Using Private and Public Context – An Approach for Mobile Discovery and Search Services

Thomas Butter, Sina Deibert*, Franz Rothlauf

{butter|deibert|rothlauf}@uni-mannheim.de

Abstract: When using context information for the customization of mobile applications, privacy issues are important as users want to limit the amount of context information that is given to service providers. Therefore, this paper distinguishes between public context that the user shares with the service provider and private context that can only be used on the mobile device for the customization of mobile applications or services. This paper illustrates how context information can be used on the mobile device for the customization of results from search/discovery services. In such services, a mobile user receives a list of search results from a service provider. The goal of the presented approach is to sort the list received from the service provider in such a way that the items which are chosen by the user are with high probability at the top of the list. We propose an adaptive optimization approach that sorts the search results with respect to the user's context information and to previous choices made by the user. The approach considers the technical limitations of mobile devices and makes optimal use of the limited resources.

1 Introduction

Since their introduction, a few years ago, mobile devices such as mobile phones or Personal Digital Assistants (PDAs) have become an ubiquitous tool for many people. However, due to the limitations of the first generation of mobile devices and communication infrastructures, most of the current applications only provide basic support for the end-user such as managing e-mails, appointments, events, contacts, etc. More advanced services such as sightseeing or restaurant guides, dating, meeting, or gaming applications, or traffic information systems that can be personalized to the preferences and wishes of users, are not yet established. With more powerful mobile devices and communication infrastructure, such new types of mobile applications can be designed that make use of available context information. Context describes information about the user and the environment or the capabilities of the mobile device, that can be used for the customization and configuration of services and applications. Based on context, applications can be adapted to the environment or to the preferences of the user therefore providing additional value.

*This work was supported by a grant from the Landesstiftung Baden-Wuerttemberg and a grant by the Ministry for Culture and Science to Sina Deibert and Thomas Butter.

In most current mobile applications, all context information is used and transferred to a server that offers the requested information or application. However, when dealing with context information, privacy issues are very important. If privacy issues are not considered appropriately, end-users will not adapt and use new services [HK03]. Therefore, mechanisms are necessary that allow end-users to control the usage of context by mobile services or applications [ADW01].

In this paper, we propose a concept for dealing with privacy issues when using context for mobile applications. The concept allows the user to declare some context as private which is not communicated to the service or application provider, and public context which can be used by the service provider for the customization of applications and services. However, as private context is not transferred to the service provider, it can not be used by the service provider for the customization of services. Therefore, private context information can only be used on the mobile device for the customization of services and applications. The mobile device receives services and applications that have been pre-configured by the service provider using public context information. On the mobile device, these services and applications are customized using private context information. To consider the private context for services and applications, the paper proposes an adaptive learning approach for the mobile device that automatically considers private context for the customization of services and applications. The proposed learning approach is developed for search services, where the user gets a list of results from the service provider. The approach uses previous user decisions for the customization of the service and considers the tight hardware restrictions that exist on mobile devices such as limited battery and processor power.

The following section deals with context in mobile applications. It defines context and describes common types of context. Section 3 discusses the quality of context. It distinguishes between private and public context and describes how the quality of context can be reduced due to problems of inaccurate measurement. Section 4 presents a concept for dealing with private and public context information in search services. Public context information is shared with the service provider and private information is only used on the mobile device for the customization of services. In Section 4.3, we propose a learning approach based on a genetic algorithm which automatically considers context information and previous decisions of the end-user for the customization of search services. The paper ends with some concluding remarks.

2 Context in Mobile Applications

In mobile applications, context describes information about the user, the capabilities of the mobile device, or the environment of the user or mobile device. Context can be used for the customization and configuration of services and applications. In the literature are many different definitions of context. Most of them define context by listing parameters and categories considered as context or not [SBG99, CK00]. The most widely known and used definition can be found in [Dey01] who states that ”Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object

that is considered relevant to the interaction between a user and an application, including the user and the applications themselves."

Context information can be used to automatically adapt services or applications and thus can remove unnecessary interactions with the user or media breaks. For example, context can be used to adjust the graphical output of a service to the technical properties of a mobile device by neglecting irrelevant information, or by changing the used output device from a text-to-speech device to a screen if necessary (e.g. in noisy environments).

