Gamework

A customizable framework for pervasive games

Christoph Stach
Universität Stuttgart
Institute of Parallel and Distributed Systems
Universitätsstraße 38
70569 Stuttgart, Germany
Christoph.Stach@ipvs.uni-stuttgart.de

ABSTRACT

Location-based applications and services gain in importance as the spread of mobile devices equipped with GPS sensors and internet capabilities increases. Moreover, there is a trend towards customizable applications in order to let the user determine e.g. the content and the look and feel. Nevertheless, there exist only a few applications combining these features.

In this paper, we propose Gamework, a framework for customizable mobile location-based games as a specialization for customizable mobile context-aware applications and services. According to their programming skills players are able to adapt a game with Gamework. This can be done by changing the context, adding user-generated content, modifying the game-flow or implementing new games from scratch by reusing existing modules of the framework or adding new ones. Therefore our framework features a reuse-oriented development methodology as well as a feedback loop analyzing all accruing user-generated content. The results of this analysis are used to automatically optimize the game with respect to frequent user-feedback. Finally, we will transfer the results back to the more general area of mobile and context-aware applications and services.

Categories and Subject Descriptors

D.2.13 [**Software Engineering**]: Reusable Software; H.2.8 [**Database Management**]: Database Applications; K.8 [**Personal Computing**]

General Terms

Design, Data analysis, and improvement.

Keywords

Mobile services, pervasive multi-player games, customizable framework.

1. INTRODUCTION

When in the early 90s the first context-aware devices such as active badges appeared [3], these devices were accessible only by a small group of researchers and served the intended purpose only. Since these early days of mobile context-aware systems (when Mark Weiser introduced his visionary idea of powerful computer systems, which integrate seamlessly into our daily routine [25]) technology made huge improvements. In fact, modern Smartphones possess not only powerful processors and plenty of main memory, but are also equipped with components such as a "big" touch-screen, GPS sensors, Bluetooth, Wi-Fi, and much more. Thus, these devices fulfill nearly all requirements of pervasive applications. Since these devices are very popular and not only accessible to a small group of experts, the number of applications for these devices is growing rapidly. In order to prevent inventing the wheel over and over again an adaptable framework with reusable components and a reuse-oriented development methodology would be extremely beneficial.

On the other hand, there is a clear trend towards so-called "crowd sourcing". [8] describes this as the next step in lean management techniques for the Web 2.0. Organizations allow their customers to customize and extend their applications and let them include their very own content data and share it with a community. While this means fun as well as new possibilities for the users, the companies can analyze this data and use the knowledge to improve and enhance applications for the benefit of their customers.

Our approach, called Gamework, brings together these two trends. We will develop a framework for mobile context-aware and community-based service technologies. We base this approach on our experiences gained with previous work on pervasive games introduced in [22]. While the Gamework approach currently targets the development process of games only, there is a clear perception that it carries over to mobile and context-aware applications in general. So it can also be useful to applications from other areas.

2. CONTRIBUTION

As mentioned above, we want to use the application area of games as a kind of "playground" for the development of new and innovative techniques for customizable mobile location-based applications and services as well as for easily gaining usage experience. This is similar to the well-accepted approaches in artificial intelligence using soccer playing robots as a playground for e.g., research on team strategies.

