

# A Comprehensive Approach for Situation-Awareness based on Sensing and Reasoning about Context

Thomas Springer<sup>1</sup>, Patrick Wustmann<sup>1</sup>, Iris Braun<sup>1</sup>, Walteneus Dargie<sup>1</sup>, and  
Michael Berger<sup>2</sup>

<sup>1</sup> TU Dresden, Institute for Systems Architecture, Computer Networks Group

<sup>2</sup> Siemens AG, Corporate Technology, Intelligent Autonomous Systems CT IC 6

{Thomas.Springer, Patrick.Wustmann, Iris.Braun,  
Walteneus.Dargie}@tu-dresden.de, M.Berger@siemens.com

**Abstract.** Research in Ubiquitous Computing and Ambience Intelligence aims at creating systems able to interact in an intelligent way with the environment, especially the user. To be aware of and to react on the current situation usually a complex set of features describing particular aspects of the environmental state has to be captured and processed. Currently, no standard mechanisms are available to model and reason about complex situations. In this paper, we describe a comprehensive approach for situation-awareness which covers the whole process of context capturing, context abstraction and decision making. Our solution comprises an ontology-based description of the sensing and reasoning environment, the management of sensing devices and reasoning components and the integration of these components into applications for decision making. These technological components are embedded into an conceptual architecture and generic framework which enable an easy and flexible development of situation-aware systems. We illustrate the use of our approach based on a meeting room scenario.

## 1 Introduction

Research in Ubiquitous Computing and especially Ambience Intelligence aims at creating systems able to interact in an intelligent way with the environment, especially the user. "Machines that fit the human environment instead of forcing humans to enter theirs will make using a computer as refreshing as a walk in the woods" [1]. Thus, a system able to recognise the current situation can adapt its behaviour accordingly. For instance, an application for supporting mobile workers during their tasks in the field could adapt the interaction modalities to improve the interaction with the user (e.g. speech input and output could be used if the workers hands are not free or gesture input could be used if the surrounding noise level is very high). In a similar way, an assistance application for elderly people could intelligently support planning of daily activities, e.g. selecting convenient connections of public transportation systems for carrying out shopping activities, visiting the doctor or meeting relatives or friends.

To create such systems, sensing the environment and adapting the behaviour according to its current state are major prerequisites. A system has to be able to capture information about the environment and the involved users based on heterogeneous and distributed information sources, e.g. sensor networks, extracted application data, user monitoring or other methods for gathering context information. The information captured in this way is usually lower-level and has to be abstracted and fused to create an understanding of the overall situation a system is currently within. A large set of schemes for reasoning about the current situation exist which include a wide range of logic and probabilistic based estimation, reasoning and recognition schemes most of which have been employed in artificial intelligence, image and signal processing, control systems, decision theory, and stochastic processes. All these schemes have their advantages and disadvantages and can be applied for different types of sensed data and application areas.

Currently, no standard mechanisms are available to model and reason about complex situations. There is no common understanding about what features are relevant for a certain situations and how such features and their interrelations can be identified and modelled. Moreover, the adoption of different reasoning schemes is in an early state, especially with respect to their combination and the overall performance.

In this paper, we describe a comprehensive approach for situation-awareness. In particular, the capturing of low-level context information based on sensor networks and further sensing devices, the abstraction of higher-level context using heterogeneous reasoning schemes and the derivation of system decisions are covered. Our solution comprises an ontology-based description of the sensing and reasoning environment, the management of sensing devices and reasoning components and the integration of these components into applications for decision making. These technological components are embedded into a generic framework and a design methodology which enable an easy and flexible development of situation-aware systems. We illustrate the use of our approach based on a meeting room scenario.

Our paper is organised as follows: Related work in the areas of context sensing and reasoning as well as for architectures and frameworks for context-awareness is discussed in chapter 2. We introduce our concepts, giving a detailed description of the requirements, the major concepts, the proposed architecture together with its components and a development methodology in chapter 3. In chapter 4 we describe the implementation of the generic framework and our feasibility study based on one example scenario is presented in chapter 5. We conclude the paper with summarising the lessons learned and giving an outlook to future work.

## 2 Related Work

Several context reasoning architectures and prototypes have been proposed in the recent past. The Sylph [2] architecture consists of sensor modules, a proxy core, and a service discovery module. A sensor module provides a standard means for initialising and accessing sensor devices. A service discovery module advertises

sensors through a lookup service, and a proxy core manages application queries and serves as a semantic translator between applications and sensor modules. The iBadge prototype, developed with the Sylph architecture, tracks and monitors the individual and social activities of children in kindergartens (loneliness, aggression behaviour, etc.) It incorporates orientation and tilt sensing, environmental sensing, and a localisation unit.

