Modeling Wireless Sensor Networks based Context-Aware Emergency Coordination Systems

Werner Kurschl
School of Informatics,
Communications, and Media
Upper Austria University of
Applied Sciences
Softwarepark 11
A-4232 Hagenberg, AUSTRIA
kurschl@fh-hagenberg.at

Stefan Mitsch
Research Center Hagenberg
Upper Austria University of
Applied Sciences
Softwarepark 11
A-4232 Hagenberg, AUSTRIA
smitsch@fh-hagenberg.at

Johannes Schönböck
Research Center Hagenberg
Upper Austria University of
Applied Sciences
Softwarepark 11
A-4232 Hagenberg, AUSTRIA
jschoenb@fh-hagenberg.at

Wolfgang Beer
Software Competence Center Hagenberg GmbH
Softwarepark 21
A-4232 Hagenberg, AUSTRIA
wolfgang.beer@scch.at

ABSTRACT

Current research on context information focuses on the use of new kinds of sensors and on the aggregation and interpretation of sensor data. Having a closer look at applications supporting emergency scenarios with context information several problems arise. Emergency activities are complex coordination tasks which involve a multitude of different roles and resources. The definition of emergency scenarios and the mapping of relevant context information is a time-consuming and error-prone task.

This work discusses how a model-based approach can support the definition of emergency scenarios at an abstract level. The abstract representation (PIM, Platform Independent Model) of an activity is then transformed into a platform-specific model (PSM), which includes a collection of context sensors. We show that the use of a model-based approach simplifies the definition of complex context-sensitive applications, and how this increases the flexibility to use different sensor platforms in different emergency scenarios.

1. INTRODUCTION

Control centers in industrial plants continuously monitor the state of the plant and its machines, but control center operators often desire additional context information of technicians working on specific tasks, such as the technicians' locations, to complete their view. The situation is further complicated if third party employees are working on the plant or visitors are given a tour. In case of an emergency, such as a gas emanation or a fire on a plant, it is to date neither possible for the operator, nor for emergency response teams, to keep track of individuals involved in the incident. Current emergency coordination heavily depends on predefined rescue routes, staging points or safe areas, and voice communication. Such emergency activities are complex coordination tasks which involve a multitude of different roles and resources. We believe that context information—like the individuals' locations, physical conditions, as well as the condition of their environment—can enhance monitoring and emergency coordination applications to improve industrial safety by continuously providing the desired information. Thereby, control center operators and emergency response team leaders are enabled to instantly react on unpredictable situation changes. Moreover, individuals directly affected by an emergency incident would also benefit from such a system. We envision, e.g., to check the state and the risk of using a specific rescue route and dynamically guide individuals at the incident's site to less dangerous routes.

Technological advances of sensor networks and mobile devices enable the implementation of an advanced emergency control system, which reduces the importance of voice communication and gains information on individuals that are not able to communicate due to their physical condition. [17] rates reduced voice communication as a good design choice, because the authors' interviews revealed that major parts of the radio communication consisted of simple "yes/no" questions which could be handled more efficiently with alternative communication media. In order to minimize the complexity often incurred with technological advances, we envision to monitor the defined activities on an abstract or situational level (e.g., the system shows the situations "individuals in danger" or it marks emergency exits as being "dangerous") instead of displaying raw sensor data. Therefore, it is necessary to transform the abstract activities into a technological description of the emergency coordination system, which contains the information about physical sensors that are needed to gather the necessary context information (e.g., temperature, smoke, or wind direction). This

iiWAS 2008, November 24–26, 2008, Linz, Austria.

work introduces necessary main components and a modeling approach in order to efficiently develop complex emergency coordination systems.

2. ENVISIONED APPLICATIONS OF CONTEXT INFORMATION

The analysis of emergency coordination suggests possible applications of context information. We describe the most interesting ones in the following subsections.

