

# Generalized Notification Services: A Simple but Versatile Paradigm for the Implementation of Mobile Data Services

Michael Decker

Institute AIFB, University of Karlsruhe  
Englerstraße 11, 76 128 Karlsruhe, Germany

e-mail: [decker@aifb.uni-karlsruhe.de](mailto:decker@aifb.uni-karlsruhe.de), web page: <http://www.aifb.uni-karlsruhe.de>

**Abstract:** *The article at hand discusses a simple but although versatile and powerful paradigm for the implementation of mobile services called generalized notification services (NS). Based on a short discussion of specific challenges for mobile services we will argue why NS are especially suited for applications with mobile and wireless terminals. We also give several examples to demonstrate that while NS are a simple paradigm there are many useful mobile services which could be considered as NS. At the end we sketch a protocol for the realization of NS which ensures certain privacy aspects.*

**Keywords:** Mobile and wireless computing/services, context-awareness, Software Engineering

## 1. Introduction

According to our definition mobile data services (MDS) are well defined sets of functionalities offered by one or more stationary computers (servers or back end system operated by service provider) to a mobile terminal (MT) as client, whereas at least the first part of the communication between client and server is realized using wireless data communication. MT are *handheld* computers like cellular phones, personal digital assistants (PDA) or smartphones<sup>1</sup>.

There is a plethora of research papers about platforms or frameworks for the implementation of MDS; many of them concentrate on how to deal with context information or architectural aspects. But often a precise description of what technical paradigm for MDS (Which interaction patterns between client and back-end may occur? What kind of messages are allowed for what part of the communication?) is supported by these frameworks is missing. One common paradigm for MDS is the pull-mode delivery of documents in markup languages (e.g. cHTML or WML) like known from the WWW. Another paradigm for MDS is the delivery of SMS whereas the user controls the service by sending simple commands via SMS to dedicated phone numbers.

In the paper at hand we discuss a paradigm for the implementation of MDS called generalized notification services (NS). Munson & Gupta (2002) mentioned the idea of generalizing notification services to different areas of application but they considered only location as context information, provided no formalization and concentrated on the problem how to provide location information for a large number of users. The basic idea behind NS is that due to the limitations of MT and the typical usage scenarios (user is “on the move”) a user cannot be expected to do much more than reading notification messages sent to him by the service based upon an initial configuration of the service. To increase the usability of the service our definition of NS also regards personal as well public context information (e.g. location of user, time, weather, profile information, available bandwidth). Although at the first glance NS seem to be limited to simple alert services, we will give examples of MDS which can be modelled as NS from different application areas. The

---

<sup>1</sup> laptop computers or notebooks are *not* mobile in this sense because such devices are used like stationary computers at different places

semi-formal description of NS given is a necessary step for the development of a software framework as mentioned in the last chapter.

The remainder of this article is as follows: in the next section we discuss two important concepts for our description of NS, namely context-awareness and push-messages. In section 3 we describe NS and argue why NS are appropriate with regard to the specific limitations of MDS. To show that despite their simplicity NS are versatile we sketch several examples of MDS found in literature or practice which could be implemented as NS in section 4. The 5<sup>th</sup> section is about a protocol for the implementation of NS with special consideration of privacy aspects. In the last section a summary and an outlook towards future work are given.

## **2. Discussion of basic concepts**

### ***2.1 Context and Context-Awareness of MDS***

The concept of context is essential in mobile computing and discussed in many publications (Chen & Kotz, 2000). According to our understanding context is information used deliberately to support a user when interacting with his MT and available at runtime of a service or application in explicit form. For MDS context-awareness is essential because MT have a poor user interface (small display, no full keyboard, see also section 3.2) so the user doesn't want to enter a lot of data. The most popular example for context-aware services in mobile computing are the so called "location based services", where the current position of the user is employed to provide the user with useful information according to his surrounding so he hasn't to browse for that information manually. Other examples for context information are the profile of the user (gender, age, fields of interest), time, weather or available resources. We distinguish between user-related and thus privacy critical context information (personal context, e.g. location, profile) and non-user-related context (public context, e.g. time, weather).

