Proximates – A Social Context Engine

Håkan Jonsson and Pierre Nugues

Lund University, LTH, Box 118, SE-221 00, Lund, Sweden
{hakan,pierre.nugues}@cs.lth.se

Abstract. Several studies have shown the value of using proximity data to understand the social context of users. To simplify the use of social context in application development we have developed Proximates, a social context engine for mobile phones. It scans nearby Bluetooth peers to determine what devices are in proximity. We map Bluetooth MAC ids to user identities on existing social networks which then allows Proximates to infer the social context of the user. The main contribution of Proximates is its use of link attributes retrieved from Facebook for granular relationship classification. We also show that Proximates can bridge the gap between physical and digital social interactions, by showing that it can be used to measure how much time a user spends in physical proximity with his Facebook friends. In this paper we present the architecture and initial experimental results on deployment usability aspects of users of an example application. We also discuss using location for proximity detection versus direct sensing using Bluetooth.

Keywords: Mobile Phone Sensing, Proximity, Social Context, Social Sensing.

1 Introduction

The purpose of middleware for social context is to simplify development of applications that use social context. By social context we mean individuals and groups in proximity of a user and the relation of the user to the individual and group, for example family, co-workers, friends, sometimes referred to as pervasive social context [17]. Modeling a user's social context is not trivial. It requires knowledge about privacy, mobile sensing, power efficient data collection, data cleaning and analysis, clustering, etc. A mobile software component that addresses all this complexity is vital to save development effort and cost. The developer will then be able to focus on the task of using social context rather than extracting it.

There has been several studies investigating the relations between online social networks such as Facebook and social networks spanned by physical proximity or co-location retrieved from mobile phones. An early major research project into using proximity data for understanding a user's social context was the Reality Mining project [9]. This project studied social changes in organizations adopting proximity based applications, but also suggested consumer oriented applications, e.g. Social Serendipity [8], but did not include integration with an online social network. The SocioPatterns [1] project combined proximity sensing using directional RFID with online social networks, including Facebook. Later [16], this is used for link prediction in the proximity

M.J. O'Grady et al. (Eds.): AmI 2013 Workshops, CCIS 413, pp. 230–239, 2013.
© Springer International Publishing Switzerland 2013

network. Cranshaw et al. [7] model the social context of locations a user visits to do link prediction in the Facebook network. To this end they use location trails collected from GPS and Wifi networks on mobile phones. However, as shown by several studies [2, 6, 14] spatio-temporal granularity makes all the difference in modelling human social interactions, and location sensing rather than proximity sensing is not granular enough for our purposes. The Lausanne Data Collection Campaign has given rise to several important studies in this area, such as [5, 10, 13] but does not include Facebook data. WhozThat [3] use both sensed proximity and social network ids to bridge the gap between physical and online social network identities. However, the simplicity of the protocol raised some serious privacy issues as noted by the author, and it was not deployed in field trials with smartphone users. SocialFusion [4] address the privacy issues of WhozThat by proposing alternatives to K-anonymity for anonymization.

Middleware to address the complexities of developing pervasive social networking applications has been the topic of several studies as well. In a survey of mobile social network middlewares [12], requirements on such middleware is defined, which we use in the description of Proximate's architecture. Mokhtar et al. [15] suggest using Bluetooth for proximity detection. They discuss different potential deployment strategies of their architecture, and results are based on simulations using Reality Mining data and a social network derived from text message interactions.

In this paper we build upon the concept developed in Serendipity, to use the Bluetooth radio transmitter as a carrier of identity. In Serendipity a separate digital social network was created. In our project Proximates, we use Facebook and phone number as the digital identity of users and bridge it to the physical identities emitted by the users' devices. This allows us to analyze the relation between Facebook friendship and physical proximity, e.g. how much time a user spends with Facebook friends. We believe that Proximates can be used across many applications and used to build a corpus of social context data that can be shared across researchers. To satisfy users need of privacy as stated in the Obvious Data Usage Principle, we need to build value in proximity data. Proximates does this by supporting the bridging of physical and digital identities with low latency. We will use Proximates to study users perceptions of privacy regarding this bridging, architectures that satisfy scaling of research applications to large numbers of users, and spatio-temporal aspects of social dynamics.

