Towards a Framework for opportunistic Activity and Context Recognition

Marc Kurz, Alois Ferscha
Institute for Pervasive Computing
Johannes Kepler University
Altenbergerstrasse 69, A-4040 Linz
{kurz, ferscha}@pervasive.jku.at

Alberto Calatroni, Daniel Roggen, Gerhard Tröster
Wearable Computing Laboratory
ETH Zürich
Gloriastrasse 35, CH-8092 Zürich
{alberto.calatroni, daniel.roggen, troester}@ife.ee.ethz.ch

ABSTRACT
Due to recent advancements in the development of sensor devices which results in increasing mobility and computing power and decreasing size and costs, this paper presents two features for a sensor-based opportunistic activity and context recognition enabling framework. Opportunistic in this context means that no sensor infrastructure has to be predefined at the design time of the system, it rather makes best use of the available sensor devices. Therefore, a framework must be capable of (i) dynamically routing the information and data between sensing and data processing entities and (ii) the dynamic incorporation of training sources for sensor devices. Besides the discussion of these two features, the paper introduces the OPPORTUNITY framework, which is a prototypical implementation of an opportunistic activity and context recognition system.

Author Keywords
Opportunistic Sensing, Activity and Context Recognition, Sensor Networks, Framework Development.

INTRODUCTION
Opportunistic activity and context recognition systems do not rely on a static and predefined sensor infrastructure, such systems rather make best use of the currently available sensor devices [8, 9]. In conventional, established and well known activity and context recognition systems (e.g. [5], [2], [11] or [10]), the sensor network topology has to be defined at the design time of the system and the corresponding classifier, training, machine-learning and feature-extraction methodologies have to be implemented and used with a specific and fixed set of sensors which is not or at least barely flexible concerning sensor disarrangements, sensor disconnects and/or connects, or general sensor malfunctions. An opportunistic system has to react properly on topological changes not only when a sensor device disconnects, but also when a new sensor appears in the sensor infrastructure. Basic idea is not to use a predefined set of sensor systems, but to use - opportunistically - those sensor systems that are currently available. Furthermore, an opportunistic system also does not rely on a fixed recognition goal, a user or an application can request a recognition goal to a system at runtime. Based on this goal, the sensor devices self-configure to sensing ensembles, which is the set of sensor systems that is best suited to execute this very recognition goal. The dynamics in the sensor infrastructure and the configuration of sensor systems according to a dynamic recognition goal bear great challenges and sophisticated requirements for an opportunistic activity and context recognition system not only in the execution of recognition goals, but also in terms of scalability, heterogeneity, redundancy and replaceability, fault tolerance mobility, and frequency of change and spontaneous availability. Despite the dynamics in the network topology, the opportunistic system always tries to have the best available set of sensor configured for executing a very recognition goal which results in high dynamics in the information and data flows within the data producing and data consuming entities in the framework and in the dynamic incorporation of learning and training sources at runtime.

This paper presents the aforementioned two features that are inevitable in an opportunistic activity and context recognition enabling framework, namely (i) the dynamic routing of information and data between sensing entities themselves and data consuming units, and (ii) the dynamic incorporation of training sources for enabling online training of sensor devices at runtime. Currently, a prototypical implementation of an opportunistic system is being developed (referred to as *OPPORTUNITY Framework*) with OSGi [1, 3] and a reference dataset has been recorded in a kitchen scenario using 72 sensor devices with 10 different modalities [6, 7]. This dataset is used mainly to evaluate the evolved opportunistic methodologies by implementing virtual sensor devices that simulate the real-world, physical sensing entities that have been used in the recording session.

In the following two sections, the two features are explained in detail, whereas the focus lies on the realization in the prototypical OPPORTUNITY framework. The last section closes the paper with a conclusion and an outlook to future work.

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.