### ***2.2 Push-Messages***

There are two basic modes for communication using digital devices: in pull-mode the user has to send an explicit request each time he wants to get information. However in push-mode the user receives information without having directly requested it — but hopefully he indirectly requested the information by giving his permission some time before. An example would be if a user enters his mobile phone number in the webpage of a merchant to receive notification about bargain offers via SMS. If one week after this the user receives the first SMS he won't perceive the message as being directly requested. Push messages have a great potential of being annoying especially if received on a MT. But due to the ubiquitous character of MT as personal communication devices using push messages many useful services can be realized, especially if time-critical information is involved; this is why NS are designed to dispatch notifications in push mode.

Push- and pull-mode can also be defined from a technically point of view: Push-communication is given if the component of a system that produces information can initiate the transmission of that information; if the information-producer has to wait till the information-consumers requests that information we have pull mode. Using technical pull communication we could realize push-communication at user level if the client application performs cyclic pulls for that information ("polling").

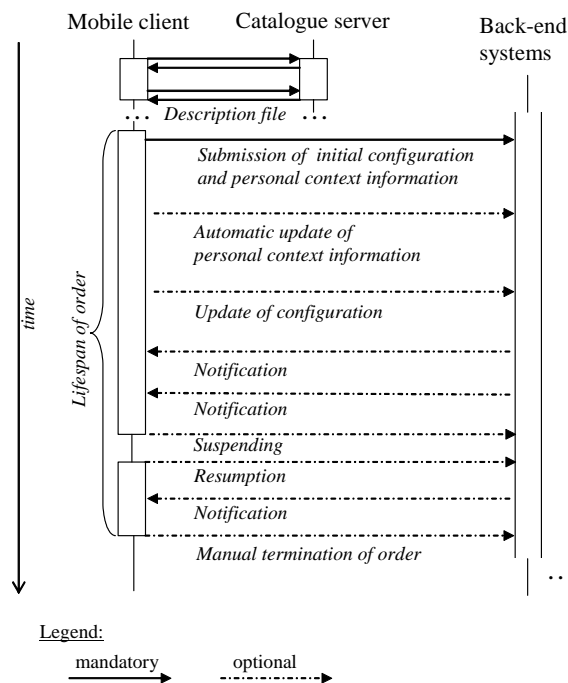
## **3. Notification Services**

### ***3.1 Description***

The basic principle of Notification Services (NS) is that the service provider (SP) sends notification messages in push-mode to the user's MT (e.g. SMS/MMS-messages, e-Mail, WAP-push or using a proprietary channel using an IP-connection) based on manually provided configuration and personal as well public context information. The notifications are presented to the user but don't affect the local

configuration; notifications might lead to further interaction (e.g. if the contain a link to a WAP-page) but this is beyond the scope of NS.

To create an instance of a NS the mobile client application needs a machine-readable description file of the respective NS type; this description file is preferably formulated using a special XML-grammar and provided by a catalogue server. The description file defines which configuration parameters have to be entered by the user manually<sup>2</sup> and which personal context parameters (e.g. location of terminal, profile information about the user, battery level) have to be provided by the client application respective the mobile network operator; the public keys of the service providers mentioned in section 4 are also included in the service description file. The required data is submitted to a stationary server of the respective SP who will create an instance of that NS type; a configured instance of a NS is denoted as “order”. Depending on the type of the order the user may suspend the order or terminate it (there are examples where this is not sensible), it might also be possible to update the configuration an arbitrary number of times. If the mobile client application detects changes of relevant personal context parameters (e.g. location of user) an update of the order will be submitted to the SP. The user can terminate an order manually.



**Figure 1: Generic sequence-diagram of a NS session**

To illustrate how NS work we take the example of location based advertising (Kölmel & Alexakis, 2002): Based on the user’s profile, his shopping list (=manual configuration) and his current location (=personal context) the service provides advertising information as the user moves through a city. If he approaches a participating store which offers a product he has on his list he will receive a corresponding advertisement message on his MT. In Figure 1 we depict the messages exchanged between the different components during the lifespan of NS in form of a UML-sequence-diagram (the sequence diagram is a generic one, not all messages will occur for all instances of NS; the only mandatory message is the submission of the initial configuration and personal context information):

1. The catalogue server stores description files for different types of NS. In certain time interval (e.g. once a week) the client application polls for new description files and stores them for later usage.