In the first section(System Architecture) we present the architecture of Proximates (Figure 1) and how it bridges the gap between physical proximity space and online social networks. In section Applications and Results we present some early experimental results from a user study and example applications that was built on Proximates for the study.

2 The Proximates Social Context Engine

The purpose of Proximates is to simplify development of mobile phone applications that use social context. By social context we mean individuals and groups in proximity of a user and the relation of the user to the individual and group, for example family, co-workers, friends. By social context classification we mean the inference of the relationship class of such an individual or group. Which user identities and social networks

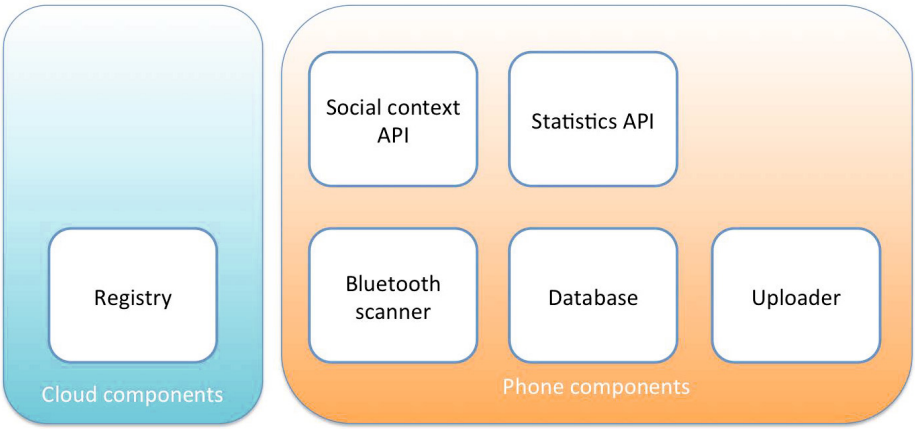

Fig. 1. Architecture of Proximates

to use is application specific, but can be shared across applications if desired. In its current deployment, Proximates use Facebook ids and phone numbers to identify users across applications.

2.1 System Architecture

Proximates consists of six components: a Bluetooth scanner, a database, two APIs, an uploader and a device registry.

Bluetooth Scanner and Social Context API Components. The Bluetooth scanner is a service that runs in the background on a mobile device. Periodically it performs a Bluetooth scan for nearby Bluetooth peers with the phone device class, and stores the result in the database. The data includes MAC ids, signal strength, and device class.

The social context API is a background service that carries out mining on the stored data in the database and triggers on events from the Bluetooth scanner. It performs smoothing of Bluetooth scans over time and group them for easy access and Bluetooth MAC ids are mapped to user ids. An API to application developers that allows applications to get notifications when a contact or group of contacts is in proximity, or when a user's social context change. The social context of a user is a ranked list of relationship labels, where the top label is the most common relation of the peers in proximity to the user, over a scan period. For example, if there are five peers in a scan where three are known to the user and two of them are classified as Family and three of them are classified as Colleagues according to the user's Facebook friend list, then the top ranked relation will be Colleagues, the second Family and the third Unknown.

To know the relation between a user and its friends, Facebook friend lists are used. When an application is notified of the proximity of a person, it can retrieve the classification of the relation to that person.

We use the labelled data to train classifiers of social context for users who don't use friend lists. The training of classifiers is ongoing work and results will be presented in future papers.

Registry Component. The purpose of the registry is to map Bluetooth MAC ids to any public ids of its owner. The public owner ids can be any application specific ids or general public ids, such as phone numbers or Facebook ids. It is up to the application that registers the device and its owner to determine which user ids to register and whether they should be hashed or not. Hashing makes it hard to make a lookup from a Bluetooth MAC id to a useful user id unless the user id is already known.

The registry is a web service and exposes a simple JSON REST API. The registry API performs all the operations on a single resource: the device. There are methods for adding, update, deleting devices, as well as retrieving a single device or a list of devices providing user ids and MAC ids as query parameters.

Applications are encouraged to cache results from device queries in order to minimize data traffic, server load, and power consumption. For known contacts, for example phonebook contacts, the Bluetooth MAC ids can be cached for a long time since they are not likely to change often. Some applications will not know the id they are looking for in advance and will need to lookup any new peers that are in proximity. These results should also be cached since transient peers often appear in at least some scans.

