MOBILE CHASE
Towards a Framework for Location-Based Gaming

Mirko Fetter
Faculty of Media, Bauhaus University Weimar, Bauhausstr. 11, Weimar, Germany
mirko.fetter@medien.uni-weimar.de

Markus Etz, Heiko Blechschmied
Graphic Information Systems Department, Fraunhofer Institute for Computer Graphics
Fraunhoferstr.5, Darmstadt, Germany
markus.etz@igd.fraunhofer.de, heiko.blechschmied@igd.fraunhofer.de

Keywords: Location-based Gaming, Pervasive Gaming, Mobile Computing, Computer Supported Cooperative Play.

Abstract: Pervasive Gaming and Location-based Games in particular have gained more and more attention recently. Researchers from a variety of fields, media artists, mobile service providers as well as the entertainment industry all seem to have their specific interests in this area. Today a couple of different games exist from basic applications that are already available to the consumer market to bleeding edge research projects. In this paper we introduce a framework for Location-based Gaming that on the one hand helps with the development of market ready games. On the other hand it serves as a toolkit for researchers aiming to rapidly develop Location-based Games, not having to deal with implementation details far away from their research interests in order to focus on their specific research aspects.

1 INTRODUCTION

Modern information and telecommunication technologies offer more and more possibilities for interaction, entertainment and recreational activities. In Pervasive Games the physical world merges with the virtual one and thus offers new adventures and experiences for the players by utilizing these technologies like mobile devices, tangible user interfaces or augmented reality. To increase the level of immersion these games incorporate the players' context by means of e.g. time, place, or personal data.

Location-based Games are one of the many areas where the concepts of Pervasive Gaming come to life. Thereby single players or teams perform tasks in specified scenarios using mobile computers like laptops, personal digital assistants or mobile phones in combination with wireless communication and location-sensing technologies, having the real world as their game board.

There are many reasons to be interested in the current development of Location-based Games. On the one hand there are clear market issues. As current video and computer games are already selling to a billion dollar market and the analyst Forrester (Forrester Research Inc., 2000) stated that Pervasive Gaming will gain a big market share there are also other interested parties besides the game development companies. The mobile service providers might see the possibility that Location-based Games could be a real driver for Location-based Services and technologies like 3G. On the other hand Pervasive Games may help researchers to test and analyze the use of a wide spectrum of technologies in game scenarios. The findings and things learned could then be transferred to other ubiquitous computing application fields.

With the framework presented in this paper we introduce a tool which can serve as a solid base for developers to implement their own gaming ideas. Therefore the framework not only takes care of data management and communication tasks but also delivers a pattern for the modelling of Location-based game content on server- and client-side.

The remainder of the paper is organized as follows: In section 2 we present work which has been done on the area of Location-based Games in several fields. In section 3 we outline the basic requirements of the framework and give a detailed overview of the technical concepts. Further on, in section 4, we describe "Mobile Chase", a prototypically-implemented game based on the framework. In the final section we outline future work and draw conclusions.

2 RELATED WORK

Pervasive Games can appear in many forms as described by Magerkurth et al. (Magerkurth, Cheok, Mandryk, & Nilsen, 2005), the genres span from smart toys over augmented tabletop games to a form that he calls Location-Aware Gaming. Thereby the approaches from different genres sometimes are mixed in one system and bring up interesting combinations that are hard to differentiate.

Location-based Games can be distinguished by the fact that they use the position of the player in the real world as a major input to the game process. In these games the real world is seen as a game board so they can be considered as the consequent answer to ideas like Live Action Role Playing Games - which bring classic Pen and Paper Role Playing Games to life in events lasting several days - Alternate Reality Games like "The Beast" (McGonigal, 2003) - that use the full bandwidth of modern media and communication technologies to blur the borders between game and reality - and modern video games.

A couple of games and systems have already been developed coming from fields like research, media art, or the industry. While last mentioned according to Jegers (Jegers, 2004) only use a small part of what the Pervasive Gaming paradigm offers and are technically reduced to fit today's user devices, some of them are already quite successful from a market perspective. For example the Tokyo based game "Mogi, Item hunt" (KDDI, 2003) where it is the player's task to collect virtual items spread over the city and to complete different collections of items. Or the first commercial Location-based Game "Botfighters" (Botfighters.com, 2005) developed by "It's alive" which appeared in 2001 and is about to be released in version 2.0 where the user plays a robot and has to solve missions hunting other players. Both games can be run on standard mobile phones supporting J2ME, using Cell-ID based positioning methods.

