobile targets according to a pre-defined angular threshold. As it turns out, determining the angular relation between a pair of mobile targets is helpful for realizing more complex HLPM functions like cluster detection [18] as well as continuous k-nearest-neighbor queries [16].

For further details on the interplay between HLPM and LLPM, which are out of the scope of this paper, please refer to [12]. For more information on the TraX software, which is constantly being advanced, please refer to the companion website <http://www.mobile.ifi.lmu.de/TraX>.

To subsume, TraX offers HLPM methods and optimizes the utilization of the data link between device and server by exchanging as few messages as necessary over the air-interface. This way, possible costs for bearer services like GPRS or UMTS packet-switched are reduced and bandwidth as well as the battery of the mobile device is saved.

To evaluate TraX a software project was carried out with a team of students, which is described in the next section.

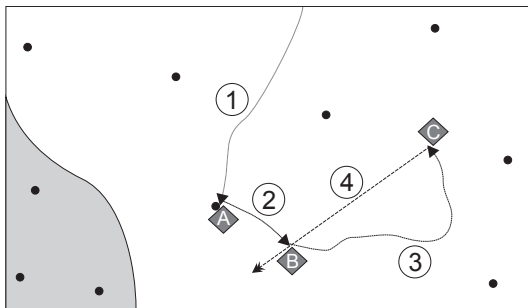
## 4 VsGolf

VsGolf is a location-based game for multiple users supported by the TraX framework. The game concept has been developed by six students during the practical course 'Mobile Gaming' at the LMU Munich in the summer term 2006. The students were free to come up with any idea, limited only by three requirements: multiple players at a time should be supported, GPS positions should be an integral part of the game, and the target platform was determined to be Java 2 Micro Edition (J2ME).

The students came up with the idea of VsGolf, a mobile game where two or more teams compete *versus* – the

“versus” giving VsGolf its name – each other over a period of time, such as a week. Each team controls a pre-defined home area, for example, a certain territory within the university campus. All around and also within the areas there lie about virtual golf balls. They are not owned by a specific team, but can be played by members of any team who happen to come across – either by accident or by searching for it. Having many golf balls located in one’s own home area means bad news for a team, because every one of these balls induces point losses as time passes. On the other hand, a team’s score increases when a member lands a ‘good shot’, which refers to sending the ball with a long stroke directly into the home area of another team.

For shooting a ball, it has to be ‘picked up’ first, which is achieved by the player (actually her mobile device) physically entering within a certain radius around the ball (in the tested scenarios 10 meters radius worked fine). As soon as a player steps into a golf ball’s ‘hot’ area, she is informed by her device that the ball is ready to be played, which is signalled either acoustically or by vibration. She is now supposed to choose a reasonable spot for placing the ‘tee’, that is, the place from where the ball will be shot. To do so, the player walks about freely and presses a button to drop the ball where preferred. Now, the player needs to take a run-up and aim. For this purpose, she departs from the dropped ball to a well-chosen position, presses the button ‘kick’, and starts running back, over and beyond the point the tee was placed, compare also Figure 2.



**Figure 2. Illustration of a VsGolf scenario: 1. The player steps into the hot area (A) of a golf ball and decides to kick it. 2. She finds a place for the ‘tee’ and drops the ball there (B). 3. The player steps back from the ball, aims at the home-area of an adversarial team and presses ‘kick’ when done (C). 4. The player runs over the spot, where the ‘tee’ was placed (B) – the golfball gets shot.**

The direction from where a player comes running over the tee spot determines the direction in which the ball takes off, whereby the force it is kicked with is derived from the

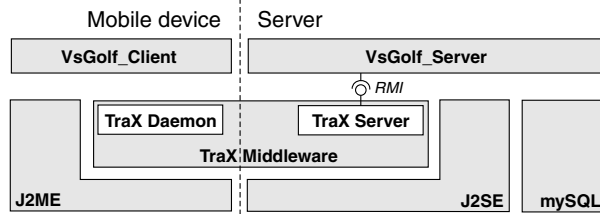
average running speed measured between pressing ‘kick’ and reaching the tee. Note that very short run-ups lead to a strong suffering from position inaccuracies. Therefore, it is worthwhile for a player to take a medium or long run in order to achieve a precise shot.