The Mediacup [3] is an ordinary coffee mug to which a programmable hardware for sensing, processing, and communicating context is embedded. The hardware is a circular board designed to fit into the base of the cup. It incorporates a micro controller, a memory for code processing and data storage, an infrared transceiver for communication, and an accelerometers and a temperature sensor for sensing different contexts. The same system architecture was used to embed a sensing system into a mobile phone. A three-layered context recognition framework is used to reason about the status of the mug and the mobile phone. It consists of a sensor layer, a cue layer, and a context layer. The sensor layer is defined by an open-ended collection of sensors which capture some aspects of the real world. The cue layer introduces cues as abstractions of raw sensory data. This layer is responsible for extracting generic features from sensed data, hiding the sensor interfaces from the context layer. The context layer manipulates the cues obtained from the cue layer, and computes context as an abstraction of a real world situation.

SenSay [4] takes the context of a user into account to manage a mobile phone. This includes adjusting the functional units of the mobile device (e.g. setting the ringer style to a vibration mode) or it can be call related. For the latter case, SenSay prompts remote callers to label the degree of urgency of their calls.

Korpiää et al. [5] exploit several sensors to recognise various everyday human situations. The authors propose a multi-layered context-processing framework to carry out a recognition task. The bottom layer is occupied by an array of sensors enclosed in a small sensor box and carried by the user. The upper layers include a feature extraction layer incorporating a variety of audio signal processing algorithms from the MPEG-7 standard; a quantisation layer based on fuzzy sets and crisp limits; a classification layer employing a naïve Bayesian classifier which reasons about a complex context. Their implementation involves a three-axis accelerometer, two light sensors, a temperature sensor, a humidity sensor, a skin conductance sensor and a microphone.

The approaches above support the dynamic binding to context sources. On the other hand, their deployment setting as well as the context reasoning task is predetermined at the time the systems are developed. This limit the usefulness of the systems as users' and applications' requirements evolve over time. Subsequently, there is a need for dynamic integration of sensing, modelling and reasoning. We build upon the experiences of previous work to support flexible context reasoning and usage.

In parallel, efforts were started to create more general models and services to model and reason about context based on ontologies. Recent projects (e.g., [6]) covered the creation of comprehensive and generic context models with the

goal of identifying and integrating characteristics of contextual information. Especially, ontology-based modeling and reasoning is addressed ([7–9]) with focus on knowledge sharing and reasoning. In [8, 7], approaches for defining a common context vocabulary based on a hierarchy of ontologies are described. An upper ontology defines general terms while domain-specific ontologies define the details for certain application domains. Both approaches use a centralized architecture and work on local scenarios from the smart home or intelligent spaces domain. These solutions focus on the modelling of context and apply particular reasoning schemes. Thus, a comprehensive approach starting with sensor integration and classification is not considered. Moreover, the systematic placement of sensors and a decomposition of situations are not addressed.

### 3 Conceptual Architecture

Real world situations usually have to be derived from a complex set of features. Thus, a situation-aware system has to capture a set of features from heterogeneous and distributed sources and to process these features to derive the overall situation. Thus, major challenges for the creation of situation-aware systems are to handle the complexity of the situation and the related features, to manage the sensing infrastructure and to find appropriate reasoning schemes that efficiently derive the overall situation from low-level context features. In the following we present a conceptual architecture for the creation of situation-aware systems. The approach is intended to be comprehensive, i.e. it comprises all components and processing steps necessary to capture a complex situation, starting with the access and management of sensing devices up to the recognition of a complex situation based on multiple reasoning steps and schemes. To handle complex situations the concept of decomposition is applied to the situation into a hierarchy of sub-situations. These sub-situations can be handled autonomously with respect to sensing and reasoning. In this way, the handling of complex situations can be simplified by decomposition.

We focus on sensors installed in the environment, i.e. they are immobile and usually not part of a mobile device carried by the user. The main idea is to exploit cheap and already available sensors to create new or extended applications. For example, buildings are more and more equipped with sensors, e.g. for measuring temperature or light intensity to enable building automation. Such installations can be extended and then exploited to create "more intelligent" situation-aware applications. We start with the discussion of the requirements for our solution and discuss our conceptual architecture in detail.

