A CONTEXT-AWARE ARCHITECTURE FOR MOBILE KNOWLEDGE MANAGEMENT

Olaf Thiele, Hartwig Knapp and Martin Schader
University of Mannheim - Department of Information Systems
68131 Mannheim, Germany

Nicolas Prat
Essec Business School - Information and Decision Systems Department
Avenue Bernard Hirsch, B.P.50105 - 95021 Cergy cedex, France

Keywords: Mobility, knowledge management and context-aware computing.

Abstract: In many professional activities (e.g., medical diagnosis, construction, or sales), the ability to retrieve and store knowledge in a mobile situation is crucial. This need, together with the progress of mobile devices, has led to the emergence of mobile knowledge management. The mobile situation imposes specific constraints on traditional knowledge management activities (e.g., knowledge retrieval or presentation). Therefore, a key research question for mobile knowledge management is how context should be taken into account.
In this paper, we propose a context-aware knowledge management architecture for mobile environments. The key aspect is the "Contextualizer" middleware that takes care of knowledge storage and retrieval as well as contextually adapted knowledge presentation on mobile devices. In contrast to existing concepts, we suggest a broader approach, where the middleware serves as the mediator to different mobile devices. We implemented the architecture in a fitted use case. Our roadside assistance use case demonstrates the architecture's strength for presenting both text and graphics stemming from several knowledge sources.

1 INTRODUCTION

The issues of knowledge management and mobility have often been explored separately. In this paper, we combine the increased importance of intellectual capital in today's business life with the rapidly changing properties and requirements of mobile devices. The world of mobile consumer electronics has changed enormously over the last years. While cell phones, for example, display high-resolution video streams and location-based services offer numerous interesting opportunities, business life, on the other hand, evolves in an analogous way but much slower. Business applications are usually more specialized and therefore more complex and more costly than standardized consumer software. Some applications like e-mail messaging or word processors can be used for both private and business tasks. Still, this kind of application is mostly implemented in a proprietary way (e.g. Blackberry). Research in this area mainly addresses just one kind of mobile device or application. Most user interfaces are constructed especially for the targeted device (e.g. special PalmOS devices; also see (Bardram and Hansen, 2004)) or functionality is limited (e.g. WML; also see (Picco et al., 2000)).

Our generalized approach towards mobile knowledge management aims at both, contextually adapted queries to knowledge sources as well as contextually adapted presentation of knowledge on mobile devices. This includes modified communication with the mobile client according to network variables. While most research focuses on certain aspects of the problem, we present a complete architecture, which builds upon previous work in the fields of knowledge management and mobility. A middleware component called the Contextualizer serves as the main building block of the architecture. The Contextualizer is complemented by components that reside on the mobile device and in the back-end. This reduces the resources needed to include a new mobile device or knowledge source into the system. The main advantage is that modifications to the knowledge source (databases, unstructured knowledge sources, etc.) or to the mobile devices need only be applied to the Contextualizer instead of modifying all knowledge sources and devices.

In the next section, we present the mobile knowledge management environment and point out the technical prerequisites. In Section 3, we highlight our proposed architecture and explain its main com-

Thiele O., Knapp H., Schader M. and Prat N. (2006).
A CONTEXT-AWARE ARCHITECTURE FOR MOBILE KNOWLEDGE MANAGEMENT.
In *Proceedings of the International Conference on Wireless Information Networks and Systems*, pages 219-226

ponents. Section 4 shows parts of our prototypical implementation while we draw the outline of other works in Section 5. Finally, we conclude with a resume and an outlook.

2 MOBILE KNOWLEDGE MANAGEMENT ENVIRONMENT AND PROPERTIES

In this section, we highlight the main aspects of knowledge management and mobility relevant to our work.