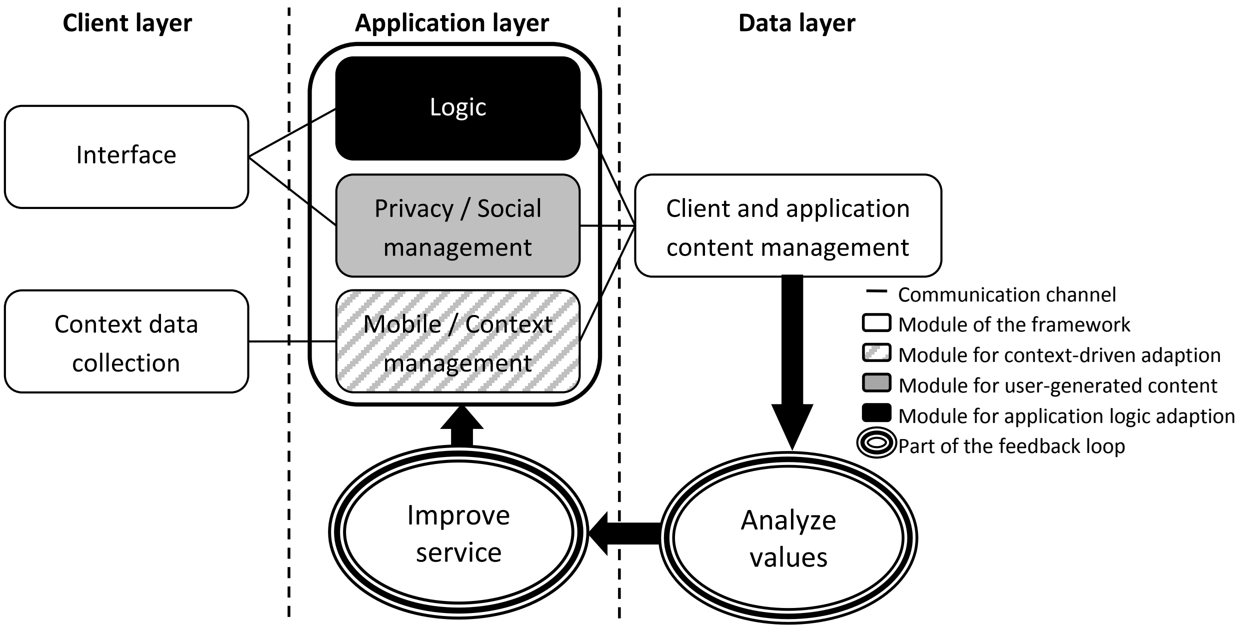
Client layer
Application layer
Data layer
Interface
Logic
Privacy / Social management
Client and application content management
Context data collection
Mobile / Context management
Communication channel
Module of the framework
Module for context-driven adaption
Module for user-generated content
Module for application logic adaption
Part of the feedback loop
Improve service
Analyze values

Figure 1: General architecture for mobile, context-aware application services

Our approach features a general framework and architecture for mobile and context-aware applications and services as depicted in Figure 1. It builds upon a three tier architecture which separates the client from application issues as well as the application logic from instantiation issues.

Supported by the framework approach and the reusability of the modules as well as by the customization-oriented development methodology, a user should be able to customize such a service to a certain degree according to his or her needs. Furthermore, the framework offers a self-improvement functionality that is realized by the content analyses feedback loop shown in Figure 1. Since all client and application data are available in the content management component at the data layer, customer-specific as well as cohort-specific or crowd-specific analyses become viable in order to improve application or service quality. Therefore, this client and application content stored in the data layer is analyzed with some mining techniques in a first step. In a second step the gained knowledge is transfered back to the application layer in order to improve the service. Examples might be the incorporation of user or crowd habits, common and repeating information demands.

In order to evaluate this approach, we concentrate on mobile location-based games. We develop a game-oriented framework, investigate basic technologies, gain experiences via evaluation of a prototype implementation and finally transfer the results gained back to the more general area of mobile and context-aware applications and services.

3. RELATED WORK

Magerkurth, et al. identify pervasive games as an attempt to bridge the real world to computer games [11]. They present the different game genres that can be augmented by context data and that vary from tabletop games (e.g. SYGo [19]) over location-aware games (e.g. Uncle Roy All Around You [2]) to augmented reality games (e.g. ARQuake [17]). Thereby, it is inevitable for the player to get in touch with his or her social surrounding [12]. Contrary to this wide range of genres, [18] pick classic board games as best fitting in the field of pervasive games.

Albeit pervasive games have existed for a while, most of them have to be created from scratch. Therefore, [24] present FRAP, a framework to support and shorten the game development process. The development of games is aided even further by the composing guidelines how a pervasive game should be designed in order to rapidly get the player into the game introduced in [26]. In contrast, [27] state that a long-term attraction to a game can only be assured by allowing a player to generate his very own game content and share it with others. So customization is extremely relevant. Anyhow, there exists no framework usable for any user independent from his or her programming skills up to now.