#### 3.1 Requirements

The goal of our research work was to develop a comprehensive approach for situation awareness which covers the whole process of context capturing, situation recognition based on context abstraction and decision making.

Because of the dynamic properties in the field of situation-awareness, systems have to be flexible and extensible. Therefore, the approach should be independent of the type of information sources involved, i.e. the types of sensors, the structure of particular sensor networks and the different real world situations which could occur. Furthermore, the approach should not depend on the application scenario, the used reasoning schemes and the type of the derived higher-level context. Based on the situation and application-specific use of the sensing infrastructure, the capturing and abstraction should be (re-)configurable, e.g. regarding the frequency of data collection, the used classification mechanisms and the aggregation of sensor values.

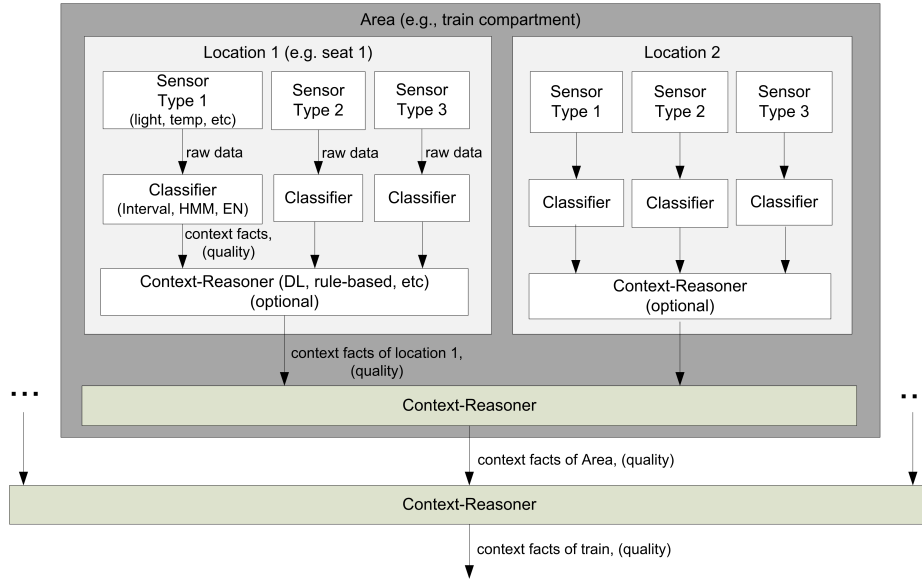
Moreover, scalability and modularity are important. That means, the approach should not restrict the amount of deployed sensors and their distribution. It should also allow a flexible combination of different schemes for reasoning of higher-level context information. For a better reuse, the resulting system should be designed in building blocks, so it is easy to replace a single building block by another one (e.g. a particular reasoning scheme with a different one). Especially, a situation should not be modelled in a monolithic way and varying availability of sensors and resulting incompleteness of information should be manageable.

Last but not least, the system should be independent from a certain application scenario. Rather, it should support the set-up of different scenarios regarding the involved sensors and their configuration, the used reasoning schemes and the overall situation captured. Thus, a formalised scenario description which can be interpreted, instantiated and exchanged should be supported by the system.

### **3.2 Conceptual architecture based on situation decomposition**

Our conceptual framework is based on the decomposition of complex situations. From previous experiments we have observed, that complex situations can be decomposed into sub-situations which can be handled independently with respect to sensing and reasoning. Each sub-situation represents a certain aspect of the overall situation and has to be fused with other sub-situations at a certain level of a hierarchical reasoning process. Below that point, a sub-situation can be handled separately. This enables especially a parallel and distributed sensing and reasoning process. Moreover, for each sub-situation the appropriate reasoning scheme can be applied. At the same time, the decomposition enables the modularization of the system, because sub-situation can be assigned to separate processing modules. Extensibility and flexibility can be supported, because during the decomposition process, alternative sub-situations can be identified and new aspects can be easily added to the hierarchical situation description.

According to our validation example, the overall situation could be the current use of a meeting room. Among others, the meeting room could be empty, used for a discussion between two people, a discussion in a larger group, a presentation, or a party. Each of these instances of the overall situation can be captured based on the aggregation and processing of different aspects of that complex situation. For instance, the number of persons in the room, their activity, the lighting, the noise level and the state of the beamer could be used



**Fig. 1.** Abstraction process for situation detection based on sensor data.

to detect the overall situation. Thus, the initial step for creating a situation-aware system according to our conceptual framework is the decomposition of the complex situation the system should be aware about. The resulting hierarchy of sub-situations can be refined or extended during the development process as well as during the whole lifetime of the system. The leaves of the situation hierarchy represent atomic sub-situation which can not or should not be decomposed further. These atomic sub-situations are the starting points for the creation of the situation-aware system.

