Adaptive Positioning for Ambient Systems

Christian Kray, Gerd Kortuem

The quality and availability of positional information (location, orientation) is a key resource that strongly influences the way in which ambient services can be provided to nomadic users. In this article, we propose a number of mechanisms that tackle resource restrictions related to positional information on two levels, the application level and the positioning level. We introduce a generic model and architecture, which allows for the development of resource-aware applications that adapt to the quality of positional information. Our approach provides location-based services with fine-grained control over the balance between resource allocation and quality of positional information. It also enables resource-unaware applications to benefit from adaptation on the positioning level. The feasibility of the approach is demonstrated by a case study using a mobile tourist guide.

1 Motivation

Increasingly computer systems consist of large numbers of interconnected mobile and embedded devices. To ensure that these systems and devices enhance our life without disrupting it we need intelligent and intuitive interfaces with the ability to recognise and respond to a user's situation and context.

Location is one of the key context parameters and thus a positioning service, i. e. a service that is able to reliably and accurately determine a user's position, is a key service for any ambient system. The requirement to minimise user disruption directly translates into the need to minimise service interruptions and increase service availability. While there are many technologies for locating a device or user (from GPS to camera-based tracking systems), none are available universally (GPS, for example, requires good 'visibility' of satellites and usually does not work indoors). This calls for an adaptable positioning service that makes use of a number of redundant or alternative location systems and which is able to dynamically combine them or switch between them.

In this paper we describe a model and implementation of an adaptive positioning service for ambient systems that dynamically and seamlessly combines three positioning methods: measurement-based positioning, inference-based positioning, and interactive positioning. This service helps to minimise user disruption caused by service interruptions: it guarantees that positioning information is available as long as one of the underlying methods is able to deliver reliable data. To enable applications (and application designers) to influence how adaptation is performed, the service provides a well-defined quality-of-service interface for specifying the required quality of positioning information. We demonstrate the feasibility and utility of this approach by describing a prototype implementation and case study.

In the remainder of this paper we provide an overview of different positioning techniques and describe a model for adaptive positioning. This is followed by a discussion of the implementation and a case study of using an adaptive positioning service in a mobile tourist guide.

2 Positioning techniques

There are a number of different ways to determine the position of a user and/or a device in space. Regardless of their respective properties and technical requirements, we have identified three distinct categories of techniques based on the main source of knowledge used to acquire positional information. These are:

- **Measurement based positioning**
  Measurement based positioning methods gather data from sensors and other positioning determination equipment and directly compute the device location. Perhaps the best known example of measurement based positioning is the Global Positioning System (GPS) (e. g. [15]). However, infrared beacons (e. g. [3]) and ultra-sound receivers (e. g. [1]) have also been used to the same effect. Furthermore, wireless communication networks such as GSM or 802.11 WLAN provide another means to determine the location of a mobile device (e. g. [5]). This is a commonly used method to acquire positional information in ambient and mobile systems.
- **Inference based positioning**
  Inference based positioning methods perform reasoning to improve the quality and/or precision of location information derived with measurement based methods. The most important example of inference based positioning is *dead reckoning.* The majority of inference based methods combine direct measurements with knowledge about past device locations and current movement patterns to infer a device's current location (cf. e. g. [14]). Inference based methods generate hypotheses: hypotheses are more or less reasonable but are not guaranteed to be correct.
- **Interactive positioning**
  Interactive positioning is a method that uses an interactive dialogue between system and user to determine the position [13], and is thus only applicable if a user is present and willing to engage into interaction. A confirmation dialogue is the most simple example for interactive positioning: the system asks the user to confirm whether they really are at the position that was computed. A more sophisticated approach is to ask the user for an initial and rough estimate of the current location and to feed this

information into an inference based method to improve accuracy. For the mobile tourist guide system Deep Map [16] we developed an approach, which uses input from the user to eliminate some of the competing position hypotheses derived by an inference based method or other means [13]. The system can determines the user's current location by proactively initiating a dialogue about the visibility of nearby landmarks and prominent buildings [13]. This dialogue is optimised in that it consists of the smallest possible number of questions required to determine the location with respect to a desired positioning accuracy.