2.1 Knowledge Management

Knowledge management is often described as a system of activities to permit the utilization of organizational knowledge by the members of that organization (Hannig, 2002). Possible knowledge management techniques and approaches usually fall into one of the five categories: expert finder, virtual teamwork, lessons learned databases, case-based reasoning, or virtual/augmented reality (Derballa and Pousttchi, 2004). The first approach is often used to identify and contact a relevant expert for a specific problem. The second one refers to work that is conducted by geographically distributed coworkers. The next approach refers to the concept of maintaining positive and negative experiences in a problem-solving process (see also (Probst et al., 1997)). The fourth looks for similar issues in the past and retrieves these solutions. The last approach assists the user by means of virtual reality. Due to the fact that knowledge management is a vast field of research, further definitions can be found in the referred work or in the books by Davenport and Prusak (Davenport et al., 1997) or Nonaka and Takeuchi (Nonaka and Takeuchi, 1995).

In this paper, we focus on four general knowledge management processes that are relevant to our more technical view. On the one hand, these are knowledge presentation and acquisition, which evolve around human interaction with the system. Knowledge presentation comprises both displaying data on the device as well as interacting with it. If this interaction leads to some sort of knowledge storage or if the user deliberately enters information through a form or just as plain text, we categorize these activities as belonging to knowledge acquisition. On the other hand, knowledge retrieval and storage are relevant to our work. We use these terms analogous to their technical counterparts.

2.2 Mobility Aspects

Classical knowledge management applications were programmed for desktop use. With the introduction of mobility, sophisticated adaptation to the mobile context becomes more important. In general, the unique properties of mobile devices are a major influencing factor in the proposed system. With a growing number of different device types, the heterogeneity grows enormously as most devices differ in their capabilities from PCs (e.g. screen size) and from other mobile device types (e.g. cell phones and PDAs). Critical factors include the display size and resolution, the memory size and accessibility, the processing power, connection protocols like GPRS, UMTS, or EDGE, interaction techniques like the stylus and supported standardized programming interfaces like Java or C (see (Lum and Lau, 2002) and more recently (Adipat and Zhang, 2005)). Key properties are the ubiquity and the ability to adapt to the user's context, given the current position and other preferences (e.g. user profiles).

To sum up, benefits of joining the fields of knowledge management and mobility are "anytime, anywhere information access" (Grimm et al., 2005), mobile-added values like ubiquity or context-sensitivity (Derballa and Pousttchi, 2004), and automatic context incorporation (e.g., knowledge is accessed in an adapted way and context variables such as location are stored automatically).

2.3 Requirements

While the limitations of technology supporting knowledge management activities have already been discussed elsewhere (see for example (Derballa and Pousttchi, 2004)), the requirements for mobile devices and their communication partners are a field, which deserves further description. As mentioned above, mobile devices are limited in several key features that are normally fulfilled by common PCs or workstations (e.g. screen size). Other mobile restrictions include network issues like bandwidth or communication protocols. Because of this fact, we introduced an intelligent middleware - the Contextualizer -, which serves the function to convert a data stream from the server side into a client-readable format and back. Therefore, the Contextualizer has a rough understanding of device types with their properties and clients transfer their specific needs (e.g., exact device identification, resolution, display size, colors, free memory, number of presentable letters in a row, etc.) during the communication's initialization phase. The middleware, which will be presented in more detailed in Section 3 receives a response from one or more of the databases and converts it into the client's contextually adapted format. In the other direction, the middleware has to translate a client's request for knowledge into a database-understandable format. This is also described in Section 3.

3 MOBILE KNOWLEDGE MANAGEMENT ARCHITECTURE

In this section, we introduce our mobile knowledge management architecture and its underlying components. Subsequently, we explain how the context mapping works and afterwards, we present the Contextualizer middleware in more detail in Section **??**.