To reflect all necessary steps to derive the overall situation from sensed information, our conceptual architecture consists of three layers: a sensing layer, a feature extraction layer and a reasoning layer. These layers are depicted in figure 1. Each of these layers will be described in more detail below.

**Sensing Layer** The sensing layer comprises solutions for two major issues, namely the integration of heterogeneous sensing devices and the placement and organization of sensing devices according to their semantic relation to sub-situations. In fact, there is a broad range of different sensors which can be considered for gathering context information like audio, video, a whole wireless-sensor network, etc. These sensors have to be accessed with the help of a specific programming interface. So the sensors deliver different types of (raw) values. Usually the interfaces are provided by the manufacturer of the sensors. Therefore mainly the sensors can be used directly with only little implementation effort, except the special implementation of a wireless-sensor network, for implementing spe-

cial sensor configurations. Usually, sensors relevant for a certain sub-situation belong to a certain location. Thus, in contrast to other approaches, which often assume a uniform or random distribution of sensing devices and calculate average values out of several values of sensors of the same type in a certain area, we identify distinct "areas of interest" which are relevant to capture sensor data relevant for a certain sub-situation. For instance, the presentation area or a table could represent such an area of interest (see figure 3). At these areas of interest different types of sensor devices should be placed and logically grouped.

Thus, in our concept each sensor is dedicated to a certain area of interest. Especially, sensors of the same type which belong to different areas of interest are handled separately in the classification and lower-level reasoning steps. The idea is, that an average value of all sensors for the light level at a certain location could be useless for capturing a certain situation, while the values of the light level at certain areas of that location could have a high relevance (e.g. at the projection area of a beamer to decide if the beamer is on or off, or at the surface of a seat to decide if the seat is taken or free). The information of the organisation of a set of sensors into an area of interest is exploited in the classification and reasoning steps described next.

**Feature Extraction Layer** Because of the heterogeneous sensing devices and the resulting different sensor values, each value has to be classified by an appropriate classifier. Classifiers are dividing the sensor data into individual, application depended classes. These classes are labelled with a symbolic name. The used classifiers can be Neural Networks, Hidden-Markov-Models, Decision Trees, Rule-Sets, Bayesian-Nets, Clustering with a matching heuristic or simply a quantisation over the data. Because of the possible use of different classifiers for the different sensors it is difficult respectively almost not possible in addition to the classified sensor data to have a common quality statement out of the different classifiers, which can be used as quality statements, which influence the reasoning steps. Therefore, the results of the classifiers are logic facts of the particular sensors on their areas of interest. These facts are forwarded then to the reasoning steps.

**Reasoning Layer** Reasoning is in our concept done in multiple hierarchical steps. Based on the facts resulting from classification new facts, i.e. new high-level context attributes, are inferred. This is done with the help of different reasoning schemes, which can be deployed separately or work in parallel. Example reasoning schemes are ontology reasoning applying description logics reasoners, rule-based reasoning, case-based reasoning, neuronal networks or bayesian nets.

Schemes like the last two can be used, but they have to be trained with a sufficient large training set. The advantages of these methods are, that contradictory facts, which are resulting from measurement failures of the sensors can be handled better. In fact, no wrong high-level context facts are reasoned out of these wrong sensor facts, because the result of these reasoning schemes are statements of the quality or the probability of the inferred context attributes.

For example, a trained Bayesian Network for a context attribute calculates the probability to that attribute. On that probability it can be decided if that attribute should be further considered in the following reasoning steps or not. The disadvantage is the application depended and high training effort in advance. The resulting new facts can be further forwarded to reasoners of next superior areas, which combine these facts to another, more abstract fact. After a certain level of abstraction is reached the context attributes of all levels can be delivered to a context-aware application, which can then involve this information for internal decision making, i.e. triggering actions, performing adaptations, etc.

To create situation-aware systems according to our conceptual architecture, the developer can intuitively decompose the relevant situation. Based on the experiences and knowledge about classification and reasoning schemes, the different layers of the conceptual architecture can be defined. Usually, the whole process of situation decomposition and the identification of the components of the different layers of the conceptual architecture are determined based on an iterative process. Starting with a small set of sub-situations, the developer can test the system and extend it stepwise to create a more-and-more complex system. Especially, our experiences show, the understanding of a complex situation grows during testing and practical trials.