The approach of Blast Theory, a group of London based artists seems in contrast very different. They developed a couple of Location-aware Games like "Can You See Me Now?" (Flintham et al., 2003) or "Uncle Roy All Around You" (Benford et al., 2004) that come to life on the streets as a sort of artistic event or interactive performance. In contrast to the commercial games mentioned before, these events take place for a fixed time only with a limited number of participants. This is somehow reflected in many details of the realisation. Professional actors interact with the players. In addition, as the devices are provided to the players for the time they play, the technological constraints to cope with are fewer than the constraints commercial game developers have to take care of, which leads to more sophisticated game interfaces and a gain in the level of immersion during play time.

Though Blast Theory collaborates with different research institutes their interests seem to derive from their artistic background in the first instance. Therefore we also want to mention three contributions made by the research community.

In "Pirates!" (Björk, Falk, Hansson, & Ljungstrand, 2001) the homonymous classic computer game from the renowned game designer Sid Meier was brought to real life. The task of the player is to "sail" between different locations, representing islands, to search treasures and battle near players all by using handheld computers, WLAN and proximity sensors.

To prevent a virus from escaping from a campus, the players of "Epidemic Menace" (Lindt, Ohlenburg, Pankoke-Babatz, Prinz, & Ghellal, 2006), a pervasive crossmedia game, are equipped with a variety of mobile devices - some of them offering mobile Augmented Reality (AR).

Another research project dealing in the area of Location-based Games was the publically funded German research project GEIST (Holweg, Schneider, & Göbel, 2004). The goal of the project was to provide pupils with historical information via means of edutainment. Virtual ghosts from the past gave the pupils tasks by which they could learn facts about the town history. Therefore the children were provided with a wearable computer connected to various positioning devices and a semi-transparent AR-display on which digital reconstructions of historical buildings were shown on their original position together with the virtual ghosts.

As we can see, a lot of work has been put into the field of Location-based Gaming, and a lot of good ideas and systems have been developed. However it seems that many implementations are tightly bound to the implemented game concept. Therefore our approach is a more generic one, not focusing on the game idea, but on the underlying concepts that are common to every Location-based Game. In the following we present our proposal for that generic approach.

3 A FRAMEWORK FOR LOCATION-BASED GAMES

In this section we describe the requirements we had concerning the framework, which will directly lead us to the concepts and implementation details.

3.1 Requirements

As mentioned it was our aim to define a generic platform for Location-based Games. Therefore a major requirement was to define the design of the framework as openly as possible in order to allow a variety of different game ideas. The framework should give the game designers the freedom to implement their concepts without being constrained to technical shortcomings or bound to specific devices or services. For example the selection of the best fitting positioning technology should be in the hands of the game developer not limiting him to a specific accuracy or the fact that the game can only be played outdoor, etc.

Therefore we defined a J2ME enabled mobile device supporting MIDP 2.0 (SUN Microsystems Inc., 2002a) and CLDC 1.1 (SUN Microsystems Inc., 2003) with network access and some positioning mechanism - which can be at a low level like an address input - as the minimal specification for a mobile player.

Though there already exist some great frameworks, middleware, and platforms for mobile multiplayer games like Exit Games Neutron 3.5 (Exit Games, 2006), Nokia's SNAP Mobile (Nokia, 2006), or the proposal of the Mobile Games Interoperability Forum (Mobile Games Interoperability Forum, 2002) , their focus does not lay on supporting Location-based Games. Therefore another requirement while planning the framework was to identify those concepts, which are specific to Location-based Games, and to concentrate on these. On the other side, this means to clearly omit services and concepts that can also be found in normal multiplayer game platforms and often just surround the actual game like e.g. Competition Management, Buddy Management or Match Making which are suggested by the Mobile Games Interoperability Forum.

