Challenges in Ubiquitous Context Recognition with Personal Mobile Devices

Marcus Handte, Umer Iqbal, Wolfgang Apolinarski, Pedro José Marrón
Networked Embedded Systems Group
University of Duisburg-Essen
Duisburg, Germany
firstname.lastname@uni-due.de

ABSTRACT

Ubiquitous computing applications are required to provide distraction-free task support by reacting on different context characteristics. With the wide-spread use of personal mobile devices, many users are in possession of a powerful platform for context recognition. This, in theory, should allow the recognition of a number of characteristics which a user experiences during the course of daily routine. However, in order to be suitable for personal mobile devices, existing systems are considering a small and static set of characteristics for a particular application. This enables the developers to manually optimize their systems. Yet, it limits the applicability of the systems to narrowly defined scenarios. We argue that context recognition systems must take heterogeneity into account in order to be practically applicable to ubiquitous computing on a large scale. Specifically, future systems must find ways to accommodate the heterogeneity of tasks and users which results in three novel research challenges, namely the dynamic integration, privacy-preserving cooperation and automatic personalization of context recognition systems. In this paper, we motivate these challenges and outline ways to address them.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures

General Terms

Context Recognition, Systems, Algorithms

Keywords

Privacy, Personalization, Integration

1. INTRODUCTION

Ubiquitous computing envisions applications which provide seamless and distraction-free support for the everyday tasks of their users. In order to realize this vision applications must be able to adapt to the dynamics of the surrounding environments and to varying user requirements. Thereby, the adaptation must be performed automatically in order to ensure that the application adaptation does not conflict with the goal of providing a distraction-free user experience. This, in turn, requires applications to consider a broad range of characteristics of the user's context at runtime. Examples for these characteristics include the user's current location, activity and goals.

The automatic recognition of such context characteristics is a non-trivial multi-step process which can be broadly divided into three phases, namely sampling, preprocessing and classification. Sampling refers to capturing of raw data from one or more sources such as samples from microphone, accelerometer and camera, etc. Preprocessing involves removal of noise, selection of samples and extraction of features, e.g., applying low pass filter on the samples from the accelerometer. Classification involves deriving meaningful information from preprocessed samples, e.g., detecting whether the features extracted from a raw audio feed refer to speech or music. In the following, we use the term recognition stack to refer to these steps altogether.

Personal mobile devices such as smart mobile phones and personal digital assistants provide a promising basis for determining user context in an automated manner on a large scale. The reasons for this are manifold. First and foremost, personal mobile devices are self-contained and do not require additional infrastructure support, but existing cellular and wireless local area networks can provide the backbone for device interaction if needed. Secondly, though these devices are resource-constrained when compared to desktop computers, the newer generations are designed to support complex tasks such as displaying a high-resolution movie. As a consequence, the devices are mostly not utilized to their fullest capacity, leaving enough resources to perform context recognition. Thirdly, with a variety of on-board sensor, personal mobile devices have access to both physical and virtual data sources which allows multi-modal context recognition with high precision. Lastly, since the devices are carried by their users continuously, the device's context is tightly correlated to the user's context.

In summary, these characteristics have contributed to the development of number of context recognition systems for personal mobile devices. The recognition stacks applied by existing systems are usually fine-tuned for specific requirements in order to provide reasonably accurate results while requiring limited resources. Although, these stacks are suit-

CASEMANS'10, Sep 26, 2010, Copenhagen, Denmark.

able for accurately detecting desired characteristics, they cover only a narrow set that can be detected by one device. Moreover, due to the resource-constraint nature of personal mobile devices, developers have usually concentrated on providing solutions for a concrete application of a particular type of user, thus, limiting the practical applicability to a particular scenario.