## **4 Generic Framework for situation-awareness**

For implementing situation-aware systems according to our conceptual model we propose a generic framework. The framework integrates several sensor devices and provides a set of classifiers and reasoners which can be adopted in different scenarios. Moreover, it supports the specification of application scenarios based on an ontology which describes the sensor infrastructure, relevant physical values, applied classifiers and reasoners together with their particular configurations and data base settings for storing the captured raw data. The architecture of our framework is depicted in figure 2. In the following sections all components of the framework are described in detail.

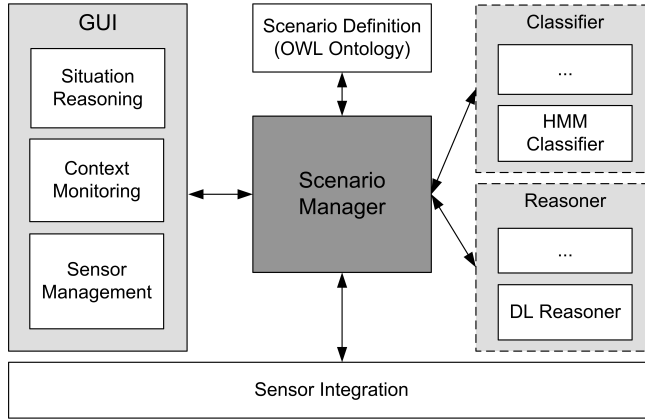
### **4.1 Scenario Manager**

The Scenario Manager is the control component of the framework. It is configured by an ontology modelling the settings, sensors, locations and physical values relevant for the current scenario. It evaluates this information, instantiates and configures the required components and invokes the components at runtime. Thus, by providing a new scenario ontology to the Scenario Manager, the framework can be completely reconfigured for another scenario.

### **4.2 Sensor Integration**

The concept of the framework supports the integration of arbitrary sensing devices. We have currently integrated two types of sensors into our framework.





**Fig. 2.** Architecture of the proposed generic framework for situation-awareness.

Firstly, we implemented a wireless sensor network (WSN) with a flexible amount of sensor nodes at different locations. These sensor nodes are able to obtain different types of sensed data, e.g. light level and temperature, at once. We have used MicaZ Motes from Crossbow in our current implementation. Crossbow offers a wide range of sensor boards which can be connected to the MicaZ and which include many kinds of sensors. For example, the basic sensor board (MTS310CA) provides the following sensors: Photo (Clairex CL94L), Temperature (Panasonic ERT-J1VR103J), Acceleration (ADI ADXL202-2), Magnetometer (Honeywell HMC1002), Microphone (max. 4KHz), Tone Detector and Sounder (4.5kHz). Based on TinyOS we have implemented basic sensor access based on the programming language NesC. We have implemented several commands for initialization, read access, reset, setting of sample rate and network statistics.

Secondly, a microphone was utilized for getting audio context information. It is cheap and easy to integrate into a sensor environment. The aim is to extract audio information from the environment, to classify these data according experience, and to identify the most likely context. The integration of the microphone in the prototype was simply done with the javax.sound package.

### 4.3 Classifiers

For performing the step of classification of sensed data the framework comprises a mechanism for the integration of classifiers and a repository containing a set of classifier implementations. Currently, we have implemented several classifiers for the different sensor types available with the WSN. The result of the classification operations are facts in form of qualitative values (e.g. dark, bright and medium for temperature). For the microphone the underlying model of the classification we used is the Hidden Markov Model (HMM). Before the HMM can be applied to audio data two the extraction of audio features followed by a quantization of these features has to be performed. Based on training data provided as wave

files for a specific scenario, the classifiers can recognise the specified situations (e.g. discussion, presentation or panic in a meeting room).

#### 4.4 Reasoners

Because we assume that all classifiers produce facts we focused on deterministic reasoning schemes. In the current implementation we support rule-based and ontology-based reasoning. Especially, for the reasoning about the overall situation we adopt an ontology-based approach. Therefore, a situation is modelled as a set of concepts and roles in the TBox of the ontology. The current values of concepts and roles related to sensor data or lower-level reasoning steps are included as individuals into the ABox by the scenario manager. To reason about ontologies, a description logic reasoner, namely Pellet [10] is applied. Especially, we use the DL reasoning service realization which works on the Abox.