Since one match of VsGolf normally lasts for a longer period of time, it can be played in different ways. In ‘active mode’ a player seeks nearby golfballs on purpose to kick them. The mobile device shows the way to the next balls and is therefore held in the player’s hand all the time. If even the searching for golf balls is not desired, a player also has the chance to enter into a virtual ‘driving range’ to train her skills independently from her team’s scores. Nevertheless, VsGolf can also be played in an ambient way. In this configuration the mobile device may be seated in the player’s pocket and proactively generates an alert as soon as she incidentally comes across a golf ball. Upon the notice, she can then decide whether to play the ball or not.

To conclude, VsGolf possesses two distinguishing characteristics. First, one ongoing match can be played in active as well as in ambient mode, which makes the game suitable for many different situations of a person’s life. Second, in VsGolf the movements of a player are the primary way of interaction, that is, the player himself becomes a sort of ‘joystick’.

VsGolf is played by multiple players concurrently on the same set of data - the golf balls. Therefore, an approach based on a centralized server was chosen to synchronize the mobile devices. For communication, the TraX middleware, which connects the server and the devices by a socket connection for exchanging position fixes, provides a kind of virtual private network connection, over which typed data elements can be transferred between the VsGolf client and the server. It serves as the sole means of telecommunication in the game. Of course, the main function of TraX is to hide the complexity of different (partly vendor-specific) libraries to access available positioning technologies like GPS. The VsGolf client is therefore based only on the J2ME runtime environment available at the mobile device as well as the TraX daemon, to which it is linked by a library. Although the selected target platform was initially fixed to Benq/Siemens SXG75 – a smartphone with integrated GPS-receiver – essentially every J2ME device supported by TraX can be used for playing VsGolf. The VsGolf server is realized as a J2SE application and connected via Java RMI to the TraX server, which manages the connections of the mobile devices. For persistent data storage, a MySQL database was chosen. Figure 3 depicts the overall architecture. VsGolf was implemented in one coherent, end-of-semester session by a student development team following an extreme-programming approach. The whole game development, comprising the definition of the overall software architecture, the actual coding and debugging

– resulting in approximately 6000 lines of code – and final testings, was all done within four days. Before the start of the session, only a rough distribution of the main tasks (two people implementing the server, two the client, and two for managing and documenting the project) and the game concept was clear, having been developed over a few afternoons during the semester.



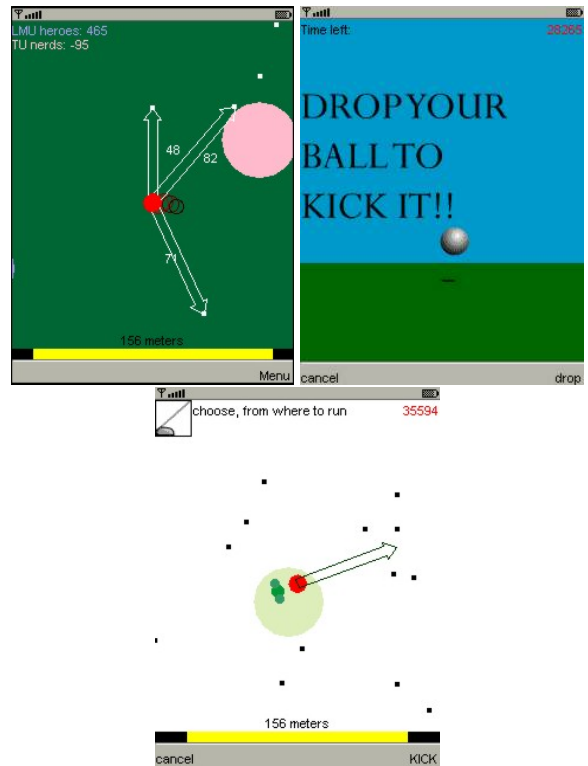
**Figure 3. Software architecture of VsGolf**

