Sharing Personal Symbolic Locations between Friends – A Location Service for Small Communities

Jörg Roth

Department of Computer Science
Univ. of Applied Sciences Nuremberg
Kesslerplatz 12, 90489 Nuremberg, Germany
Joerg.Roth@Ohm-Hochschule.de

Abstract: In this paper we introduce a novel community service based *on personal symbolic locations*. Users can share their location with other people using simple textual distribution mechanisms without the need of a geographic map display service. Small communities can set up a light-weight infrastructure and do not need to send private locations to big central service centers.

1 Introduction

People often want to know *where* their friends or family members currently are. Many public services such as *Google Latitude* or *facebook* support communities if they want to share their locations. Such services are mainly based on so-called *physical position*s (i.e. geographic coordinates) and usually show locations on a geographic map.

People have some difficulties with physical positions. Only few people know their current physical co-ordinate or the co-ordinate of their home. Instead they use terms such as 'at home' to indicate their location. Locations that are described as texts are called *symbolic* locations. Some community services already support symbolic locations when users check in e.g. into a restaurant. Such symbolic locations are defined by a global directory that contains publicly relevant points of interests.

We go into another direction: our approach is based on the idea of *personal symbolic locations* which are defined by users and only have a meaning for small communities. Typical personal symbolic locations are 'home', 'office' or 'shopping'. In contrast to physical locations, they can be distributed and displayed using purely textual mechanisms.

2 Related Work

Location-based services such Google Latitude allow users to track friends. Small icons represent the friends' locations painted on a background map that is loaded from, e.g., Google Maps. Users define a bidirectional *friend* relation and can decide to temporarily

sign off from the service if they want to become 'invisible'. Friends automatically see updated position without any interaction of the tracked user. A manual 'send my location' function allows a user to distribute her or his location to any other user (not only friends) by e.g. email or SMS. The receiver gets an URL that opens a Google Maps view in which a marker shows the user's position.

Often, social services are combined with so-called *check-in services*. Places to check in are, e.g. restaurants, cinemas or public places. *Google Latitude* uses its social network *Google+*; services such as *facebook*, *foursquare* or *Gowalla* follow a similar idea. Users can automatically check-in based on their precise measured position or can manually check in via a web-based front end. The latter approach ensures that the user really resides at a certain location. Typical locations are small compared to position measurements errors, thus due to the measurement jitter, a full automatic system may fail. In addition, GPS fails indoors, but most check-in-locations are indoor locations.

Other projects try to identify symbolic locations from physical positions. In [Ve09] three major contexts are distinguished: 'home', 'work' and 'on the move'. In [BC+08] visited locations are stored in a kind of diary. This can by classified later by a Bayesian network that also takes into account the time of visit. A similar approach follows [LW+06], but it first maps physical locations to semantic locations with the help of a central geo database. [AS03] presents an approach to geometrically cluster GPS locations of multiple users to identify interesting locations. Projects such as *CybreMinder* [DA00], *Nexus* [HH+10] or [SH+06] provide a general platform to capture and to distribute location information. Usually they base on a centralized infrastructure that collects information and executes the rules to trigger events. *ContextPhone* [RO+05] primarily provides a system to plug-in context components on a mobile device. If a platform relies on central services, privacy issues become more important.

Connecto [BB+08] assigns names to physical locations that can be shared with friends. Each name still represents a single point in space. In contrast, our personal symbolic locations represent (potentially large) areas, also cities, city centers or districts. This difference has implications on privacy issues, as friends only know that users are *somewhere* inside an area, not the exact position. Privacy can also be achieved by rules that control how to share locations in groups. In [JH+02] the 'Principle of Minimum Asymmetry' is suggested that is in short: users that provide useful location information to the group will get the same amount of information back from other users. We do not support such mechanisms as they are often difficult to manage and to understand by users. Especially, different users have certain roles in communities and often cannot achieve the required symmetry between receiving and providing data. We strongly believe that our notion of personal symbolic locations is much easier to understand and allows communities to control privacy issues themselves without 'magic' and potentially intransparent control mechanisms.