2.1 Tracking the Location of Individuals and Assets in Heterogeneous Environments

Continuously tracked outdoor and indoor location information with adequate precision is the basis for emergency response team guidance. By that team members do not need to search the plant for potential casualties, but instead the location tracking system can guide them directly to injured individuals. Additionally, the position of team members could be monitored throughout their mission and, thus, increase their safety as well. Of course the location tracking system cannot guarantee the absence of individuals within a critical section; still it offers a valuable source of information for emergency response teams as also stated in [17]: firefighters “generally want to see information about the local incident scene” and “they will frequently want to know locations of the fire, stairways, elevators, standpipes, and other personnel”.

2.2 Sensing Environmental and Vital Parameters

Environmental parameters, like temperature, wind direction, carbon monoxide and carbon dioxide content, can either be captured by sensors installed in the working environment or by sensors worn by individuals (integrated in items needed for their daily work or in their working clothes, see [6] and [14]). These environmental parameters can support commanders in planning evacuations, to extinguish fire more efficiently, and to control emergency exit signs to disable dangerous emergency exits. Some of the preconfigured emergency paths may become invalid, if environmental data shows, e.g., high temperature or dense smoke. Lifenet described in [9] uses a similar approach to guide fire fighters to the next appropriate exit. In addition to environmental parameters vital parameters can be used to monitor the physical condition of firefighters and keep firefighters at a scene just as long as it is not harmful to them. This aspect is backed by [17], where firefighters mentioned that they “do not want to be pulled from a scene prematurely when they feel ok”, but they also fear, that pension/insurance benefits will decline if they have pushed themselves beyond reasonable limits after being warned to stop.

2.3 Interpreting the Sensed Parameters and Deriving Actions

Huge data volumes of sensed data in large-scale incidents could easily overburden commanders. It is therefore necessary to derive more significant, high-level context data (i.e., compute semantics based on quantitative low-level sensor data). Rules to initiate actions—e.g., a warning message to firefighters or notify emergency exit labels to visually block certain exits—can then be defined easily.

2.4 Analyzing Tracks of Individuals

The data collected during incidents or fire practices can provide a high potential for post-mortem analysis of emergency response plans, because the dynamic behavior of each individual as well as the development of the incident can be studied in detail. A replay mechanism could help to identify strengths and weaknesses in existing emergency response plans (e.g., by measuring the mean distance covered by individuals, or the time taken to get out of the building).

These four application scenarios led to the definition of a context-aware emergency coordination system, which is described in the next section.

3. MAIN COMPONENTS

The proposed context-aware emergency coordination system consists of the following main components depicted in figure 1. A wireless sensor network (which consists of location tracking, health, and environmental sensors that ideally form an ad-hoc mesh network for increased reliability) delivers low-level context data to a context recognition framework (the sensor abstractions). These sensor abstraction components integrate heterogeneous hardware platforms (location tracking systems, environmental and vital sensors) by mapping from platform-specific data formats to a common platform-independent context data model. The knowledge base also adheres to this context data model and accumulates the provided sensor information. A rule engine follows a set of user-defined rules to infer actions and inform client applications about low-level context data. Different client applications—based on a geographic information system (GIS) component—visualize building plans augmented with additional information like mission overview, alarms, and notifications.

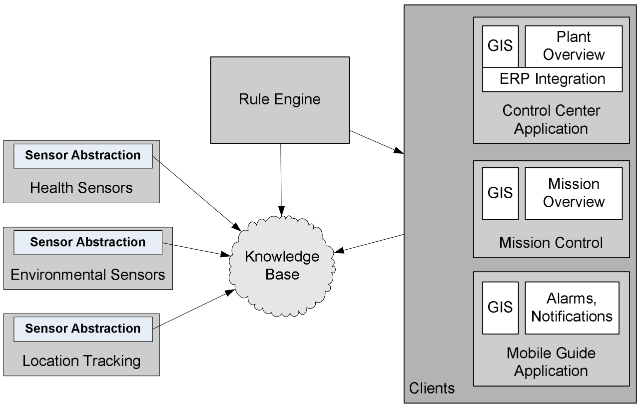

Figure 1: System Main Components

The remainder of this section describes the system components in more detail.