Each of these three techniques for positioning has advantages and disadvantages. Measurement based methods can be very accurate but in some cases are not very reliable. Measurements and sensor information can exhibit systematic yet difficult to predict errors due to the characteristics of the operating environment. For example, bad weather, narrow aisles and dense vegetation can severely deteriorate the reception of GPS signals. Infrared suffers from reflections and shielding, while ultra-sound is prone to interference. Approaches based on network cells frequently face reception problems, e. g. in crowded places.

Inference based methods can work in situations where measurement based methods fail. In Deep Map, for example, we used a dead reckoning algorithm that takes into account contextual information such as previous locations, the current means of transportation, and the user's age and physical constitution to derive hypotheses about the user's speed, direction of movement and, ultimately, location. This way, we were able to determine the user's location in situations in which measurement based methods alone would fail [12]. On the downside, it is oftentimes difficult to predict the accuracy of the derived location information. In addition, inference based methods can be computationally expensive and might not lend themselves to implementation on less powerful mobile devices.

Finally, interactive positioning is able to deliver location information even if no positioning determination equipment is available. For example, we have demonstrated that it is possible to accurately determine the user's current location by asking the user a small number of very specific questions regarding the visibility of landmarks [13]. Interactive methods, however, are generally more intrusive and time consuming, and they can depend heavily on advanced language processing capabilities of a system. Thus their application has to be carefully evaluated.

3 Adaptive positioning

The discussion above highlights that there is no single method that is able to deliver the best possible position information in all circumstances. To address this issue we developed a new method and system for *adaptive positioning*. In particular, we developed a resource-aware adaptive positioning service, a component for ambient and mobile systems that can be used by location-aware applications to acquire location information with varying quality of service attributes. The goal of this new approach is to use adaptation to increase availability of location information and to deliver location information with the quality of service as required by a particular application.

3.1 Quality of service model

Our quality of service model for resource aware adaptive positioning has three main constituents: location aware applications, a positioning service and positioning resources. One or more *location aware applications* request location information from a central *positioning service*. This service relies on a (flexible) number of *positioning resources* involved in delivering location information: a series of *positioning sensors* (such as a GPS receiver), computing resources (such as a CPU), networking resources (such as a wireless network), storage resources (such as system memory) and - most notably - the user. Users are modelled as a resource because their collaboration is required for interactive positioning. An application can access the positioning service by means of an positioning application programming interface (positioning API). This interface allows applications to request location information with specific QoS characteristics. Applications send *qualified requests* to the positioning service and in return receive *qualified responses*. The abstract structure of these messages is as follows:

- **get-current-position***(time, distance, confidence, confidenceAngle, confidenceMotion)*
  The parameters describe the required quality of information in terms of recency (time), maximum deviation (distance) as well as confidence with respect to location (confidence), orientation (confidenceAngle), and motion (confidenceMotion). All parameters are optional. In case they are not passed, they are assumed to be set in the least restrictive way, e. g. confidence values are set to 'very low' and recency is set to 'anytime'.
- **current-position**: *position, errorX, errorY, viewDirection, viewAngle, viewInclination, viewDepth, speed, confidence, confidenceAngle, confidenceMotion*
  The corresponding reply contains the location (position) with deviation in x/y-direction (errorX/Y), various information describing the field of view of the user (viewDirection/Angle/Inclination/Depth), the user's current speed (speed) and the corresponding confidence values (confidence, confidenceAngle, confidenceMotion).

3.2 Adaptive positioning service

When a positioning service combines measurement, inference, and interaction to determine the user's current location, it gains the ability to adapt not only to the availability of sensor data but also to other resources such as time or memory. For example, if there are severe time constraints, the number of user interactions may be reduced (at the expense of precision). This applies especially to processes aimed at reducing the uncertainty as they are usually iterative. Memory usage can be slashed by skipping certain processes completely (again at the expense of precision).