<sup>2</sup> the description of what data has to be configure manually should be defined using a subset of the Xform-language (Dubinko, 2003) in the service description file

2. When the user wants to create an instance of this service he has to configure it: the client application will present a form where he can specify what kind of products/services he is interested in (shoes, restaurants, entertainment events, tourist attractions, etc.), price-level (bargain, medium, high-quality) and maximum number of advertisement messages allowed each day; which attributes have to be defined is specified in the description file. The description file also says that the client application has to provide the personal context parameters “gender” and “age” (so an elderly user doesn’t receive advertising from a boutique for teenagers) and the current location with a precision of 200 meters. The client application will submit the configuration to the SP’s server.
3. As the user moves through the city the client application sends updates of the current position retrieved using a GPS-receiver; if the personal context parameter “battery level” falls below a critical threshold the order is suspended automatically to save energy for more important applications.
4. If the SP determines that the user is approaching a store with offers matching his shopping list and profile he sends the notification messages with the advertisement message.
5. The user might suspend the advertising service and resume it later.
6. In the example the user manually terminates the services; the service configuration might also contain an expiry date for an order.

A m-advertising service like this could also evaluate public context information like the time (don’t send offers from stores that are already closed) and weather (don’t send offers concerning “good-weather-products” like sun glasses when it is raining all day long).

### **3.2 Appropriateness of NS for services for MT**

To argue about the appropriateness of NS for scenarios with MT we have to highlight the limitations which arise from the mobility or portability of terminals and wireless data communication (see also Forman & Zahorjan, 1994):

Mobility of terminals leads to devices of small size so the display has limited quality and there are only rudimentary means for data input. Thus it is very cumbersome to interact with the MT and when designing a MDS one cannot expect the user to enter a lot of data or to browse through large sets of information. This leads to the postulation of “one touch” interaction (Zobel, 2001): a MDS or application should provide useful information without requiring long user sessions, in the ideal case the user turns on his device and sees the information helpful for him in his current situation. NS meet this postulation: in most cases it is sufficient to perform an initial manual configuration of the service and read the push messages sent by the service afterwards.

The small size of mobile devices leads also to a limited energy supply; while we saw an impressive growth of available CPU-power and memory space the limited capacity of accumulators seems to be a bottle neck for the next years. Since intensive computations lead to heavy power consumption and therefore services shouldn’t require the performance of such computations on MT. Again NS meet this postulation since the actual business logic of the services is executed on the stationary backend systems.

Wireless data communication is relative slow, unreliable, expensive and has high latency when compared to wired data communication. Therefore a MDS shouldn’t need to send or receive large amounts of data and temporary connection dropouts should be hidden from the end user as far as possible. When using “mobile web pages” connection dropouts lead to times of waiting for the user (long delays till the next page is loaded) or even the necessity to manually resubmit requests. Since the configuration of NS is based on a description file there is no need for network interaction during the manual configuration of the order; submission of configurations or updated context information can be postponed and retried later without impairing the user experience. Also NS do not require large amounts of data to be sent as long as the notification messages don’t have to contain multimedia elements (this might be necessary in the m-advertising example).

The usage of MT involves privacy concerns: the wireless data transmission could be eavesdropped or someone could track the user’s position. In section 4 we therefore show that it is simple to state a protocol

which guarantees that the mobile network operator and service provider see only data in clear text that is actually required by the respective party.