In the field of pervasive applications some approaches aim to analyze all accruing sensor data (e.g. [1]). According to [20] these data streams are rough to handle because of their immense I/O usage and the requirement for huge storage devices especially in multi-user environments. Therefore, our approach focuses on data analysis working with any content a user creates by customizing a game or a service in order to realize feedback loop of our framework. In doing so, we will examine existing mining techniques in the Web 2.0 area (e.g. [23]).

4. PRELIMINARY WORK

For the last couple of years, the Collaborative Research Centre “Nexus” has been working on related topics. One important goal of Nexus is the development of a federated platform for generating, managing and providing global world models. [10] present methods for different problems in integration and sharing such context models in an open, global platform of context providers. Information about the physical world must be managed and provisioned for different pervasive applications in variable accuracy levels. [7] present

Table 1: Review of the four customization techniques

Game aspect	Context in games	User-generated content	Game-flow adaption	Creation of new games
Framework support	Geographic functions	Graphical editor	FSM	Modular structure
Required game support	Very high	High	Moderate	None
Programming skills required	None	None	Moderate	High
Degree of customization	Low	Moderate	High	Unlimited

an infrastructure which not only deals with these difficulties but also allows some synergetic benefits such as data reuse and sensor sharing. When applications use sensor data, they get no discrete values but continuous data streams. [6] present NexusDS, an approach to build data-flow graphs for stream processing. [13] introduce a service-oriented architecture supporting location-based applications in a federated context. [4] study how such context data could be used in applications for the Web 2.0. NexusScout [14] demonstrates how different use cases of location-based services can be facilitated by the Nexus platform.

Based on this rich variety of techniques and experiences with mobile and context-aware services and applications, the team of the Collaborative Research Center “Nexus” started already in 2001 with the work in the area of mobile and context-aware games. [16] investigate what special demands are made by pervasive games concerning hardware features as well as data infrastructure. Building on these results [15] developed the NexusRallye, a virtual scavenger hunt, as a sample for a location-based game.

This considerable piece of work on location-based methods and techniques is now taken over and extended in various ways in this approach. [21] deals with pervasive games running in common web browsers. That approach concentrates on games using just a plain browser and no additional plugins or extensions. In this work we identified special features regarding conceptual as well as technical issues. Furthermore, we present an initial framework to facilitate the development of pervasive browser games. This framework is built with a conventional three tier architecture. The front-end runs on a client, realizing the presentation and, in the special context of pervasive games, the provisioning of context data. In the back-end resides a database storing all game relevant content such as user data or game logic. The actual game runs on a server connecting those two layers. In addition to the execution of the game logic it also manages user accounts as well as context data. TreasureCache and T^4 – *TicTacToe in Teams*, two concrete prototypes for location-based browser games using this framework, are presented in [5].

In [22] we extended this initial framework in order to allow players to initially customize their games or add user-generated content. It is a specialization of the generic architecture given in Figure 1. Albeit, it already provides different customization techniques, the implementation of the framework is very plain. Therefore, it has to be redesigned from scratch for planned doctoral thesis. Also, the unique feedback loop is not a part of that approach. Our proposed work packages are presented in Section 5 in detail.

5. PROPOSED WORK

With this dissertation proposal we want to continue the successful work we have already started and that we have briefly described in Section 2 and in the last paragraph of Section 4. In the following, we will describe the major steps on how to proceed in that work:

- Support for various customization techniques
- Step-wise development of a comprehensive framework approach
- Prototype framework realization
- Evaluation and refinement
- Generalization from the games scenario to general mobile applications and services

These issues are covered in the following subsections.

5.1 Customization

Customization techniques are very important for any viable framework concept. Hence, our framework approach will support different kinds of adaption or customization techniques at different levels as shown in Table 1. We assume auto-adaptive games, e.g., games that automatically position the game field to the current location of the player (for location-based games this is in most cases the map clipping at the current position). Hence, the player gets well served, whereas the designer has to ensure that this functionality is available. For these games, our framework will support the designer with basic geographic functionality such as calculating the relative distance to a given location or converting relative coordinates to absolute ones.

User-generated content means adding new information to the game (e.g. hints in a location-based quiz) as well as expanding existing levels by new sub-ordinate targets as far as creating user-specific levels, e.g., by adding targets to an empty or new map. Therefore, our framework will have to include a map-editor with an easy to operate graphical user interface. This way, even players without any programming skills will be able to customize the game. To develop such an editor, control over item management is needed.