3.1 Introduction

Our proposed architecture for mobile knowledge management, shown in figure 1 on the next page, is divided into four layers. The first layer contains the mobile users or mobile agents. This human agent interacts with the system to access knowledge. The human agent then utilizes this knowledge, transfers it to others or enters some new knowledge back into the system. The knowledge management activities, knowledge presentation and acquisition are the interface between layers one and two. The second layer consists of Java-enabled mobile devices like mobile phones, PDAs, or laptop computers. These three device types are the most common today and usually all devices of a certain type share similar properties (e.g., reduced set of keys on cell phones compared to a complete keyboard on notebook computers). We programmed client-side pieces of software for each device type. They serve as small agents on the device, which automatically determine device properties such as screen resolution, color table, or bandwidth. If the device supports determination of the location (by GPS), this data is collected as well and sent to the Contextualizer, which resides on the third layer. The implemented middleware serves as the connecting link between the adjoining layers - it is the main focus of our work. It translates the data flow coming from the mobile devices to a database query and the response is then adapted according to the mobile context and sent back to the device. The interfacing activities between the third and fourth layer are knowledge storage and retrieval. The fourth layer contains the knowledge source which is built up of different database types. We chose the three most popular types: relational (SQL), object-oriented (OODB), and XML databases. Due to the fact that most of the intelligence and complexity of our application reside on the third layer, we explain this component more detailed in the next subsection.

3.2 Outline of the Contextualizer

The Contextualizer is a middleware component, which serves as a mediator between the human agent and existing knowledge bases. It takes care of communication with the knowledge base (knowledge storage and retrieval) as well as linking the server and client side according to the context (knowledge presentation and acquisition). In general terms, if the Contextualizer receives an XML query from a mobile device, it parses the request and analyzes the context according to the following four context elements: user dependency, technological environment, situation, and task. Subsequently, all relevant databases are queried, the results are transformed according to the four context elements, and, finally, the result is sent back to the client application. All four context elements are described thoroughly in the following subsections. Table 1 gives on overview of typical context elements.

Table 1: Overview of Context Elements by Category.

Context Elements	**Subelements**
User Dependency	Role in the Organization (function, hierarchical position), User Preferences (technical, application specific)
Technological Environment	Handheld Device (screen size, processing power, available memory), Network (bandwidth, current network load)
Situational Elements	Time (time and time zone), Location (automatically via GPS or manually by zip code)
Task-Specific Elements	Heavily dependent on the task and therefore hard to pre-determine

Not all context elements need to be considered for all knowledge management activities. The role of a user within the organization, for example, has significant impact on what type of knowledge can be accessed (retrieval of knowledge) but is independent of the knowledge presentation. Furthermore, context elements may change during the execution of a task and therefore need special attention. Again, the role of the user will usually remain stable but the location or

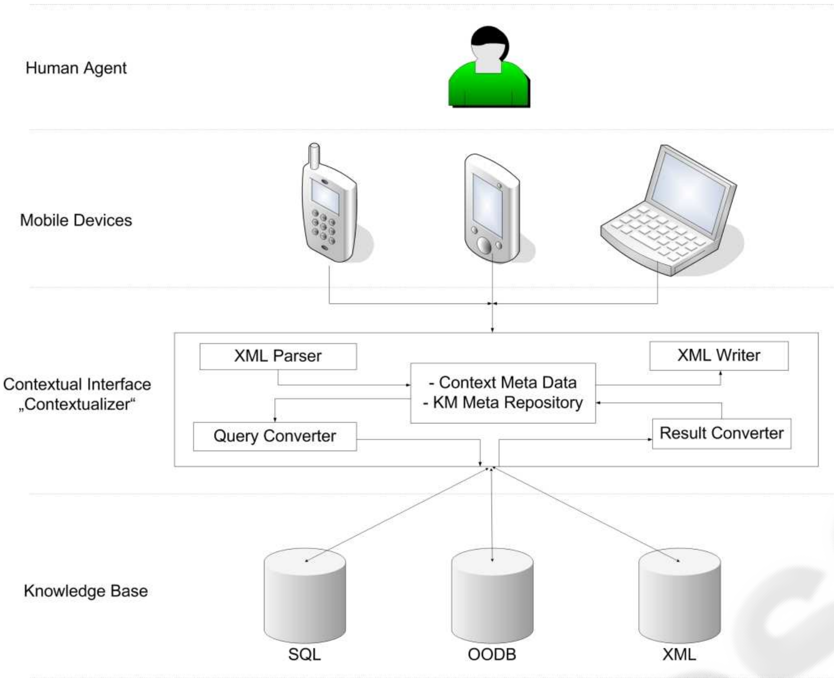

Figure 1: Mobile Knowledge Management Architecture.