<b>Name of example</b>	<b>Configuration</b>	<b>Context information</b>	<b>Notifications</b>
Location Based Advertising (Kölmel & Alexakis, 2002)	Wish list, price level, maximum number of notifications/day	Location, battery level, profile; public: time, weather	Advertisements with offers
Virtual Memo/Graffiti (Brown, 1995)	Text of message to deposit, expiry date	Location	Messages deposited by other users near current location
m-Payment (Pousttchi, 2004)	Amount to pay, code of recipient	SIM-card authentication	Confirmation of payment
Weather Alert (Fraunhofer, 2006)	Types of relevant weather situations	Location	Warning about menacing weather
Tourist Guide (Hinze & Voisard, 2003; Sampat et al., 2005)	—	Location, profile	Explanations concerning sight in surrounding
POI-Finder (D'Roza & Bilchev, 2003)	Category of point-of-interest (POI), e.g. pharmacy, restaurant, ...	Location; public: time, weather	Instructions how to drive/walk
Community support (Burak & Sharon, 2004)	Nickname, identifier and pass phrase for group	Location	Name of friend in nearer surrounding
Blind Date Finder	—	Location, Profile	Notification that potential partner is around
m-Ticketing (Hussin et al., 2005)	Specification of ticket (public transport: destination; theatre: category of seat)	SIM-card authentication, possibly location	m-ticket with digital signature
Alert-Services (Adya et al., 2002)	Relevant events (e.g. identifier and threshold value of stock quote), pass phrase if confidential information	—	Notification with time-critical information
Two-Factor-Authentication for online-banking (Wüest, 2005)	Challenge code	SIM-card authentication	Response code, details of transaction
Support for acquisitions of purchase orders at customer's site (Villanen et al., 2004)	Details about purchase order (customer, items, amount)	—	Confirmation

**Table 1: Examples for MDS considered as NS**

### 3.3 Examples of NS

In this subsection we give examples for NS from different fields of applications. The MDS mentioned here are not novel services but they weren't considered as one common class of services; also implementations concentrated on providing one distinct solution instead of a generalized solution that can be easily adapted to obtain MDS from different fields of application. All services discussed have "mobile value", off course:

they satisfy needs typically arising when the user is “on the move” like (1) being informed about time-critical events, (2) killing time or (3) doing something useful during waiting times on journeys or (4) getting information needed in mobile situations (Anckar & D’Incau, 2002). There are two further settings with mobile value: (5) support of mobile workers and (6) MT as authentication credential. For each example service we state the required configuration and context parameters (personal context if not stated otherwise) along with the meaning of the notifications (see Table 1).

### **Virtual Memo/Graffiti**

Using this service mobile users can deposit virtual memos/graffiti at their current position; if another user approaches that location he gets a notification with the message of the memo (e.g. “visit the museum in this street, it’s lovely”). Mobile value: (4) or if the messages are of less serious character (2).

### **M-Payment**

M-Payment means payment procedures where at least the payer uses a MT to execute a payment. To pay at point-of-sale (checkout in supermarket, vending machine) the user enters the amount to pay and the recipient code provided by the vendor. The SIM-card authentication is used as credential for the identity of the payer. Mobile value: (6)

### **Weather alert**

Weather alert is a location based alert service that warns people when a potentially menacing weather is about to occur at their location. The configuration is to specify what kind of weather the user (home owner, motorist, outdoor sportsman) is interested in, e.g. thunderstorms, hail or temperatures beyond the freezing point. Mobile value: (1) and (4)

### **Tourist guide**

A MDS for visitors to new places (e.g. cities, parks, exhibitions, museums): when the user approaches certain locations he receives a notification message with information about that location, e.g. name and artist of a monument. Mobile value: (4)

### **Community support (“Friend Finder”)**

A group of friends chooses an identifier (string) and pass phrase; if members of that group approach each other they receive notifications with the name of the counterpart. A similar service called “FriendZone” was developed by SwissTelecom. If instead of an identifier the profile is used for matching we obtain a “blind-date finder”. Mobile value: (2) and (4)

### **m-Ticketing**

m-Tickets are electronic documents sent to MT to certify that the owner has the right claim a certain service (e.g. permission to entertainment event, public transport). The configuration is required to specify the type of service (what category of seat in theatre, what kind of ticket for public transport); for some scenarios location-awareness could be employed (e.g. location of start station for public transport ticket). Mobile value: (6)

### **Alert-services**

Alert-services send notifications to users when a critical event was detected (e.g. stock quote below a certain level, etc). Meanwhile there are even mobile alert-services for critical values detected by business intelligence systems (e.g. alert if stock of important assembly part went below minimum inventory). Mobile value: (1) and when used for professional purposes (5).