Another important area for context usage is to relieve the user from the explicit definition of context and to automatically generate or derive context information from sensors [GS01]. For example, the context property "user is at work" can be automatically derived from the location of the user. As long as the user is located in or next to his office or working place, the user is at work (with high probability) and it is not necessary for the user to define this context property manually. If the description of the user's situation can be done automatically using context information, it is not necessary for the user to define the context by hand. For example, when searching for a train connection in a city, information about the current time (important for departure) or the location of the next station can be automatically derived by context information already available and it needs not to be specified by the user. Another important aspect is the discovery of new services which may be relevant to the user in his current situation. Finally, context changes may also be used to automatically trigger actions and inform the user of newly available possibilities or to start new services. To enable the usage of context information context providers (e.g. sensors) and the services need to have a common notion for the used context attributes. The following paragraphs describe some relevant types of context.

Environmental Context Environmental context is anything that describes the situation of the user but has not to be directly caused by the user or his mobile device. Common environmental attributes are constructed for example from nearby objects, weather or traffic [SV04].

Some context-aware systems generate environmental context information from the position of the user and a projection of the real world into a virtual world. Therefore, nearby objects can easily be found using the location of the user. An example is determining the location of near-by stations, restaurants, or taxis. Other context-aware systems find nearby objects and environmental conditions by using RFID readers and tagged objects [RSMDd03]. Often environmental changes occur fast and they are used for triggering automatic actions for the user. However, such changes are difficult to determine reliably (compare 3.1) and the reliability of such changes should be taken into account before triggering any action based on environmental context.

Device Capabilities Capabilities of mobile device are valuable criteria for the adaptation of applications to the technical properties of the mobile device. Such parameters do not change as often as environmental context parameters since the device is changed seldomly. Relevant context parameters are for example resolution, display size, input and output capabilities and the capabilities of an web browser. These may restrict or forbid the use

of some services or applications entirely and so may be used for the selection of suitable services for a user.

There are also dynamic properties in this category such as the currently used network connection, the available bandwidth, or the latency and type of payment for the current network connection. Changes in these context attributes can be used for example to trigger automatic synchronizations when a cheap network connection (flat rate) is available or to use graphics of smaller sizes if only a low bandwidth is available or traffic is expensive.

User Status The current activity and the role of the user are described by the user status. This context information contains attributes such as work status (free-time versus work time) or attending an event (for example a conference). Furthermore, information regarding the role of the user is important (for example there is a large difference is the user whether only attending a talk or is giving the talk). Such context attributes often determine the availability of the user and can be used to change the types of event notifications or restrict services by the time of availability. An example is to switch off event notifications if the user is involved in a presentation (which can be determined from the user's calendar).

User Profiles Profile information is user-specific and is pre-specified by the user. Such context information determines the way a user wants to interact with his mobile device or properties of the user. This information can be synchronized between multiple devices or stored on a server. Profile information is the most reliable and important source of context information because the user enters this informationdirectly. However, as profiles contain private and sensitive information it is important to protect it and to allow users to restrict their usage to some services and applications [JLTT03]. There are three basic categories of profile information. The first are preferences/likings of the user. Personal information like the gender, age or family status constitue the second group. Furthermore relationships to other users may be used together with their status information which is the core of instant messaging systems such as ICQ, and others.

History Previous decisions of the user are also important context that can be used for the customization of services. Therefore, it is useful to save the history of user decisions and to use this information for forecasts, determination of context, or to predict user behaviour.

3 Quality of Context

As mentioned in the previous section there are multiple sources of context information. Most of these source have some technical limitations on accuracy and resolution or are based on inference from other context sources which in turn may degrade the resolution and trustworthiness of context information [BKS03].

From a users point of view, the use of inaccurate or wrong context information is only a larger problem than neglecting it. Furthermore, derived context attributes rely on a minimum quality of other context attributes they are based on. For example, when determining

context information about the role of a user in a presentation (e.g. to determine the type of user notification), the accurate measurement of the user's position can be important. It can be assumed that the user is currently speaking if his position is next to the presentation devices. In constrast, it can be assumed that he is only listening to a presentation if he is constantly located more than five metres away from the presentation devices. Therefore, an accurate measurement of the user's position is important for determining the derived context information "user is giving/not giving a presentation". In contrast, when navigating the user to the room for the presentation an accuracy of five metres may be accurate enough to guide the user to the conference room.