As it turned out, the TraX platform made the ambitious plan possible by delivering the following features: TraX’s LLPM takes care of filtering position fixes and generating position updates. This way it is automatically detected when a player enters the hot area around a golf ball. Also, TraX manages the data transfer between server and client by initiating and maintaining communication channels and providing means for reliable transport. For achieving both necessities, location-awareness and mobile connectivity, only a few of the provided interfaces on the client as well as the server-side needed to be implemented.

Since TraX is event-driven, and realtime graphics in video games classically rely on a loop-based, single-threaded approach, VsGolf resulted to be a hybrid of both concepts: In ambient-mode, the software is inactive until an event (a user-input or the entering of a golf ball’s hot area) occurs. In active mode, however, the visualizations of the player’s position relative to the golf balls and the team areas were implemented by a single repeating rendering function.

To evaluate the game, several field tests were conducted. Two teams, each with two members, played against each other. It turned out that the rapidly developed prototype worked perfectly and even sophisticated features like step less zooming and sound and video animation were included. Although the players had seemingly a lot of fun, two major problems regarding GPS were discovered early in the tests:

First, obtaining highly-accurate position fixes seemed crucial for a positive gaming experience. Since GPS works best when there are no obstacles around, like buildings or trees, we became clear about the fact that VsGolf can not be played everywhere. For example, in the English Garden in Munich already the occasionally occurring shadows of trees increased the GPS inaccuracy up to 50 m, which made it hard to hit once-dropped golf balls at all. Also, when line-of-sight to the GPS satellites was actually given, the



**Figure 4. Left top: 'active screen'; the central red dot represents the user’s position; arrows point to the next three balls and show the corresponding distances. Right top: 'drop-screen'; the player places her 'tee'; Bottom: 'aiming-screen'; the player has to find an adequate spot to run from; the arrow indicates the driving direction, when the user starts running from her current position.**

received position data turned out to be quite jittery (with an average variation of about 10 meters when standing). This made it necessary for players to step quite far back from the point where they placed their tee, in order to achieve somewhat predictable drives.

Second, since the used mobile devices are not equipped with a compass, all map-displaying screens were simply oriented north. This showed the disadvantage that players – especially, when they just switched from ambient mode to active playing – could not estimate in which direction on the map they were heading. In combination with imprecise position data from the GPS device, this resulted in disoriented players walking at first in the wrong direction before realizing their mistake. On the only rainy day we tested, some users failed to orientate themselves completely. This problem could be partly solved by showing the user a history of her last positions (see the small circles near the player’s

position in the left screenshot of Figure 4). Still, having a compass seems to be the better option.

Experienced problems with the used GPRS/UMTS data connection turned out to be tolerable. Interrupted or slow transmissions – which did occur – usually did not disturb the game flow, because the golf balls are transferred in the background. Lags of tens of seconds not even got noticed by the players. The whole process of finding and hitting the ball is controlled locally by the VsGolf client and therefore yielded very short response times, resulting in a good experience of interaction.

The massive battery consumption of the GPS receivers that are integrated in the used devices reduced the overall standby time during ambient mode strongly. One possible solution, which, however, has not been implemented so far, is to dynamically adjust the frequency of generated position fixes to the state of the game and the distance to the next golf balls in reach. This way battery resources could be saved.

## 5 Conclusions and Future Work

This paper presented a framework for proactive multi-user location-based services, called TraX, which is well suited for supporting location-based multi-player games. All raised challenges during the development of such a game, like support of proactivity, synchronization and exchange and correlation of position data, are handled by this framework.

