2 A Formal Model for Mobile Map Adaptation

Martin Raubal[1] and Ilija Panov[2]

[1] Department of Geography, University of California, Santa Barbara, CA, USA
[2] Institute for Geoinformatics, University of Münster, Germany

Abstract

Computing has become increasingly mobile and pervasive, which implies that services must be aware of and adapt to changing contexts in highly dynamic environments. Services that require a lot of user interaction have less potential of being used, because they tend to be obstructive. Thus, context-awareness and adaptation are important research issues in the area of mobile computing. A major goal is to minimize user interaction through service adaptation, and to provide context-sensitive and personalized information to the user. Adaptations for mobile map applications must consider a wide range of factors – from technical requirements to cognitive abilities and goals of the user. However, specifying contextual facts in an accurate and traceable manner is challenging. Initial approaches have focused on information visualization for mobile map applications through context information. These typically focus on simplifying and generalizing route segments rather than adapting to personal information. In this paper we propose a formal conceptual model for automatic mobile map adaptation that can be employed for different applications, such as pedestrian navigation. This model is composed of three components – a *context* model, a *user* model, and a *task* model. Through specified adaptation operations it aims at a reduction of both the user interaction with the service and the cognitive load for the user.

Keywords: mobile computing, location-based services, mobile maps, adaptation, context-awareness, formal specifications

2.1 Introduction

Computing has become increasingly mobile and pervasive, and the emerging technologies provide 'anytime/anywhere' information. These changes imply that applications and services must be aware of and adapt to changing contexts in highly dynamic environments. A mobile user is potentially more distracted, and different constraints and limitations exist, such as small display, and limited energy and bandwidth. Users often need to make decisions on the spot and therefore require current personalized and context-sensitive information on their mobile devices, i.e., 'the presentation [...] must be conditioned by the users' activities and by the state of the world around them.' (Lake 2001, p.1) A prime example is pedestrian navigation. Finding ways for services to adapt appropriately within a wide range of possible user situations in order to best support human–computer interaction has been identified as an important research problem (Dey & Abowd 2000). It has been pointed out though that research on accurately discovering and efficiently disseminating contextual information is still at an early stage (Strang & Linnhoff-Popien 2004). Thus, context-awareness and adaptation comprise key research topics in the area of mobile computing and location-based services (LBS) (Raper, Gartner, Karimi, & Rizos 2007).

The user and her activities in a particular context define the amount and detail of necessary information, the degree of generalization, and the way such information is visualized on a mobile map. First attempts of adapting visualization for mobile services have been described in (Zipf 2002). Maps are of great value for people as they have the potential to represent large amounts of information about an area of interest within a single frame in a comprehensible form. Examples of where maps are useful for pedestrians range from searching for points-of-interest (POIs) to navigating in unfamiliar environments. These different tasks and circumstances require a large amount of user interaction, such as changing program settings and receiving personalized information. Therefore services have less potential of being used, because they tend to be obstructive. Consequently, a major goal in the field of mobile computing is to minimize user interaction through service adaptation, and to provide context-sensitive and personalized information to the user in a changing environment. In this paper, we propose an abstract formal model for mobile map adaptation, which takes these issues into account. Based on a number of adaptation operations this model aims at reducing both the user interaction with a service and the cognitive load for the user.

Section 2.2 presents a use case and derives current problems with respect to user interaction with mobile LBS. In *Section 2.3* we discuss previous work on context and adaptation. *Section 2.4* develops an abstract conceptual model for mobile map adaptation based on three components – a context model, a user model, and a task model. In *Section 2.5,* this model is applied to a LBS for pedestrian navigation by formally specifying the model components and their operations. *Section 2.6* compares the formal model to the use-case service. The final section presents conclusions and directions for future work.

2.2 Use Case and Problem Statement

This section introduces a use case for mobile map adaptation and identifies current problems regarding user interaction with mobile LBS.

2.2.1 Scenario

The presented scenario is based on previous work regarding prototypical implementations of a user-oriented pedestrian navigation service[1] *(utopian)* and a *HotelFinder* service (Rinner & Raubal 2004). *Utopian* is a LBS for recreation facilities and gastronomy offers combined with a navigation service for pedestrians. LBS assist users in the performance of spatio-temporal tasks and provide location-dependent information.

Alice visits the city of Münster for the first time and wants to stay for a few days. She has arrived at the train station and starts the *HotelFinder* software on her mobile device. This service supports Alice in finding a suitable hotel. After the decision-making process with the *HotelFinder, utopian* starts to navigate Alice from her current position to the chosen hotel. This navigation service provides a series of pictures with landmarks of every decision point along the way. Brief written instructions provide Alice with directions of turns. On her way through the city Alice passes several historical buildings. She is very interested in historical monuments, buildings and places, and therefore wants to get some information about them. The navigation tool does not support any kind of information retrieval beyond the pure navigation task. Therefore, Alice has to start a standard search engine to get more information. Continuing her way to the hotel, Alice receives the instruction to turn right at the next landmark. She reaches a place with a large building. The large and scattered environment confuses Alice and she is unsure what ‘right at the building’ exactly means. To get an overview of the current situation Alice has to switch to the digital map to verify her current position and look for the direction of the next decision point. This map always provides Alice an overview of the complete route from her starting point to the destination. The map scale does not deliver a realistic impression of her current location on the map. She has to manually zoom in to get a detailed view of the location and then zoom out to get back to the route overview.

[1] http://utopian-online.de

Alice feels uncomfortable when following the navigation instructions, because she must orient herself several times. This is due to the fact that the map is always oriented to the North direction and not aligned with her current walking direction.

2.2.2 Problems

Based on the described use case, we identify several problems that occur during user interaction with mobile LBS regarding map extent and alignment, zooming, personalized information, and time. These issues have also been recognized by others (Radoczky 2003, Wealands, Miller, Benda & Cartwright 2007).

Problem 1: Map Alignment with North Direction

Most maps are aligned with the North direction. When using an analog map, users often turn the map around every few minutes or at every direction change, because this facilitates orientation (Nivala & Sarjakoski 2003, Wealands et al. 2007). This is also true when using digital maps on mobile devices, which are aligned with the North direction *(Figure 2.1)*. Therefore, one of the most useful pieces of context information is the user's walking direction, which could be measured by the Global Positioning System (GPS) or different types of sensors (Baus & Kray 2002). 86% of the participants in the work by Radoczky (2003) stated that a track-up oriented map is indispensable.

Fig. 2.1. Initial map extent after starting *utopian* with North alignment.