Database, Statistics API and Uploader. The database stores collected sensor data and events. The statistics API allows the application developer to query historical information, for example retrieve the most frequently occurring people, groups of people or social contexts of the user, over a specific time frame. The uploader pushes the stored data to a server. The uploader is an optional component that is deployed if Proximates is used for research applications, for example in computational social science, where extensive data logging is needed for analysis.

2.2 Requirements

As a middleware intended for real world deployment, Proximates needs to fulfill several requirements that are common to middleware for mobile social networking and pervasive social context. In the survey of mobile social middleware [12], the aspects below are analyzed, and we use them here as reference requirements. We also use the two social context modeling requirements [18] defined by Tran et al.

Simplification of Development Process. Modelling social context requires knowledge about privacy, mobile sensing, power efficient data collection, data cleaning and analysis, stream processing, clustering, etc.The social context API for detecting proximity of people, groups and social context is a very simple and high-level API.

Energy Efficiency. Power consumption is a major concern for opportunistic sensing applications. Recent availability of dedicated sensor processing subsystems in smartphone chipsets is improving the situation by allowing continuous sensing with low

power consumption. However, application sensing of network data from Bluetooth, 3G and Wifi still needs to be done in the application CPU.

Proximates uses Bluetooth to detect proximity. By using Bluetooth, proximity is sensed directly rather than indirectly through a translation to location coordinates and distance calculation. Bluetooth consumes less battery than GPS and Wifi, and only needs to be sampled periodically. Using Bluetooth rather than GPS or Wifi avoids both translation to location coordinates in the case of Wifi, and more importantly removes the need for frequent uploading of location data for all users for which we want to detect proximity.

The power consumption of scanning can be traded off with latency. If an application needs to be notified of a nearby person with low latency, power consumption increases. The latency in Proximates is configurable through setting of the scan rate. The default rate of 2 minutes makes the power consumption very low and in general not noticeable to the end user.

Privacy. By using hashed identifiers for people, for example Facebook IDs and phone numbers, Proximate applications require their users to already know the identities of the people they want to detect proximity of. This means they already need to be friends on Facebook or to already have their phone numbers. This is a more secure approach than the one used in WhozThat which transmits Facebook IDs in clear text. It is up to the application to determine whether to used hashed IDs or not, depending on requirements. For applications that do not need the access to the registry to be open, for example in the case where the ID is entirely application specific, strict access control to the registry component can be enforced rather than providing shared access across applications.

However, the attitudes of users regarding mapping their phones' Bluetooth MAC ids to personal identities, such as Facebook identities, is unknown. As far as we know, no such studies has been done. Several studies on privacy aspects of location sharing has been done, but we cannot assume they apply directly to proximity. We believe that sharing of your social network identity connected to your Bluetooth MAC id is less sensitive than sharing location data, since within Bluetooth range it is hard to hide your identity anyhow. At least in the non-public case, where only people who know you can detect you when in proximity. This remains to be verified by user studies.

Scalability and Distributed Architecture. The architecture of Proximates is very simple compared to most mobile social networking platforms since it focuses on a specific problem, uses proximity sensing, and delegates management of social networks to the original social network services rather than aggregates. Using direct proximity sensing rather than via location makes it possible to do the sensing directly on the device. This eliminates scalability problems associated with pairwise distance computations. The registry is a centralized component, but it is not subject to heavy loads since registrations seldom change which allows for long caching in clients.

Heterogeneity and Dynamicity of Mobile Environments. Performing the proximity detection directly on the device removes the need for continuous network coverage and data connection, making Proximates insensitive to networks signal strength fluctuations. Regarding heterogenous evironment, this is ignored by selecting Android 4.0.4 or later as the target environment.

Social Context Modelling. Tran et al. defined requirements [18] for social context modeling. We show how Proximates fulfill these and elaborate on them:

> Social context needs to explicitly capture constructed relationships and interaction constraints between actors. This set of constructed relationships and constraints needs to be managed, and modeled subjectively from an actor's perspective. The architecture of context-aware systems needs to externalize the management of social context from the implementation of actors.