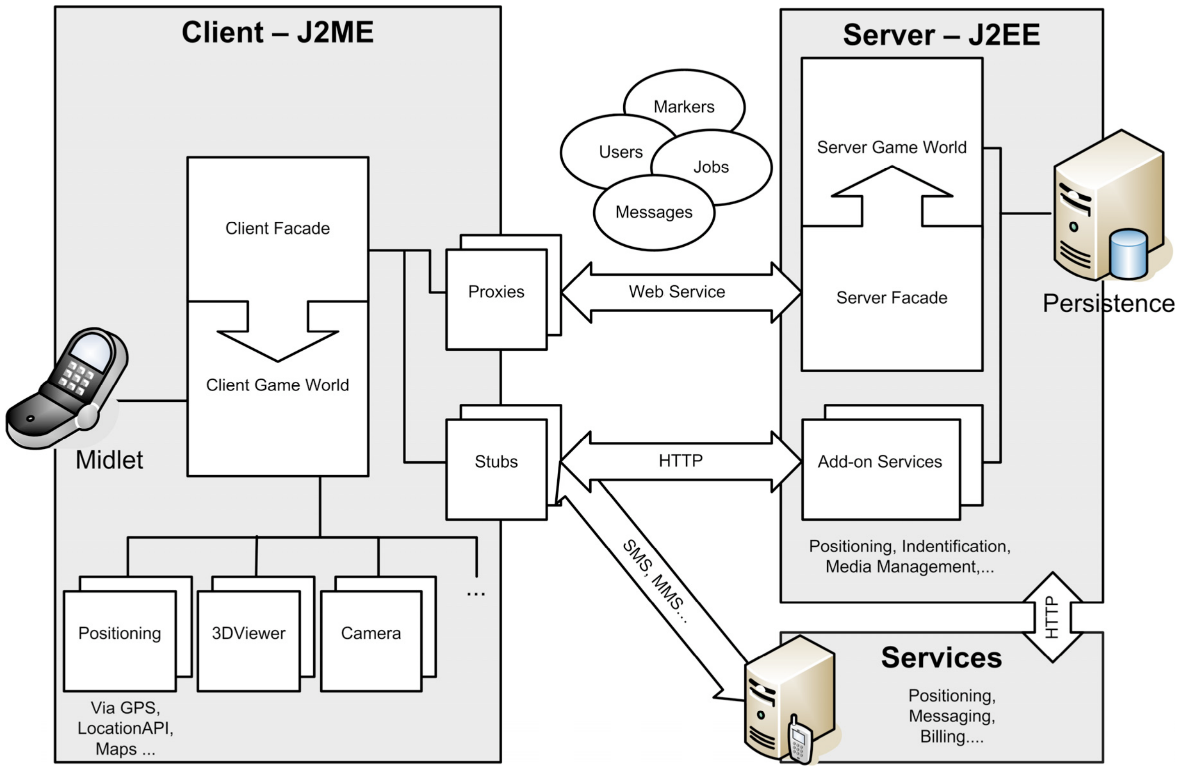

Figure 1: Schematic overview of the framework and its interfaces.

3.2 The Framework

The framework is completely implemented in Java whereby the basic concepts are implemented in a way that they can be run on devices supporting J2ME, J2SE or J2EE. The base of the framework builds a domain model that maps the concepts, entities and their relationships that can be found in Location-based Games.

Therefore we have identified four basic entities that are common to most of the Location-based Games: `User`, `Marker`, `Job` and `Message`. The class `User` represents a player, which would normally be a human, mobile player. But the concept also allows to model e.g. virtual players, controlled by stationary players or the CPU, that interact with players in the real world. The `Marker` class models places or objects with a position relevant to the game. Thus a marker can be an actual object like a house, a monument, or a public display or just mark a position where a virtual object can be found like a waypoint or some virtual item that can be collected by passing by. As `User` and `Marker` both have a spatial element, their position is manifested in the class `PositionableObject` both inherit from. Besides the current and last position stored in the WGS 84 format (NIMA, 1991) the class holds information like the name, team name, or an image that can be used for multiple purposes like an avatar, additional information like the photo of a place, or visual tags to identify a player or marker via computer vision.