The vision of ubiquitous computing, however, extends beyond the boundaries of a single scenario as it envisions seamless support for everyday tasks. As a consequence, by supporting a particular scenario, existing context recognition systems only make the first step towards ubiquity. The reason for this is threefold. First, different tasks may be sharing the same space and time. Second, a number of tasks may have to be handled cooperatively by multiple users. Third, different users may handle similar tasks in different ways. Thus, the next step in enabling context recognition on a large scale is to develop concepts that can deal with the heterogeneity of tasks and users inherent in the vision of ubiquitous computing. Dealing with this heterogeneity results in three novel research challenges. The challenges can be summarized as the dynamic integration, privacy-preserving cooperation and automatic personalization of context recognition systems. In this paper, we motivate and describe these challenges and outline possible ways to address them.

The remainder of this paper is structured as follows. Next, we describe existing context recognition systems for personal mobile devices and we outline their capabilities and scope. In Section 3, we motivate the need for supporting the heterogeneity of tasks and users in order to enable ubiquitous context recognition on a large scale. From this motivation, we derive three novel research challenges and we outline ways to address them in Section 4. Finally, in Section 5, we conclude the paper with a summary and an outlook.

2. RELATED WORK

There exist a number of context recognition systems for personal mobile devices. Some examples are [14], [15], [19], [12], [11], [21], [4]. These and many alike systems are manually fine-tuned for particular application and therefore are able to detect the fixed set of characteristics that has been determined at design time. As a result, these systems cannot be adapted to dynamic environments which a user might experience in a daily routine. These systems use built-in sensors in mobile phones to recognize the required context. For instance, CenceMe [14] uses microphones and accelerometers to determine user context which is then injected into social networking websites. Nericell [15] uses accelerometers and microphones to detect road conditions. Vtrack [19] uses location sensors to identify road traffic congestions. SoundSense [12] uses a microphone to identify different types of sounds in the surrounding.

Apart from the systems mentioned above, there also exist a number of rapid prototyping tools for developing context recognition applications. Examples include [18],[2],[9]. For instance, [18] is targeted at context recognition with pre-deployed sensors and provides a uniform abstraction for applications to access and use context information by hiding the actual context sensing and its interpretations from the applications. [2] is targeted towards motion recognition using wearable systems. It provides functionalities to develop distributed context recognition systems as well as reusable components, parametrizable algorithms, filters and classifiers. BetelGeuse [9] is a data gathering and processing open source platform targeted towards mobile phones with varying hardware capabilities. It consists of a minimal core that can be extended by plug-ins. The core relies on a black-board architecture that can be configured during start up phase, however, at runtime it cannot be adapted.

These and many other existing context recognition systems are well-suited to provide accurate information of their desired context. And since they are fine tuned, they require a minimal set of resources to perform their tasks. But despite this fact, they can only detect a narrow set of context characteristics. The main limiting factor is the resource constraint nature of personal mobile devices. Recently, there has been some work for minimization of power consumptions for context recognition systems on personal mobile devices. Examples include [8],[20],[3] which try to avoid unnecessary computations by deactivating parts of a recognition stack that are not needed. To do this, they identify the required stack on the basis of the current context or additional models of the user behavior.

The above mentioned approaches prove the applicability of personal mobile devices for context recognition, but, as we discuss in the following, they do not provide a generic solution that supports the everyday tasks envisioned by ubiquitous computing.

3. UBIQUITOUS RECOGNITION

Although, the existing systems clearly indicate the viability of context recognition with mobile devices in specific scenarios, the vision of ubiquitous computing extends beyond the boundaries of a single scenario as it envisions seamless support for everyday tasks. As a consequence, practical context recognition for ubiquitous computing must be able to support a broader range of tasks and users. To clarify this, we analyze the characteristics of a number of motivating examples in the following and discuss arising issues. Thereby, we start with issues resulting from task heterogeneity, followed by a discussion of heterogeneity with respect to users.