Besides symbolic and physical locations there is a third type to define friends' locations: the *mailing address*. Actually, it could be summarized under symbolic locations, as it also can be transmitted via purely textual channels. Mobile platforms have easy-to-use APIs to get the mailing address from a physical location. The drawbacks: only users that

are close to buildings with a mailing address can be located. Second: often mailing addresses do not have a meaning for other users, thus the recipient again has to use a map service to locate an address.

3 The Zonezz Location Service

The *Zonezz Location Service* picks up the idea of *personal symbolic locations*. It can be installed on a smart phone (currently an Android phone). It is deployed as app, but mainly provides a background tasks that runs without any user intervention. It periodically detects the user's current symbolic location that is distributed to community members.

The Zonezz Location Services is based on the *Zonezz Core* as presented in [Ro11]. The Core was created to provide a local provider of personal symbolic locations for other apps that run on the respective smart phone. This is useful, if, e.g. the smart phone's home screen should change between a Job and Hobby view or if a calendar app should present job-related dates more intrusively when at work. The original Zonezz Core only locally communicates with components on the same smart phone. Our new extension, the Zonezz Location Service now communicates across the network and allows users to share their current location with other users.

As described in the related work section, the basic idea of sharing locations is realized in many platforms. However, the Zonezz Location Service has some important differences to other services:

- We use symbolic locations instead of physical positions. The locations are transmitted as text string such as 'job' or 'home' and not as physical location such as 49.465778 North, 11.158749 East.

- The symbolic locations are user-defined and *not* globally unique. Whereas other platforms map physical to symbolic locations with the help of a central database of *Points of Interests*, our symbolic locations are individually defined. Each user defines his own set of locations that are only stored on his smart phone. There is no central repository of locations at all. Location names may be *non*-disjoint, e.g. two users may define a location 'home', but with a different meaning.

- We only consider small communities. Each community creates its own distribution infrastructure that is *not* shared with other communities. This means, there is no big central server that holds locations for thousands or millions of people.

- We incorporated the notion of *trigger events* into the concept. Not only to *be* somewhere is important – to *change* the location is also important.

- The mobile user has full control over the distributed data and can adjust the published location information in a fine-grained manner. She or he can decide which locations to be presented to other users and which not.

We believe that especially the distribution of user-defined texts provides a significant benefit compared to existing map-based systems that only share physical locations.

As smart phone platforms do not provide a fully stable environment, there cannot be any guarantee for location information delivery. Thus, Zonezz is *not* designed for mission-critical applications, payment or access control. An example for the intended usage is a family that wants to get overview about each other's locations. Useful events could be: 'Child John enters school at 7:55', 'Dad leaves the office at 17:33', 'Ma enters the city center at 12:05'.

3.1 The Architecture

Zonezz starts without user intervention whenever the smart phone boots up. It installs itself as a background service, i.e. it periodically runs its code even though the smart phone is in stand-by. Usually, after a short configuration, user can 'forget' this program as it never requests any attention.

It is important that the service does not consume too much CPU power and battery load. The background task wakes up every 2 minutes to check the location. Zonezz especially supports the 'network' positioning system that consumes considerably less battery than GPS. Between the checks, the phone is switched to low power mode. The check as such does not significantly affect battery lifetime. Note that on typical smart phones a lot of background services periodically work.

Fig. 1 presents the architecture with the following components:

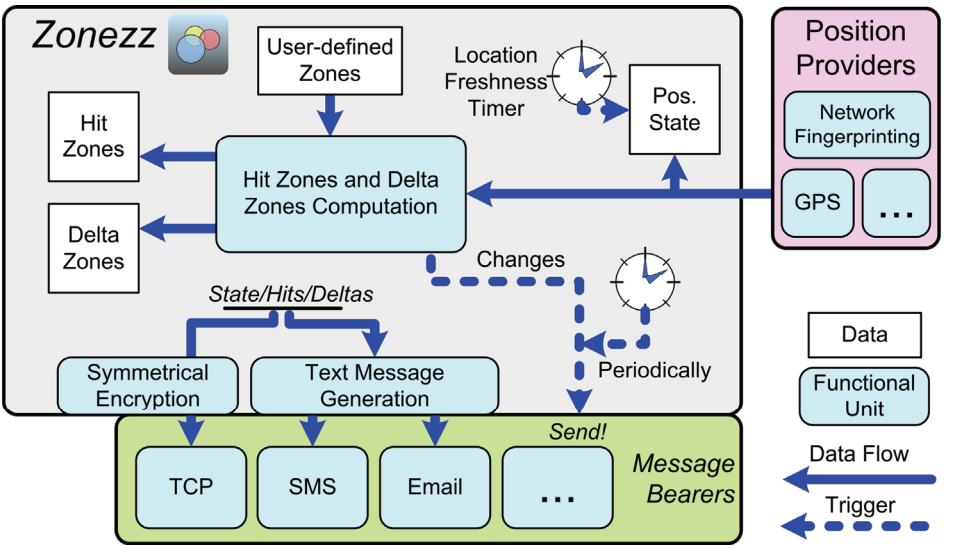

Figure 1: Service architecture

Position Providers are components of the smart phone operating system that measure the current physical position. Typical smart phones have a GPS provider and a 'network' provider which detects the current position with the help of signal strength fingerprints (of WLANs and Cell phone systems).

A *Position State* register contains information about the last measurement. State information may be: *positioning system error*, *last fix too old* or *last measurement is new and valid*.

The component *Hit Zones and Delta Zones Computation* maps a physical position to personal symbolic locations called *zones* (see section 3.2). It geometrically checks if one or more user-defined zones cover the current physical position. That leads to a set of *hit zones*. The *delta zones* reflect changes of the hit zone set. If we have two hit zone sets hit_n, hit_{n+1} of consecutive time stamps t_n, t_{n+1}, the delta zones contain the two sets $entered_{n+1}=hit_{n+1} \setminus hit_n$ and $left_{n+1}=hit_n \setminus hit_{n+1}$.

Periodically or whenever $entered_{n+1}$ or $left_{n+1}$ are not empty, the position state and the sets of hit zones and delta zones are transmitted using the *Message Bearers*. The bearers can easily be extended. A bearer has to implement a software interface that requests two functions: one to send location data and one to open a configuration dialog, to set e.g. the target address.

The transmission of location information can be done in two ways: *textual* or *raw*. Textual means: the transmitted message is human-readable without any further processing. This requires a mapping of the location information into a readable text such as 'Jörg enters the office at 8:03'. This is performed on the sending smart phone with the component *Text Message Generation*. Examples for this usage are emails or SMS messages that Zonezz can automatically emit. If in contrast the receiver again is a software component (especially our *Aggregator*, see section 3.3), the bearer uses a raw mode that only transmits the location data without any additional explainable texts. In raw mode, the location information is symmetrically encrypted using a community password concatenated with an initialization vector. This provides a basic security.

3.2 Expressing Personal Symbolic Locations with Zones

In [Ro06, Ro08] we already presented the benefits of symbolic locations for communities. In this paper we now endorse *personal* symbolic locations as a perfect tool to share locations in small communities such as families or between friends. Each user defines a set of *zones* which are two-dimensional circular regions on the Earth's surface with arbitrary center and radius (fig. 2). The zone geometry is only stored on the user's device and never published. Each zone has an arbitrary zone name that is the only data transmitted to other users. Usually a user defines zones such as 'home', 'job' or 'office', 'shopping', or hobby zones such as 'gym'.