A `Job` describes a single objective that has to be accomplished. Each `Job` has inter alia an owner who has to complete the task and a subject. To fulfil the requirement, to restrict the game design as less as possible, a `Job` could comprise and be assigned to either a `User` or a `Marker`. This offers the possibilities to design games in which it can be the job of a player to catch another player, or to find a waypoint. It is even possible to design games where a `Marker` is in charge of fulfilling a job. For example some sort of digital artefact like a Bluetooth beacon could identify players passing by.

The class `Message` is modelled in the same open way to give players the possibility to send messages to their team mates, to public displays represented by a `Marker`, and so on. It is in the hands of the game designer whether the text field of the class is just used to send plain text messages or to utilize this system and for example XML-Serialization to let the players exchange collected items, etc. via messages.

Besides these four basic concepts we have defined one central abstract class, the `GameWorld`, which finds its manifestation on the server and the client side and has to be implemented by the game designer. The most important task of the `ServerGameWorld` and the `ClientGameWorld` is to provide the basic game logic and rules and act as a Mediator between the single instances. Besides the methods addressing the entities like e.g. `getJobsByUserID`, `getLocations`, or `setJob`, etc. which are already implemented the game designer has to realize a couple of methods dealing with the game logic and presentation. Depending on where the computation can be done these methods have to be implemented on the client or the server side in the specific `GameWorld` class. These methods are dealing with the provision of the playing field as a mobile 3D model or 2D graphic or the effects it has to the game process when players update their position or identify other `Users` or `Markers`.

3.3 Technical Details of the Framework

As mentioned before the framework has been implemented in Java designed to be as less demanding on resources as possible. In addition the use of open standards makes it easier for the later game developer to make modifications and use standard tools. Therefore the communication between the server and the clients is realized as a web service using SOAP (Gudgin, 2003) instead of defining a proprietary low-level exchange protocol of its own.

The server side offers all the methods that are intended for the client encapsulated in the `ServerFacade` utilizing Apache Axis (The Apache Software Foundation, 2006). The J2ME clients use kSoap (Haustein, Morgan, Beatty, & Mcdaniel, 2003) to consume the web service and also make use of the Façade pattern in combination with the Proxy pattern. In this way the flow of the framework-relevant data is tightly packaged and offered at clear endpoints.

If it is necessary to access other services e.g. looking up street addresses on GIS servers for positioning, applications identifying visual tags, or billing services this has to be done bypassing the framework's communication flow - as can be seen in Figure 1.

4 THE GAME "MOBILE CHASE"

Based on the principle of the good old paper chase, we all will remember from our childhood days, we implemented the game "Mobile Chase" in order to test and refine the functionality of the framework. In the following we explain how we adapted the principle of the classic game and adapted it to a Location-based Game and give some detailed information on the implementation.

4.1 The Game Principle

Exactly as in the classic archetype a team of players runs through town marking their route each time they change direction. A team of pursuers starts with a several-minute delay in order to track down the first group. Instead of using chalk arrows, in "Mobile Chase" the first team has to mark each shift in direction with a photo, which is then geo-coded using GPS and uploaded to the server by the cellular phone.

Figure 2: Students Playing "Mobile Chase".

The pursuing team can use GPS to verify if they have reached the next 'marking'. Once the pursuers are at the right spot, the phone fetches the next picture from the server, which is superimposed in correct positional arrangement on a three-dimensional playing field on the phone's display (see Figure 3). This way the pursuing team follows the route of the first team trying to catch up. When the first team of players is finally in sight, the pursuing team can 'catch' them via Bluetooth, and the time is taken. The teams can now change role, and compete for shorter times.

4.2 Implementation

The client-side implementation of "Mobile Chase" was done in Java ME, thus allowing, on the one side, the use of the technologies provided by the framework and, on the other side, the porting of the game to a variety of devices.

As mentioned above various tasks had to be fulfilled in order to implement the game play planned for "Mobile Chase". One of the key tasks was to retrieve the players' location during the game. This location was used, on the one side, to georeference the taken images and, on the other side, to compare the positions of the pursuing team with the markings. Here two methods were evaluated during the implementation phase, one based on the Location API (JSR 179) (SUN Microsystems Inc., 2005) provided by cellphones in combination with positioning data retrieved from the cellular network, the other one using a satellite-based localization technology. While from a theoretical point of view the location data retrieved from the cellular network could have been used for a first rough localization of the players, tests showed that the usage of this data depended on a collaboration with network providers, a prerequisite which was not met during this project phase. Therefore the localization in the final implementation was done using GPS. Technical obstacles connected to GPS like the impreciseness related to shadowing effects in city areas were overcome by allowing a reasonable degree of inexactness during the game play.