New context information can also be derived from other context attributes by inference [SNPL04]. For example, a context attribute which contains the current city can be derived from the geographical location of the mobile device and a city database. For determining new context information, the reliability and accuracy of the used information context is important. When determining the city from the user's position, determining the user's position with low accuracy near a city border would lead to a low probability value for the city.

An important issue of context information is private and public context information. Usually, the user wants to restrict the usage of private context information by the service or application provider. Common ways to restrict the usage is to either not give context information to the service provider or to reduce the quality of the context information that is given to the service provider ("blurring").

3.1 Quality Losses of Context due to Measurement

The most prevalent accuracy loss of context information is caused by the technical limitations of the measurement devices. For example, GPS devices for determining the users location have an accuracy of about 15 metres in open field while having a much worse accuracy within cities with limited sky visibility. Therefore, it is important for services and applications that make use of context to know about the quality of the context information. Determinants of quality of context information are accuracy, reliability, and time lag.

Accuracy The resolution or accuracy of context information usually depends on the technical capabilities of the sensors. In the simplest form, accuracy can be expressed by the specification of a range of values where the correct value is inside. Depending on the type of context matching, a context-aware system could then consider this information by either assuming the center of the interval as the correct value or ignoring it entirely. An advanced form for considering inaccurate context information is to use probability distribution functions. However, such functions are often not available and difficult to evaluate.

Reliability Most context attributes (especially if expressed as Boolean values) have a level of reliability. For example, the reliability of the context information "user is / is

not attending a presentation" depends on other context information and may be more or less reliable. To deal with the reliability of context information, probability values can be assigned to context information to consider the trustworthiness during context evaluation. Trustworthiness also plays a role when data entered by other users is used as context information. Here an anticipated probability of the trustworthiness of the other user can be helpful for deciding if the value is important.

Time Lag Not every context sensor may be able to constantly supply all relevant context data all the time. Therefore, context information can be outdated or change over time. To solve problems with time lags, the time of context acquisition should be explicitly noted and transmitted to better utilize the context information. Combining time lag of context sensors with the history of context changes can be helpful for making use of context information with time lag.

3.2 Private and Public Context

Many context attributes are naturally very personal and considerably breach the privacy of the user. To overcome the user's fears of privacy loss, it must be possible for the user to declare context information as private or public. Public information may be used by service and application providers, but not private information. There are also intermediate states of context information possible by reducing the quality of context information ("blurring"). Then, the user intentionally reduces the quality of context information. A common example is disguising the user's identity when using specific services by using a slightly changed age or location.

Blurred and public context information is often enough for the service provider to preconfigure a service while still making it hard to track the exact context of an user. The existence of blurred and private context results in a situation that the service provider can consider less context information than the mobile device for the customization of services. However, the mobile device often does not have the technical capabilities to perform extensive configurations. Therefore, mechanisms are necessary that consider the tight hardware restrictions of mobile devices and which can use the private or blurred context information for a user-specific configuration of services on the mobile device.

4 Using Private Context in Mobile Services and Applications

When searching for reasons that explain the low usage of advanced mobile applications and services [HK03], privacy issues are seen as an important, but often neglected, barrier for a widespread use of such services. Users do not want to give private information like user profiles, information about the environment, or other types of context to a service provider but restrict or prohibit the usage of such information. We develop a concept that allows the user to restrict the context information that is sent to the service provider and

to use the private context information to customize the services or applications directly on the mobile device. In the following sections, we show the basic concept and demonstrate how private context information can be used on the mobile device to customize a service. As an example, we use a discovery/search service.

4.1 Distinguishing between Private and Public Context

As already discussed, we distinguish between private and public context information. When using a service or an application, public context information can be sent to a service provider and used for the personalization and customization of the service. In contrast, private information is not sent or substantially degraded in its quality before sending.

Each user chooses which context attributes are private and which may be used outside of his mobile device. Furthermore he may choose to deteriorate the quality of some attributes to decrease the possibility to be tracked by a service provider.