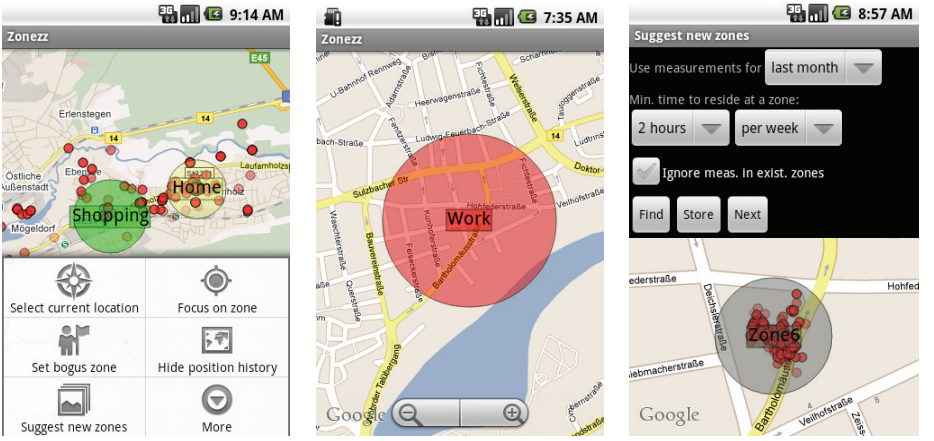

Figure 2: Definition of zones

The model is easy to understand by the mobile user and allows the execution of efficient algorithms. To define a new zone, a user points on a map and specifies name and radius. For the radius, there exist discrete values 50m, 100m, 200m,…, 50km. It is important to note that the map is only required during zone configuration.

One could argue that circles usually do not precisely represent location borders, but more complex areas such as polygons are difficult to enter and to administrate on a mobile device. In addition, for circular regions the system can easily suggest new zones based on a position history (fig. 2 right). The user can query the system for position clusters that can be added to the zone list.

Another benefit of circular regions: Position measurements are Gaussian distributed around a real position. Areas that include all position measurements of a certain area tend to circles anyway, even though the original areas are not circles. Thus, the actual benefit for more complex geometries would be small.

Every time when a measured position is inside such a zone circle, the corresponding zone is considered as part of the current location. As zones may overlap, a location is defined by a set of hit zones. We expect the hit zone with the smallest radius as most specific for the context, thus the set of hit zones is ordered ascending by their radius.

Crossing the border of a zone is important information, sometimes more meaningful than the static set of hit zones. If between two consecutive timestamps the delta zones are not empty, the user changed at least one of his zones. This always means movement. Movement often is more interesting for community members. E.g., to be in the office at a working day is obvious. But leaving the office at 18:10 is an important information, e.g. for family members. Here, our concept has a huge advantage compared to services that are based on physical positions: as physical positions virtually change permanently due to measurement fluctuations real movement detection is more difficult.

3.3 Aggregating Personal Symbolic Locations

Transmitting Emails or SMSs is an easy way to send locations and change triggers without any additional processing component. However, if multiple users should be informed, this type of transmission is not suitable. For this scenario a so-called *Aggregator* can collect locations of a certain community and serves as a central information point.

The Aggregator performs simple access control, stores the positioning states, hit zones and delta zones of all community members and emits trigger events. In addition, it creates formatted information pages (e.g. HTML) that reflect the current situation.

From the network point of view, an Aggregator in fact is a server. But compared to, e.g. a Web server, the aggregator is very light-weight with only 18 kB of Java code. Thus, it could even run on small or embedded devices such as a router, any PC or on the *Infowall* screen (see below). In principle, the Aggregator could also run on a user's smart phone, but currently, mobile phone networks do not support server sockets on mobile devices.