The second main technical task in the implementation of "Mobile Chase" was the visualization of the playing field. Here one of the main tasks, which had to be fulfilled, was the playability of the game and the satisfaction of the players. On the one side, the locating of the markers should not be made too easy for the pursuing team as this would reduce the "fun factor" of the game and make it too boring for the players. On the other side the team should be provided with enough hints allowing to find the markers' position in the real world and to catch the first team of players.

We decided to use a three-dimensional playing field, an adaptation of the "Mobile 3D Viewer"-Technology developed at Fraunhofer-IGD (Blechschmied, Coors, & Etz, 2006), which is based on the Mobile 3D Graphics-API (JSR 184) (SUN Microsystems Inc., 2002b) and allows the visualization of three-dimensional virtual worlds on mobile devices. As described in chapter 4.1, the markers are located in correct position relative to each other on the playing field, allowing the players to deduce information out of the playing field like the distance and the direction from one shown place to the next one. Technically the visualization of the markers was realized by using textured cuboids which were added to the 3D scene. By that it was possible, on the one side, to support the pursuing team in their task without, on the other side, providing the players' with too much information like in a regular street map where the additional information like the road names would have affected the complexity of the players' task.

Figure 3: Zoomed view of the next Marker and a view of the 3D display of the playing field showing four Markers and the game menu.

Another main technical element of the client-side application was the implementation of a module, which allows the first team to use the camera integrated in the mobile phone to generate the markers and to transfer the pictures to the game server. For this task the Mobile Media API (JSR 135) (SUN Microsystems Inc., 2006b) was used. Here the central class `Manager` of the API allows the access to the resources of the mobile device like the video stream of the integrated camera. After having made a photography, the user is shown the picture on the screen, giving him the possibility to accept the picture and send it to the game server using a HTTP-connection. As the photos which are taken by the first team are to be integrated into the playing field as textured cuboids, the transferred photos are afterwards scaled to a size of 128 x 128 pixels on the server. This is done due to the prerequisite of the Mobile 3D Graphics-API only allowing textures with a side length of 2^x.

To geo-reference the taken photos, an external GPS device was connected to the mobile phone using the Bluetooth API (JSR 82) (SUN Microsystems Inc., 2006a). Here the data retrieved from the GPS device in form of the NMEA-protocol - a protocol containing amongst others the retrieved position - were parsed on the mobile device and used to determine the position of the players. The identification of the first team at the end of the game was another task where the Bluetooth-technology was used. By using the class `BTDeviceDiscovery` the pursuing team starts a so called discovery process scanning for other Bluetooth-enabled devices in its range, usually a few meters. The scanned Bluetooth-addresses are sent to the game server, where the addresses from `ServerGameWorld` and `ClientGameWorld` are compared. If one of the scanned addresses corresponds to the Bluetooth-address of the first team's device, the job is closed and the pursuing team wins the game.

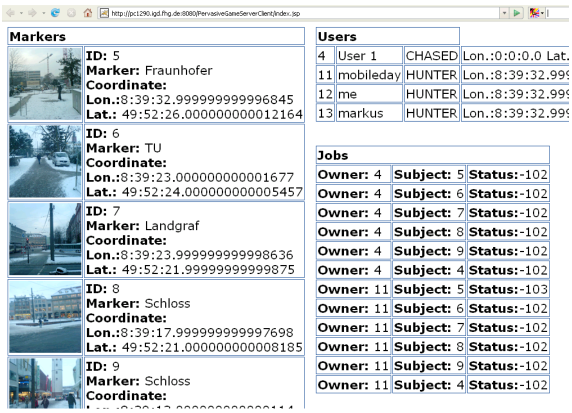

Figure 4: Web-interface for the game-player.