Problem 2: Manual Zooming and Static Map Extent

When using *utopian* the user has to constantly change the map scale to get an overview of a larger area or a detailed view of an area of interest. In order to do this, the user must interact with the device by spanning a rectangle with the stylus on the screen. Some experience and skills are needed to get the correct map scale with this technique. In general, it is not known how to fit maps on small screens and which technique is best (Gutwin & Fedak 2004). Therefore, zooming at decision points is one important aspect of the model presented in our work. When the user reaches a decision point an automatic zooming function delivers a detailed view. In the survey done by Radoczky (2003), 45% of the participants stated this function as indispensable.

After the route calculation, an overview map for a general presentation of the whole route has been found vital by 64% of the participants in the survey by Radoczky (2003). But this overview map of the calculated route should only serve as a starting point at the beginning of the guiding process. In our use case the service constantly delivers the overview map with the entire route. The user must therefore manually zoom in and out to get more detailed views and to get back to the entire route. This results in a large amount of user interaction with the device. Additionally, a small-scale map results in a high cognitive load for the user. A smaller viewable map extent would reduce the cognitive load, because it reduces the amount of visualized information. The following map extent should depend on the velocity of the user, so that the actual map extent shows the area that the user can reach within a certain amount of time.

Problem 3: Visualization of Personalized Information

In *utopian* the user gets information about the POIs the service calculates for the tour. This means that the user only gets information about locations she is interested in for the current tour. Such information about short-term interests is not visualized automatically, but through direct interaction with the context menu of the service. The user has no facility to determine and set personal long-term interests, e.g., through preference settings via a user profile.

Problem 4: Daytime-Independent Landmark Visualization

People make use of salient objects in the environment to orient themselves and navigate through space (Denis, Michon & Tom 2006, Lovelace, Hegarty & Montello 1999). *Utopian* provides landmark-based navigation instructions using point landmarks at decision points. 73% of the survey participants in (Radoczky 2003) voted for multi-encoded navigation instructions, in particular the integration of landmark

Fig. 2.2. Same facade by day and night. Illumination at night increases visual attraction.

photographs in case of decision points. Raubal & Winter (2002) provided a formal measure to specify landmark saliency of buildings and mention visibility as one of the components. The visibility of buildings and other kinds of salient objects is different for day and night (Winter, Raubal & Nothegger 2005). Whereas some facades have low visual attraction during daytime, their visual attraction and saliency increases when illuminated at night *(Figure 2.2)*. *Utopian* does not support this functionality of switching landmarks depending on the time of day.

2.3 Related Work

This section reviews *context* and *adaptation* from the perspective of geospatial mobile applications. The ability of services to use context information allows for the adaptation of the available information in order to generate a benefit for the user.

2.3.1 Context

The dynamic changes of service and user states cause a change in context, and therefore context-awareness is an important factor in mobile computing. Several definitions for mobile computing regard context as the changing execution environment, which is divided into *computing* (e.g., computation resources), *user* (e.g., social situation), and *physical environment* (e.g., weather). Dey & Abowd (2000, pp.3–4) presented a more generic context definition for ubiquitous and mobile computing: ‘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 applications themselves.’ It offers two important advantages for designing a context model for the pedestrian navigation task described in *Section 2.2.1.* First, if a piece of information can be used to characterize the situation of the user or task, then this information is context. Location and time are examples for useful information within the model. Second, the definition allows for context to be either explicitly or implicitly indicated by the user. For example, the location of nearby objects can be detected implicitly by the service or explicitly through user input. The main aspects in the above definition are identity (user), activity (interaction with service and environment), location, and time (as temporal constraints). These aspects build the basis for *context-aware computing* – 'the ability of a mobile user's applications to discover and react to changes in the environment they are situated in.' (Schilit & Theimer 1994, p.3)

2.3.2 Adaptation

One of the goals of LBS should be to take up as little of the user's attention as possible. There are mechanisms available to fit the services to the current situation and make them flexible, called *adaptation.* Adaptation is therefore the answer to a changing context (Reichenbacher 2003). There are two ways of achieving knowledge sharing between a service and its user – making the service *adaptive* or *adaptable* (Fischer 1993, Oppermann 1994) *(Table 2.1).* One of the dominant factors for adaptation is the user's task. All relevant factors need to be formally represented within the service.

Table 2.1. Characteristics of adaptive and adaptable services, modified from (Fischer 1993).

	Adaptive	Adaptable
Definition	dynamic adaptation by the service to current task and user	user changes functionality of the service
Strengths	little (or no) effort by the user	user is in control
Weaknesses	loss of control	user must do substantial work

The visualization of geoinformation and its interactive use on mobile devices is adapted to either one or all components of the actual context (user, location, information, etc.) (Gartner 2004). However, the visualization does not need to adapt to all factors at once. Zipf (2002) argues that when adapting maps for mobile services it is insufficient to focus only on technical parameters such as device characteristics, but maps have to be dynamically generated according to a wider range of variables, including user preferences, task, and location. To achieve this goal the service needs to exploit a user model and context knowledge. Different tasks have different requirements regarding map design. While it is important for an overview map to show many features (but not necessarily in great detail), a route map must display important decision points or landmarks (Agrawala & Stolte 2001). Incorporating adaptation within the visualization process solves several usability problems encountered in the mobile environment.

2.4 A Conceptual Model for Mobile Map Adaptation

This section presents a conceptual model for mobile map adaptation, which will be applied to the use case of navigation and formally specified in *Section 2.5.* The adaptation model is designed in an abstract way to be used for different tasks, constraints, and requirements in the domain of mobile LBS.

2.4.1 Design Considerations

Our main hypothesis in this research is that *a formal model for mobile map adaptation predicts a reduction of user interaction and cognitive load during location-based tasks.* Different design decisions have to be made with respect to the following questions:

- What is the user's task?
- What are the user's requirements, needs, and preferences?
- Which context features are needed for the task and satisfy the user's preferences?
- Which kinds of operations are needed, so that the user can successfully accomplish the task?

Reduction of Cognitive Load

Cognitive load can be understood as the amount of work needed to acquire and use information. In the case of mobile pedestrian navigation services this corresponds to the visualized information and navigation instructions displayed on the screen. The *cognitive load theory* (CLT) offers designers of different services a way of assessing and affecting some critical components during the design process of digital maps (Bunch & Lloyd 2006). Several aspects of CLT were implemented in the area of mobile digital maps (Mayer & Moreno 2002). These techniques concentrate on the reduction of the visualized information. The pitfall in this information reduction is that it stands in contrast to adaptation methods, because the key to successful adaptation is to collect as much information about the user and her environment as possible (Hampe & Paelke 2005). Therefore, the challenge in the reduction of cognitive load is to find the appropriate amount of information, i.e., determining the right *level of detail* (LoD). Here, this LoD will be achieved through a task-driven adaptation model. The model and its components will be filled with features required for the specific task.