In many services and applications that are offered for mobile devices, the main computational effort is performed on centralized servers (for example search engines or routing discovery). In our concept, services and applications are customized in two phases (compare also Figure 1 as an example of a search service for restaurants). The first phase happens on the server using only the public context attributes the user wants to make available for the service provider. The service provider can use different mechanisms to match the context information with the service descriptions and to adapt the results to the context of the user. When delivering the service descriptions or the application to the mobile device, it adds a list of context criteria which were used for matching and further context criteria which could have been used if they had been sent by the client. These additional criteria will be used in the next phase.

The second phase is performed on the mobile device. An application running on the mobile device scans the context criteria which could have been used for customization and checks whether there are context criteria that have been declared as private to be checked. If private context information exists, it is used for customization. A simple, but illustrative example is the context information “available time”. This context information can be declared as private and is only used on the mobile device for customization or narrowing of search results (of course, only if it makes sense to use this context information). Then, for example it is possible to use the context “available time” on the mobile device to determine if a proposed action is feasible for the user but it is not necessary to give this information to the service provider.

4.2 Search and Discovery Services

We want to illustrate the basic concept for a service searching for restaurants. This example should be used as a representative for discovery services that are initiated by a mobile user and return a list of search results (e.g. a list of restaurants). Discovery services are offered

by service providers and only the outcome of the search – a list of search results – is sent to the mobile device of a user. When using discovery or search services, usually user-specific context information like location, preferences, time, and other context information is considered for the search. We want to assume the following scenario:

The mobile device knows the location of the user with an accuracy of some metres and knows from the profile that the user is a smoker and only wants to pay with credit card. The user declares the context information "pay with credit card" as private. Furthermore, he does not want to give his exact location to the service provider but reduces the accuracy of the context information "location" to a few hundred metres. The context information "smoker" is used as public context.

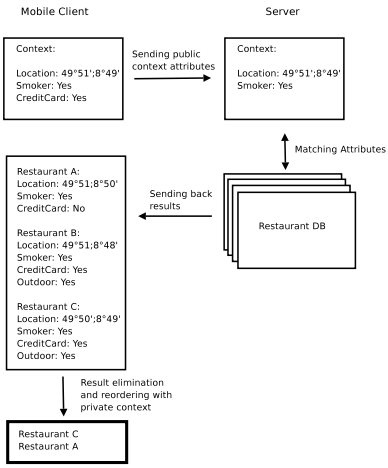

Figure 1: Search Service

After sending the request (including information about his position and smoking preferences) to the service provider, the service provider uses these results to create a list of appropriate restaurants that do not contradict these attributes [vSPK04, HV03]. The service provider notes the context attributes that are considered for the search and that influenced the order of the restaurants in the list that is sent back to the mobile device. Because these attributes contain a validity range, it is possible to re-evaluate the match without understanding the semantics of the service description by comparing the received ranges and the locally available values for each attribute. For the client, private context is considered and all restaurants where no credit cards can be used are excluded from the list. Furthermore, the list can be re-ordered as the correct position of the user (and not the blurred information sent to the service provider) can be used for calculating the distances to the restaurants. An overview of the process is given in Figure 1.

Each context attribute which is not sent to the server may increase the size of the response since the servers ability to filter the results is reduced. While marking single binary attributes (e.g. "pay with credit card") as private is expected to double the number of results the omission of the location may result in a non-feasible increase in the number of results if there are a lot of location-specific services available. Therefore attributes having a big influence on the number of results and continuous values should preferably be blurred to allow a prefiltering on the server while still preserving the privacy of the user.

4.3 Learning the Individual Importance of Context Attributes in Search and Discovery Services

The available information regarding the influence of context attributes on the matching process for each search result and the possibility to repeat the matching process with ad-

ditional attributes on the mobile client makes it possible to re-customize the results of the search on the mobile client. In our prototype implementation we try to find out which attributes that have not yet been considered by the service provider influence the choice of the user and how important each context attribute is for the user. If the user prefers search results from the list that always have some specific context attributes, the application observes this behavior, learns it, and considers the preferences of the user for future searches. The following paragraphs describe in detail how the user preferences are considered for the ranking of search results on a mobile device.

4.3.1 An Adaptive Optimization Approach for Considering Context

When dealing with results from search services like a restaurant search, the order of the restaurants in the list is important. Users want to have a list where their favorite choices are at the top of the list so may do not have to scroll or click until they find their favorite search result. However, when receiving the list of restaurants from the service provider, all private context information has not yet been used for ordering the list. Therefore, in a first step the client discards any results from the list which can be eliminated due to existing private context attributes. In one next step the mobile device weights the relevance of context attributes by using historical data to better anticipate the decisions of the user. The goal of such mechanisms is to detect the relevant context characteristics and to re-order the list such that the search results (restaurants) preferred by the user are at the top of the list.