Proximates explicitly models relationships through the integrated social networks, defined by the application. Currently integrated networks are Facebook and phone contacts. The user manages his Facebook relations using the Facebook service and his phone contacts through the phone book application, and are thus externalized. Interaction constraints are not specifically captured, but are left to the application since these are application specific. The relationships are modeled subjectively since Facebook friend lists is managed by the user and only visible to him.

> The architecture also needs to support the adaptability of social context, and needs to be easily deployable.

The complexity and cost of integrating and deploying a social context engine in commercial applications must be low. Many companies are yet to understand the potential benefits of context aware applications. This means that a small and simple component that solves a specific problem in existing infrastructure is preferable to a complex system that solves a wide range of problems. Additional context information should be added through integration of additional simple components that integrate well. It should also utilize existing infrastructure and services, e.g. Facebook for management of a user's social graph. Furthermore, the social context model must be simple and usable across several applications. Proximates fulfill these requirements through its simplicity and integration with existing services.

Additional Requirements. In addition to the requirements used in [12] we have further requirements on Proximates.

- Low latency for proximity detection is a requirement for some applications, for example reminder applications triggered by proximity to a person. This requirement makes it impossible to use location for proximity detection.
- Robustness, i.e. not needing to rely on GPS satellite visibility, availability of nearby wifi access points and a network data connection, also makes us choose direct proximity sensing.
- Finally, accuracy of location based methods is at best 10 meters indoors which is not enough to detect actual social interactions. Using Bluetooth, 10 meters is the maximum distance. Also, as shown by Cattuto et al. [2, 6] spatio-temporal granularity makes all the difference in modeling human social interactions, and location sensing rather than proximity sensing is not granular enough for our purposes.

3 Applications and Results

3.1 SmartTodos

Proximates was used to develop SmartTodos, a contextual reminder application, that for allows a user to add a reminder that will trigger when he is in close proximity of a specific contact or Facebook friend, in addition to reminders based on time and location triggers. It was distributed to about a 100 users who installed it and who could also send invites to others. We sampled seven users out of this population to make a usability study. The users in the sample were advanced smartphone users with university degrees.

Task completion time and requests for help was measured for seven different tasks as shown in figure 2. Task 1 is the task of setting up Proximates, while the other tasks are related to SmartTodos as such, for example the creation of location alarms. It is clear from the study that setting up Proximates is a hard task. It includes the following manual steps for the users: Signing into Facebook and accepting requested permissions, entering phone number, accepting enabling of Bluetooth and location services if not enabled, and setting Bluetooth visibility timeout to infinity. The most complicated step is the last one. This step should not be needed at all, but exists due to a bug in Android. Entering of phone number is often needed since most operators do not provide the information on SIM cards. Preloading of Proximates, an option only available to phone OEMs, can remove some of these obstacles, but it is clear that it is important to reduce the complexity of these tasks in Proximates.

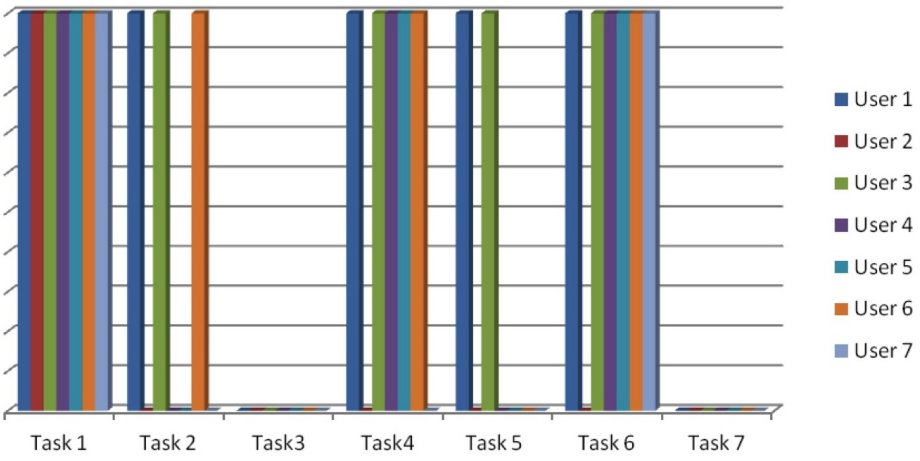

Fig. 2. Requests for help

3.2 Data Collection

In the SmartTodos trial we deployed the application with the upload component. Proximates collected proximity and Facebook data and uploaded them to a server where they were stored for analysis. Bluetooth scans were performed every minute, and only

Bluetooth with device class "phone" were stored for analysis. In the period July 2012 to March 2013, 2,466,036 Bluetooth scans containing 84,793 peers were collected by 135 devices. 161 users were registered with Facebook id in the registry, and they had 29,979 friends in total. 99 users had no user defined friend lists while the median number of friend lists was 9 among the 62 others. In addition to the traditional informed consent trough terms of service agreement, The Obvious Data Usage Principle [11] was applied in the application design to make it clear to users what information was being collected and how it was used. All data collected is anonymized through hashing.