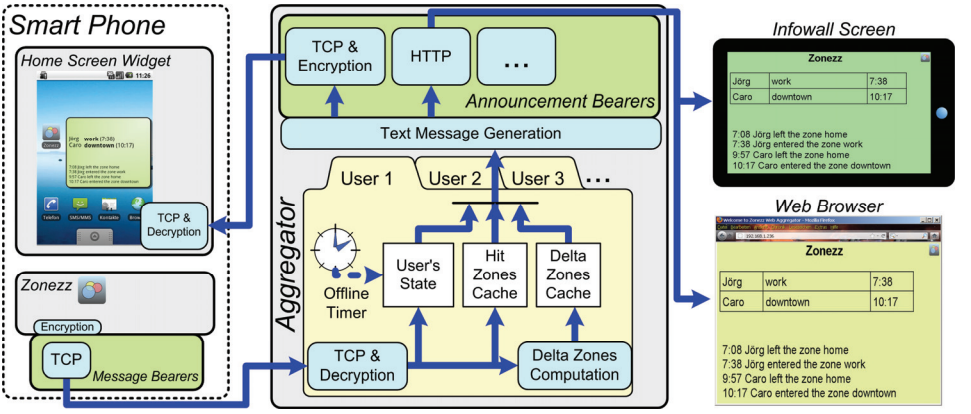

Figure 3: Distributing locations with the Aggregator

Fig. 3 shows the flow of data. After decrypting incoming messages, all location information (state, hit zones and delta zones) are cached. An *Offline Timer* detects if the last received message was too old. As each smart phone periodically sends a message (and not only in case of a change), the Aggregator can detect devices that are offline. Offline devices may have no wireless connection or may be completely switched off. In the current version, the transmission period is 30 minutes and the offline timeout 90 minutes.

Even though the smart phone already computes the delta sets, the Aggregator additionally computes them. This is necessary because messages may be lost due to network problems and the Aggregator caches may run out of sync.

Table 1: User states

No Position	The user is online but the smart phone cannot measure a position, because, e.g. no GPS satellites can be received.
Old Position	The user is online but transmitted zone sets are based on an old position (older than 10 minutes).
No Zone	A new position is available, but the hit zone set is empty.
Zones Available	A new position is available that resides at least in one zone.
Offline	The last update message is too old. The device is considered as offline.

Table 1 shows the possible user states. Note that **No Position** and **No Zone** have a different meaning. Even though **No Position** also leads to an empty hit zone set, it indicates a failure of position measurement, whereas **No Zone** indicates a properly running system.

3.4 Privacy

The approach provides some means to respect the user's privacy:

- Zones are mainly expressed by user-defined strings, thus provide a basic obfuscation. The *meaning* of the same string (e.g. 'job') is different for different users, thus cannot easily mapped to geographic positions.

- As each community sets up its own Aggregator, there is no big organization that can access location information of all users. Thus, users do not have to be afraid of large companies running data mining over all users.

- Encrypted communication between smart phones and Aggregator prevents attacks against the message transfer.

- If a special location is too 'private', the user is free not to model it as a zone. As zones have a certain geometric extent, other user cannot reveal the exact position.

Users can be members of multiple communities and may share different locations with each community. Let Z be the set of user-defined zones. For each Aggregator A, the user defines a set $Z_A \subseteq Z$ that indicates all publishable zones. Consistently, the sets *hit*, *entered* and *left* are mapped to hit_A, $entered_A$ and $left_A$ that are published to an Aggregator A: $hit_A = hit \cap Z_A$, $entered_A = entered \cap Z_A$ and $left_A = left \cap Z_A$.

As an example: a professor is member of the communities 'family' and 'university'. With the latter he wants to run a service for students that always indicates, whether he currently is at the university or not. For his family, he sets $Z_{family} = Z$, but for its students, he only sets $Z_{university} = \{\text{'Office'}\}$. Thus, the aggregator for students is not able to distinguish any zone outside the university. It is important to note that the filter Z_A is executed on the smart phone, not on the Aggregator. Thus, this mechanism is not vulnerable against Aggregator attacks.

4 Presenting Zonezz Locations

We support different tools to present the Aggregator's results (fig. 4).

We first tested to present the locations on our *Infowall service* (fig. 4 top). The *Infowall* screen is a fixed mounted wall screen or a bed table device that displays daily information, e.g. the local weather report, news messages, the message of the day or a calendar sheet with important dates. It can also display instant messages or reminders for family members. The screen periodically changes through all available information pages without user interaction.