network load may well change. Thus, the distinction between static and dynamic context elements is important to our architecture. Finally, some context elements need to be determined either on the client or the server side. While the user's role has to be retrieved on the server side to prevent misuse, the location (e.g. by GPS use) is usually retrieved on the mobile device. Some elements require server and client side effort (e.g. measuring network load).

We carefully designed this integrated architecture to meet all our needs. Lei and Georganas distinguish three different areas (client, proxy, or server), where an adaptation could be placed in a context-aware architecture (Lei and Georganas, 2001). Several arguments support our approach to include some logic on the client and the back-end with a sophisticated mediator like the Contextualizer in between. Thus, we combine all three approaches with an emphasis on the proxy. For one, we need to collect as much context parameters as possible. These are only available on the client. Therefore, we need a minimum application on the client if not a more complex agent software. Furthermore, a unified language for accessing knowledge sources allows for selecting chunks of knowledge as they are needed for the actual context. Driving instructions could be textual, verbal, sketch-based, or full-size maps. Performing all transformations on the client, for example, would exceed the processing power of most cell phones, whereas an implementation centered around only one knowledge source would hinder spreading the architecture to different tasks.

3.3 User Dependency

User dependent elements form the first type of context elements. They include technical preferences as well as the agent's role within the organization. Most technical preferences are represented through the user's profile. A typical user profile includes data on preferred color themes as well as favored font sizes. Profile information is stored on the client and is transferred to the Contextualizer in the initialization phase. In addition to the profile, the role of the user within the organization needs to be taken into account. On the one hand, not all members of an organization work with all available knowledge. The users' role determines what knowledge they are allowed to and need to access. Authentication and authorization is incorporated within the startup phase. For example, all available data would lead to an information overflow for executive members of the organization. They need an abstract overview of the accessible knowledge.

3.4 Technological Environment

The technical environment comprises elements of the device and the network. The device plays a major role for the Contextualizer. Variables like the screen size, processing power, available colors, or programming interfaces are key determinants for putting the knowledge into context. Two major types of applications need to be considered: text and graphical knowledge representations. Text can be displayed in many ways. Font size, text formatting and navigational elements within texts (e.g. hyperlinks) as well as ways for changing screens (e.g. scrolling) can be varied between mobile devices depending on their capabilities. As for graphical knowledge representation, display facilities vary a lot between devices. While some devices like laptops or PDAs offer powerful standardized programming interfaces, smaller devices like cell phones or smartphones usually have limited custom interfaces.

The network connection is another important influencing factor. The available bandwidth determines the transferable data size. Pictures, for example, need to be compressed or simplified before transfer. Furthermore, interaction techniques usually rely on broadband connections. But the available bandwidth may differ due to increased network traffic. In this case, the bandwidth parameters need to be dynamically adjusted. Finally, the costs for network traffic might be a limiting factor. Some users might prefer a low-end knowledge representation over a more expensive one.

3.5 Situational Elements

Situational elements are those connected with time and location. The time variable includes the actual time as well as the corresponding time zone. Depending on the time of day, knowledge queries might return different results. Moreover, the time zone is important when retrieving time-critical information from databases in far away countries.

The other important situational element is the location. The position of the user on the globe is one of the most prominent contextual elements and is used in many mobile applications. The location can be determined automatically (via GPS) or manually. The position of the user is especially interesting for graphical knowledge representation in maps or sketches. The dynamic nature of this element is also important in navigational tasks. Both the Contextualizer and client need to keep up with a driving vehicle to deliver timely knowledge.