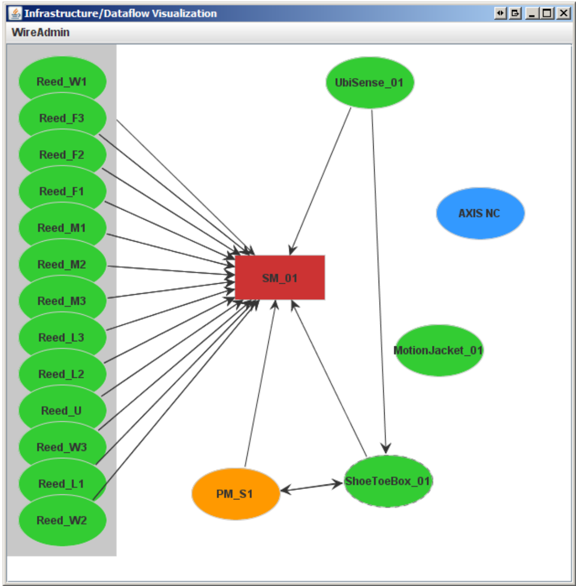

Figure 1. Screen dump of the graphical visualization of the dataflow in the OPPORTUNITY framework.

DYNAMIC ORGANIZATION OF DATAFLOW

As already mentioned, the OPPORTUNITY framework is developed as prototypical implementation of an opportunistic activity and context recognition framework using the Open Services Gateway Initiative (OSGi)[1] [1, 3]. Reasons for using OSGi as base for the framework implementation are (i) the universality of code deployment, (ii) the profound life cycle management capabilities, (iii) the portability and compatibility with various different hardware platforms, (iv) the modularity and component oriented paradigm, and (v) the ability to install, uninstall, restore, start and stop applications at the runtime of the system. The framework partly utilizes software simulations of physical sensor devices which were used at the OPPORTUNITY dataset recording sessions [6, 7]. The sensor simulation device acts as if the physical sensor would be attached to the system, with the same sampling rate and the same data samples. The only difference to a physical sensor is that the data is not achieved by measuring some environmental quantity at real time, but replayed from a data file that stores the pre-recorded data samples. From the framework's point of view it makes no difference, whether a real, physical sensor delivers data, or a software simulation. This makes the constitution of exemplary scenarios very flexible by mixing physical with simulated sensor devices. Furthermore, to simulate sensor disconnects as they might happen in the sensor infrastructure, each software simulation of a sensor device can be comfortably stopped and re-started over a framework controller interface.

Figure 1 presents a screen dump of the graphical visualization of the sensing infrastructure in the OPPORTUNITY framework. The colored ellipses are sensor devices that are currently connected and accessible by the system. The rectangle in the middle of the figure represents the sensing mission, which is a machine-readable translation of the dynamically requested recognition goal. To enable the dataflows between the entities in the framework, the implementation uses the OSGi Wire Admin service specification[2]. The arrows in Figure 1 visualize the currently available information/data flows.

[1]http://www.osgi.org

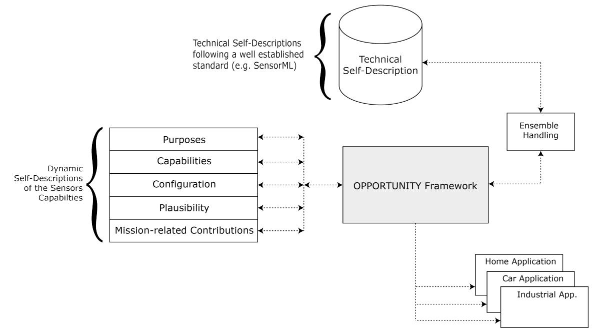

Figure 2. The sensor's self-description concept, illustrating the separation between the technical and the dynamic, changeable sensor self-description.