Customization of the game-flow and game logic is overly important. In order to do this, we exploit finite state machines (FSM) and interpret the flow-graph as an FSM. Each node represents a particular game situation and an edge mimics the next possible courses of action. With this, customizing the game logic can be accomplished by simply changing the FSM. An appropriate graph editor allows manipulating the

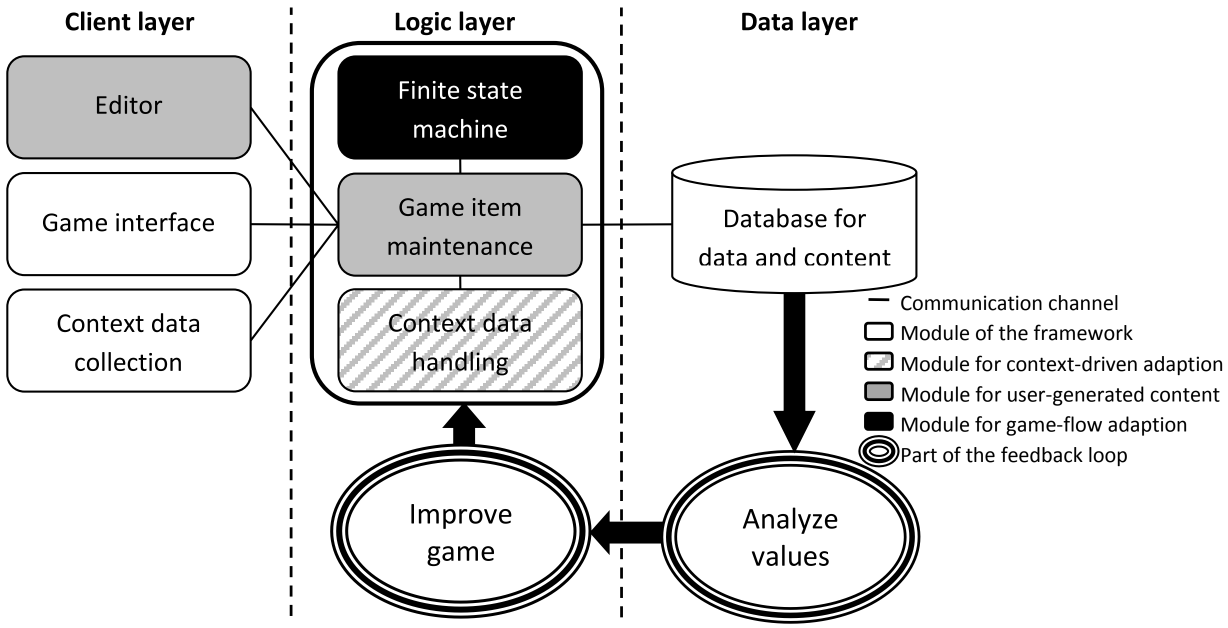

Figure 2: Gamework's architecture (drawn as a specialization of the general architecture from Figure 1)

FSM such that inexperienced users are able to customize the game logic. Finally, our framework reflects a modular design to ensure that each module or component can be replaced without any additional overhead. So creating new games will be supported by existing game modules and their customizability. For instance, if the game uses Cell-IDs instead of GPS to get the position of the players, only the context data collection module has to be exchanged, while the rest stays untouched. Even though (auto-) adaption techniques (as presented in [9]) are popular in current game projects, we may ignore them in this context because they are irrelevant for applications without artificial players.

Table 1 summarizes the prominent features of the four mentioned customization techniques and the four customization levels. As shown, the degree of customization possibilities increases from left to right moving the customization responsibilities from the pure player to the game designer.

5.2 Initial Framework Design

In order to enable the different customization issues mentioned above, the framework has to reflect a clear and modular architecture. As shown in Figure 2, we divide the modules of the framework into three separate layers. The main modules of our approach rest on a central server, while the mobile device is simply used as a thin client. Thus, in addition to the game interface the client side comprises an editor for customization purposes and a context data collection component.