Our idea was to integrate the Zonezz locations as a separate page inside the Infowall service (fig. 4 bottom left). The upper half of the page displays the most important (i.e. smallest) hit zone of each user and the time when the user entered this zone. The lower half shows the delta messages as a message log.

The Infowall decides when an information page is not interesting any more. Zonezz pages are automatically removed when they are not changed for two hours. If a Zonezz page changes, the Infowall service automatically speaks aloud the last delta message using Text-to-Speech function. Thus, a user that is in range of the Infowall screen can *hear* the most important message without reading the text.

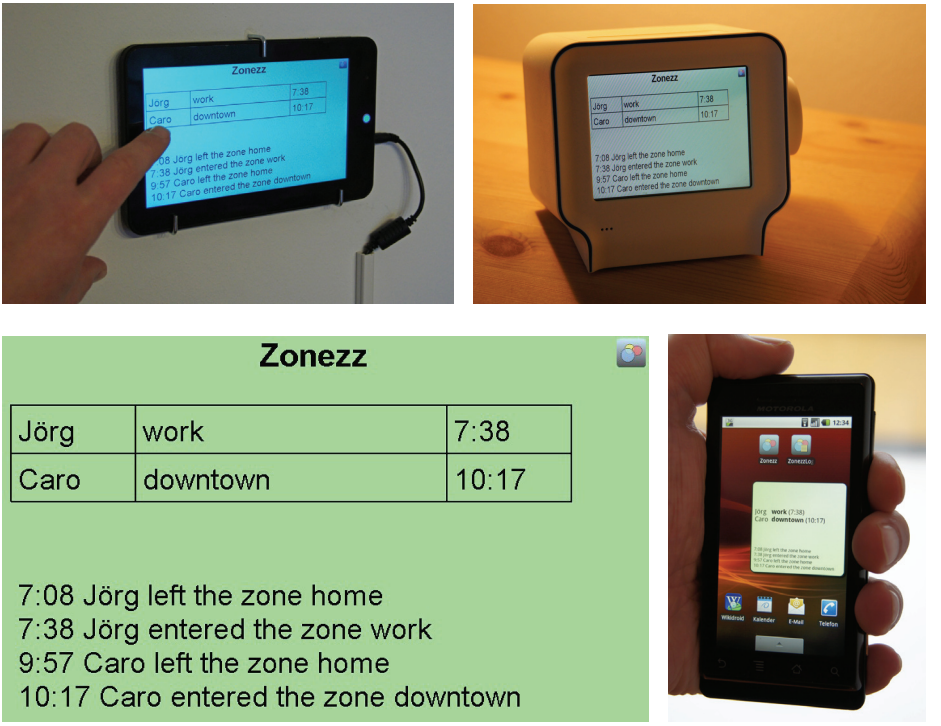

Figure 4: Different ways to present the location output

Mobile users require an additional tool to view Zonezz messages (fig. 4 right). A user thus can install a home screen widget on the smart phone that presents the information page.

5 Issues and Open Problems

Even though all described components are fully implemented and run as desired, we had to face some problems. Here, we describe the two most important ones:

1. Android Background Issues: Android supports periodic background execution and services that are started at boot time. However, the main goal of Android is to run usual apps and *not* background tasks. As a result, the stability of background code is not as desired. To give some examples:

- Network connections are not always established in the background. E.g., if the phone enters a known WLAN, sometimes it does not automatically connect.
- Background tasks are sometimes removed for memory reasons. Even though they are re-instantiated at the next cycle, sometimes irreproducible null pointers occur.
- Some operating system components are not fully error-free. If errors occurred in usual apps, the user simply would restart the app, but for background services this is a critical problem as the user does not even notice a problem.
- Sometimes the positioning systems do not properly work in background. On some devices, e.g., GPS is switched off in background.

2. Location Fluctuations: The concept of zones already is robust against positioning errors. A zone may be large enough to cover imprecise measurements at this location. However in rare cases, the current measurement jumps too far. We can distinguish two problems.

- *Enter fluctuation*: the measured location jumps inside another zone that is not the actual zone.
- *Leave fluctuation*: the measured location jumps outside the current zone.