3.1 Sensors and Sensor Abstraction

Location tracking systems used in industrial environments have to fulfill diverse requirements (see [12]). Industrial plants cover large areas comprising indoor and outdoor areas. Having a detailed look on specific tracking technologies (see [13] and [7]) it is obvious that a single technology can hardly satisfy these requirements as certain technologies cannot cope with specific environmental influences. It is necessary to integrate several systems into a common location tracking system to provide the possibility to choose the optimal system for a specific environment. In [13] a software prototype is presented that allows the seamless integration of different location tracking sensors.

Current location tracking systems hardly include any context information. This might be due to the fact that most available tracking hardware cannot gather additional context information. In [11] wireless sensor networks are shown to be suitable for industrial tracking solutions. This tracking system could easily integrate context information by adding sensor boards to the motes used for location tracking. To save hardware and network costs the location tracking and sensing hardware should be combined. Nevertheless this is no prerequisite of the described system as the location information and the additional context information are stored in a context model (see section 3.2), which is able to find links between all sorts of context information (including locations).

Wireless sensor networks enable sensing, aggregating, processing, and communicating of environmental and vital parameters. Current sensors are extremely specialized in their functionality: almost every parameter can be measured with a specialized sensor type. Given the diversity of environmental and vital parameters envisioned in the emergency activities described in section 2, one can easily imagine the diversity of sensors needed in an emergency coordination system. Therefore, each sensor type needs a sensor abstraction component that ensures compatibility with the system's knowledge base context model (described in the next section).

3.2 Knowledge Base

Data provided by different tracking and sensing components needs to be represented in a uniform context model. [1] and [18] present surveys on context models that vary from simple key-value models to ontologies. The Aspect-Scale-Context Information (ASC) model described in [19] as part of the CoOL ontology and the context atom attributes listed in [1] are a flexible and general basis for building a context model describing low-level context (i.e., raw sensor data), but the model lacks a possibility for detailed descriptions of system entities (hardware components, users, individuals, etc.). Figure 2 on the next page shows the extended ASC context model for describing data in the knowledge base.

Every sample describes a certain Aspect (e.g., location or temperature). Depending on the hardware the sample's value is interpreted according to a particular Scale. Scales additionally offer methods to convert between different scales of one aspect. This is for example useful to convert geometrical (e.g., WGS-84 coordinates) to symbolic location information (e.g., "room 123") or to treat different temperature units (e.g., Celsius and Fahrenheit).

A context information sample not only represents a raw value obtained from a sensor (the context information sample's source), it also characterizes a particular entity. [3] distinguishes between social, physical, and computational entities. We extend their ontology with an *isRelatedTo* relation type, which associates entities to each other according to some relation (e.g., temporal or spatial). During system deployment stationary entities need to be configured in the system: e.g., spatial relations between rooms in a building (i.e., the building's map), or the spatial relations between rooms and location beacons (i.e., the beacons' installation locations) must be set up. In our previous work (see [10]) we describe how spatial relations between rooms and location beacons can be obtained automatically to simplify location tracking system deployment.

During runtime preconfigured and sensed relations between entities enable the emergency system to reason about the current situation. We can thereby answer questions like "How many individuals are in a room with a temperature higher than 70°C" or "What is the shortest path, if any, between room A and room B, avoiding rooms with a CO concentration higher than some threshold value".

3.3 Model-Based Context Framework Development

To gather low-level context data that conforms to the ASC context model specific hardware for each aspect is needed. Today there are numerous hardware sensors from different vendors using incompatible software platforms available. Here the question arises how to handle these different platforms efficiently. The main goal of Model Driven Architecture (MDA) (see [15]) is to model a system at an abstract, platform independent level first. By using model transformations the abstract model can be adapted to a certain platform. Often MDA is combined using Domain Specific Languages (DSL). Thus we are providing a graphical DSL that enables programmers to model applications based on a pipes-and-filters architectural style (see [4]).

The platform independent meta model of our DSL is depicted in Figure 3: it defines the basic elements for modeling situation inference components.