**Realization:** Given an individual  $I$ , Abox  $A$  and Tbox  $T$ , return all concepts  $C$  from  $T$  s.t.  $I$  is an instance of  $C$  w.r.t.  $A$  and  $T$ , i.e.,  $I$  lies in the interpretation of concept  $C$ . If all Realization is performed for all individuals in the Abox, we speak about the Realization of the Abox.

The Abox updates are implemented based on the Semantic Web Framework Jena. Based on realization, concepts can be identified which represent the situation according to the facts in the ABox. For instance, the concept `BeamerOn` can be defined for the meeting room example.

#### 4.5 Scenario Settings

To ensure the configurability of the framework we introduce an ontology-based scenario description. The description consists of two parts: an upper ontology defining general concepts and roles for the scenario description and a scenario-specific ontology, containing concepts, roles and individuals for the definition of a concrete scenario.

The **upper ontology** defines basic concepts for modelling application scenarios, i.e. available sensors, sensor locations, physical values measured by the sensors, and the assignment of sensors to sensor notes (if existing). The concept *Sensor* represents the sensors available in the environment. Each sensor is described by a certain location which is assigned to the sensor with the role property *locatedAt*. The concept *Location* allows the modelling of semantic locations relevant in the scenario. The sensors available at a certain location are modelled by the property role *hasSensor*, which is the inverse role of *locatedAt*.

Furthermore, each sensor is described by the physical value, which can be measured by the particular sensor type. The concept *PhysicalValue* represents a value captured from the environment, e.g. temperature or light intensity. The relation between a sensor and its physical value is modelled by the property *measures*. Further scenario settings are related to available reasoners, a microphone, the database for storing sensed values and general settings for the scenario. The values are defined as datatype properties for the concepts.

In the **lower ontology**, which is specific to a certain scenario, the general concepts of the upper ontology can be refined by deriving scenario specific concepts. To define concrete instances of reasoners or classifiers, individuals have to be defined in the lower ontology. To add new sensors, classifiers or reasoners, appropriate components have to be implemented and registered with the Scenario Manager. If these components require additional settings, the upper ontology and the controller have to be extended as well. All changes can be done easily based on the current framework implementation. Especially, as long as the upper ontology is extended without changing existing concepts and properties, all existing scenario settings can remain usable without any changes.

#### 4.6 User Interface

For the visualization of the functionality of the framework and the easy setup of demonstrations, a graphical user interface was created. The user interface consists of three parts enabling the configuration of the WSN, the monitoring of sensor data and captured context and to reason about the current situation.

### 5 Validation Example

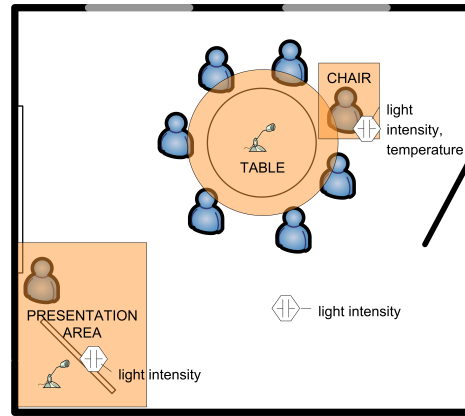
In the following the usage of the proposed conceptual architecture and the generic framework is illustrated by an application scenario, namely the capturing of the current situation in a meeting room. In most companies and universities, conference and lecture rooms, special places in cafeterias and lounges, and other public places should be reserved in advance to conduct public meetings, presentations, parties, etc. This ensures that business is conducted as planned and no inconvenience or conflict of interests occurs which inhibits the overall goal of the company or university. On the other hand, this well planned and well organised execution of business should accommodate rooms for impromptu or short term decisions to organise get together, staff meetings and bilateral discussions.

For the meeting room scenario the goal is to determine the current activity in the room based on the state of devices (e.g. the beamer), the noise level and the occupation of seats (number of persons) in the room. The activity inside a room can be a meeting, a casual discussion of two or three people, a presentation involving many people, a party, a person who is reading or idle. Based on this knowledge, the usage of that meeting room could be optimized. For instance, an empty meeting room could be detected for a spontaneous meeting or the location of a certain meeting could be detected.

#### 5.1 Areas of Interest

The identified areas of interest are shown in figure 3. To distinguish between several activities in the room (e.g. one person reading, two persons discussing, a working meeting, a presentation or a party) and an empty room the individual chairs were identified as areas of interest similarly to the train compartment

application scenario. For each seat area we installed a temperature and a light sensor to measure the light level and the temperature directly at the surface of the seat. Based on the measured values we can distinguish between the occupation by a person or by an object and an empty seat.