As an opportunistic system neither presumes (i) a static and fixed sensor infrastructure, nor (ii) predefined recognition goal(s), the OPPORTUNITY framework has to autonomously deal with the occurring dynamics in the sensor topology and the goal. For a recognition goal, the best available set of sensors has to be found and configured and the system has to react on topological changes at runtime. This also includes the automatic handling of the information/data flows in the system among the sensing devices themselves to exchange information and among sensing devices and sensing missions, which consume the data and try to classify the data samples into a set of output classes (recognized activities). The *wires* - as we call the connections in the OPPORTUNITY framework - are dynamically organized by an internal planner that decides which information/data flow is relevant at a certain time spot according to a given recognition goal. Therefore, the planner uses different pieces of information that are available within the framework to decide whether a wire is created or not:

- The sensor devices are able to self-describe by providing a machine-readable file. This self-description can be divided in two parts (see Fig. 2):
  - a technical and thus static self-description containing physical properties (e.g. size, weight, dimension), working characteristics, power requirements and available interfaces.
  - a changeable over time and thus dynamic self- description containing spatial information (location, position), the sensors capabilities related to sensing missions it has contributed in the past, sensor specific configuration details and the sensor's self-awareness concerning the plausibility of the delivered data samples.

[2]http://www.osgi.org/About/Technology

- The translated recognition goal in form of a sensing mission. The system knows what has to be recognized by applying a knowledge-base. By using semantic relations between the sensor systems and their self-descriptions and the steps in the recognition goal, the system can decide whether a sensor might be useful for a very sensing mission or not.

- Once a sensor is part of an ensemble that is configured to execute a given sensing mission, the sensor's contribution to this very mission can be quantified and stored in its self-description and used as experience value for future recognition goals.

- The information, how the sensor infrastructure is currently configured, which sensor systems are available, and which wires are currently defined can also be a valuable source of information for the automatic organization of wires in the framework.

DYNAMIC INCORPORATION OF TRAINING SOURCES

For a sensing mission to be correctly executed, each sensor, queried by the framework, will have to deliver data, which can simply be raw measurements of some physical quantity, but also a more elaborate piece of information, like an activity or a gesture which the user is executing. The traditional way for a sensor to deliver such elaborate pieces of information is to have the raw data processed by a feature extraction block and the features classified with a pattern recognition algorithm. This approach needs the classifiers to be trained beforehand to learn the matching between example features and activities. This is normally done offline in an ad hoc phase, were the training is done with manually labeled examples, which makes it time consuming and makes the system static. On the contrary, in an opportunistic system, where new sensors appear and are included in the sensing mission at runtime, we also have to conceive the training phase as something seamlessly integrated into the operation of the system. The feasibility of this from the machine learning point of view has been shown already in [4], where the relevant requirements are stated.

Our framework structure features data producers and data consumers and wires routing the data when needed and where needed: this allows for online dynamic incorporation of training sources, e.g. labels coming from users or output coming from other entities, like sensors delivering data to classifiers, which in turn output activitiy classes. The structure of the framework allows a great flexibility: a label source can be connected by a wire to a sensor, equipped with a classifier capable of incremental training and this acts as a learning entity, and associates the sensor data to the labels coming from the label source. In the meantime, the classifier can deliver the activity classes for which it has already enough training data. The ease with which wires can be connected between entities allows the framework to train a new sensor as it appears, and nearly immediately start using it also as a source of information, while it can still learn from other entities.

A more advanced role of the framework is the planning of how the data have to be routed, including labels. The planner will decide which data producers can be used immediately, which can be exploited after some training and which are probably not relevant for the sensing mission that has to be executed. The conceptual flow of events for the incorporation of a new training source is the following:

- The framework knows which resources are available and where, because
    a. Sensors provide the technical self-description;
    b. Sensors self-advertise where they are (for example on a certain body segment, on an appliance etc.);
    c. Sensors self-advertise their capabilities, i.e. what they can eventually recognize (in combination with a classifier) in terms of activities;
- A new sensor appears and provides all the pieces of information in terms of the self* properties just listed;
- The planner decides if there is benefit in integrating this new sensor, based on its self-characterization, and eventually decides which of the existing resources can be used to provide training for this sensor;
- In case of a positive answer, a wire is connected to route the relevant labels to the new sensor, so that incremental learning occurs;
- When the new sensor has enough training, the sensing ensemble is reconfigured in order to incorporate the new sensor;
- Eventually, the "changeable" part of the self-description can be updated, for example including the new sensor capabilities.