When considering context information for re-sorting the list, it is unknown how strong the different context criteria influence the preference and choice of the user. Some context information (like smoking/non-smoking restaurant) may be very important for the user's choice whereas other context information (such as distance to restaurant) is less important. To consider the influence of the different context criteria on the preferences of the user, we have developed an adaptive search algorithm that learns which context characteristics have been important for previous choices of the user. Since sending the decision history to the server would give away private and personal information, the whole weighting approach is performed on the mobile device.

In our approach, a weight w_i is assigned to each context information i. w_i represents the importance of context i for the user. The higher the weight w_i of a context information is, the higher the rank of those search results (e.g. restaurants) that match this context property i. For the ordering of the list of displayed search results on the mobile device, the sum $\sum_i x_i w_i$ is used. x_i denotes if the context attribute i matches the property of the search result ($x_i = 1$), or not ($x_i = 0$). The list presented to the user is ordered such that the search results with the highest sum are presented at the top of the list.

The ordering of the list offered to the user depends on the weights w_i. The goal is to determine the weights w_i such that in all previous searches the search results that have been chosen by the users would have been near to the top of the list. Then, we can assume that for the current search the user's preferences are considered appropriately and the search result that will be chosen by the user appears at the top of the list with a high probability. For this task, an optimization algorithm is necessary that determines the weights w_i such

that the search results (e.g. restaurants) that are preferred by the user are on the top positions of the list. Such an optimization algorithm can use the previous choices of the user for the evaluation of different combinations of w_i.

When the user uses a search service for the first time, there is no information available regarding the importance of the different context characteristics. Therefore, when using a search service for the first time, we assume that all context characteristics have the same importance and the list of results that is received from the service provider is re-ordered in such a way that all context criteria are considered equally. For example, for a non-smoker who does not want to use a car to get to a restaurant, a non-smoking restaurant is as attractive as a restaurant where he does not need a car.

4.3.2 Implementation

For the optimization algorithm that should find the optimal w_i, tight technical restrictions are imposed by the limited capabilities of mobile devices. The most important are:

- The algorithm must have low heap memory usage and small code size.
- It must be possible to partition the algorithm in very short execution intervals to not interfere with user tasks.
- It should be possible to use old weights w_i as a starting solution.
- The algorithm should not rely on floating point arithmetic since these may be slow on mobile devices.

To find the optimal w_i, linear optimization methods can not be used as the quality of a solution (a solution can be described by a parameter setting for the w_i) is non-linear and depends on the user's choices. To solve the problem of finding the optimal values of w_i, we use a genetic algorithm [Gol89]. This type of optimization algorithm can solve non-linear problems, it can work without any noticeable drawbacks for the user, it is possible to implement it with very few lines of code, it is possible to stop the execution after every single fitness evaluation without the need to hold extensive state information in heap, and meaningful results are available anytime.

We implemented the genetic algorithm for the weight optimization on a mobile device in a Java class file with a size of about 5kB. Such a small application can easily be held in memory all the time. The history (previous user choices) is read in sequential access for every fitness evaluation (one fitness evaluation calculates the quality of different w_i using the history, where a top-ranked service means high fitness and a low-ranked low fitness) which allows efficient loading for any storage medium.

The genetic algorithm uses a population size of 50, simple standard recombination and mutation operators and fitness proportional selection with elitism to make sure that the best ever found solution is always available [Gol89]. The algorithm gets some processor time for fitness evaluations during short idle intervals like the time waiting for a reply over the network. Using such time slots does not affect battery life that much because an active network connection does not allow a mobile device to use a battery saving state.

Furthermore, in such time intervals the user has to wait for the reply so it won't slow down user interaction.

4.3.3 Adding Additional Context Attributes from History

The mobile application tries to find patterns in the usage of services by also considering context attributes which are not explicitly specified in a rule for the service description. These may be not specified because the service provider did not know about the possibility to sense a specific context attribute at the time of service creation or because the context attribute only concerns some individual users. One example for an individual influence for just some users would be that a user usually visits certain restaurants only for lunch.