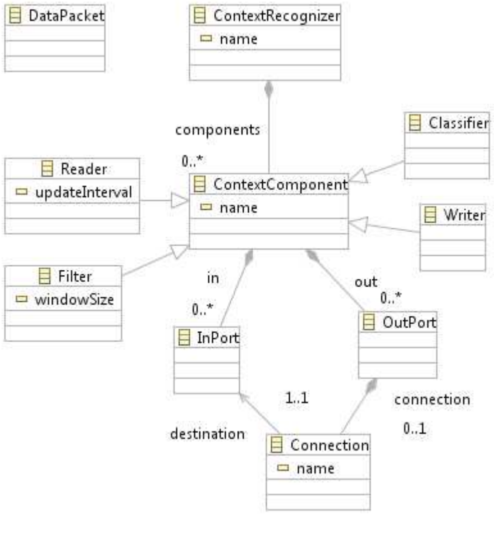

Figure 3: Platform Independent Meta Model

The class *ContextComponent* acts as an abstract base class for all components; concrete classes (not shown within the figure) provide specific behavior, e.g., a *MinFilter* component calculates the minimum value of incoming *DataPackets*. DataPackets hold the actual data delivered by a physical sensor. ContextComponents can be connected via *InPorts* and *OutPorts* using *Connections* that transfer data from one component to another. If components run on different hardware devices or processes specialized connections hide

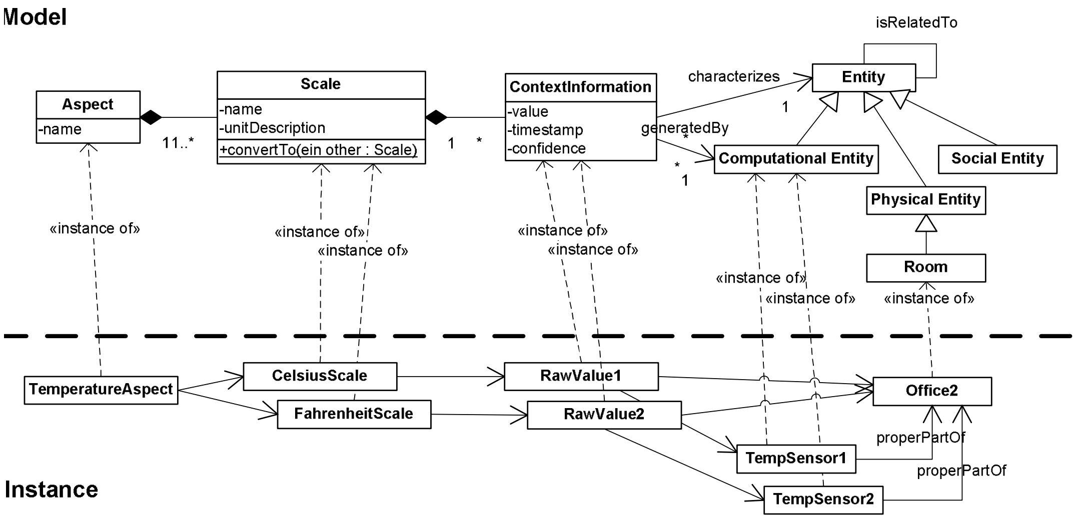

Figure 2: ASC Context Model

the specifics of remote calls. The *ContextRecognizer* class aggregates all ContextComponents and Connections within a certain model and thus forms the whole pipeline system.

The meta model provides four different abstract base type components, namely Reader, Writer, Filter and Classifier. A reader provides data from a specific source (e.g., a sensor), whereas a writer forwards data to a specific sink (e.g., file, database or actuator). Filters extract features from raw sensor data. These features can be used in classifiers to recognize situations.

We provide a graphical editor (a graphical representation of the DSL) using the Graphical Modeling Framework[1] of the Eclipse foundation to model context-aware applications in a platform independent way. The platform independent model (PIM) forms the basis of the whole application. This model can be transformed, as depicted in figure 4 on the next page, to several platform-specific models (PSMs).