**Fig. 3.** Areas of interest in the meeting room scenario.

Moreover, the table represents an area of interest. In that area we installed a microphone to capture the audio data from the room which helps us to distinguish between a discussion between two people, a larger discussion, a party and a presentation (i.e. one person is speaking from a certain distance to the table). A light sensor installed in that area captures the ambient light in the room.

Additionally, we identified a so called presentation area, the area where the image of the beamer is projected on and where the person is standing while giving a talk. In that area we installed a light sensor to capture the beamer state (i.e. on or off). In combination with the light level in the room we can identify the situation that the beamer is on and the ambient light in the room is dimmed as it would be the case during a presentation.

## 5.2 Lower Ontology

For the scenario the sensors *TemperatureSensor*, *LightSensor* and *AudioSensor* are defined. These sensors measure the physical values *Temperature*, *Light* and *Audio* respectively. For *Temperature* the three qualitative values *HighTemperature*, *MediumTemperature* and *LowTemperature* are defined. For *Light* four qualitative values are defined, namely *Dark*, *Medium*, *Bright* and *VeryBright*. The values for *Audio* represent an activity captured by microphone measurements in the room. Currently we distinguish between *Discussion* and *Presentation*. By adding more training examples and extending the ontology more values can be added. To reason about the situation in a meeting room we modelled the chairs

around a table in the meeting room. *Table* and *Seat* as well as *PresentationArea* are modelled as sub concepts of *MeetingroomLocation* which itself is a sub concept of *Location*. Free seats are modelled by the following concept:

$$\begin{aligned} \text{FreeSeat} = & \exists \text{hasLightSensor}(\exists \text{uso} : \text{hasValue}(\text{Bright} \vee \text{Medium})) \wedge \\ & \exists \text{hasTemperatureSensor}(\exists \text{uso} : \text{hasValue}(\text{LowTemperature} \vee \text{MediumTemperature})) \end{aligned}$$

For occupied seats we distinguish between seat occupied by persons and by objects. A seat occupied by a person is described as follows:

$$\begin{aligned} \text{PersonOccupiedSeat} = & \exists \text{hasLightSensor}(\exists \text{uso} : \text{hasValue}(\text{Dark})) \wedge \\ & \exists \text{hasTemperatureSensor}(\text{uso} : \text{hasValue}(\text{HighTemperature})) \end{aligned}$$

### 5.3 Reasoning

The reasoning about the situation in the meeting room is based on the individual *MyMeetingroom*. Again, the type of situation is determined by computing the types of this individual based on the reasoning service realisation. *MyMeetingRoom* belongs to the concept *MeetingRoom*. The concept represents the overall situation. All partial situations, namely *BeamerOff*, *BeamerOn* and *PresentationMeetingRoom* are modelled as sub concepts of *MeetingRoom*. Based on the light sensor installed at the presentation area, the state of the beamer can be determined:

$$\begin{aligned} \text{BeamerOff} = & \exists \text{hasPresentationArea}(\exists \text{hasLightSensor}(\exists \text{uso} : \text{hasValue}(\text{Bright} \vee \text{Medium} \vee \text{Dark}))) \end{aligned}$$

$$\begin{aligned} \text{BeamerOn} = & \exists \text{hasPresentationArea}(\exists \text{hasLightSensor}(\exists \text{uso} : \text{hasValue}(\text{VeryBright}))) \wedge \exists \text{hasRoom}(\exists \text{hasLightSensor}(\exists \text{uso} : \text{hasValue}(\text{Medium} \vee \text{Dark}))) \end{aligned}$$

In combination with the audio signal the overall situation is determined:

$$\begin{aligned} \text{PresentationMeetingRoom} = & \text{BeamerOn} \wedge \\ & \exists \text{hasRoom}(\exists \text{hasAudioSensor}(\exists \text{uso} : \text{hasValue}(\text{Presentation}))) \end{aligned}$$

New scenarios can be implemented by creating a new scenario ontology based on the upper ontology for configuring the generic framework.

## 6 Summary and Outlook

The major result of our work is a comprehensive solution for modelling and reasoning about complex situations. The solution is comprehensive in the sense that it starts at the sensor layer and comprises all steps necessary to abstract the captured low level sensor values to an overall notion of a complex situation. In

our solution we considered the installation and access of sensors, the classification of captured sensor data, several steps of abstracting sensor data and reasoning about sub-situations and the overall situation.