3.1 Heterogeneous Tasks

During the course of day, most users are involved in several tasks that stretch multiple scenarios such as working at the office during the day, shopping in the afternoon and reviewing papers at home in the evening. When looking at a more fine-grained resolution, the tasks are not necessarily tied to a particular location and they may be interleaved in an arbitrary order. In fact, sometimes they may be even executed in parallel. For example, a user may extend the shopping list in the office if additional items come to the user's mind. Similarly, the user may print the papers for review in the morning and may try to finish some office work at home. As a consequence, it is not possible to separate different scenarios such as context recognition *at work* and *at home* using space or time. Instead of supporting them separately, they must be treated in an integrated manner which can be complicated, especially, when considering that different types of users require the integration of different types of scenarios. In fact, when looking at a larger time scale, even the same user may require this. As a simple example consider a user that switches jobs.

Another issue arises from tasks that are performed cooperatively by multiple users. For such tasks, it may be hard or even impossible to detect the relevant characteristics of the

context using one personal mobile device. Instead, it may be necessary to combine inputs from multiple devices. As an example consider a presentation. Given a single device, it may be hard to determine the presenter. However, when combining the context gathered by multiple devices, detecting the presenter can be significantly simpler. To do this, the devices of the participants could cooperatively try to detect the presenter as the person whose device is receiving the speaker with the highest signal level or as the device that is facing the audience. However, the collaborative recognition requires that devices are able to interact with each other spontaneously.

3.2 Heterogeneous Users

In addition to the heterogeneity of tasks, another source of heterogeneity are users. Even when looking at a particular task of a single scenario, different types of users may handle them differently. As a result, the recognition stack that is used for context recognition may have to be customized specifically for each type of user to achieve a satisfactory precision. As an example consider two types of office workers, a manager and a staff member, whose meetings should be recognized automatically. Usually, the manager organizes meetings in order to distribute information to a number of staff members. Thus, they are scheduled well in advance and the meetings take place in a special meeting room. The staff member, on the other hand, participates in ad hoc meetings that are used to coordinate the current work. Thus, the meetings are not planned ahead of time but they are called in when they are needed. Furthermore, the meetings have only few participants and they might be held in an office of one of them. Clearly, detecting these two types of meetings require different recognition stacks as they require the detection of different low-level context characteristics in order to properly recognize a meeting. For the manager, a classifier could combine information from a room reservation system with the current time and location of the manager. For the staff member, a classifier could combine the number of staff members in the office. Obviously, for two types of workers, it may be possible to derive a common classifier that combines the low-level characteristics to detect both types of meetings. However, with an increasing number of user types, this approach cannot be applied without a considerable waste of resources.

4. CHALLENGES

In the following, we identify three novel challenges that arise when applying context recognition with personal mobile device to ubiquitous computing. Thereby, we discuss how they are tied to the specifics of ubiquitous recognition described previously. The challenges can be summarized as dynamic integration, privacy-preserving collaboration and automatic personalization of context recognition systems. After identifying these challenges, we structure them more clearly and we outline possible approaches to tackle them in the consecutive subsections.

As indicated previously, the interleaving and parallel execution of tasks from different scenarios requires an integrated treatment of multiple scenarios. The simplest form of integration would be to execute the recognition systems for all scenarios at all times. While this approach would be easy to realize, it is clearly not efficient with respect to resource utilization. The reason for this is that in many cases, the recognition systems would try to detect similar context characteristics which would result in duplicate measurements and computations. To avoid this, it is possible to manually integrate different systems. However, this approach suffers from the associated cost in terms of development effort. Furthermore, given a large number of scenarios, supporting all possible combinations is simply not feasible. To combine the efficiency of manual system integration with the simplicity of parallel execution, it is necessary to develop system software that takes care of the *dynamic integration* of different context recognition systems in an efficient manner.