Reduction of User Interaction with the Device

The usability and usefulness of mobile map services is highly dependent on the appropriate *graphical user interface* (GUI) design including the visualization of spatial and non-spatial information. The visualization of the different elements is constrained by the limited resolution and small display size, therefore the GUI design on a small display must balance space requirements of both a map and a set of tools (Rinner, Raubal & Spigel 2005). Further constraints are imposed through the limited processing power and low resolution of pointing devices. Methods and techniques for GUI design for mobile devices have been proposed by Cartwright et al. (2001). Here, we focus on minimizing the use of pointing devices to achieve a reduction of user interaction.

2.4.2 Adaptation Model

The *AdaptationModel* (*A*) is composed of three submodels – the *ContextModel (C),* the *UserModel (U),* and the *TaskModel (T).* These models are classified into dynamic and static elements *(Figure 2.3).* The *ContextModel* represents the dynamic elements of the model, because in most cases the *Situation (S)* of using a mobile device implies that the surrounding context changes (e.g., the user's position), whereas the user and the task remain the same (Zipf & Jöst 2006).

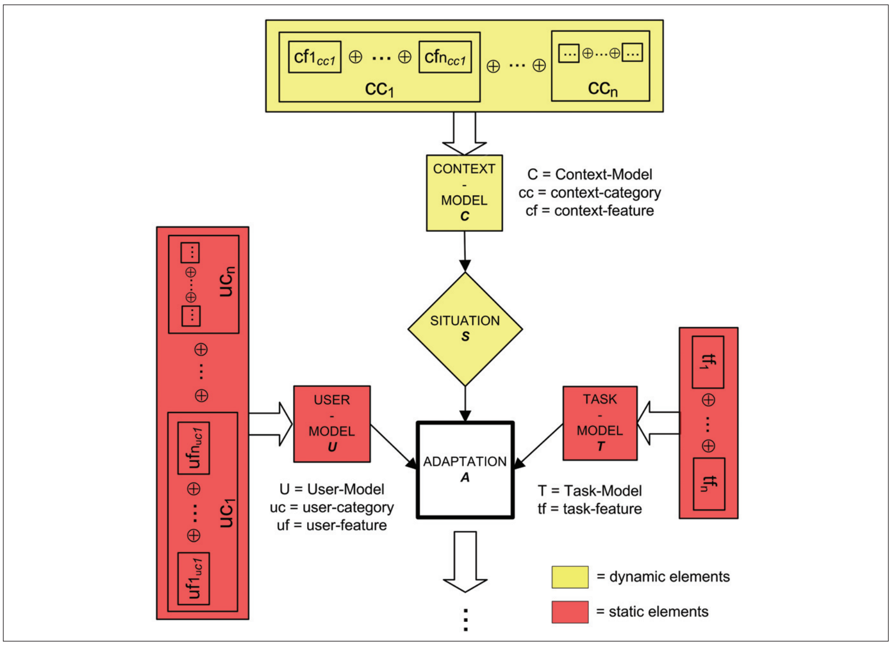

Fig. 2.3. Abstract *AdaptationModel* for modeling context-sensitive and user-centered LBS.

The *ContextFeatures (cf)* underlie dynamic actions and are always available in an explicit form, because they can be sensed by the device (e.g., position through GPS) or derived through other *ContextFeatures.* Their values change each time a *Situation* changes. In contrast to the *TaskModel,* both the *ContextModel* and the *UserModel* consist of different categories, and each category consists of an arbitrary number of features (e.g., the *ContextModel* consists of a *context-category cc1* with features *cf1,...,cfn)* (Schmidt, Beigl & Gellersen 1999). The categories serve as structural units to classify the features of the model. The values of these features serve as input parameters for the different types of operations, which can be either adaptive or adaptable, and strongly depend on the specific task.

The ContextModel

The *ContextFeatures* in the *ContextModel* describe the user's current situation with its various characteristics corresponding to the real world. With regard to the context definition given in *Section 2.3.1* this model should cover the two most important features, i.e., position and time, supplemented by further useful features such as direction of movement. The *Situation (S)* in *Figure 2.3* can be conveyed as the user being situated in a dynamic environment described by the *ContextFeatures* $(S = (cf_1, ..., cf_n))$ (Reichenbacher 2003). The ⊕ symbol represents a linking between several *ContextFeatures* (see *Figure 2.5* for examples). More available features make a more detailed description of the user's *Situation* possible. To manage the amount of possible and useful *ContextFeatures,* the *ContextModel* is classified into categories $(C = (cc_k, cf_l)$ *where* $1 \leq k, l \leq n)$. The classification into different categories is particularly useful when using a large number of required features for the task.

The UserModel

The service's representation of the user is incorporated through a *UserModel* that describes the user with predefined information about her preferences. This information is represented by the *UserFeatures* $(uf_1, ..., uf_n)$. These features represent all characteristics, which fall under the identity–category of the context definition *(Section 2.3.1).* The model should capture different types of information, such as user needs, preferences, and interests. The ⊕ symbol stands for a linking between several *UserFeatures,* which are organized into categories $(U = (uc_i, uf_j)$ *where* $1 \leq i, j \leq n)$. The different features representing the *UserModel* are static elements, because it is unlikely that they change during the navigation process (e.g., a user's preference for historical buildings).

Another approach to user modeling involves detecting patterns in their behavior (Zipf & Jöst 2006). This is a complex approach based on artificial intelligence and ubiquitous computing. The outcome of the current work focuses on a complementary approach where the service designer decides which changes in the *Situation* should lead to service adaptation based on available *ContextFeatures* and the predefined user information (Göker & Myrhaug 2002). It is a personalization approach where the service lets the user specify her own settings for how it should behave (Barkhuus & Dey 2003); e.g., the service designer specifies different recreation types or categories and the user chooses her favorite ones[2].

The TaskModel

One of the most challenging parts of the *AdaptationModel* is accounting for the user's purpose of using the map. The task mainly affects the determination of adaptive and adaptable operations based on the *ContextFeatures* and *UserFeatures.* To clarify the relationship between task and operations an appropriate approach is needed. The structure of the model in *Figure 2.4* is a simplification of the activity theory for cartography (Dransch 2002). It is a hierarchical framework where the activity builds the root element supplemented by goals, subgoals, and actions to accomplish the different activities. To reach the goal, several activities must be performed, which comprise the interactions of the user with the environment and the service. Hence, the operations depend on the specific task with its activities. Take, for example, a simple wayfinding task from a starting point A to a destination B. The goal is to reach B. An action could be that the user has to orient at a decision point and find the correct walking direction as a subgoal. The actions are represented by the different operations, which are either adaptive or adaptable.