To demonstrate the flexibility of the framework an additional web-interface was realized. This interface based on the functions offered by the `ServerFacade` allows the retrieval of the current markers, users and their jobs and their representation in table form and the reset of the game world. While the different teams are playing "Mobile Chase" on their mobile devices, this web-interface allows the online administration of the game via a game master or the inclusion of online players watching the progress of their teams (see Figure 4).

5 FUTURE WORK

Of course one major future task will be to implement a variety of different games that will focus on different research aspects and get them played. Based on ethnographic studies, like first-hand observations or interviews etc. of people playing these games we expect insights in different fields like playability and usability on the one side but also effects on heterogeneous research topics.

One of these topics is how digital tools can help mobile people working together and collaborating in different environments and how these tools can be improved. Today we have to deal with uncertainties like e.g. unsatisfactory network coverage or with mobile devices that are not specialized to specific tasks, and therefore are highly complex to use. Having people play these games, we will probably see what they will accept and where current lacks in acceptance have to be addressed.

Another main research focus is the usage of the developed technologies in “serious” location-based technologies like locating and navigating systems on mobile devices. As an example, the visualization of markers on the playing field seem to be a good base for representing Points-of-Interest in a Location-based Service (LBS) while the visualization of a three-dimensional map and the usage of this map inside of the game could provide us with information about the usability and the navigational help provided by 3D map representations. Concerning the localization of an LBS-user, future work will be done in order to overcome the challenges resulting from restricting factors like the inexactness of cell-based location data or the absence of satellite-based locating data inside of a closed building. These insights will also give us a deeper understanding on how the physical world and electronic spaces can be better matched as described by (Gross & Specht, 2001) in the field of context-aware applications (Schilit, Adams, & Want, 1994) to provide a seamless movement between these worlds.

In the end, all this knowledge will of course help us refine the framework and integrate these new findings. In this way, we want to extend the area of application of the framework from supporting games to support a variety of different location-aware, cooperative applications.

6 CONCLUSIONS

As suggested by Starner et al. (Starner, Leibe, Singletary, & Pair, 2000) games seem to be an ideal arena to test new concepts and prototypes from the field of human-computer interaction. And as described by Davies et al. in the introduction of “Rapid Prototyping for Ubiquitous Computing” (Davies, Landay, & Hudson, 2005) we also see the dilemma concerning the implementation of ubicomp scenarios where researchers have to deal with the implementation of broad systems even if they just want to focus on a specific area. In this spirit we wanted to make our contribution by providing our framework. In our eyes the development and things learned from Location-based Games today might give us insights and ready-to-use technologies for a variety of application fields tomorrow. Our framework is designed to help researchers to implement technology and scenarios in order to get early insights in feasibility and the acceptance and usage by the potential users.

ACKNOWLEDGEMENTS

The work presented in this paper has been realized during the stay of Mirko Fetter at the Fraunhofer IGD under the supervision of Professor Horst Grauer from the University of Applied Science Wuerzburg. We want to thank everybody involved in facilitating this cooperation.

REFERENCES

Benford, S., Flintham, M., Drozd, A., Anastasi, R., Rowland, D., Tandavanitj, N., et al. (2004, Jun. 3-5). *Uncle Roy All Around You: Implicating the City in a Location-Based Performance.* Paper presented at the Proceedings of Advances in Computer Entertainment 2004, Singapore.

Björk, S., Falk, J., Hansson, R., & Ljungstrand, P. (2001, Jul. 9-13). *Pirates! Using the Physical World as a Game Board.* Paper presented at the Proceedings of Interact, Tokyo, Japan.

Blechschmied, H., Coors, V., & Etz, M. (2006). Interaction and Visualization of 3D City Models for Location-Based Services. In S. Zlatanova & D. Prosperi (Eds.), *Large-scale 3D Data Integration : Challenges and Opportunities* (pp. 117-135). Boca Raton, FL: CRCpress, Tailor & Francic Group.

Botfighters.com. (2005). Botfighters. http://www.botfighters.com.

Davies, N., Landay, J., & Hudson, S. (2005). Guest Editors' Introduction: Rapid Prototyping for Ubiquitous Computing. *IEEE Pervasive Computing, 4*(4), 15-17.

Exit Games. (2006). Exit Games Neutron 3.5 (Vol. 2006). http://www.exitgames.com/.