To support collaborative tasks, it is necessary to extend recognition systems with the ability for cooperation. Intuitively, this raises a number of traditional questions that are typically addressed in the research area of distributed systems. Starting from basic communication, the issues encompass dynamic device discovery and support for spontaneous and interoperable interaction. Yet, such issues have been addressed already by a number of existing middleware systems for ubiquitous applications such as [7] or [1]. In addition to basic communication it is necessary to address consistency issues stemming from distributed observations which can be tackled by means of time synchronization [17], for example. A novel issue raised by collaboration results from the potentially sensitive nature of information that needs to be shared. As a result, achieving acceptance for collaboration among a broad spectrum of users requires additional concepts for *privacy-preserving cooperation*. A key issue thereby is support for a maximum degree of automation in order to enable safe as well as distraction-free collaborative recognition.

To efficiently support a broad range of users, it is necessary to adapt the context recognition to their behavior. Given the limited amount of resources available on most personal mobile devices, it seems to be hard – if not impossible – to develop generic recognition systems for more complex characteristics. An alternative to genericity is personalization. Thus, instead of developing generic systems it becomes necessary to develop specialized variants that work for different types of users. Intuitively, when the system development and optimization must be done manually by a developer, this approach results easily in an impossibly high development effort. To avoid this overhead it is necessary to develop tools to support *automatic personalization*.

4.1 Dynamic Integration

As mentioned above, the dynamic integration of different context recognition systems is required to provide an integrated treatment of parallel or interleaved tasks. In order to support dynamic integration, different context recognition systems are required to coexist and interoperate on the same device. Yet, due to the fact these systems are usually developed in an ad hoc manner, they do not provide means for cooperation with one another, thereby, limiting the required interoperability. However, simply gluing together various context recognition systems on a single device is not feasible due to the resource constraints of personal mobile devices. A closer look into context recognition systems reveals that there are a number of common functionalities performed by their recognition stacks mostly located at the lower level which can be exploited to perform integration in an efficient manner.

To dynamically integrate different context recognition systems in a more efficient manner, it is necessary to expose the

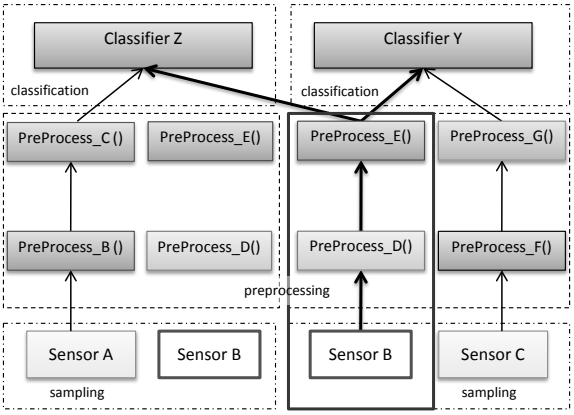

Figure 1: Component-based approach for dynamic integration

system structure in such a way that it can be analysed at runtime. This allows the automated detection of redundant or similar functionalities in the recognition stacks employed by different systems. In addition, the systems need to expose tuning knobs that enable modifications to the structure. These can then be used to remove the redundant and similar functionalities to reduce the waste of resources stemming from duplicate measurements and computations.

To support both, the exposition and the modification of the structure, it is possible to use a uniform component-based approach for stack development in which the algorithms used for context recognition are implemented as reusable components that are wired to detect a particular characteristic. This, on one hand, allows the development of recognition stacks by different developers in isolation. On the other hand, it provides a low analysis effort for recognition stacks at runtime. A simple example for this is shown in Figure 1 which depicts two context recognition stacks to detect the characteristics Y and Z. By runtime analysis of the components and their wiring, we can identify redundant functionalities such as the chain of components consisting of Sensor B, PreProcess_D() and PreProcess_E() which are common in both stacks. By rewiring the components at runtime, it is possible to avoid duplicate components, thus, reducing wasteful duplicate computations.