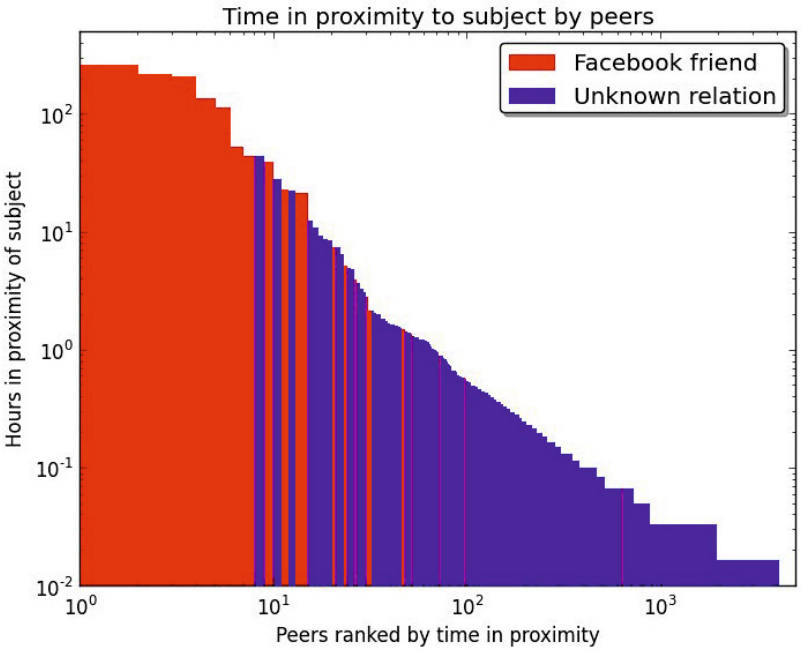

Fig. 3. Time in proximity of a single subject, and Facebook relationship

Figure 3 is a log-log plot of the time spent in proximity to one specific user versus the peers ranked by the same measure. In the plot we have colored each peer according to its relationship to the user. Red indicates a Facebook friendship relation, while blue means that the relationship is unknown, since that peer is not registered in the whoowns registry. A peer colored in blue could still be a Facebook friend of the user, but we have no information about this. This means that the time this user spent with Facebook friends was significantly larger than with friends with unknown relationships. An interpretation of this is that Facebook friendship is not only used to keep in touch with distant friends, but also a relationship users have with the people they actually spend time with. More work is needed to determine the generality of this result to other users.

4 Conclusion and Future Work

We have presented an architecture for proximity based services on smartphones, that is power efficient, easy to deploy, delegates social network management and scales well due to using direct proximity sensing rather than location and distance calculations. It also has other privacy properties than location sensing that needs further investigation.

We have shown that Proximates can be used in real world applications by means of the SmartTodos application.

Furthermore, we have shown that Proximates can be used for research applications by showing that we can measure the time users spend in proximity of Facebook friends. We also reported on how many users actually use Facebook friends lists. Further work is needed to investigate if Facebook friends list labels has the potential to be used as ground truth for classification of social context when Facebook friends labels are not available. We will also study how to improve usability by reducing the complexity of the setup of Bluetooth visibility and user ids in different social networks. We are about to launch SmartTodos on Google Play in order to scale up the amount of users and data collected.

Acknowledgements. This work was partly funded by the Industrial Excellence Center EASE – Embedded Applications Software Engineering, (http://ease.cs.lth.se) and the European 7th Framework Program, under grant VENTURI (FP7-288238).