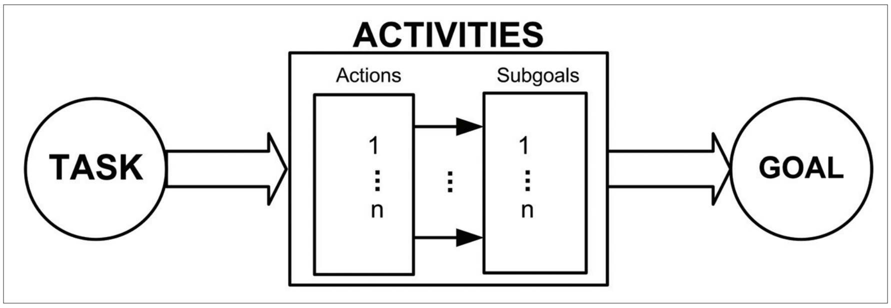

Fig. 2.4. Task-dependent activities to reach the goal. Simplified *activity theory* for cartography (Dransch 2002).

The representation of the *TaskModel* within the *AdaptationModel* is similar to the other submodels. The *TaskModel* is a static element of the *AdaptationModel,* but in contrast to the former two, a combination or selection of several *TaskFeatures* *(tf)* is not allowed *($T = (tf_n)$ where $n = 1$).* This means that the $\oplus$ symbol should be read as a XOR-operator (exclusive disjunction: either tf_n or tf_{n+1}*).* This makes the definition of operations – and the required *ContextFeatures* – less complex than for an entire application such as *utopian.*

[2] See http://www.heidelberg-mobil.de for examples.

2.5 Application and Formal Model

This section demonstrates the applicability of the conceptual model for mobile map adaptation using the scenario of a pedestrian navigation service. The main focus lies on the formal specification of the adaptation operations. Our method of formalization uses algebraic specifications, which have proven useful for specifying data abstractions in spatio-temporal domains (Frank 2000, Raubal & Kuhn 2004). Entities are described in terms of their operations, depicting how they behave. The tool chosen here is Hugs, a dialect of the purely functional language Haskell (Hudak 2000). The result is a formal model that can be used as a basis for implementing mobile map adaptation for pedestrian navigation.

2.5.1 Adaptation Model

The specification of the adaptation model follows the conceptual model introduced in *Section 2.4.2.* We assign explicit features to each of its components regarding context, user, and task *(Figure 2.5).*

The pedestrian passes through different *Situations* during the wayfinding process, such as different decision points. These are described exclusively by the *ContextModel.* Therefore, the operations depend mainly on the explicit *ContextFeatures.* The *ContextFeatures* are subsumed in the spatio-temporal *context-category.* For the adaptive and adaptable operations the spatio-temporal category contains the features *Position, Time, Velocity,* and *Direction.* The ability to determine the *Position* of a mobile device is a direct requirement for every LBS. Nowadays, many mobile phones have integrated GPS modules, which provide accuracies from a few meters in stand-alone mode to sub-meter in differential mode (DGPS) (Gartner 2004). A major limitation of current LBS is that they do not consider temporal properties (Raubal, Miller & Bridwell 2004). The closest Café, for example, may not be open. *Time* can be represented at different scales. Here, we focus on time of day because it is the main parameter for deciding whether to visualize day- or nighttime pictures of landmarks. This context information can be obtained automatically by the built-in clock of the mobile device. *Velocity* is defined as the rate of positional change and can be calculated based on GPS positions in time. It is an important feature for calculating the current map extent.

The majority of people who use a mobile map rotate the device while walking (Schmidt et al. 1999). This is due to the fact that most mobile map services do not support an automatic track-up orientation of the digital map. Constantly orienting the current viewable map to the direction of travel may be confusing to the user and requires permanent attention. Therefore, we argue that the track-up orientation of the mobile map should depend on the projected walking *Direction* from one deci-

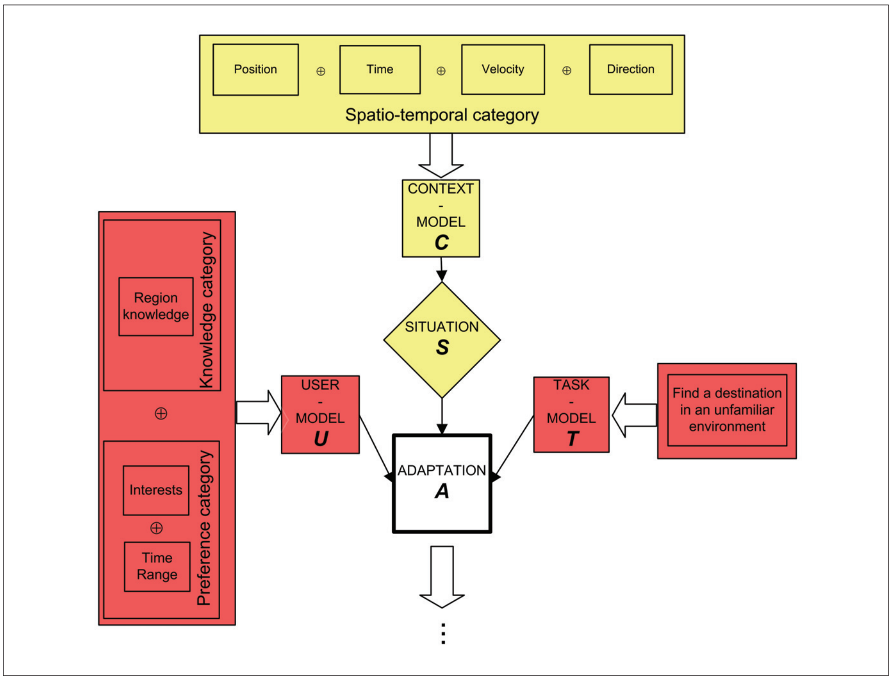

Fig. 2.5. The *AdaptationModel* for pedestrian navigation services.

sion point to the next. Such direction can be measured by determining the user's current position and the next decision point to be reached.

One of the most difficult characteristics to be interpreted and modeled is the user. Users vary with regard to physical abilities, cognitive and perceptual abilities, as well as in terms of personality differences (Shneiderman 1992). The service could automatically employ various pieces of user information to improve navigation instructions. Here, we consider information about the user's region knowledge and her interests, both classified in separate categories within the *UserModel (Figure 2.5)*. Even if people are unfamiliar with a particular environment, they may have conceptual representations about the location type (e.g., general structure of urban areas). The *Preference* category contains the features *Interests* and *TimeRange. Interests* determine the kinds of nearby objects to be visualized on the map, such as historical buildings or other attractions. This work focuses on the long-term interests (Zipf & Jöst, 2006), which do not change during the guiding process. *TimeRange* is responsible for the determination of the LoD *(Section 2.4.1)*. It enables the user to manipulate a parameter for both the *MapExtent* and the *Zooming* operations *(Section 2.5.2)*. The user can determine the preferred *TimeRange* (e.g., 10 minutes) and thereby the viewable map extent of the map. Similarly, the user can set the *TimeRange* for the *Zooming* operation (e.g., 1 minute) to predefine the local detail at decision points.