Although, this simple example already indicates the overall idea, when considering more realistic stack configurations it should be clear that the presence of identically configured chains of components is likely to be an exception. Instead, we can expect that the isolated development of recognition stacks will usually rather result in similar wirings with components that are parametrized differently. As a simple example consider that one system might try to detect the location of the device more frequently than another one. In order to deal with such cases, it is necessary to merge the parametrizations in a desirable manner. For example, one could merge localization stacks by detecting location at the higher frequency. However, in the general case, finding an appropriate configuration that satisfies the needs of all stacks can be complicated. Thus, doing this requires the development of appropriate parameter models for different types of components as well as the development of the associated analysis algorithms.

4.2 Privacy-preserving Cooperation

To support the detection of complex context characteristics, it may be unavoidable to perform the context recognition collaboratively using several personal mobile devices that may not be owned by the same user. For such a collaboration, devices must share their gathered sensor data with each other. An unlimited sharing that includes all available information is on the one hand very inefficient and on the other hand imposes several privacy issues. In many cases, the information may contain highly sensible information that can be used to guess the current context of a user. A built-in microphone, for example, may allow other devices to overhear spoken words and therefore get private information. Thus, it is necessary to restrict the access to shared information only to trusted devices. Thereby, it is noteworthy to mention that different devices might have different trust levels, so that some information will only be shared with a few trustworthy devices whereas other information may be shared more freely.

Context-sharing enabled devices must be able to answer the question which information should be shared with whom. This question can be automatically answered, if the device has a fine-grained privacy policy that contains both the trusted devices and the context characteristics allowed for sharing. Additionally, a device needs mechanisms that enforce this policy as shown in Figure 2.

The contents of a policy are typically user and thus, device dependent. Many users have different opinions about what kind of context should be regarded as private and not every device supports all context characteristics. As a consequence, we can expect that some policies might be more restrictive than others. To make things worse, the current situation of the user might have an influence on his current context sharing policy, so it must be updated regularly. This dynamic nature and the dependency on the user does not allow to create one static policy that is valid for every device, user and situation. Instead, a user dependent policy is necessary that can be updated according to the changes.

The manual creation of this kind of privacy policy, is already a tough task for the user. For manual creation, a user must think of all different context characteristics that could appear and define a fine-grained access control scheme. Additionally, groups of users must be defined and be given different access rights to the characteristics. If new users appear, they must be inserted in the present scheme without creating inconsistencies. A similar process will occur, if a new context characteristic is discovered. Thus, using a manual approach for policy creation, the user would be busy creating a policy and the achievable benefit from collaboration may be less than the loss of time that was needed for creating the policy.

To avoid the overhead of manual policy creation while supporting the privacy-preserving sharing of context information, it is necessary to automate the generation of the privacy policy. Using social networking sites, where users already define their privacy preferences with regard to several types of context information (including current location and private images), it could be possible to obtain user groups and access rules for context information. In fact, there already exist approaches such as the privacy wizard described in [5] which try to extract policy information from a social networking site. However, these approaches must be extended in order to support the automatic generation of con-

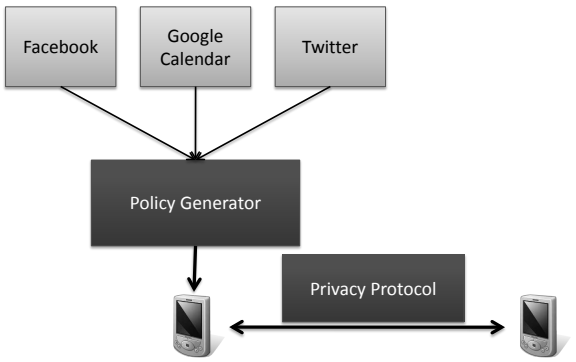

Figure 2: Generation and usage of privacy policy during the sharing of context information

text sharing policies since they only consider the aggregated context of social networking sites and not the fine-grained context of physical sensors. Several approaches that share sensor data without considering privacy, like [10], show that this challenge is still not fully addressed by the research community and therefore remains a valid research challenge.