Figure 1 (right) illustrates how the QoS model described in 3.1 was implemented in the Deep Map prototype to provide for robust and adaptive positioning [12]. Measurements are used to directly compute the current position, and to seed the other processes with an initial hypothesis. One of these processes is inference, which consists of a context-aware dead reckoning algorithm. The inference process is tightly linked to interaction related processes. Whenever the result of the algorithm is not adequate for the task at hand – e. g. when it lacks precision – the

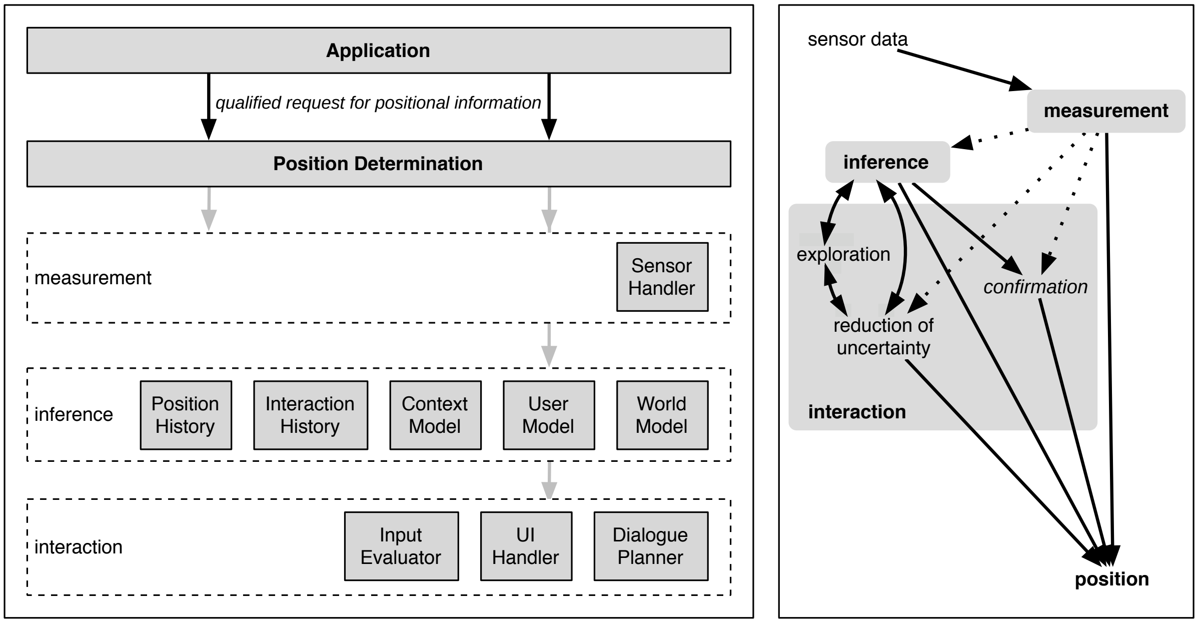


Figure 1: Example positioning architecture implementing the QoS model (left), and processes involved in determining the user's position: measurement, inference and interaction (right).

output of dead reckoning is passed on to interaction for further processing. Similarly, when new information is acquired through interaction, the inference process is triggered in order to evaluate whether the user's current position can now be determined in accordance with the specified service requirements.

The combination of these methods allows for the dynamic adaptation to the quality of sensor data. As long as the output of the sensors is sufficient to determine the user's position, no process except measurement is triggered. When the quality of sensor data degrades or when there is not enough data available, inference and/or interaction are triggered. This results in higher resource consumption, e. g. in terms of time, CPU power or memory, hence establishing an inverse relationship between quality of sensor data and resource consumption.