3.6 Task-Specific Elements

All other context element implementations, as described above, are universal and can be used for most task types, but elements specific to the task need to be adjusted for each task type. To ease the programming effort needed for implementing new tasks, we developed a semantic XML description format (similar to Topic Maps), which can be used to semantically express what knowledge should be retrieved from which database. The semantic description is situated in the part named KM meta repository and further includes information on interrelated topics. Basically, the client sends a request to the server, which determines the resources to access by using these descriptions. Within the Contextualizer, all data is represented in a uniform XML format, regardless of the data source (relational, object-oriented, or native XML).

4 PROTOTYPE IMPLEMENTATION

We implemented a sample use case prototype to validate our mobile knowledge management architecture and to illustrate its features. First, a technical overview of the implementation is given. Subsequently, we then present the basic idea of our use case and we introduce the implemented clients as well as the corresponding server side.

4.1 Overall Implementation

Generally, all applications (client and server side) are programmed in Java. The many advantages of Java include the wide variety of available client programming interfaces as well as a free choice of the server side operating system. The often mentioned performance drawback is not as significant for our applications but might play a role when using more or larger images. The client-side application has been programmed using the J2ME programming interfaces. The widely available components include APIs for handling images as well as storing information on the device for caching purposes. The server side and especially the Contextualizer were programmed using J2EE. The Java 2 Enterprise Edition APIs offer the capabilities for handling multiple connections with clients as well as databases. The Contextualizer is therefore scalable and performant.

4.2 Roadside Assistance Use Case

The main idea behind the use case is helping both professionals and car drivers in assessing problems related to a car that broke down. The look and feel of the application is depicted in figure 2.

The welcome screen is shown in the left half and a typical GUI element in the right half of the screenshot. Due to the high screen resolution and processing power of modern cell phones (here the Nokia6230i)

Figure 2: Screenshots from the application GUI.

we present these screenshots. The adaptations for PDAs and laptop computers look similar to those known from analogous applications (e.g. (Yue et al., 2005)). The main idea of the use case led to two basic scenarios: In the first one, a professional mechanic drives to a remote location to solve a car breakdown. In order to find the location, he uses a map given by our application. Our implementation is shown in figure 3.

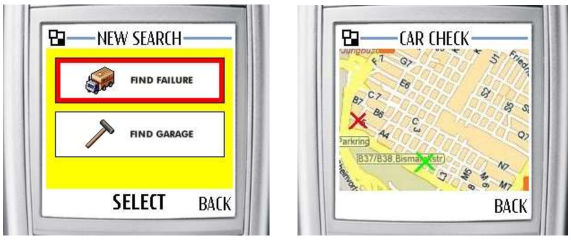

Figure 3: Illustrations from Scenario 1.

Both, the actual position and the location of the accident or breakdown are shown on the map. The cursor keys may be used to navigate on the map. Once the mechanic arrives at the destination, the application helps him to spot the failure by querying several databases (e.g. the manufacturer's one and reports by colleagues). The mechanic as well as the car driver might use either a cell phone, a PDA, or a laptop to access the knowledge management system. The second scenario includes the driver of the car, who wants to find a nearby garage while the car is still on the road or he intends to mark his position on a map. Furthermore, the driver might describe his situation using the client, which sends the request to the Contextualizer. There, the request is parsed and sent to the databases according to the predefined semantic map. The results are then transferred back to the mobile client. A possible request to the system is shown in figure 4.

First, the user describes the problem in short keywords. Afterwards, several databases are questioned and the results are then transferred back to the device. In both scenarios, some knowledge on possible solutions is transferred back into the system. The outside temperature, for was helpful in solving the problem

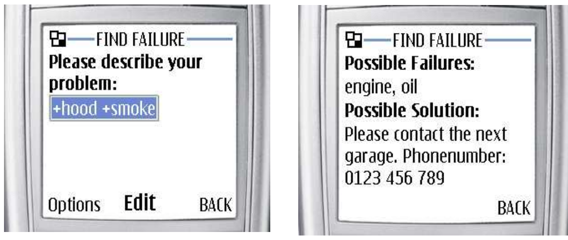

Figure 4: Screenshots of Scenario 2.

might ease future use of the system.