The game item maintenance module represents the core of the framework and belongs to the middle tier, the logic layer. This module manages all game objects, i.e., it generates, consumes or destroys them. A game object is modeled by its characteristics such as its name, value, measurements, etc. Additional attributes can be added to any object, too. A game object can be both, a game item as well as an object representing a player or a user. This information will be forwarded to the context data handling component which provides functions such as geographic data support. The game item maintenance module also forms the interface to the modules of the client layer and to the database component at the data layer. A finite state machine models all game states (for instance login → start → . . . → end) as well as its different stages and even every task within these stages. Finally, the persistence module transfers all relevant game data to the data layer in order to store them in or collect them from the back-end database. This enables a simple load / save capability for a game as well as it is the basis to an analyses feedback loop that will be used to (auto-) improve the game.

Obviously, there is a huge congruence with the abstract approach for applications presented in Figure 1, not only in architectural matters but also in the modules provided by the framework. With this, we ensure that knowledge gained by the game framework can be easily transferred to general mobile and context-aware applications and services. This generalization step is addressed below in Subsection 5.6.

5.3 Completion of the Framework

The state of the art with respect to this framework approach is given in [22]. It presents a first draft of an architecture for our game framework, called Gamework. There we also sketch an initial framework prototype that, however, misses many important features and components, such as the customization editor, a clear context data separation, and the feedback loop among others.

The design and extension of the customization interfaces, which is done in this phase, happens in two steps. In the first step, all APIs of the modules will have to be revised in order to simplify their use. In a second step, adequate editors are implemented handling the elements in the logic layer using their APIs.

A viable prerequisite for the feedback loop is the realization of the data and content management that involves the persistence module and the database back-end as well as the component for game item maintenance.

The feedback loop introduces a kind of self-improvement functionality: Since all client and application data are available in the content management component at the data layer, customer-specific as well as cohort-specific or crowd-specific analyses become viable in order to improve application or service quality. Examples are the incorporation of user habits or crowd habits, common and repeating information demands.

A number of algorithms for analyzing game data are determined and subsequently evaluated and assessed. E.g. in a quiz this could be an algorithm which detects massive user-generated hints concerning a single question or issue.

5.4 Prototype Development

A concrete prototype of a customizable framework for pervasive games will be realized and furthermore used for the development of mobile and context-aware games. Within the Nexus project we have already worked on some pervasive games. We will be able to fall back on some proposed implementations in the area of customizable mobile location-based games. In addition, we will look for more games and realize them based on our framework and development approach. Special attention will be given to the outstanding features of our approach:

- Customization interfaces and editors
- Data and content management at the data layer
- Feedback analysis

5.5 Evaluation and Refinement Step

In the evaluation phase, the game implementations will be assessed from various points of view. There will be a technical evaluation, a design evaluation and a usability evaluation, again with special attention to customization, content management and feedback issues. For this evaluation we will identify various user groups differing in their knowledge in the field of computer science in general and programming techniques in particular. They will be allowed to work with our framework for an observed testing period. The results of this assessment, gained by observation as well as analyzing their feedback afterwards, will be processed in a framework refinement step.

In this phase, we also want to evaluate our approach and our game implementation in workshops for top talent upper school students as well as their teachers. In doing so, we hope to receive manifold views from different perspectives varying in age, experience, etc.

5.6 Generalization from games to mobile applications and services

Learning from (playing) games is the topic in this phase. The results obtained from our game work will be transferred to mobile and content-aware applications and services. For this work, we will significantly benefit from the work we already did in the Nexus project, where the focus is on general context-aware applications and services and where there is already a running platform that comprises context management, stream processing, application modeling etc.

We assume that the most interesting and also useful results of this generalization and transfer step concern the customization issue, the content management issue, and the feedback analysis.

6. SUMMARY

In our approach, we present Gamework, a framework for customizable mobile location-based games as a specialization for customizable mobile context-aware applications and services. With Gamework, a player may, according to his or her programming skills, adapt a game by changing the context, adding user-generated content, modifying the game-flow or implementing new games from scratch by reusing existing Gamework modules or implementing new ones.

In order to evaluate our approach, we develop a framework for games that supports a reuse-oriented development methodology and reusable components. Furthermore, we investigate basic technologies, gain experiences via evaluation of a prototype implementation and finally transfer the gained results back to the more general area of mobile and context-aware applications and services. Our strategy is to exploit the application area of games as a kind of "playground" for the development of new and innovative techniques towards customizable mobile and context-aware applications and services.