In general, the positioning service uses the application's QoS parameters (mentioned in the previous section) as guidelines and makes reasonable effort to deliver the requested level of service. This means if the positioning service is able to deliver the QoS level as requested by the application it will do so. If however the requested QoS level exceeds the maximum deliverable QoS level, the response will be marked accordingly. For example, an application might request location information that is accurate to within a few centimetres, yet available positioning sensors may only deliver information that is accurate to within meters. In this case, the errorX/errorY fields would reflect the higher granularity of the available information.

In order to facilitate application development, requests are interpreted by a simple algorithm. It analyses the parameters passed in the request and tries to satisfy the constraints by consuming as few resources as possible. For example, if sensor data meets the criteria specified in a request, no further resources are used to generate the reply. Applications can also decide to not pass some parameters, which are then interpreted as being set to the least restrictive values. Consequently, by not specifying any parameters, an application can effectively access positional information at the quality that is currently available without the usage of further resources. This information can then serve as a basis for deciding, which adaptation strategy to apply. For example, in case an application wants to invest more resources into improving the quality of positional information, it can resubmit its original request with some parameters set to tighter constraints.

3.3 Application level adaptation

The adaptive positioning service is an example of application-aware adaptation [17]. There is a collaborative partnership between location-aware applications and the positioning service. The application sets the required QoS level and the positioning service determines how best to satisfy the request. In case the QoS request cannot be satisfied the service delivers the best possible level and informs the application. The application can then make a strategic decision of how best to deal with this situation. If the positioning service cannot deliver the QoS level requested by an application, the requesting application can then react in one of four ways, i. e. by applying one of the following strategies:

1. **Allocate more resources to improve the current position hypothesis**
   For example, if the initial request only asked for positional information of any quality, another request with tighter constraints may be sent to the positioning service. This will result in the allocation of further resources on the positioning level.
2. **Use more resources to still provide its service at its optimal quality**
   Depending on the service an application is providing, it may be possible to still provide the service even if the

quality is less than desired. For example, a mapping service may decide to generate a map that is more detailed than the one it would produce if the quality of positional information was adequate. The increased level of detail will then result in higher resource consumption, e. g. in terms of computing power or memory usage.

3. **Degrade the quality of the service it provides**
   An example for this strategy is an augmented reality application that would switch from a first person perspective to a birds-eye view when the current location hypothesis is not precise enough.
4. **Fail at providing its service**
   This is the standard 'strategy' of non-adaptive applications. Obviously, this is not a satisfying solution for the user of a system, who may be willing to accept either a lower-quality of service or longer response times to avoid failure.

All of these strategies are supported by the model and architecture described here.

3.4 Positioning architecture

In order to simplify the realisation of these strategies, we designed an architecture that supports not only resource-aware adaptation but also provides transparent access to the resource-adaptive determination of the user's current location. Figure 1 (left) depicts the positioning service architecture. An application in need of positional information sends a qualified request (see 3.1) to a single component in charge of determining the user's current location (position determination). The qualified reply (see 3.1) enables it to decide whether it needs to apply an adaptation strategy, and if so, which one.

If the current position hypothesis does not meet the requirements in the context of an application, it can decide to trigger the inference process and/or interaction. This will result in an improvement of the quality of positional information at the expense of further resources. While the second strategy listed above is application-dependent, the implementation of the third one (degradation of service quality) can benefit from the approach described in the previous section. The distinction of several different processes related to positioning lends itself to the definition of several levels of quality and confidence. These levels provide a basic scale of location quality, which facilitates the definition of distinct levels of degradation. For example, an application can define an adaptation strategy that degrades the quality of its service to the lowest level in case sensor data is of very bad quality. It can define another one that optimises the quality of its service for a scenario, where location information is of excellent precision but has a low confidence value.

The architecture also enables resource-unaware application to transparently benefit from adaptation to sensor data of varying quality. Such an application can request positional information of the highest quality level, which will result in the activation of all processes related to positioning depending on the quality of sensor data. The positioning service will then adaptively apply all necessary processes in order to attain the request quality of service. The holonic structure [7] of our architecture renders this access completely transparent. Resource-adaptive applications are also shielded from the complexity of position determination while being provided with fine-grained control over resource consumption. The following section will present examples from a prototypical implementation of the presented architecture.