2.5.2 Adaptation Operations

The system designer is responsible for deciding which operations are required (task–dependent) and how they should behave (adaptive or adaptable). The determination of the *ContextFeatures* and *UserFeatures* for each operation is shown in *Table 2.2*. The *TaskFeatures* are not listed as input parameters for the operations; although they affect the kinds of operations, they do not deliver explicit features as input parameters for them. *Figure 2.6* gives an overview of the adaptive and adaptable operations.

Table 2.2. Required *ContextFeatures* and *UserFeatures* to formalize the suggested operations.

	ContextFeature				**UserFeature**
Operation	Position	Time	Velocity	Direction	TimeRange
Zooming	X				X
MapExtent	X	X	X		X
POIVis	X				
TrackUp	X			X	
LandmarkSwitching		X			

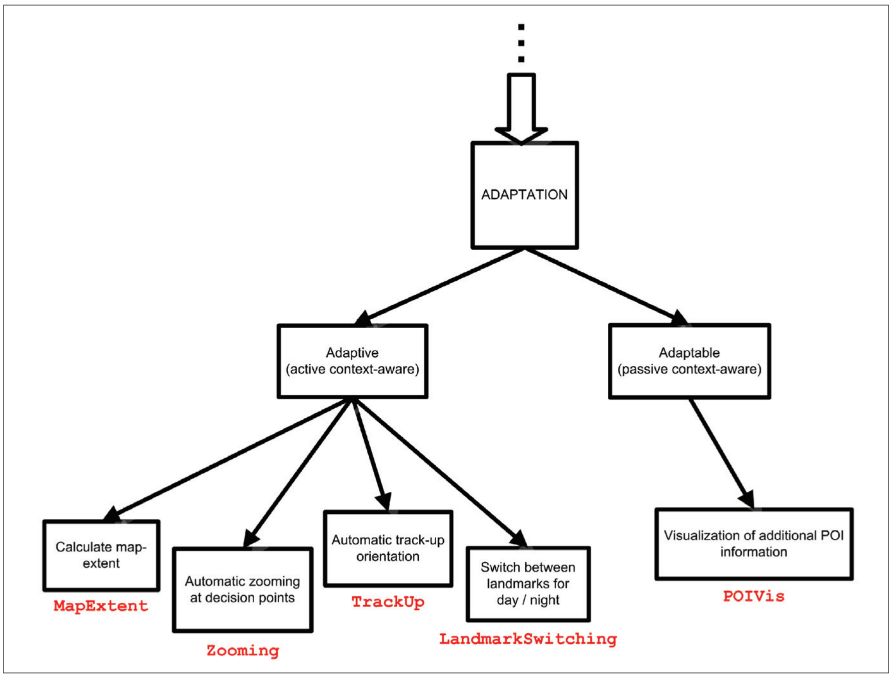

Fig. 2.6. Operations for mobile map adaptation in pedestrian navigation services. Automatic zooming at decision points.

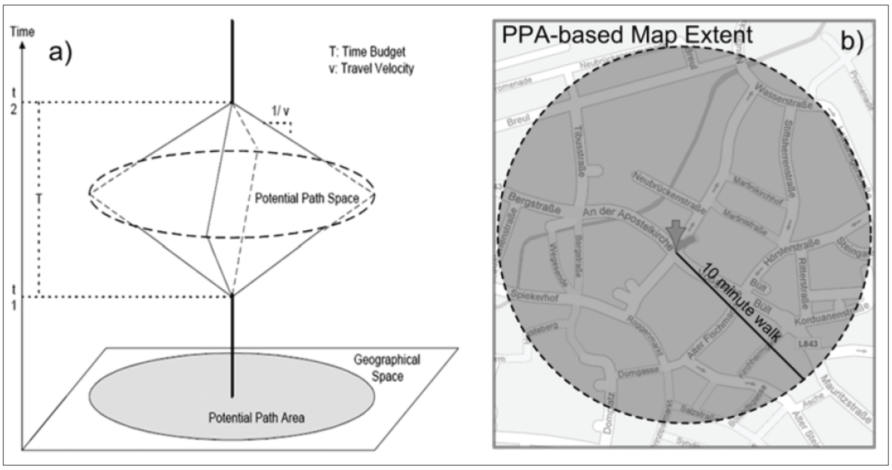

Fig. 2.7. a) Potential Path Space and Area. b) Example for a viewable map extent on a mobile device derived from the PPA.

Automatic Zooming at Decision Points

The *Zooming* operation provides the user with smooth multiple scaling at decision points. When reaching a decision point the service automatically zooms to local detail, predefined through the *TimeRange.* The small displays of mobile devices necessitate the use of different map scales ranging from overview to local detail. Additionally, a landmark will be displayed in a separate window. The correct direction can be represented through an arrow on the map or by written instructions.

Calculate Map Extent

Previous investigations have considered parameters for the selection and generalization of visualized information, but not for map scale and viewable extent (Hampe & Elias 2004). In this work the calculation of an appropriate map extent is essential, because other operations such as *POIVis* are directly related to it. The calculation of an appropriate map extent is necessary after leaving the start point of the route, while walking between decision points, and before arriving at the destination. It depends on the traveler's current *Velocity* (e.g., 5 km/h), her current *Position,* and a *TimeRange* (e.g., 10 minutes). Time geography (Hägerstrand, 1970) offers a way to calculate the map extent: the Potential Path Space (PPS) delimits all locations in space and time that an individual can possibly occupy, assuming some travel velocity. The Potential Path Area (PPA) results by projecting the PPS to the two-dimensional geographic plane (Miller 1991) *(Figure 2.7).*

Automatic Track-Up Orientation

The *TrackUp* operation delivers a map visualization depending on the walking direction *(Section 2.2).* Therefore, the viewable map extent is aligned to the direction between two decision points but not to the viewing direction. This allows for orienting the map in a way that may be easier to interpret by the user.

Distinction Between Landmarks for Day and Night

Because the visibility of salient features varies according to time of day (e.g., day/night), and because landmarks are so important in human wayfinding, a time-dependent distinction between landmarks is required. Each landmark is linked to a decision point. Every node consists of at least two landmarks, one for day and one for night. Depending on the current time, the appropriate landmark will be displayed.