This statement was evaluated and confirmed by the development of an innovative location-based multi-player golf game, which was completely developed in four days by a team of six non-briefed students. TraX was sufficiently flexible for the new application domain and the students found it easy to become familiar with the design principles and interfaces. Thus, it enables the rapid development of sophisticated mobile games permitting to shift more resources to game play related issues.

Future work will include user studies on usability and usage patterns, but also the design and integration of new mobile gaming specific interfaces and features in the TraX framework like the management of mobile virtual objects and methods for controlling an avatar via movement of a person.

**Acknowledgment.** The authors would like to thank Florian Huber, Walter Kammergruber, Antonius Weinzierl and Nora Zelhofer for their inspiring ideas and their efforts during the development of Vs. Golf.

## References

[1] Botfighters, <http://www.botfighters.com/botfighters2/>. Website. Accessed September 09, 2006.

- [2] Gameboy on Wikipedia, <http://de.wikipedia.org/wiki/gameboy>. Website. Accessed September 22, 2006.
- [3] Gps::tron, <http://gps-tron.datenmafia.org/>. Website. Accessed September 09, 2006.
- [4] Mogi, <http://www.mogimogi.com/mogi.php?language=en>. Website. Accessed September 09, 2006.
- [5] Explorer, <http://rex-regensburg.de/im-stadtraum/rexplorer/>. Website. Accessed September 09, 2006.
- [6] Skeeters, <http://www.yewsoft.com/product.php?productid=15>. Website. Accessed September 09, 2006.
- [7] S. Bjrk, J. Falk, R. Hansson, and P. Ljungstrand. Pirates! - using the physical world as a game board. In *Proceedings of Interact: Conference on Human-Computer Interaction*, Tokyo, Japan, July 2001. IFIP.
- [8] A. D. Cheok, S. L. Teo, L. Cao, and L. N. Thang. Capture the flag: A multiplayer online game for phone users. In *ISWC '05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, pages 222–223, Washington, DC, USA, 2005. IEEE Computer Society.
- [9] E. G. JSR-179. Location api for j2me 1.0. September 2003.
- [10] A. Küpper. *Location-based Services — Fundamentals and Operation*. John Wiley & Sons, Aug. 2005.
- [11] A. Küpper and G. Treu. From location to position management: User tracking for location-based services. In *Tagungsband der ITG/GI-Fachtagung Kommunikation in Verteilten Systemen (KiVS 05)*, volume 61 of *Lecture Notes in Informatics (LNI)*, pages 81–88, Kaiserslautern, Germany, Februar 2005.
- [12] A. Küpper and G. Treu. Efficient proximity and separation detection among mobile targets for supporting location-based community services. *ACM SIGMOBILE Mobile Computing and Communications Review*, 10(3), July 2006.
- [13] A. Küpper, G. Treu, and C. Linnhoff-Popien. Trax: A device-centric middleware framework for location-based services. *IEEE Communications Magazine*, 44(9), September 2006.
- [14] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Proceedings of Pervasive 2005*, volume 3468 of *Lecture Notes in Computer Science*, pages 116–133. Springer-Verlag, Munich, Germany, May 2005.
- [15] A. Leonhardi and K. Rothermel. Protocols for updating highly accurate location information. In A. Behcet, editor, *Geographic Location in the Internet.*, pages 111–141. Kluwer Academic Publishers, 2002.
- [16] K. Mouratidis, D. Papadias, S. Bakiras, and Y. Tao. A threshold-based algorithm for continuous monitoring of k nearest neighbors. In *IEEE Transactions on Knowledge and Data Engineering*, volume 17, pages 1451–1464. IEEE Educational Activities Department, November 2005.
- [17] O. Rashid, I. Mullins, P. Coulton, and R. Edwards. Extending cyberspace: location based games using cellular phones. *Comput. Entertain.*, 4(1):4, 2006.
- [18] Z. Xu and H.-A. Jacobsen. Efficient constraint processing for location-aware computing. In *Proceedings of the 6th International Conference on Mobile Data Management*, pages 3–12, New York, NY, USA, 2005. ACM Press.