4 A case study

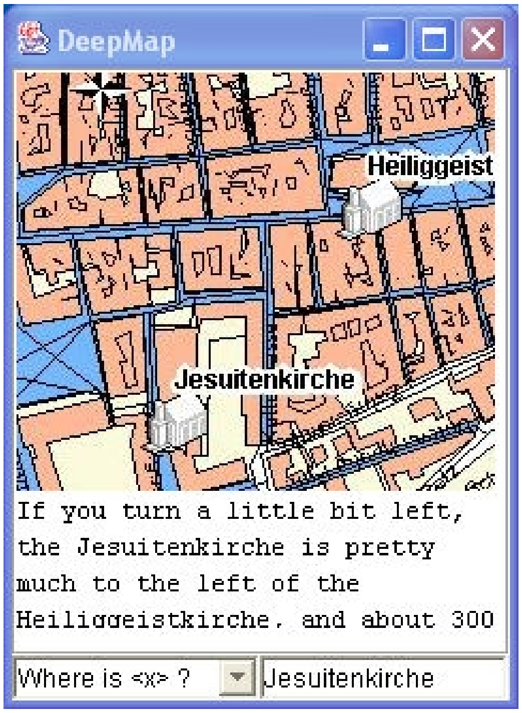
Deep Map [16] is a mobile tourist guide that provides a number of (location-based) services, which it adapts to the situational context. Users can request information on objects in the city of Heidelberg and ask for localisation of these. Furthermore, the system can incrementally guide them to arbitrary locations or provide complete directions. Additionally, Deep Map can generate tailored maps of any location in the city. The system provides a multi-modal interface that combines speech, text, images, and animations. Within the Deep Map project, one major research goal was to realise adaptive positioning, and consequently we implemented most of the architecture described in the previous sections.

One specific service provided by the Deep Map system is the generation of adapted you-are-here maps. In this case, the user asks for a map of their immediate surroundings, and the system generates a map of the area, which depicts landmarks as well as objects that are of interest to the user.[1] This application is an example for the first adaptation strategy: the component realising it will allocate more resources to improving the quality of positional information if it is insufficient. More specifically, it will first request the user's location based on current sensor data. If this information is not precise enough to warrant the generation of a you-are-here map, it will then instruct the positioning subsystem to apply further means (e. g. inference, interaction) to position the user more precisely.

Deep Map also provides a service for the localisation of arbitrary objects, where it generates a multi-modal description of the position of the target object (see Figure 2). This is an example for the application of the second adaptation strategy. In order to compensate for less than desired quality of positional information, more resources are used to still provide the service at a high quality. In this case, imprecise positional information is accounted for by using induced frames of reference [12]. These frames require the listener/user to first perform a mental or physical re-location and/or re-orientation before the attached localisation can be decoded. In the example shown in Figure 2, the use of an induced frame of reference results in the initial phrase "If you turn a little bit left". Hence, the user first has to perform this operation before being able to understand the following localisation. As one would expect, the use of induced frames of reference requires more resources than simply relying only on direct frames of reference. Thus, the system maintains its level of service quality by allocating more resources to cope with low quality positional information.

In order to demonstrate the general feasibility of interactive positioning, we also implemented an approach based on object visibility (cf. [13]). Within Deep Map, this process is currently triggered in the context of the service that generates personalised you-are-here maps. In case there is no sensor data available at all, the system will resort to a series of questions about the visibility of objects that are located somewhere near the last know

[1] The latter ones are selected according to known interests of the user as well as their individual interaction history.



If you turn a little bit left, the Jesuitenkirche is pretty much to the left of the Heiliggeistkirche, and about 300

Where is <x> ? Jesuitenkirche

Figure 2: Localising an object using an induced frame of reference

street the user was on. An underlying matrix linking potential positions and objects is used to optimised the dialogue in terms of information gain and shortness. Initial results are promising as they allowed us, for example, to locate a user to within 10 meters on an urban street of approximately 170 meters length. This required only three interactions.