Whenever the user chooses an entry from the list, all currently available context attributes are stored, together with this entry, to record the context of the user's decision. If an entry is chosen multiple times in different searches, the context attributes which are equal in all decisions for this entry will be used as additional, implicit context attributes for further rankings. Those additional, implicit context will be used just like the explicit context attributes. For the lunch example, restaurants previously selected at lunch time would be favored only at lunch time. To save storage space for the user history, context attributes which have been different for multiple choices of the same entry are discarded and added to a blacklist for this entry, so they won't have to be considered again for this entry.

5 Conclusions

Using context information for the customization of mobile services is a promising direction to increase the usage of mobile devices. However, when dealing with context information, privacy issues are very important. Therefore, mechanisms are necessary that allow the user to control the usage of context. In this paper, we distinguish between private and public context information. Public context information can be used by service providers for the customization of mobile services. Private context information can be used on the mobile device for further adaption of mobile services with respect to the users context.

We illustrated for search/discovery services how context information can be split into public and private context. In search services, a mobile user receives a list of search results from a service provider which can be sorted according to the context of the user. The goal is to sort the list in such a way that the items that are chosen by the user are at the top of the list. We propose an optimization approach that orders the results delivered from the service provider with respect to the user's private context information and to previous choices of the user. Due to tight technical restrictions and to the non-linearity of the problem, we use a genetic algorithm. This approach requires only minimal computational effort and makes optimal use of the limited resources of mobile devices.

Currently, we have implemented the approach and already tested it for some test examples and a few test users. The first results are promising and we are planning to evaluate the approach for larger scenarios. Open questions are the quality of the reordering (however, the first results are encouraging) and the scalability of the approach.

References

[ADW01] Mark Ackerman, Trevor Darrell und Daniel J. Weitzner. Privacy in Context. *Human-Computer Interaction*, 16(2/4):167–176, 2001.

[BKS03] Thomas Buchholz, Axel Küpper und Michael Schiffers. Quality of Context: What It Is And Why We Need It. In *Workshop of the HP OpenView University Association 2003 (HPOVUA 2003)*, 2003.

[CK00] Guanling Chen und David Kotz. A Survey of Context-Aware Mobile Computing Research. Bericht TR2000-381, Dartmouth College, Computer Science, Hanover, NH, November 2000.

[Dey01] A. K. Dey. Understanding and Using Context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.

[Gol89] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.

[GS01] Philip D. Gray und Daniel Salber. Modelling and Using Sensed Context Information in the Design of Interactive Applications. In *EHCI*, Seiten 317–336, 2001.

[HK03] Shuk Ying Ho und Sai Ho Kwok. The Attraction of Personalized Service for Users in Mobile Commerce: An Empirical Study. *ACM SIGecom Exchanges*, 3(4):10–18, Januar 2003.

[HV03] Annika Hinze und Agnès Voisard. Locations- and Time-Based Information Delivery in Tourism. In *SSTD*, Jgg. 2750 of *Lecture Notes in Computer Science*, Seiten 489–507. Springer, 2003.

[JLTT03] Sirkka L. Jarvenpaa, Karl Reiner Lang, Yoko Takeda und Virpi Kristiina Tuunainen. Mobile commerce at crossroads. *Commun. ACM*, 46(12):41–44, 2003.

[RSMDd03] Kay Römer, Thomas Schoch, Friedemann Mattern und Thomas Düben-dorfer. Smart Identification Frameworks for Ubiquitous Computing Applications. In *PerCom*, Seiten 253–. IEEE Computer Society, 2003.

[SBG99] Albrecht Schmidt, Michael Beigl und Hans-Werner Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[SNPL04] A. Shehzad, H. Q. Ngo, K. A. Pham und S.Y. Lee. Formal Modeling in Context Aware Systems. In *Workshop on Modeling and Retrieval of Context, CEUR*, 2004.

[SV04] Jochen H. Schiller und Agnès Voisard, Hrsg. *Location-Based Services*. Morgan Kaufmann, 2004.

[vSPK04] Mark van Setten, Stanislav Pokraev und Johan Koolwaaij. Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In *AH*, Jgg. 3137 of *Lecture Notes in Computer Science*, Seiten 235–244. Springer, 2004.