References

1. Alani, H., Szomszor, M., Cattuto, C., Van den Broeck, W., Correndo, G., Barrat, A.: Live social semantics. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 698–714. Springer, Heidelberg (2009), http://link.springer.com/chapter/10.1007/978-3-642-04930-9_44
2. Barrat, A., Cattuto, C.: Temporal networks of face-to-face human interactions. Temporal Networks, pp. 50–55 (2013), http://link.springer.com/chapter/10.1007/978-3-642-36461-7_10
3. Beach, A., Gartrell, M., Akkala, S., Elston, J., Kelley, J., Nishimoto, K., Ray, B., Razgulin, S., Sundaresan, K., Surendar, B., Terada, M., Han, R.: WhozThat? Evolving an ecosystem for context-aware mobile social networks. Network IEEE 22(4), 50–55 (2008), http://dx.doi.org/10.1109/MNET.2008.4579771
4. Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S., Seada, K.: Fusing mobile, sensor, and social data to fully enable context-aware computing. In: Proceedings of the Eleventh Workshop on Mobile Computing Systems Applications HotMobile 2010, p. 60 (2010), http://portal.acm.org/citation.cfm?doid=1734583.1734599
5. Blom, J., Gatica-perez, D., Kiukkonen, N.: People-Centric Mobile Sensing with a Pragmatic Twist: from Behavioral Data Points to Active User Involvement. In: Mobile Devices and Services, pp. 381–384 (2011)
6. Cattuto, C., Broeck, W.V.D., Barrat, A., Colizza, V., Pinton, J.F., Vespignani, A.: Dynamics of person-to-person interactions from distributed RFID sensor networks. PloS One 5(7), 1–9 (2010), http://dx.plos.org/10.1371/journal.pone.0011596

7. Cranshaw, J., Toch, E., Hong, J.: Bridging the gap between physical location and online social networks. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing (2010), http://dl.acm.org/citation.cfm?id=1864380
8. Eagle, N.: Can Serendipity Be Planned? MIT Sloan Management Review 46(1), 10–14 (2004), http://www.angelfire.com/ab8/iissungminshane/smr.pdf
9. Eagle, N., Sandy Pentland, A.: Reality mining: sensing complex social systems. Personal and Ubiquitous Computing 10(4), 255–268 (2005), http://www.springerlink.com/index/10.1007/s00779-005-0046-3
10. Frank, J., Mannor, S., Precup, D.: Generating storylines from sensor data. In: Mobile Data Challenge Workshop (2012)
11. Jonsson, H.: The Data Chicken and Egg Problem. In: Proceedings of the 3rd International Workshop on Research in the Large, pp. 9–12 (2012)
12. Karam, A., Mohamed, N.: Middleware for mobile social networks: A survey. In: 45th Hawaii International Conference on System Sciences Middleware, pp. 1482–1490. IEEE (2012), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber+6149064, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6149064
13. Kiukkonen, N., Blom, J., Dousse, O., Gatica-perez, D., Laurila, J.: Towards rich mobile phone datasets: Lausanne data collection campaign. In: Proceeding of International Conference on Pervasive Services ICPS (2002)
14. Liu, S., Striegel, A.: Accurate Extraction of Face-to-Face Proximity Using Smartphones and Bluetooth. In: 2011 Proceedings of 20th International Conference on Computer Communications and Networks, ICCCN, pp. 1–5. IEEE (2011), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6006081
15. Mokhtar, S.B., Mcnamara, L., Capra, L.: A Middleware Service for Pervasive Social Networking. In: Proceedings of the International Workshop on Middleware for Pervasive Mobile and Embedded Computing, pp. 1–6 (2009)
16. Scholz, C., Atzmueller, M., Stumme, G.: New Insights and Methods for Predicting Face-to-Face Contacts. In: 7th Intl. AAAI Conference on Weblogs and Social Media (2013), http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/viewPDFInterstitial/6097/6396
17. Schuster, D., Rosi, A., Mamei, M., Springer, T., Endler, M., Zambonelli, F.: Pervasive Social Context - Taxonomy and Survey. ACM Transactions on Intelligent Systems and Technology 9(4) (2012)
18. Tran, M.H., Han, J., Colman, A.: Social context: Supporting interaction awareness in ubiquitous environments. In: MobiQuitous, vol. 9, pp. 1–10. IEEE (2009), http://researchbank.swinburne.edu.au/vital/access/services/Download/swin:15475/SOURCE1