Visualization of POI Information

The *POIVis* operation is the only adaptable operation. It informs the user about nearby POIs, but does not automatically visualize additional information about them. The difficulty in developing this operation results from the meaning of ‘nearby’. The location of nearby objects falls in the category of proximate selection (Schilit, Adams & Want 1994). It involves entering two variables, the ‘locus’ (user’s current *Position)* and the ‘selection’ *(Interests* as nearby objects). To overcome and simplify the issue of what ‘nearby’ means to the user, the visualization of *Interests* as nearby objects depends on the calculated map extent. Using this approach, we follow the definition by Schilit et al. (1994): in context-aware services the most usefully located objects are close at hand, either co-located or requiring a short time to get to. The visualization of POIs that are close at hand depends on the area the user can reach during the predefined *TimeRange.* POIs are represented in the current map extent through icons. The user can click on the icon and receive the requested information.

2.5.3 Formalization

The *AdaptationModel* is formally represented[3] as a data type, which is constructed from different types. *Edge* represents the *Direction* of the *ContextModel,* because of the track–up orientation of the digital mobile map along the *Edge.* We consider the *UserFeature* region knowledge an implicit feature that does not directly affect

[3] The complete Hugs code is available at http://www.geog.ucsb.edu/~raubal/Downloads/MobileMapAdaptation_Hugs.rar. Hugs interpreters can be downloaded freely from http://www.haskell.org.

the formalization process. Because the perceived distances shrink when the user knows the area better (Zipf 2002), it is left to the user to define the two *TimeRanges* to receive appropriate map extents. The *MapState* is specified through three components. *ViewableExtent* changes at every *Node* (zooming) and for the calculation of the map extent. *Angle* represents the orientation of an *Edge.* Also, the appropriate *Landmark* for each *Node* is represented. The constructor of the *UserModel* gets the type *TimeRange* twice, representing the parameters for the *MapExtent* and the *Zooming* operations.

```
data AdaptationModel =
AdaptationModel ContextModel UserModel TaskModel MapState
  data ContextModel = ContextModel Position Edge Time Velocity
  data UserModel = UserModel Interests TimeRange TimeRange
  data TaskModel = TaskModel Task
  data MapState = MapState ViewableExtent Angle Landmark
```

The wayfinding environment is formally specified as a graph with *Nodes* and *Edges,* denoting decision points and transitions between them. The data type *Environment* is constructed from a list of *Edges.* Every *Edge* is constructed from two *Nodes* and an *Angle,* which provides the *Direction* of the *Edge.* Every *Node* has a *Position* represented by geographic coordinates and a list of *Landmarks.* These are constructed from a *Name* as an identifier and a specified *TimeSpan* to allow for a distinction between day and night.

```
data Environment = Environment [Edge]
  data Edge = Edge Node Node Angle
  data Node = Node Position [Landmark]
  data Landmark = Landmark Name TimeSpan
  data TimeSpan = Day | Night
```

The abstract type signatures for the operations are implementation independent and can be implemented for different types of pedestrian navigation services. In the following, the operations will be implemented for the data types *AdaptationModel* and *Environment* as presented above.

The *Zooming* operation represents the changing of viewable map extent to local detail at decision points. For this implementation it is the same as the *MapExtent* operation. Applying the function has the following effects:

1. The operation checks whether the destination has been reached (this is initially done by all operations and because the code is the same, we only represent it here). If yes, then the wayfinding task is completed.

2. The changes in the *ViewableExtent* depend on the current *Position*, *Velocity,* and *TimeRange*. The function *getTimeRangeAtNode* determines the extent of local detail defined by the user.

```
instance AdaptationModels AdaptationModel where
  zooming environment (AdaptationModel cm um tm (MapState ve a lm))
     = if isDestination (Node (getPosition cm) (Landmark “default” Day:
     [Landmark “default” Night])) environment
     then error (“The destination is reached.”)
     else (AdaptationModel cm um tm (MapState veChange a lm)) where
        veChange = ViewableExtent (getPosition cm) (getVelocity cm)
        (getTimeRangeAtNode um)
```

The *TrackUp* operation delivers a map visualization depending on the calculated walking *Direction* between decision points. The viewable map extent is aligned to the *Direction* between two decision points along an *Edge.* The function *getCurrentEdge* provides the *Edge* related to the current *Position* of the user. The *Angle* of the map will be replaced by the orientation of this *Edge.*

```
instance AdaptationModels AdaptationModel where
  trackUp environment (AdaptationModel cm um tm (MapState ve a lm))
     = if isDestination …
     else (AdaptationModel cm um tm (MapState ve aChange lm)) where
        aChange = getOrientation (getCurrentEdge cm)
```

The *LandmarkSwitching* operation enables a time-dependent extraction of landmarks for a decision point. The operation checks which *Node* is related to the current *Position* of the user. The *getTime* operation delivers the current *Time,* so that the required *Landmark* can be extracted.

```
instance AdaptationModels AdaptationModel where
  landmarkSwitching environment (AdaptationModel cm um tm
     (MapState ve a lm))
  = if isDestination …
  else (AdaptationModel cm um tm (MapState ve a lmChange)) where
     lmChange = getRecentLandmark (getNodeAtPosition environment
        (getPosition cm)) (dayOrNight (getTime cm))
```

The *POIVis* operation gives the user information about nearby POIs. The *Interests* of the user are visualized using the preselected *Interests,* which are retrieved by the *getInterests* function. The *getVisablePOIs* function uses an auxiliary function *(liesWithin)* to check whether *Nodes* are within the current map extent. The operation retrieves those *Nodes* in the list that are POIs and returns them.

```
instance AdaptationModels AdaptationModel where
  poiVis environment (AdaptationModel cm um tm (MapState ve a lm))
  = if isDestination …
  else (AdaptationModel cm um tm (MapState ve a lm)) where
     poiChange = getVisablePOIs (getInterests um)(MapState ve a lm)
```

2.6 Discussion

The reduction of user interaction with the device is mainly achieved through the *Zooming, MapExtent,* and *POIVis* operations. In order to quantify the effects of the model, we compare and measure the user interactions for *utopian* and the applied *AdaptationModel.* Interaction is represented through a pointing device such as a stylus. User interaction for both is measured by calculating/counting the individual clicks. If a survey were performed, one could also measure the interaction time by using methods such as the one based on the keystroke-level model (Haunold & Kuhn 1994). Our comparison consists of three actions, which are performed with *utopian.* These actions correspond to the formal operations of the *AdaptationModel. Table 2.3* gives an overview of the results.

Table 2.3. Comparison of *utopian* and the applied *AdaptationModel* regarding user interaction.