5 Related work

Advanced work on location systems focuses on sensor fusion techniques to improve accuracy and availability of location information. The Location Stack [9, 11], for example, performs multi-sensor fusion using Bayesian filtering [8] and adaptive particle filtering [10]. In addition it provides a uniform programming interface to applications. While similar in motivation, the adaptive positioning framework described in this paper differs in a number of important respects: rather than fusing sensor data on a low level, it combines already processed and improved information from multiple independent location systems. This allows the use of specialised and tailored algorithms for each location system. Second, it provides an explicit quality of service model rather than a probabilistic one. We claim that this improves the ease of use from a programmer's point of view. Third, our framework supports resource-aware adaptation by taking into account the current status of network and computing resources. Finally, we are able to integrate interactive positioning as a method of last resort when there is no direct sensor data available.

Adaptive positioning has also been explored in the context of mobile navigation systems and tourist guides. The GUIDE system [6] provides visitors of a city with information adapted to their interest and location. The system uses a WIFI-based location system and supports a simple form of interactive positioning. In the absence of a wireless network connection GUIDE asks users to identify their position by picking a nearby sight or prominent building from a list. In a similar way, adaptation is realised in the LoL@ [2] system. It relies on GPS for positioning but it has been designed to use the position information provided by third generation mobile phones. Whenever LoL@ is unable to precisely determine the user's current position from sensor readings, it dynamically creates a list of street segments and ask the user to select the one they are located on. This list consists of ranges of house numbers along with the name of the street.

Human attention and cognitive load are one of the most important resources when it comes to interactive positioning. Resource-aware adaptation in the context of cognitive resources has been explored in the REAL project [4]. The IRREAL system generate interactive maps of the user's environment, and selects the level of detail and scale according to how precisely the user's location is known. For example, very coarse information results in a map that depicts a larger area with few details whereas very fine-grained information triggers a very detailed map of the immediate environment. The user can also click on specific icons embedded in the map to tell the system about their current location. This information is then used to improve the level of detail of the map. While all these systems exhibit innovative resource-aware and interactive position methods, they are missing a general framework for adaptive positioning. Our approach provides such a general framework that makes it easy to integrate additional location technologies and to dynamically switch between them depending on resource availability.

Most positioning technologies deliver data with a specific quality range. We can make use of this fact by pre-adapting services to low-precision positional information instead of trying to pinpoint the user's position more precisely (cf. e. g. [18]). This method represents an alternative to resource-adaptive positioning. However, pre-adaptation restricts the system to a given quality of service, and thus prevents it from making full use of better positional information.

6 Conclusion

Location-enhanced services are poised to become the first real-world example of ubiquitous computing and ambient intelligence. With so many different location technologies available it is becoming important to develop generic frameworks for combining and integrating them into one coherent system. This is especially true if one considers the enormous range and variety of available positioning technologies.

When developing a positioning service for ambient environments we must aim to maximise service availability in order to minimise user disruption. To achieve this goal we developed a framework for adaptive positioning that dynamically and seamlessly combines three positioning methods: measurement-based positioning, inference-based positioning, and interactive positioning. The framework enabled us to implement an adaptive positioning service that guarantees that positioning information is available as long as one of the underlying methods is able to deliver reliable data. To enable applications (and application designers) to influence how adaptation is performed, the service provides a well-defined quality-of-service interface for specifying the required quality of positioning information. We demonstrated the feasibility and utility of this approach by describing a prototype implementation and case study in the context of a mobile tourist guide.

The work presented in this paper can be extended by making use of anytime algorithms (cf. e. g. [19]), allowing for an even finer level of control. We expect that this extension will result in a system, where an application can assign a certain amount of resources to positioning, and where the quality of the resulting location hypothesis directly depends on the amount of resources.