In particular, we developed a systematic approach for the decomposition of complex situations. Based on the identification of sub-situations we have shown that it is feasible to place sensor devices in so called areas of interest. Each area of interest can then process the captured sensor data separately (at least the aggregation and classification of sensed values). In higher level reasoning steps the sub-situations are then integrated into the overall situation.

To demonstrate the feasibility of our approach we designed and implemented a generic framework for situation awareness. The framework comprises implemented components of every level of situation awareness as described in the third chapter of this document (see figure 1). Each layer is extensible regarding to new components and technologies (i.e. sensors, classifiers and reasoners).

Our solution enables an easy, fast and flexible development of situation-aware applications. New scenarios can be implemented by creating a new scenario ontology based on the upper ontology for configuring the generic framework. By using an OWL ontology, scenarios and sensor configurations are clearly defined, easy readable and easy to understand.

## 6.1 Lessons learned

From our approach we learned the following lessons:

1. It is possible and reasonable to decompose complex situation into partial situations. At least in all scenarios considered in our project it was possible to decompose complex situations in a way that each sub-situation characterises a certain aspect of the complex situation and is by itself meaningful. Moreover, the partial situations can then be composed to infer the overall situation even if only a subset of all modelled partial situations is considered, i.e. information about environment is incomplete.
2. The identification of areas of interest and the well defined placement and combination of sensors in that area seems to be reasonable and efficient. Compared to an equal distribution of sensors in the environment, the placement of sensors in areas of interest is more focused and environmental state can be captured systematically.
3. The combination of several classifiers and reasoners seems to be possible. In our generic framework a set of classifiers and reasoners is available and we combined them in different ways in the two scenarios. Based on the ontology-based scenario definition, the classification and reasoning algorithms can be configured and thus reused for different scenarios to some degree.
4. We found out that the programming of WSNs and the placement/organisation of sensors is highly related to the scenario. Usually WSNs have to be reprogrammed and sensor placement has to be adopted for different scenarios. Thus, the reusability of particular sensor networks and sensor settings is limited. Nevertheless, the generic framework itself is configurable and thus can be reused in different scenarios.

## 6.2 Future Work

The results demonstrate that our approach is reasonable and advantageous compared to scenario specific and monolithic approaches. While our approach is more flexible, it assumes a closed sensing infrastructure. Especially, the sensors should be under control of the system developer. To exploit existing sensor infrastructures, our approach could be extended to support sensor discovery and dynamic integration of sensing devices. Especially, a location-based discovery of sensors should be considered. Another important point is a further decoupling of all components and layers of our framework. Classifiers and reasoners are available in a local repository and can't be distributed in the current implementation. Thus, a distribution of these components as well as a discovery and integration mechanism similar to the sensor integration will be considered in the future. Moreover, we will adopt our approach to further application scenarios.

## References

1. Mark Weiser. The Computer for the 21st Century. *Scientific American*, pages 66–75, Sep 1991.
2. M. Srivastava, R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments. In *The 7th Annual international Conference on Mobile Computing and Networking*, pages 132–138, 2001.
3. V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa. Computational auditory scene recognition. In *International conference on acoustic speech and signal processing*, 2002.
4. D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. Wong. Sensay: A context-aware mobile phone. In *The 7th IEEE international Symposium on Wearable Computers*, 2004.
5. P. Korpipää, J. Mäntyjärvi, J. Kela, H. Kernen, and E-J. Malm. Managing context information in mobile devices. *IEEE Pervasive Computing*, 2(3):42–51, 2003.
6. K. Henriksen S. Livingstone and J. Indulska. Towards a hybrid approach to context modelling, reasoning and interoperation. In *Ubi-Comp 1st International Workshop on Advanced Context Modelling, Reasoning and Management*, pages 54–61, 2004.
7. H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments, 2003.
8. Tao Gu, Hung Keng Pung, and Da Qing Zhang. Toward an osgi-based infrastructure for context-aware applications. *IEEE Pervasive Computing*, 3(4):66–74, 2004.
9. Eleni Christopoulou, Christos Goumopoulos, and Achilles Kameas. An ontology-based context management and reasoning process for ubicomp applications. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 265–270, New York, NY, USA, 2005. ACM Press.
10. Evren Sirin and Bijan Parsia. Pellet: An OWL DL reasoner. In V. Haarslev and R. Möller, editors, *Proc. of the 2004 Description Logic Workshop (DL 2004)*, number 104, 2004. See also <http://www.mindswap.org/2003/pellet/index.shtml>.