Operation	AdaptationModel	Utopian
Zooming	no interaction needed	minimum 4 clicks
MapExtent	no interaction needed	minimum 4 clicks
POIVis	1 click	3 clicks

Section 2.2 has shown that in *utopian* basic functionalities are only reachable via the context menu. This means that the user is forced to perform 2 clicks for each action – one to open the context menu and another to select the desired operation. In order to change the current map extent the user must span a rectangular area on the map. This rectangle determines the desired map scale and extent, and requires 2 more clicks. The applied *AdaptationModel* does not require a context menu for these frequently used operations. The *MapExtent* and *Zooming* operations are adaptive and do not necessitate any user-device interaction.

The second comparison deals with the visualization of POIs. As with the other two operations in *utopian* an interaction with the context menu is needed. The third click on the desired POI delivers the corresponding information. The *AdaptationModel* provides an adaptable functionality for the visualization of POIs. If a POI is located within the current map extent, an icon will be visualized on the map. A click on the icon delivers the requested information.

Evaluating the effects of changes in the cognitive load for the user also requires a quantification. Such reduction can be measured by using different subjective and objective measurement techniques (Bunch & Lloyd 2006). Both are based on surveys with participants. The *AdaptationModel* presented here does not support surveys, because it is not an implemented service that is executable on a mobile device. The reduction of cognitive load will therefore be evaluated through existing survey results (Radoczky 2003, Wealands et al. 2007). As mentioned in *Section 2.2.2,* many participants voted for multi-encoded navigation instructions *(SwitchingLandmark* operation) and a track-up oriented map *(TrackUp* operation). Both operations are not provided by *utopian.* In combination with the *MapExtent* operation, the number of perceived elements in the map is lower. Hence, a reduction of cognitive load is achieved. The *Zooming* operation provides the user with a detailed cartographic image. This reduces the amount of perceived elements for the user at decision points. Therefore the discussion so far indicates a verification of the hypothesis. An implemented pedestrian navigation service, which provides navigation instructions based on the *AdaptationModel* adapts to the user's situation, instead of forcing the user to adapt to the service.

2.7 Conclusions and Future Work

We presented an abstract formal model for mobile map adaptation, which can be used as a basis for implementing context-aware LBS. The useful combination of adaptive and adaptable functionality achieves a user-centered design. The *AdaptationModel* can act as a guideline for both simple (e.g., basic routing functionalities) and complex (e.g., manifold functionalities such as with *utopian)* LBS. The classification into submodels and components regarding context, user, and task makes the *AdaptationModel* manageable and flexible. To demonstrate the latter two characteristics the concrete *AdaptationModel* for a pedestrian navigation service was developed. Concrete operations were specified to represent user interaction with the service. The identified features of the submodels served as input parameters with the aim of achieving context-sensitive map adaptation. The functional programming language Haskell was used to express the algebraic specifications in a formal manner. The comparison of *utopian* to the operations of the developed model confirmed the reduction of both the user interaction with the device and the cognitive load for the user. We come to the conclusion that taking a formal view of context-aware computing enables reasoning about the foundational relationships that process context. In particular, the formal model in this work serves as a rigorous basis for further development of a formal framework to design and evaluate context-aware services.

We made various simplifying assumptions and only those features necessary to confirm the hypothesis were taken into consideration. One of the most important aspects for future work is the implementation and evaluation of the *AdaptationModel*. After implementing the operations in an executable pedestrian navigation service, a survey could be done to evaluate the proposed adaptation features. Further investigations are necessary to decide whether adaptive user interfaces may be confusing for the user or not, especially when the user is not aware of the underlying mechanisms and algorithms. With a GIS behind the map representation for a mobile device, the potential types and amount of information for display are unlimited. Additionally, digital mobile maps have the ability to display different levels of detail. More research is needed on how people interact with digital maps on mobile devices.

Another direction for future work is making the *AdaptationModel* more dynamic, i.e., realizing the *UserModel* as a dynamic part similar to the *ContextModel*. Design environments offer possibilities for integrating adaptive and adaptable components to increase the shared knowledge between users and computers. This could be realized through a user profile. Behavioral user data could be acquired automatically through sensors. To provide personalization and adaptation capabilities, systems need to be able to reason about their users (e.g., applying methods such as neural networks or machine learning techniques). There are also important user privacy and ethical issues that need to be addressed. 'Perfect' privacy guarantees are in general hard to achieve, therefore a balance between service enhancement and privacy concerns has to be found.

In the current version the *AdaptationModel* only supports one task to determine the required features. A next step could be the consideration of an entire application such as *utopian*. A possible incorporation of several tasks will lead to an increased number of features and operations. Complex applications will also require a deeper investigation of the relationships between the three submodels. Controlling the information visualization on the small mobile device may require weighting the available features. A research goal in this respect is finding appropriate weights for the different input parameters (i.e., adaptation targets).

Acknowledgements

The comments from two anonymous reviewers provided useful suggestions to improve the content of the paper. We also thank Gwen Raubal for improving the grammar and style of this work.

References

Agrawala M, Stolte C (2001) Rendering effective route maps: improving usability through generalization. In Pocock L (Ed.), *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)* (pp. 241–249). New York: ACM Press.

Barkhuus L, Dey A (2003) Is context-aware computing taking control away from the user? Three levels of interactivity examined. In Dey A, Schmidt A,. McCarthy J (Eds.), *Proceedings of 5th International Conference on Ubiquitous Computing 2003 (Ubicomp), Seattle, Washington* (Vol. 2864, pp. 159–166). Berlin: Springer.

Baus J, Kray C (2002) *Frames of Reference, Positional Information and Navigational Assistance.* Paper presented at the Fifteenth International Florida Artificial Intelligence Research Society Conference (Flairs 2002), Pensacola Beach, Florida, USA.

Bunch R, Lloyd R (2006) The Cognitive Load of Geographic Information. *The Professional Geographer, 58*(2), 209–220.

Cartwright W, Crampton J, Gartner G, Miller S, Mitchell K, Siekierska E, Wood J (2001) Geospatial Information Visualization User Interface Issues. *Cartography and Geographic Information Science, 28*(1), 45–60.

Denis M, Michon PE, Tom A (2006) Assisting Pedestrian Wayfinding in Urban Settings: Why References to Landmarks are Crucial in Direction-Giving. In Allen G (Ed.), *Applied Spatial Cognition – From Research to Cognitive Technology* (pp. 25–51). Mahwah, NJ: Lawrence Erlbaum.

Dey A, Abowd G (2000) Towards a Better Understanding of Context and Context-Awareness, *CHI 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness*. The Hague, The Netherlands.

Dransch D (2002) *Handlungsorientierte Mensch-Computer-Interaktion für die kartographische Informationsverarbeitung in Geo-informationssystemen*: Fachbereich Geowissenschaften, Freie Universität Berlin.