Besides from policy generation, the mechanisms that are necessary to enforce the policy should also be fully automated. Thereby, the mechanisms should enforce the policy while preserving the privacy. For example, if it is necessary to detect whether a device belongs to a specific user group, doing so should not reveal the existence of the group nor the associated access rights. Therefore, new protocols need to be developed which enforce the policy, i.e. are gathering the needed information from other devices, looking up the access rights specified in the policy and finally share this information in a way that it cannot be gathered by unauthorized devices.

4.3 Automatic Personalization

To support a broad range of different user types, it is necessary to automate personalization with the help of off-line tools. Just like dynamic integration, this requires a common structure for recognition stacks that can be generated or manipulated in a generic fashion. Furthermore, it requires models to express desirable balances between different goals such as resource consumption or recognition accuracy. In addition, automation requires representative traces of sensor readings that are augmented with descriptions of the characteristics that should be detected at different times. However, existing systems such as MyExperience [6] already provide the ability to generate such augmented traces.

Based on the ground truth provided by the traces, a tool for automatic personalization could try to optimize an existing stack or ideally even synthesize a number of alternative stack configurations to recognize the desired characteristics considering various trade-offs. If the stacks are modelled as a parametrized wiring of components, the tools would have to modify the parametrization to optimize an existing stack and they would have to compose a wiring to synthesize new stacks as shown in Figure 3. Clearly, for both cases, the number of alternatives grows fast and exponentially with the number of parameters and the number of components that have to be considered. Thus, it is necessary to develop effective search algorithms.

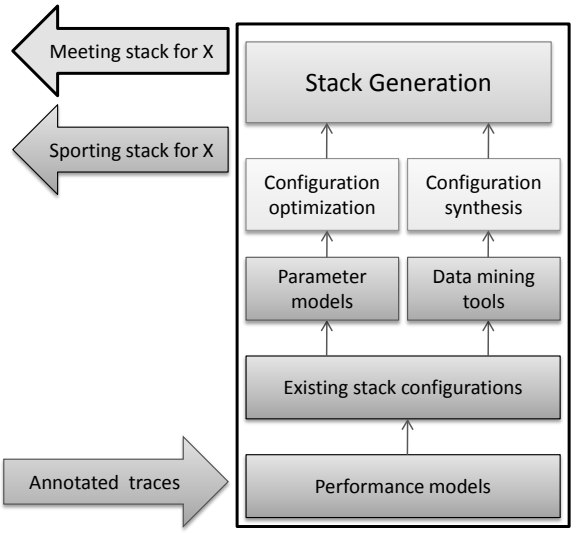

Figure 3: Tool support for automatic personalization

A simple approach may be the reuse of existing search algorithms and heuristics which can be found in the domain of artificial intelligence and operations research. However, to avoid both, the overhead of a complete search and possible imprecisions of heuristics, a better alternative could be the development of parameter models for optimization as well as the integration of data mining tools such as RapidMiner [13] for synthesis. A parameter model could express dependencies between parameters and goals or specific parameter behavior such as monotonicity to restrict the search to relevant parts of the search space. As a simple example consider that reducing the sampling rate of a sensor is not likely to improve the detection accuracy, in general. Similarly, it is unlikely that increasing the sampling rate results in a reduced resource consumption. For the synthesis, the integration of data mining tools could limit the search to compositions that use data that is correlated to the detectable characteristic. Although this might not be sufficient to enable a fully automated synthesis of a recognition stack, combined with a toolbox of frequently used sub-wirings, it may be sufficient to simplify the process such that it can be done by an advanced user.

5. CONCLUSIONS

Ubiquitous computing envisions seamless and distraction-free support for the everyday tasks of users. To achieve this objective, ubiquitous applications must be able to adapt to the context of their users. This raises the need for automated context recognition. Personal mobile devices provide a promising basis to support automated context recognition on a large scale. However, existing context recognition system usually focus on recognizing a small set of characteristics in a narrowly defined scenario.