In addition to create a user-friendly, easy understandable, ergonomic design for our framework we identified the feedback loop as the major challenge in this dissertation. So we are aware of the fact that we will only be able to develop (auto-) adaptive solutions for given use cases. Nevertheless, our approach will be able to make a contribution not only to create new pervasive services but also to improve existing ones.

7. REFERENCES

[1] K. Aizawa, D. Tancharoen, S. Kawasaki, and T. Yamasaki. Efficient Retrieval of Life Log Based on Context and Content. In *Proceedings of the the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, CARPE '04, pages 22–31, 2004. http://doi.acm.org/10.1145/1026653.1026656.

[2] S. Benford, M. Flintham, A. Drozd, R. Anastasi, D. Rowland, N. Tandavanitj, M. Adams, J. Row-Farr, A. Oldroyd, and J. Sutton. Uncle Roy All Around You: Implicating the City in a Location-Based Performance. In *Proceedings of the 2004 International Conference on Advances in Computer Entertainment Technology*, ACE '04, pages 1–11, 2004. http://www.blasttheory.co.uk/bt/documents/Implicating%20the%20city.pdf.

[3] I. Bokun and K. Zielinski. Active Badges—The Next Generation. *Linux Journal*, 1998(54):24–26, Oct. 1998. http://www.linuxjournal.com/article/3047.

[4] A. Brodt, D. Nicklas, S. Sathish, and B. Mitschang. Context-Aware Mashups for Mobile Devices. In *Proceedings of the 9th International Conference on Web Information Systems Engineering*, WISE '08, pages 280–291, 2008. http://dx.doi.org/10.1007/978-3-540-85481-4_22.

[5] A. Brodt and C. Stach. Mobile ortsbasierte Browserspiele. In *INFORMATIK 2009: Tagungsband der 39. GI-Jahrestagung, 28.9. – 2.10.2009, Lübeck*, volume 154 of *LNI*, pages 1902–1913, 2009. http://cs.emis.de/LNI/Proceedings/Proceedings154/gi-proc-154-153.pdf.

[6] N. Cipriani, M. Eissele, A. Brodt, M. Großmann, and B. Mitschang. NexusDS: A Flexible and Extensible

Middleware for Distributed Stream Processing. In *Proceedings of the 2009 International Database Engineering & Applications Symposium*, IDEAS '09, pages 152–161, 2009. http://doi.acm.org/10.1145/1620432.1620448.

[7] M. Großmann, M. Bauer, N. Hönle, U.-P. Kappeler, D. Nicklas, and T. Schwarz. Efficiently Managing Context Information for Large-Scale Scenarios. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, PerCom '05, pages 331–340, 2005. http://dx.doi.org/10.1109/PERCOM.2005.17.

[8] J. Howe. The Rise of Crowdsourcing. *Wired Magazine*, 14(6):1–4, June 2006. http://www.wired.com/wired/archive/14.06/crowds.html.

[9] R. Hunicke. The Case for Dynamic Difficulty Adjustment in Games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '05, pages 429–433, 2005. http://doi.acm.org/10.1145/1178477.1178573.

[10] R. Lange, N. Cipriani, L. Geiger, M. Großmann, H. Weinschrott, A. Brodt, M. Wieland, S. Rizou, and K. Rothermel. Making the World Wide Space happen: New challenges for the Nexus context platform. In *Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*, PerCom '09, pages 1–4, 2009. http://dx.doi.org/10.1109/PERCOM.2009.4912782.

[11] C. Magerkurth, A. D. Cheok, R. L. Mandryk, and T. Nilsen. Pervasive Games: Bringing Computer Entertainment Back to the Real World. *Computers in Entertainment*, 3(3):1–19, July 2005. http://doi.acm.org/10.1145/1077246.1077257.

[12] C. Magerkurth, T. Engelke, and M. Memisoglu. Augmenting the Virtual Domain with Physical and Social Elements: Towards a Paradigm Shift in Computer Entertainment Technology. *Computers in Entertainment*, 2(4):1–20, Oct. 2004. http://doi.acm.org/10.1145/1037851.1037870.