### **Two-Factor-Authentication**

An approach to hinder “phishing”-attacks with regard to online-banking is two-factor-authentication using the MT as additional credential. When the user has entered the necessary details for a bank transfer

(amount, bank code of recipient, secret transaction number (TAN)) on the website of his bank the website displays a “challenges code”. This challenge code is the configuration input for a NS, which will send a response code along with the details of the pending transaction as notification. To actual execute the transaction the user has to enter the response code on the banks website. The idea behind this kind of two-factor-authentication is that it is very unlikely that an attacker compromises the computer and the user’s MT. Mobile value: (6)

#### **Support for acquisitions of purchase orders**

When a commercial traveller makes an acquisition of a purchase order at customer’s site he enters the details as configuration of a NS, the notification is the confirmation of the order. This helps to speed up the whole process and to eliminate errors since the traveller hasn’t to transfer the details of an order from paper into a computer. Mobile value: (5)

### **4. High-level architecture and protocol for the implementation of Notification Services**

We assume a simple but realistic service provisioning model (Figure 2) where all traffic between the SP and the MT is routed over a trusted zone (Bettini et al., 2005). In our case the trusted zone is interpreted as the core network of the mobile network operator (MNO) because the core network is not a public network like the internet and a subscriber has to “trust” his MNO to a certain degree because the MNO can track the user’s position (he knows the position of the currently used base station or can calculate the position based on the run-time differences of the signal received by several base stations) as long as his MT is switched on. As we will show in this section the MNO hasn’t to know about the content of the communication. This model also has the notion in mind that the MNO concentrates on his core business (provision of wireless communication capacity) and the actual services are provided by third-party companies. We also assume that for the wireless communication (access network) a standard is used that provides security (e.g. UMTS-build-in encryption).

In the “Trusted Zone” there are three components of the infrastructure for a NS:

- An *application router* which routes orders or updates of them to the respective SP.
- The *notification dispatcher* sends the actual push-messages to the MT.
- The *catalogue server* is required to supply the description-files for the services.

An order or update is send from the MT to the application router (1) which forwards it to the respective SP (2). If the SP has to dispatch notifications he sends a message to the notification dispatcher (3) which will trigger the actual notification (4).

The protocol is based on asymmetric cryptography (Schneier, 1996). We denote the ciphertext of a message  $m$  encrypted with the public key  $SP(i)$  of the service provider with index  $i$  as follows:  $\{m\}_{SP(i)}$ . Only the responsible SP can decrypt this message by applying his private key. If the same  $m$  is decrypted at different points in time with the same public key the resulting ciphertexts are *not* identical<sup>3</sup>. Other placeholders in the protocol are:

- *Orderdef*: Configuration of an order (type, manual configuration data, personal context information) or update of an order.
- *Z*: The part of the personal context information that has to be provided or altered by the MNO. This is the case when the MT cannot find out its position (no GPS-receiver attached) and thus the MNO has to provide the location. But even if the MT can find out its position it might provide that information in *Z* so the MNO can perform some kind of cloaking of this information to provide location privacy like proposed by Gruteser & Grunwald (2003) or Gedik & Liu (2004).

---

<sup>3</sup> this is realized by prepending a random bit string before encryption which is removed after decryption; this measure ensures that an attacker cannot perform known plaintext attacks or detect if a message is sent again

- *Address*: User's end address to be used for notification, e.g. phone number for SMS/MMS or e-mail-address.
- *T*: Time at encryption of message; is included to prevent "replay-attacks": if an attacker just replays an encrypted message he intercepted but couldn't understand the timestamp will show that the message is outdated and has to be ignored.
- *SID*: Randomly chosen session ID for the whole lifespan of an order, so the SP can associate updates (or commands for termination or suspension) to the respective order.