4.3 Contextual Adaptation

The implemented use case adapts knowledge according to all four context elements. User dependent elements include a user profile and organizational role. The user is able to choose a certain skin and save favored top menu items. The organizational role determines, which parts of the system may be accessed by the user. The mechanic has more access to different knowledge sources than a regular driver (e.g., excerpts from a professional database are not easily understandable). Technical context elements in the use case include the color, interaction style, screen size, and bandwidth. The images sent to the device are adapted according to the given color table. Certain keywords in responses are highlighted if colors are supported. The graphics are sent in total if the device supports interaction types like the stylus or mouse movements. Only parts of a picture are sent to a cell phone due to the limited interaction capabilities and the smaller screen. The screen size and bandwidth determine how much of the map is sent at a time. Devices with a slow network connection receive smaller pieces of the map. The bandwidth of a regular cell phone posed a problematic situation when moving fast while viewing high resolution images. Situational elements include the time and place. Problems differ at night, especially in winter time. Furthermore, a much smaller set of garages is open after hours. The location plays a major role in this scenario and is monitored constantly. The task specific elements were mainly described above and included map navigation and problem solving.

5 RELATED WORK

While knowledge management and mobility are well-established research fields, the combination of both raises questions and uncertainties. On the more technical side, for example, we face the problem that connections between mobile devices and databases are not dependable; a connection and session management has to be introduced to guarantee a complete and timely data exchange. Some (Holliday et al., 2002) attend to the question how a database query might be answered completely and discuss several disconnection modes while others (see for example (Chan and Roddick, 2003)) dwell on weak or partial connection modes. Further work revolves around the question how the amounts of data sent back and forth can be minimized in order to avoid large data transfers in non-broadband networks (see (Chang et al., 2004) or (Lindemann and Waldhorst, 2004)).

Work similar to our approach was carried out by Fagrell et al. as early as 2000 (Fagrell et al., 2000). As mobile devices had much less processing power back then, context elements such as user dependency or automatic location detection were not considered. The work of Wei and Prehofer focuses more on distributed context repositories, which are queried for decision making (Wei and Prehofer, 2003). Derballa and Pousttchi present the idea of mobile-added values (Derballa and Pousttchi, 2004). Their approach is rather abstract and they do not offer a technical implementation roadmap. Focusing on similar issues are Grimm et al., who are working on the MUMMY project (see (Grimm et al., 2005) or (Grimm et al., 2002)). They developed applications for several tasks (e.g. mobile facility management, trade fair information). Finally, Adipat and Zhang develop an adaptable framework for mobile knowledge management (Adipat and Zhang, 2005). Their main focus lies on web content adaptation according to user preferences.

Further research on contextual awareness is carried out in the field of human computer interaction. Dourish (Dourish, 2004) gives a comprehensive overview of the field's history and current research. An approach towards mobile knowledge management from that research field is presented by Shen (Shen, 2003) or Bardram and Hansen (Bardram and Hansen, 2004). Finally, work on mobile knowledge management exists that centers around peer to peer exchange of knowledge. Schwotzer and Geihs present an architecture for topic map exchange (Schwotzer and Geihs, 2002).

6 CONCLUSION

Mobile knowledge management is a key application area for mobile computing in general and context-aware computing in particular. In this paper, we have presented a context-aware mobile knowledge management architecture. The key component of the architecture is the Contextualizer. This middleware implements the knowledge management activities (e.g., knowledge storage, retrieval, presentation, and acquisition) by taking into account the context elements, namely the user, the technical environment, the situation (i.e. time and place) and the specific task at hand. Moreover, in contrast to existing approaches, the Contextualizer comprises a meta-repository, which decides dynamically where the requested knowledge resides. We have implemented a prototype of our mobile knowledge management architecture and applied it to a scenario.