We allow the programmer to transform some components of the PIM to one PSM (e.g., NesC) and others to a different PSM (e.g., Java). This is due to the fact that resource intensive calculations (e.g., a classification algorithm) cannot be executed on the resource-constrained sensor hardware, whereas others, like simple filter operations, can be done on the hardware itself. This so called in-network data processing can improve the scalability and can reduce the energy and other resource consumption, since it can significantly reduce the data volume that has to be routed through the network (see [16]).

The PSM is then used to generate platform specific code. Figure 5 shows the template based code generation process using the XPand[2] template language. The code generator uses an instance of the PSM and mainly generates the glue code. The implementations of the components are provided by an archive file implemented in the platform-specific language, which is referenced by the generated code.

We currently defined the meta model in Ecore using the Eclipse Modeling Framework. NesC and Java are supported as different platforms. To integrate a new platform the programmer needs to specify the transformation from the above presented PIM to the new PSM by using a transformation language like the Atlas Transformation Language[3]. Based on the Eclipse plug-in and extension point mechanism code generators for new platforms can be easily integrated

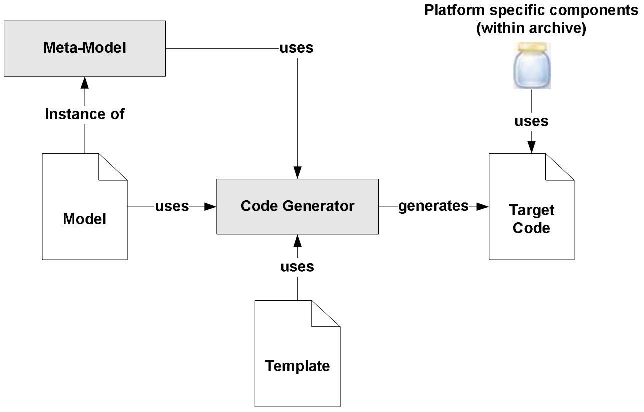

Figure 5: Code generation

4. RELATED WORK

The Context Recognition Network (CRN) Toolbox (see [2]) describes a C++ framework integrating hardware abstraction, filter algorithms, feature extraction components and classifiers in a configurable runtime to support rapid development of context recognition applications. It is designed for deployment to embedded devices that support the POSIX runtime environment. Although a graphical editor is provided it is not based on a formal meta model and thus the CRN Toolbox cannot fully benefit from a MDSD approach. As we base our toolbox on a formal meta model we are able to generate code for various platforms, including the CRN Toolbox, by exchanging the generator templates.