When starting or updating an order the client application on the MT sends the following message to the MNO:

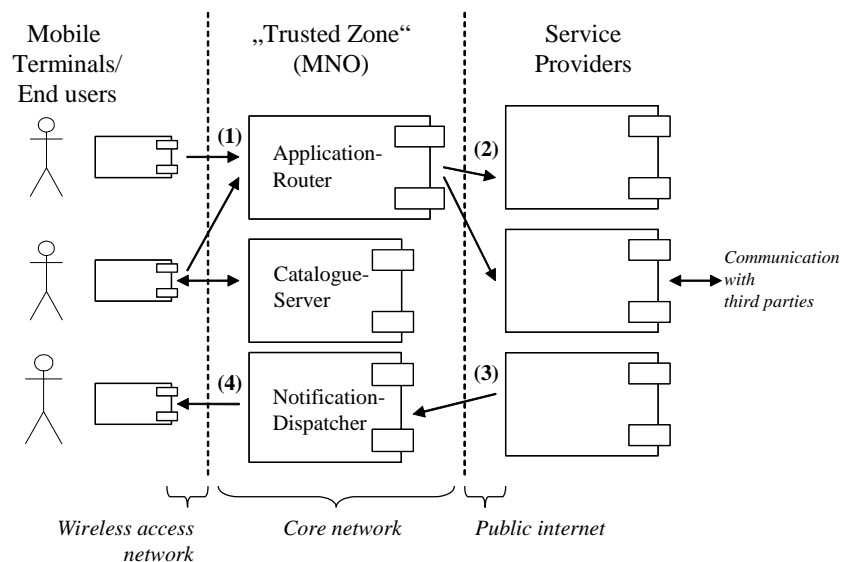
$$i, Z, \{orderdef, t, SID, \{address\}_{MNO}\}_{SP(i)}$$

Looking at *i* the application router knows to which SP he has to forward this message, but he cannot find out what the details of the order are. The respective SP can "unwrap" the message and obtain:

$$Z, orderdef, t, SID, \{address\}_{MNO}$$

If *SID* is new the SP creates an instance of the requested NS; if there is already an instance with *SID* he updates it according to *orderdef*. The SP cannot find out about the identity of the user, because he cannot decrypt the address. If he has to send a notification with content *Y* to the user he sends the following to the MNO:

$$\{Y, t, \{address\}_{MNO}\}_{MNO}$$



**Figure 2: High-level architecture for NS**

The MNO (respective the dispatcher component) decrypts this message to obtain *Y* and also decrypts the address of the user to dispatch the notification. Depending on the requested channel for notification it is not possible for the SP to encrypt *Y* to hide the content of the notification from the MNO; this is the case for

SMS/MMS. For notification channels that allow encryption (e.g. e-Mail, proprietary channel) the SP can decrypt  $Y$  with a symmetric encryption algorithm using a key derived from  $SID$ .

Not for all mentioned examples it is reasonable to hide the user's identity from the SP, e.g. for the online-banking example where the SP (the bank) uses the challenge code included in *orderdef* to look up the details of the transaction and to calculate the response code.

## 5. Summary and Outlook

We discussed a simple paradigm for the modelling and implementation of mobile data services called generalized notification services (NS). It was argued why NS are appropriate with regard to the limitations and habits of users in scenarios with mobile terminals like cellular phones or personal digital assistants. Several examples of different fields of application were given to show that NS are not limited to conventional alert services.

The precise description of a paradigm for the implementation of MDS is a necessary starting point for the development of a software framework in the sense of Johnson & Foote (1988). Frameworks in this sense are generic applications that can be customized to obtain specific applications. This customization is done by configuration of the black-box subsystems or extending the white-box subsystems of the framework (Pree, 1997). Instead of calling functions like with conventional libraries when using a framework the code written by the developer is called by the framework ("inversion of control"); the framework thus handles the control flow. Using a framework applications/services (in our case: NS) can be developed with less effort since not only parts of the implementation are reused but also the whole design. Applications/services developed based on a framework are also easier to maintain.