The framework is then able to deal with open-ended systems also with respect to the use of resources to provide training to others, without interrupting the normal operation of the system and without having to enter any special mode of operation.

CONCLUSION AND FUTURE WORK

In this paper we have shown two fundamental elements of the OPPORTUNITY framework. The dynamic organization of the data flow in the framework by connecting wires between modules to run a specific sensing mission is required due to unknown and changing sensor network topologies and the ability of the framework to handle abstract recognition goals that are stated at runtime. The second feature, the dynamic incorporation of training sources enables the seamlessly performing of incremental training of classifiers that are connected to sensors by just routing new wires between producers of labels and new resources that need to improve their capabilities.

The next step is the more precise definition and implementation of the planner, which orchestrates all the system dynamics. The planner will have to decide which resources have to improve their capabilities to enable a reconfiguration of the ensemble. Since a reconfiguration can be costly, there will always be a tradeoff between effort in doing the training and reconfiguration and the benefit of the new ensemble for the current sensing mission. Additionally, the dynamic part of the sensor self-description, which is needed by the planner to generate dynamic connections between entities in the framework, has to be further advanced.

ACKNOWLEDGEMENTS

The project OPPORTUNITY acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 225938.

REFERENCES

1. O. Alliance. OSGi Service Platform Release 4. *architecture*, page 9, 2000.

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

3. N. Bartlett. OSGi in practice. *Bd. January*, 11, 2009.

4. A. Calatroni, C. Villalonga, D. Roggen, and G. Tröster. Context cells: Towards lifelong learning in activity recognition systems. In *Proceedings of the 4th European Conference on Smart Sensing and Context (EuroSSC)*, LNCS. Springer, 2009.

5. A. K. Dey and G. D. Abowd. The Context Toolkit: Aiding the Development of Context-Aware Applications. pages 434–441. ACM Press, 1999.

6. P. Lukowicz, G. Pirkl, D. Bannach, F. Wagner, A. Calatroni, K. Förster, T. Holleczek, M. Rossi, D. Roggen, G. Troester, J. Doppler, C. Holzmann, A. Riener, A. Ferscha, and R. Chavarriaga. Recording a complex, multi modal activity data set for context recognition. In *Proceedings of the 1st Workshop on Context-Systems Design, Evaluation and Optimisation (CosDEO 2010)*, Hannover, Germany, February 2010. VDE Publishing House.

7. D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, F. Wagner, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, M. Creatura, and J. del R. Milln. Walk-through the opportunity dataset for activity recognition in sensor rich environments, May 2010.

8. D. Roggen, K. Förster, A. Calatroni, A. Bulling, T. Holleczek, G. Troester, P. Lukowicz, G. Pirkl, D. Bannach, A. Ferscha, A. Riener, C. Holzmann, R. Chavarriaga, and J. del R. Milln. OPPORTUNITY: activity and context awareness in opportunistic open-ended sensor environments. In *Poster at the 1st European Future Emerging Technologies Conference (FET 2009)*, Prague, Czech Republic, April 2009.

9. D. Roggen, K. Förster, A. Calatroni, T. Holleczek, Y. Fang, G. Troester, P. Lukowicz, G. Pirkl, D. Bannach, K. Kunze, A. Ferscha, C. Holzmann, A. Riener, R. Chavarriaga, and J. del R. Milln. OPPORTUNITY: Towards opportunistic activity and context recognition systems. In *Proceedings of the 3rd IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2009)*, Kos, Greece, June 2009. IEEE CS Press.

10. T. Stiefmeier, D. Roggen, G. Troster, G. Ogris, and P. Lukowicz. Wearable activity tracking in car manufacturing. *Pervasive Computing, IEEE*, 7(2):42–50, April-June 2008.

11. M. Tentori and J. Favela. Activity-aware computing for healthcare. *IEEE Pervasive Computing*, 7:51–57, 2008.