Acknowledgements

The work presented in this article was mainly funded by the Klaus Tschira Foundation under the grants SISTO and SPACE. Additional funding was provided by the EPSRC through the RELATE project.

References

[1] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56, 2001.

[2] H. Anegg, H. Kunczier, E. Michlmayr, G. Pospischil, and M. Umlauft. LoL@: designing a location based UMTS application. *Elektrotechnik und Informationstechnik*, 119(2):48–51, 2002.

[3] Jörg Baus, Antonio Krüger, and Wolgang Wahlster. A resource-adaptive mobile navigation system. In Yolanda Gil and David B. Leake, editors, *IUI 02 – 2002 International Conference on Intelligent User Interfaces*, pages 15–22, San Francisco, California, 2002. ACM Press.

[4] Andreas Butz, Jörg Baus, Antonio Krüger, and Marko Lohse. A hybrid indoor navigation system. In *Proceedings of IUI 2001*, pages 25–32, New York, 2001. ACM Press.

[5] Keith Cheverst, Nigel Davies, Adrian Friday, and Christos Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of CHI 2000*, pages 17–24, Netherlands, 2000.

[6] Keith Cheverst, Nigel Davies, Keith Mitchell, and Adrian Friday. Experiences of developing and deploying a context-aware tourist guide: the GUIDE project. In *Mobile Computing and Networking*, pages 20–31, Boston, U.S.A., 2000.

[7] Klaus Fischer. Holonic multiagent systems – theory and applications. In *Proceedings of the 9th Portuguese Conference on Progress in Artificial Intelligence (EPIA-99)*, volume 1695 of *LNAI*, pages 34–48. Springer, Berlin, Heidelberg, New York, 1999.

[8] Dieter Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. Bayesian filtering for location estimation. *Pervasive Computing Magazine*, 2(3):24–33, 2003.

[9] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[10] Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In Nigel Davies, Elizabeth D. Mynatt, and Itiro Siio, editors, *Proceedings of Ubicomp 2004*, pages 88–106, Berlin, Heidelberg, New York, 2004. Springer.

[11] Jeffrey Hightower, Barry Brumitt, and Gaetano Borriello. The Location Stack: A Layered Model for Location in Ubiquitous Computing. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, pages 22–28, Callicoon, NY, USA, 2002.

[12] Christian Kray. *Situated Interaction on Spatial Topics*. PhD thesis, Computer Science Department, University of Saarland, Saarbrücken, Germany, 2003.

[13] Christian Kray and Gerd Kortuem. Interactive positioning based on object visibility. In Stephen Brewster and Mark Dunlop, editors, *Proceedings of Mobile Human-Computer Interaction 2004*, pages 276–287, Berlin, Heidelberg, New York, 2004. Springer.

[14] Bu-Sung Lee, Wentong Cai, Stephen J. Turner, and L. Chen. Adaptive dead reckoning algorithms for distributed interactive simulation. *International Journal of Simulation*, 1(1–2):21–34, 2000.

[15] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. Rapid prototyping of mobile context-aware applications: The Cyberguide case study. In *Mobile Computing and Networking*, pages 97–107, 1996.

[16] Rainer Malaka and Alexander Zipf. Deep Map - challenging IT research in the framework of a tourist information system. In D. R. Fesenmaier, S. Klein, and D. Buhalis, editors, *Information and communication technologies in tourism 2000*, pages 15–27. Springer, Berlin, Heidelberg, New York, 2000.

[17] B. D. Noble, M. Price, and M. Satyanarayanan. A programming interface for application-aware adaptation in mobile computing. *Computing Systems*, 8(4), 1995.

[18] Sanjay Sood, Kristian J. Hammond, and Larry Birnbaum. Low-fidelity location based information systems. In Nuno Jardim Nunes and Charles Rich, editors, *Proceedings of IUI 04*, New York, NY, USA, 2004.

[19] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI Magzine*, (17):73–83, 1996.