Flintham, M., Benford, S., Anastasi, R., Hemmings, T., Crabtree, A., Greenhalgh, C., et al. (2003, Apr. 5-10). *Where On-Line Meets On-The-Streets: Experiences With Mobile Mixed Reality Games.* Paper presented at the Proceedings of the Conference on Human Factors in Computing Systems, Fort Lauderdale, Florida.

Forrester Research Inc. (2000). *Pervasive Gaming Goes Mainstream* (No. FOR275086).

Gross, T., & Specht, M. (2001, Mar. 5 - 8). *Awareness in Context-Aware Information Systems.* Paper presented at the Mensch & Computer - 1. Fachuebergreifende Konferenz, Bad Honnef, Germany.

Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., Nielsen, H.F. (2003). SOAP Version 1.2. http://www.w3.org/TR/soap12/.

Haustein, S., Morgan, B., Beatty, J., & Mcdaniel, S. (2003). Enhydra Ksoap Project. http://ksoap.objectweb.org/: Enhydra.

Holweg, D., Schneider, O., & Göbel, S. (2004, Oct. 11). *GEIST - History Experience at Face (GEIST - Geschichte vor Ort erleben).* Paper presented at the Conference Electronic Imaging & the Visual Arts, Berlin.

Jegers, K. (2004). Usability of Pervasive games. http://www.ipsi.fraunhofer.de/ambiente/pergames2005/papers/Usability of Pervasive Games.pdf.

KDDI. (2003). Mogi, item hunt (Vol. 2006). http://www.mogimogi.com/.

Lindt, I., Ohlenburg, J., Pankoke-Babatz, U., Prinz, W., & Ghellal, S. (2006, Apr. 22-27). *Combining Multiple Gaming Interfaces in Epidemic Menace.* Paper presented at the Conference on Human Factors in Computing Systems, Quebec, Canada.

Magerkurth, C., Cheok, A., Mandryk, R., & Nilsen, T. (2005). Pervasive Games: Bringing Computer Entertainment Back to the Real World. *Computers in Entertainment (CIE), 3*(3), 4-4.

McGonigal, J. (2003, May 19 - 23). *This Is Not a Game: Immersive Aesthetics and Collective Play.* Paper presented at the MelbourneDAC, the 5th International Digital Arts and Culture Conference., Melbourne.

Mobile Games Interoperability Forum. (2002). Mobile and Wireless Gaming White Paper - Version 1.0. http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/mgif/MGIF_Whitepaper_v1.0.pdf: Open Mobile Alliance.

NIMA. (1991). *Department of Defense-World Geodetic System 1984 Technical Report (and supplements), 2nd Edition* (No. TR 8350.2): National Imagery and MappingAgency.

Nokia. (2006). SNAP Mobile. http://snapmobile.nokia.com/.

Schilit, B., Adams, N., & Want, R. (1994). Context-Aware Computing Applications. *Workshop on Mobile Computing Systems and Applications,* pp. 85-90.

Starner, T., Leibe, B., Singletary, B., & Pair, J. (2000, Jan. 9-12). *MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Game.* Paper presented at the Proceedings of the 2000 International Conference on Intelligent User Interfaces, New Orleans, LA, USA.

SUN Microsystems Inc. (2002a). JSR 118: Mobile Information Device Profile 2.0, Final Release. http://java.sun.com/products/midp/index.jsp.

SUN Microsystems Inc. (2002b). JSR 184: Mobile 3D Graphics API for J2ME. http://jcp.org/en/jsr/detail?id=179.

SUN Microsystems Inc. (2003). JSR 139: Connected Limited Device Configuration 1.1, Final Release. http://java.sun.com/products/cldc/index.jsp.

SUN Microsystems Inc. (2005). JSR 179: Location API for J2ME. http://jcp.org/en/jsr/detail?id=179.

SUN Microsystems Inc. (2006a). JSR 82: JavaTM APIs for Bluetooth. http://jcp.org/en/jsr/detail?id=82.

SUN Microsystems Inc. (2006b). JSR 135: Mobile Media API. http://jcp.org/en/jsr/detail?id=135.

The Apache Software Foundation. (2006). Apache Axis 1.4. http://ws.apache.org/axis/: The Apache Software Foundation.