Our prototype architecture needs to be extended in order to take into account all mobile knowledge management activities (e.g. better knowledge acquisition techniques like forms and recording audio). The scalability of the architecture requires further testing (a wider range of devices and media types) and applications to real-life situations (e.g. handheld audits). We are currently working on these issues.

REFERENCES

Adipat, B. and Zhang, D. (2005). Developing adaptive and personalized mobile applications: A framework and design issues. In *Proceedings of the Eleventh Americas Conference on Information Systems (AMCIS 2005)*. Omaha, Nebraska.

Bardram, J. E. and Hansen, T. R. (2004). The aware architecture: supporting context-mediated social awareness in mobile cooperation. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 192–201, New York, NY, USA. ACM Press.

Chan, D. and Roddick, J. (2003). Context-sensitive mobile database summarisation. In *ACSC 2003. Twenty-Sixth Australasian Computer Science Conference*.

Chang, T.-Y., Velayutham, A., and Sivakumar, R. (2004). Mimic: raw activity shipping for file synchronization in mobile file systems. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 165–176, New York, NY, USA. ACM Press.

Davenport, T. H., Prusak, L., and Prusak, L. (1997). *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston, MA, USA.

Derballa, V. and Pousttchi, K. (2004). Extending knowledge management to mobile workplaces. In *ICEC 2004. Sixth International Conference on Electronic Commerce*, pages 583– 590.

Dourish, P. (2004). What we talk about when we talk about context. *Personal Ubiquitous Comput.*, 8(1):19–30.

Fagrell, H., Forsberg, K., and Sanneblad, J. (2000). Fieldwise: A mobile knowledge management architecture. In *CSCW 2000*, pages 211– 220.

Grimm, M., Tazari, M.-R., and Balfanz, D. (2002). *Lecture Notes in Computer Science 2569*, chapter Towards a Framework for Mobile Knowledge Management, pages 326–338. Springer.

Grimm, M., Tazari, M.-R., and Balfanz, D. (2005). A reference model for mobile knowledge management. In *Proceedings of I-KNOW 05*. Graz, Austria.

Hannig, U. (2002). *Knowledge Management and Business Intelligence*, chapter Zwei Welten wachsen zusammen. Springer, Berlin.

Holliday, J., Agrawal, D., and El Abbadi, A. (2002). Disconnection modes for mobile devices. In *Wireless Networks 8*, pages 391– 402.

Lei, Z. and Georganas, N. (2001). Context-based media adaptation in pervasive computing. In *Canadian Conference on Electrical and Computer Engineering, 2001*, pages 913–918 vol.2. IEEE Press.

Lindemann, C. and Waldhorst, O. P. (2004). Exploiting epidemic data dissemination for consistent lookup operations in mobile applications. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(3):44–56.

Lum, W. Y. and Lau, F. C. M. (2002). A context-aware decision engine for content adaptation. *IEEE Pervasive Computing*, 1(3):41–49.

Nonaka, I. and Takeuchi, H. (1995). *The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press.

Picco, G. P., Murphy, A. L., and Roman, G.-C. (2000). Developing mobile computing applications with lime. In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 766–769, New York, NY, USA. ACM Press.

Probst, G., Raub, S., and Romhardt, K. (1997). *Wissen managen*, chapter Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Gabler, Wiesbaden.

Schwotzer, T. and Geihs, K. (2002). Shark - a system for management, synchronization and exchange of knowledge in mobile user groups. In *2nd International Conference on Knowledge Management (I-KNOW '02)*, pages 149–155. Graz, Austria.

Shen, J. (2003). Utilizing mobile devices to capture case stories for knowledge management. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 688–689, New York, NY, USA. ACM Press.

Wei, Q. and Prehofer, C. (2003). Context management in mobile environments. In *Proceedings of ANwire Workshop, Paris*.

Yue, W., Mu, S., Wang, H., and Wang, G. (2005). Tgh: a case study of designing natural interaction for mobile guide systems. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 199–206, New York, NY, USA. ACM Press.