Fischer G (1993) Shared knowledge in cooperative problem-solving systems – integrating adaptive and adaptable components. In Schneider-Hufschmidt M, Kühme T, Malinowski U (Eds.), *Adaptive user interfaces: Principles and practice* (pp. 49–68). Amsterdam: North-Holland.

Frank A (2000) Spatial Communication with Maps: Defining the Correctness of Maps Using a Multi-Agent Simulation. In Freksa C, Brauer W, Habel C, Wender K (Eds.), *Spatial Cognition II – Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications* (Vol. 1849, pp. 80–99). Berlin, Heidelberg, Germany: Springer.

Gartner G (2004) *Location-based mobile pedestrian navigation services – the role of multimedia cartography.* Paper presented at the International Joint Workshop on Ubiquitous, Pervasive and Internet Mapping 2004 (UPIMap), Tokyo, Japan.

Göker A, Myrhaug H (2002) User context and personalization. In *6th European Conference on Case Based Reasoning and Personalization (ECCBR 2002), Aberdeen, Scotland*. Berlin: Springer.

Gutwin C, Fedak C (2004) Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In Heidrich W, Balakrishan R (Eds.), *Proceedings of the 2004 Conference on Graphics interface, 17.–19. May 2004, London, ON, Canada* (pp. 145–152). Ontario, Canada: Canadian Human-Computer Communications Society.

Hägerstrand T (1970) What about people in regional science? *Papers of the Regional Science Association, 24*, 7–21.

Hampe M, Elias B (2004) *Integrating Topographic Information and Landmarks for Mobile Applications*. Paper presented at the Symposium on Location Based Services and Telecartography, Vienna, Austria.

Hampe M, Paelke V (2005) *Adaptive methods for mobile applications*. Paper presented at the 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2005), Salzburg, Austria.

Haunold P, Kuhn W (1994) A Keystroke Level Analysis of a Graphics Application: Manual Map Digitizing. In Adelson B, Dumais S, Olson J (Eds.), *CHI'94 Conference on Human Factors in Computing Systems in Boston, MA* (pp. 337–343): ACM.

Hudak P (2000) *The Haskell School of Expression: Learning Functional Programming through Multimedia*. New York: Cambridge University Press.

Lake R (2001) *Location-Based Services & GML* (Technical report). Vancouver: Galdos Systems.

Lovelace K, Hegarty M, Montello D (1999) Elements of Good Route Directions in Familiar and Unfamiliar Environments. In Freksa C, Mark D (Eds.), *Spatial Information Theory – Cognitive and Computational Foundations of Geographic Information Science, International Conference COSIT '99, Stade, Germany* (Vol. 1661, pp. 65–82). Berlin, Heidelberg, New York: Springer.

Mayer R, Moreno R (2002) Aids to computer-based multimedia learning. *Learning and Instruction, 12*(1), 107–119.

Miller H (1991) Modeling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information Systems, 5*(3), 287–301.

Nivala A, Sarjakoski L (2003) Need for Context-Aware Topographic Maps in Mobile Devices. In Virrantaus K, Tveite H (Eds.), *ScanGIS, The 9th Scandinavian Research Conference on Geographical Information Science, 4.–6. June 2003, Espoo Kivenlahti, Finland,* (pp. 15–29): Springer.

Oppermann R (1994) *Adaptive User Support Pod: Ergonomic Design of Manually and Automatically Adaptable Software*: Lawrence Erlbaum Associates.

Radoczky V (2003) *Kartographische Unterstützungsmöglichkeiten zur Routenbeschreibung von Fußgängernavigationssystemen im In- und Outdoorbereich*. Unpublished Master Thesis (German), Vienna University of Technology, Vienna, Austria.

Raper J, Gartner G, Karimi H, Rizos C (2007) A critical evaluation of location based services and their potential. *Journal of Location Based Services, 1*(1), 5–45.

Raubal M, Kuhn W (2004) Ontology-Based Task Simulation. *Spatial Cognition and Computation, 4*(1), 15–37.

Raubal M, Miller H, Bridwell S (2004) User-Centred Time Geography For Location-Based Services. *Geografiska Annaler B, 86*(4), 245–265.

Raubal M, Winter S (2002) Enriching Wayfinding Instructions with Local Landmarks. In Egenhofer M, Mark D (Eds.), *Geographic Information Science – Second International Conference, GIScience 2002, Boulder, CO, USA, September 2002* (Vol. 2478, pp. 243–259). Berlin: Springer.

Reichenbacher T (2003) *Adaptive Methods for Mobile Cartography.* Paper presented at the 21st International Cartographic Conference (ICC 2003), Durban, South Africa.

Rinner C, Raubal M (2004) Personalized Multi-Criteria Decision Strategies in Location-Based Decision Support. *Journal of Geographic Information Sciences, 10*(2), 149–156.

Rinner C, Raubal M, Spigel B (2005) *User Interface Design for Location-Based Decision Services.* Paper presented at the 13th International Conference on GeoInformatics, 17–19 August 2005, Proceedings (CD-ROM), Toronto, Canada.

Schilit B, Adams N, Want R (1994) *Context-aware computing applications.* Paper presented at the Workshop on Mobile Computing Systems and Applications.

Schilit B, Theimer M (1994) Disseminating Active Map Information to Mobile Hosts. *IEEE Network, 8*(5), 22–32.

Schmidt A, Beigl M, Gellersen H (1999) There is more to context than location. *Computers and Graphics, 23*(6), 893–901.

Shneiderman B (1992) *Designing the user interface: strategies for effective human-computer interaction*. Boston, MA: Addison-Wesley.

Strang T, Linnhoff-Popien C (2004) A Context Modeling Survey. In *Workshop on Advanced Context Modeling (UbiComp 2004), Nottingham, U.K.* (Vol. 3205): Springer.

Wealands K, Miller S, Benda P, Cartwright W (2007) User Assessment as Input for Useful Geospatial Representations within Mobile Location-Based Services. *Transactions in GIS, 11*(2), 283–309.

Winter S, Raubal M, Nothegger C (2005) Focalizing Measures of Salience for Wayfinding. In Meng L, Zipf A, Reichenbacher T (Eds.), *Map-based Mobile Services – Theories, Methods and Implementations* (pp. 127–142). Berlin: Springer.

Zipf A (2002) User-Adaptive Maps for Location-Based Services (LBS) for Tourism. In Wöber K, Frew A, Hitz M (Eds.), *Proceedings of the 9th International Conference for Information and Communication Technologies in Tourism, Innsbruck, Austria*. Heidelberg, Berlin: Springer.

Zipf A, Jöst M (2006) Implementing Adaptive Mobile GI Services based on Ontologies – Examples for pedestrian navigation support. *Computers, Environment and Urban Systems (CEUS), 30*(6), 784–798.