In both cases, trigger events are emitted that may confuse other users. From the cases above, the leave fluctuation is more critical. As an example: if my position wrongly jumps outside my job zone, a delta message 'Jörg leaves the office' will be distributed, thus friends assume that I leave off work.

We could address fluctuation problems with location filters on the smart phone. E.g. a zone change has to be stated by multiple measurements before distributed to others. This however, would delay each trigger event for a certain time.

5 Conclusion

This approach to share locations in small communities is based on *personal symbolic locations*. They can be presented by purely textual mechanisms without the need of a map service. The underlying light-weight service can even run on small devices. In addition to the current location, we identified the change of locations as important information for community members. We integrated the idea into an information screen that could e.g. used by a family to know each other's current location.

References

[AS03] Ashbrook D., Starner T. 2003. Using GPS to Learn Significant Locations and Predict Movement across Multiple Users. Personal and Ubiquitous Computing archive Vol. 7 No. 5, Oct. 2003

[BB+08] Barkhuus L., Brown B., Bell M., Sherwood S., Hall M., Chalmers M. 2008. From awareness to repartee: sharing location within social groups, in Burnett M., Costabile M.F., Catarci T., de Ruyter B., Tan D., Czerwinski M., Lund A. (eds.): Proceeding of the 26th SIGCHI Conference on Human Factors in Computing Systems, ACM, 497–506

[BC+08] Bicocchi N., Castelli G., Mamei M., Rosi A., Zambonelli F. 2008. Supporting location-aware services for mobile users with the whereabouts diary. 1st intern. conf. on Mobile Wireless Middleware, Operating Systems, and Applications

[DA00] Dey A., Abowd G. 2000. CybreMinder: A Context-Aware System for Supporting Reminders, Proc. of the Handheld and Ubiquitous Computing 2000, Bristol, UK, Springer LNCS 1927, 172-186

[HH+10] Häussermann K., Hubig C., Levi P., Leymann F., Simoneit O., Wieland M., Zweigle O. 2010. Understanding and designing situation-aware mobile and ubiquitous computing systems. Proc. of intern. Conf. on Mobile, Ubiquitous and Pervasive Computing, March 2010, 329-339

[JH+02] Jiang X., Hong J.I., Landay J.A. 2002. Approximate information flows: socially-based modeling of privacy in ubiquitous computing, in Borriello G., Holmquist L.E. (eds.): UbiComp 2002: Ubiquitous Computing, Springer, Vol. 2498, 176–193

[LW+06] Liu J., Wolfson O., Yin H. 2006. Extracting Semantic Location from Outdoor Positioning Systems. 7th intern. conf. on Mobile Data Management

[RO+05] Raento M., Oulasvirta A., Petit R., Toivonen H. 2005. ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. IEEE Pervasive Computing, April-June 2005, Vol. 4 No. 2, 51-59

[Ro06] Roth, J. 2006: Detecting Identifiable Areas in Mobile Environments, 21st Annual ACM Symp. on Applied Computing, 23.-27. April 2006, Dijon, (France), ACM Press, 986-991

[Ro08] Roth, J. 2008. A Probabilistic Approach for Context Reasoning. Use In Context, GI Informatik 2008, Munich, Germany, Sept. 12. 2008, Proceedings 134, Vol. 2, 802-807

[Ro11] Roth, J. 2011. *Context*-aware Apps with the Zonezz Platform, ACM MobiHeld 2011, Proc. of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, Cascais (Portugal), 23. Oct. 2011

[SH+06] van Sinderen, M. J., van Halteren, A. T., Wegdam, M., Meeuwissen, H. B., Eertink, E. H. 2006. Supporting context-aware mobile applications: an infrastructure approach. IEEE Communications Magazine, Sept. 2006, Vol. 44, No. 9, 96-104

[Ve09] Verkasalo H. 2009. Contextual patterns in mobile service usage. Personal and Ubiquitous Computing. Vol. 13, No. 5, 331-342