## References

- Adya, A., Bahl, P., & Qui, L., 2002. Characterizing Alert and Browse services for Mobile Clients. Proceedings of the USENIX Annual Technical Conference, Monterey, CA, USA, 343-356.
- Anckar, B., & D'Incau, D., 2002. Value-added Services in Mobile Commerce: An Analytical Framework and Empirical Findings from a National Consumer Survey. Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35'02), Hawaii, USA, IEEE.
- Bettini, C., Wang, X., & Jajodia, S., 2005. Protecting Privacy against Location-Based Personal Identification. Proceedings of the Conference on Secure Data Management (SDM '05), Trondheim, Norway, 185-199.
- Brown, P. J., 1995. The stick-e document: a framework for creating context-aware applications. Electronic Publishing, Vol. 8, 259-272.
- Burak, A., & Sharon, T., 2004. Usage Patterns of FriendZone — Mobile Location-Based Community Services. MUM '04: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, Maryland, USA, ACM, 93-100.
- Chen, G., & Kotz, D., 2000. A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Dartmouth College, Hanover, USA.
- Dey, A. K., 2001. Understanding and using Context. Personal and Ubiquitous Computing Journal. Vol. 5(1), 4-7.
- Dubinko, M., 2003. XForms Essentials. O'Reilly, Upper Saddle River, USA.
- Forman, G.H., & Zahorjan, J., 1994. The Challenges of Mobile Computing, IEEE Computer, 27(4), 38-47.
- Fraunhofer, 2006. @ptus weather — Weather Information on Demand. Brochure of Fraunhofer ISST,

Berlin, Germany.

- Gedik, B., & Liu, L., 2004. A customizable k-Anonymity Model for Protecting Location Privacy. Proceedings of the 25th International Conference on Distributed Computing Systems (IEEE ICDCS 2005), Columbus, Ohio, USA, 620-629.
- Gruteser, M., & Grunwald, D., 2003. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. Proceedings of the First International Conference on Mobile Systems, Applications and Services, San Francisco, USA.
- Hinze, A., & Voissard, A., 2003. Combining Event Notification Services and Location-based Services in Tourism. Technical Report B 03-06, Freie University Berlin, Germany.
- Hussin, W., Coulton, P., & Reuben, E., 2005. Mobile Ticketing System Employing TrustZone Technology. Proceedings of the International Conference on Mobile Business (ICMB '05), Sydney, Australia, IEEE.
- Johnson, R. E., & Foote, B., 1988. Designing Reusable Classes. Journal of Object-Oriented Programming, Vol. 1, 22-35.
- Kölmel, B., & Alexakis, S., 2002. Location Based Advertising. Proceedings of the International Conference on Mobile Business (ICMB), Athens, Greece.
- Munson, J. P., & Gupta, V. K. (2002). Location-Based Notification as a General-Purpose Service. Proceedings of the 2nd international Workshop on Mobile Commerce, New York, NY, USA, 40-44.
- Pousttchi, K., 2004. An Analysis of the Mobile Payment Problem in Europe. Mobile Business Systems, Mobile and Collaborative Business, Techniques and Applications for Mobile Commerce (TAMoCO), Essen, Germany, 260-268.
- Pree, W., 1997. Komponentenbasierte Software-Entwicklung mit Frameworks (in German). dpunkt.verlag, Heidelberg, Germany.
- Sampat, M., Kumar, A., Prakash, A., & McCrickard, D. S., 2005. Increasing Understanding of a New Environment using Location-Based Notification Systems. Proceedings of the 11<sup>th</sup> International conference on Human-Computer Interaction, Las Vegas, NV, USA.
- Schneier, B., 1996. Applied Cryptography, 2<sup>nd</sup> Edition. Wiley, New York, USA.
- Villanen, J., Modée, K., Koivula, J., Pousttchi, K., & Gumpp, A., 2004. Mobile Enterprise in Germany — State-of-the-art, Expectations and Perspectives for Mobile Business Processes in Small and Medium-sized Enterprises on the German Market. Study of Finpro Munich (Finland Trade Center) and Mobile Commerce Working Group of University of Augsburg, Munich, Germany.
- Wüest, C., 2005. Phishing In the Middle Of The Stream — Today's Threats to Online Banking. Proceedings of the 8<sup>th</sup> Association of Anti-Virus Asia Researchers Conference (AVAR 2005), Tianjin, China.
- Zobel, J., 2001. Mobile Business und M-Commerce (in German), Hanser Publishing, Munich, Germany.

The author studied industrial engineering with focus on computer science and operations research and is member of the mobile business research group of the Institute for Applied Informatics and Formal Description Methods (AIFB) at University of Karlsruhe. His current research focus is on the development of mobile and wireless services.