---

[1] `http://www.eclipse.org/modeling/gmf/`

[2] `http://www.eclipse.org/gmt/oaw/doc/4.1/r20_xPandReference.pdf`

[3] `http://www.eclipse.org/m2m/atl/`

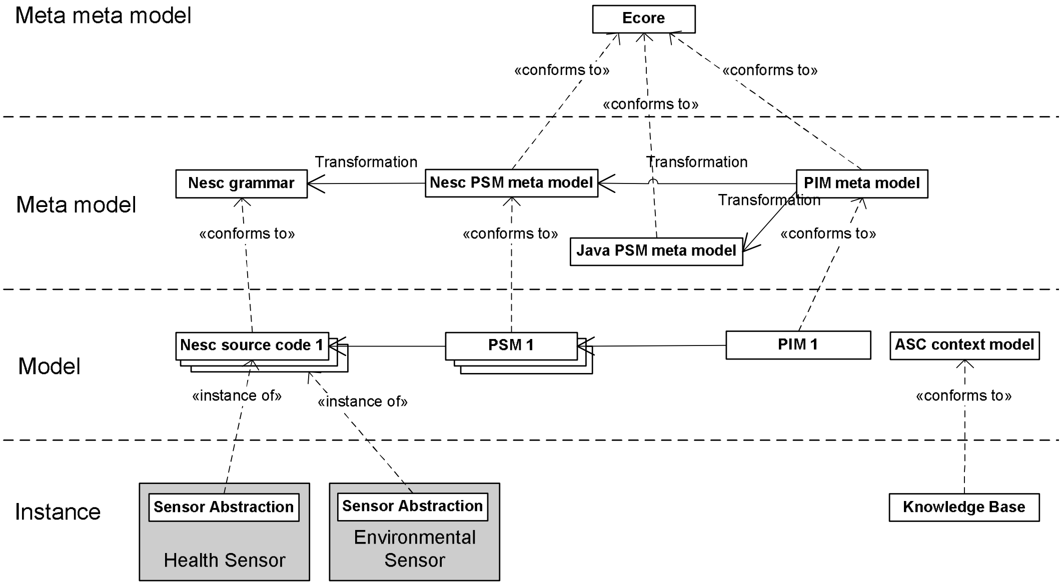

Figure 4: MDSD approach

Chaczko an Ahmad describe in [5] a wireless sensor network based application to detect fires in rural areas. It uses temperature, humidity, and smoke sensors. The main focus of their work was to develop a reliable group communication with limited transmission power and minimal computational resources. The self-organizing network uses prioritized data network transmission in the case of an emergency or a fire, but the system does not contain any notion of location and high-level modeling or processing of context data.

The Siren framework proposed by [8] tries to support tacit communication between firefighters. Environmental data is collected over a redundant wireless network by sensors, which are either placed at strategic locations (e.g., at smoke detectors) deployed on the fly by firefighters. Firefighters carry PDAs that exchange sensor information on a peer-to-peer basis. A built-in rule engine interpretes raw sensor data to generate alerts (which is a very weak way of deriving semantics); signal processing to enhance raw data quality and reasoning upon the gemerated semantics are completely ignored. Moreover, Siren obviously lacks formal data models and application meta models. Their data model—a tuple space—uses fixed relations between entities (e.g., locations can only be related to each other with one of four directions); thus, applications are not able to define their own specialized relations (e.g., to introduce spatial concepts like locations being near each other or far apart). The authors also assume a single platform and therefore do not provide an application meta model to develop applications for different platforms.

The LifeNet navigation system for firefighthing described in [9] uses sensor networks to provide relative positioning under limited visibility (which is their main distinguishing feature in comparison to Siren). The navigational aid is limited to directional indications so that firefighters are guided from one sensor node to the next on a trail leading out of a building. To date, LifeNet communicates only very little context information to firefighters; moreover, it does not process context information to derive semantics, though at least it seems to be achievable within the system. Just like Siren it does not support platform-independent modeling and different platforms.

5. CONCLUSION AND FURTHER WORK

Wireless sensor networks have emerged as a suitable technology for collecting environmental data and they can successfully be applied to obtaining vital data of individuals. Tracking solutions also matured in the past few years, so that they can be applied in harsh industrial environment, but wireless sensor networks are rarely used to augment them. An analysis of emergency coordination revealed possible application scenarios. In this paper we presented methods how wireless sensor networks and tracking solutions can be combined to a context-aware emergency coordination system. We also introduced necessary components and a modeling approach to efficiently develop such systems. We believe that the use of a model-based approach simplifies the implementation of complex context-aware applications; aditionally, it provides flexibility to use different sensor platforms during different emergency scenarios. The combination of numerous sensor types provides detailed insight into emergency situations; moreover, a context-aware system effectively unburdens commanders in their work during large-scale incidents. Envisioned improvements of the system further elaborate the use of high-level context in that domain to predict an incident's near future (e.g., which areas will be affected next).

6. REFERENCES

[1] M. Baldauf, S. Dustdar, and F. Rosenberg. A Survey on Context-Aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[2] D. Bannach, O. Amft, and P. Lukowicz. Rapid Prototyping of Activity Recognition Applications. *IEEE Pervasive Computing*, 7(2):22–31, 2008.

[3] Y. Bu, J. Li, S. Chen, X. Tao, and J. Lu. An enhanced ontology based context model and fusion mechanism. In *LNCS 3824 : Proceedings of IFIP 2005 International Conference on Embedded and Ubiquitous Computing (EUC2005)*, volume 3824 of *Lecture Notes in Computer Science (LNCS)*, pages 920 – 929. Springer Verlag, Noveember 2005.