[13] B. Mitschang, D. Nicklas, M. Grossmann, T. Schwarz, and N. Hönle. Federating Location-Based Data Services. In T. Härder and W. Lehner, editors, *Data Management in a Connected World: Essays Dedicated to Hartmut Wedekind on the Occasion of His 70th Birthday*, pages 17–35. Springer Berlin Heidelberg, 2005. https://doi.org/10.1007/11499923_2.

[14] D. Nicklas, M. Großmann, and T. Schwarz. NexusScout: An Advanced Location-Based Application on a Distributed, Open Mediation Platform. In *Proceedings of the 29th International Conference on Very Large Data Bases*, VLDB '03, pages 1089–1092, 2003. https://doi.org/10.1016/B978-012722442-8/50112-9.

[15] D. Nicklas, N. Hönle, M. Moltenbrey, and B. Mitschang. Design and Implementation Issues for Explorative Location-based Applications: The NexusRallye. In *Proceedings for the VI Brazilian Symposium on GeoInformatics*, GeoInfo '04, pages 167–181, 2004.

[16] D. Nicklas, C. Pfisterer, and B. Mitschang. Towards Location-based Games. In *Proceedings of the International Conference on Applications and Development of Computer Games in the 21st Century*, ADCOG '01, pages 61–67, 2001.

[17] W. Piekarski and B. Thomas. ARQuake: The Outdoor Augmented Reality Gaming System. *Communications of the ACM*, 45(1):36–38, Jan. 2002. http://doi.acm.org/10.1145/502269.502291.

[18] C. Schlieder, P. Kiefer, and S. Matyas. Geogames: Designing Location-Based Games from Classic Board Games. *IEEE Intelligent Systems*, 21(5):40–46, Oct. 2006. http://dx.doi.org/10.1109/MIS.2006.93.

[19] M. Schmatz, K. Henke, C. Türck, C. Mohr, and T. Sackmann. SYGo – a location-based game adapted from the board game Scotland Yard. In *INFORMATIK 2009: Tagungsband der 39. GI-Jahrestagung, 28.9. – 2.10.2009, Lübeck*, volume 154 of *LNI*, pages 1891–1901, 2009. http://cs.emis.de/LNI/Proceedings/Proceedings154/gi-proc-154-152.pdf.

[20] C. Shahabi, G. Barish, B. Ellenberger, M. R. Kolahdouzan, N. Jiang, S.-R. Nam, and R. Zimmermann. Immersidata Management: Challenges in Management of Data Generated within an Immersive Environment. In *Proceedings for the 5th Workshop on Multimedia Information Systems*, MIS '99, pages 107–114, 1999.

[21] C. Stach. Mobile ortsbasierte Browserspiele. Diplomarbeit, Universität Stuttgart : Sonderforschungsbereich SFB 627 (Nexus: Umgebungsmodelle für mobile kontextbezogene Systeme), Germany, Jan. 2009.

[22] C. Stach. Gamework – A Customizable Framework for Pervasive Games. In *Proceedings of the 2010 IADIS International Conference Game and Entertainment Technologies*, MCCSIS '10, pages 45–52, 2010.

[23] D. Thorleuchter, D. Van den Poel, and A. Prinzie. Extracting Consumers Needs for New Products – A Web Mining Approach. In *Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining*, WKDD '10, pages 440–443, 2010. http://dx.doi.org/10.1109/WKDD.2010.14.

[24] J.-P. Tutzschke and O. Zukunft. FRAP: A Framework for Pervasive Games. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '09, pages 133–142, 2009. http://doi.acm.org/10.1145/1570433.1570459.

[25] M. Weiser. The Computer for the 21st Century. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3):3–11, July 1999. http://doi.acm.org/10.1145/329124.329126.

[26] R. Wetzel, R. McCall, A.-K. Braun, and W. Broll. Guidelines for Designing Augmented Reality Games. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, Future Play '08, pages 173–180, 2008. http://doi.acm.org/10.1145/1496984.1497013.

[27] S. Wolff and B. Grüter. Context, emergent game play and the mobile gamer as producer. In *INFORMATIK 2008: Tagungsband der 38. GI-Jahrestagung, 8.9. – 13.9.2008, München*, volume 133 of *LNI*, pages 495–500, 2008. https://subs.emis.de/LNI/Proceedings/Proceedings133/gi-proc-133-087.pdf.