The heterogeneity of tasks and users originating from the vision of ubiquitous computing raises a number of novel research challenges in context recognition with personal mobile devices. In this paper, we identified the dynamic integration, the privacy-preserving cooperation and the automatic personalization of context recognition systems as up-

coming challenges in ubiquitous context recognition. Furthermore, we have outlined how these challenges can be addressed.

Currently, we are working towards solutions for the dynamic integration of context recognition systems as well as the privacy-preserving cooperation. The results of this research will be incorporated into the *NARF Adaptive Recognition Framework* [16] which aims at providing a generic context recognition platform for ubiquitous applications.

6. ACKNOWLEDGEMENTS

This work has been partially supported by CONET (Cooperating Objects Network of Excellence) funded by the European Commission under FP7 with contract number FP7-2007-2-224053.

7. REFERENCES

[1] E. Aitenbichler, J. Kangasharju, and M. Mühlhäuser. Experiences with mundocore. In *Pervasive Computing and Communications Workshops*, volume 0, pages 168–172, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[2] D. Bannach, O. Amft, and P. Lukowicz. Rapid prototyping of activity recognition applications. *IEEE Pervasive Computing*, 7:22–31, 2008.

[3] A. Y. Benbasat and J. A. Paradiso. A framework for the automated generation of power-efficient classifiers for embedded sensor nodes. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 219–232, New York, NY, USA, 2007. ACM.

[4] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. The bikenet mobile sensing system for cyclist experience mapping. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 87–101, New York, NY, USA, 2007. ACM.

[5] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *WWW '10: 19th international conference on World wide web*, pages 351–360, New York, NY, USA, 2010. ACM.

[6] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay. Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *MobiSys '07: 5th international conference on Mobile systems, applications and services*, pages 57–70, New York, NY, USA, 2007. ACM.

[7] M. Handte, C. Becker, and G. Schiele. Experiences - extensibility and minimalism in BASE. In *Workshop on System Support for Ubiquitous Computing (UbiSys) at Ubicomp*, 2003.

[8] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song. Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 267–280, New York, NY, USA, 2008. ACM.

[9] J. Kukkonen, E. Lagerspetz, P. Nurmi, and M. Andersson. Betelgeuse: A platform for gathering and processing situational data. *IEEE Pervasive Computing*, 8(2):49–56, April-June 2009.

[10] N. D. Lane, H. Lu, S. B. Eisenman, and A. T. Campbell. Cooperative techniques supporting sensor-based people-centric inferencing. In *6th Int'l Conf. on Pervasive Computing*, 2008.

[11] M. Lawo, O. Herzog, P. Lukowicz, and H. Witt. Using wearable computing solutions in real-world applications. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3687–3692, New York, NY, USA, 2008. ACM.

[12] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: Scalable sound sensing for people-centric applications on mobile phones. In *MobiSys '09: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, pages 165–178, New York, NY, USA, 2009. ACM.

[13] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: rapid prototyping for complex data mining tasks. In *KDD '06: 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, 2006. ACM.

[14] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell. Cenceme - injecting sensing presence into social networking applications. In *EuroSSC*, pages 1–28, 2007.

[15] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys '08: 6th ACM conference on Embedded network sensor systems*, pages 323–336, New York, NY, USA, 2008. ACM.

[16] Networked Embedded Systems Group, University of Duisburg-Essen. Narf adaptive recognition recognition framework homepage, August 2010.

[17] K. Römer. Time synchronization in ad hoc networks. In *MobiHoc '01: 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 173–182, New York, NY, USA, 2001. ACM.

[18] D. Salber, A. K. Dey, and G. D. Abowd. The context toolkit: aiding the development of context-enabled applications. In *CHI '99: SIGCHI conference on Human factors in computing systems*, pages 434–441, New York, NY, USA, 1999. ACM.

[19] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 85–98, New York, NY, USA, 2009. ACM.

[20] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 179–192, New York, NY, USA, 2009. ACM.

[21] J. Ward, P. Lukowicz, G. Troster, and T. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1553–1567, 2006.