[4] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-Oriented Software Architecture. A System of Patterns*. John Wiley & Sons, 1996.

[5] Z. Chaczko and F. Ahmad. Wireless sensor networks based systems for fire endangered areas. In *Proceedings of Third International Conference on Information Technology and Applications (ICITA)*, pages 203–207. IEEE Computer Society, 2005.

[6] A. Gaggioli, M. Vettorello, and G. Riva. From cyborgs to cyberbodies: The evolution of the concept of techno-body in modern medicine. *Psychnology*, 1(2):75 – 86, 2003.

[7] J. Hightower and G. Boriello. A survey and taxonomy of location systems for ubiquitous computing. Technical report, Technical Report UW-CSE 01-08-03, 2001.

[8] X. Jiang, N. Y. Chen, J. I. Hong, K. Wang, L. Takayama, and J. A. Landay. Siren: Context-aware Computing for Firefighting. In *Proceedings of the Second International Conference on Pervasive Computing (PERVASIVE 2004)*, pages 87–105, Linz/Vienna, Austria, 2004. Springer-Verlag.

[9] M. Klann, T. Riedel, H. Gellersen, C. Fischer, M. Oppenheim, P. Lukowicz, G. Pirkl, K. Kunze, M. Beuster, M. Beigl, O. Visser, and M. Gerling. Lifenet: an ad-hoc sensor network and wearable system to provide firefighters with navigation support. In *Adjunct Proc. Ubicomp 2007*, pages 124–127, September 2007.

[10] W. Kurschl, W. Gottesheim, S. Mitsch, R. Prokop, and J. Schönböck. Automatic Position Determination of Fixed Infrastructure Sensor Network Nodes based on Topology Sensing and Maps. In *Proceedings of The 2008 International Conference on Wireless Networks (ICWN), Las Vegas, NV, United States of America*, 2008.

[11] W. Kurschl, W. Gottesheim, S. Mitsch, R. Prokop, J. Schönböck, and W. Beer. A Two-Layered Deployment Scheme for Wireless Sensor Network based Location Tracking. In *Proceedings of 5th International Conference on Information Technology : New Generations, Las Vegas, NV, United States of America*, 2008.

[12] W. Kurschl, W. Gottesheim, S. Mitsch, R. Prokop, J. Schönböck, and W. Beer. Large-Scale Industrial Positioning and Location Tracking - Are We There Yet. In *Proceedings of 7th International Conference on Mobile Business, Barcelona, Spain*, 2008.

[13] W. Kurschl, W. Gottesheim, S. Mitsch, R. Prokop, J. Schönböck, and W. Beer. Towards a Unified Location Tracking System for Heterogeneous Industrial Environments. In *Proceedings of IEEE Indin 2008, 6th International Conference on Industrial Informatics, Daejeon, Korea*, 2008.

[14] Z. S. Nakad. *Architectures for e-Textiles*. PhD thesis, Virginia Polytechnic Institute and State University, 2003.

[15] Object Management Group. OMG Model Driven Architecture. `www.omg.org/mda`. last accessed 07.2008.

[16] K. Römer. *Time Synchronization and Localization in Sensor Networks*. PhD thesis, Swiss Federal Institute of Technology Zürich (ETH Zürich), 2005.

[17] D. Steingart, J. Wilson, A. Redfem, P. Wright, R. Romero, and L. Lim. Augmented cognition for fire emergency response: An iterative user study. In *Proceedings of the 1st International Conference on Augmented Cognition, Las Vegas, NV*, 2005.

[18] T. Strang and C. Linnhoff-Popien. A Context Modeling Survey. In *First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004, Nottingham, England, September 7, 2004*, 2004.

[19] T. Strang, C. Linnhoff-Popien, and K. Frank. CoOL: A Context Ontology Language to enable Contextual Interoperability. In J.-B. Stefani, I. Dameure, and D. Hagimont, editors, *LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, volume 2893 of *Lecture Notes in Computer Science (LNCS)*, pages 236 – 247, Paris/France, November 2003. Springer Verlag.