From Sensors to Assisted Driving – Bridging the Gap

Marcus Tönnis[1], Jan-Gregor Fischer[2], Gudrun Klinker[1]
[1] Department of Computer Science, Technische Universität München,
Boltzmannstraße 3, 85748 Garching b. München, Germany
[2] Siemens AG, Corporate Technology,
Otto-Hahn-Ring 6, 81730 München, Germany
Email: {toennis, klinker}@in.tum.de, jan-gregor.fischer@siemens.com

Abstract— **Increasing traffic density enforces development of Advanced Driver Assistance Systems to cope with safety aspects. Such systems require serious amount of sensor data to deduce spatial relationships. Not only car mounted sensors, but the combination with environmental tracking systems can fulfill the demand for obstacle surveillance.**

Fusion of sensor data, federation to environmental models and reasoning about that data allows for a broad spectrum of new in-car systems. From collaborative and informing systems up to part or fully automated driving, various assistance systems generate a demand for such spatial knowledge. With an integrated system, dependable data can be delivered to any user-related assistance system, and as side effect, reduce workload in already loaded in-car computer systems.

To face these data aggregation and analysis issues, we developed SCORE, a Spatial Context Ontology Reasoning Environment. We illustrate our approach of a distributed ad-hoc infrastructure that collects and disseminates tracking data of environmental objects and thus allows for vehicle- and ontology-based reasoning.

In extend, we illustrate how such systems can gather data and where such a system can help in spatially related driver assistance systems.

Index Terms: Advanced Driver Assistance Systems, Ontology-based Knowledge Acquisition, Spatial Context, Augmented Reality, Time-Critical Systems

I. Introduction

Driver inattention is a major reason for vehicle collisions, contributing to 78% of crashes and 65% of near-crashes [1]. Reasons can be attributed to driver absent-mindedness, driver distraction due to interaction with an in-car information system, and to increasing amount of traffic on the road. Traffic density becomes higher and drivers have to cope with more unexpected situations.

Advanced Driver Assistance Systems (ADAS) aim at supporting drivers with their driving task [2]. One of these, for instance, is the Adaptive Cruise Control (ACC, [3]), that automatically senses cars in front and controls speed such that a safe braking distance is maintained. Yet, ACC is only allowed to operate within carefully set safety margins. Extreme braking operations are excluded, causing ACC to turn itself off and alert the driver that there is no more longitudinal support. One common reason for such behavior occurs when a neighboring car switches lanes just in front, thereby being temporarily in very close range. Such switch-offs could be avoided by more sophisticated ADAS systems. If they could monitor the car's wider environment and thus could deduce traffic situations or other drivers' behavior. These systems then could provide a wider range of information for the driver about timely and accurate safety-related information, use in automatic course adjustment or even intervene in emergency situations. We are exploring such concepts in a system that supervises location, speed and trajectory of other vehicles and deduces relevant behavioral traffic data to be used within a vehicle-centric service-oriented architecture.

Mandatory basis for the analysis of traffic behavior is a rich sensing system. GPS, Galileo as well as in-car dead-reckoning systems indicate that car motion will be trackable within individual lanes of a road. We assume that stationary systems along the road will be available to gather such data from passing cars and integrate it into local traffic models. For the area of pervasive computing, challenges of a highly dynamic environment in conjunction with dynamic short range network connections are addressed. It is the responsibility of the individual cars to gather and interpret accumulated traffic data from such stationary providers along the road – enhanced by data from on-board sensors such as radar, near and far infrared and range scanning devices [4].

ADAS systems then need to infer critical events from such traffic data and provide appropriate feedback to the driver. Among other types of feedback, those systems have to provide intuitive and minimally distractive information to the driver of a car. To focus on the user interface, such systems require underlying deduction of a car's spatial context. To enable ADAS systems to receive such spatially related information about traffic situations, we designed and implemented a distributed system to meet the general issues of such systems.

This paper is based on "Ontology-Based Pervasive Spatial Knowledge for Car Driver Assistance", by M. Tönnis, J. Fischer and G. Klinker, which appeared in the Proceedings of the 15th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops '07, White Plains, New York, USA, March 2007; The First International Workshop on Pervasive Transportation Systems, PerTrans 2007. © 2007 IEEE.

Our context deduction system, *SCORE*, a Spatial Context Ontology Reasoning Environment, constitutes, what pervasive distributed systems need: a peer to peer middleware in conjunction with an applied contextual deduction system. The system consists of separate autonomous components which federate and reason about contextual data that is related to spatial properties [5], [6]. With SCORE we show that Semantic Web technologies and logic/rule-based systems are applicable to a spatial context. Federation components collect explicit spatial data coming from distributed information sources. To support applications that access structured information, reasoning systems deduce spatial knowledge by querying data from the federation components. Multiple instances of this system can set up short range ad-hoc networks and can thereby maintain up-to-date models of moving cars in the vicinity.

As part of an continuing research project towards creating and evaluating novel assistance systems for car drivers, the SCORE system has helped us overcome the limitations of sensory equipment in current cars and look a few years ahead. In this article, we illustrate the approach of our context deduction system and the underlying middleware. We illustrate our system in the context of its own dependencies. Our groups work in underlying sensor systems is presented together with other approaches on top level ADAS systems for future use in intelligent transportation systems. Especially the way from sensor data to ADAS systems, that use Augmented Reality for egocentric and therefore minimally distractive information presentation is described. We close with a description of the implementation, results from applying and testing SCORE, and end with an outlook of future work.

II. Related Work

Research projects as the project INVENT [7] deal, among other things, with the subject of car to car communication. Stationary sensors as well as mobile sensors mounted to cars acquire data for use in a traffic routing system. Diverse and heterogeneous sources of data are combined to obtain a prognosis of traffic state. In this manner, a comprehensive knowledge base is built up to support optimal individual route guidance. To reconstruct the traffic state based on fused data coming from different sources methods such as computer simulation of traffic flows are used. Route planning then is performed using a roadway network with dynamic attributes. The network combines criteria from a static digital map with attributes obtained from a reconstruction of the current traffic state, from a forecast of the future traffic state, from knowledge of traffic management and control strategies. Waypoint- and graph-theory algorithms are used in a in-car computer system. Here each car deduces its own relevant information for use in 3rd generation navigation systems.

The goal of the work of Kosch [8] is to provide precise and up-to-date information to car drivers, according to the needs of their individual situation. The developed CARISMA system independently and autonomously self-organizes direct wireless transfer of sensor data between automobiles. Varying system analysis and simulations clarify the behavior of the resulting vehicle ad-hoc network and provide insight to its characteristics. By adding new protocols below and besides the TCP/IP stack, the used protocol design takes special network properties and very dynamic topology into account. The developed methods allow targeted exchange of information between vehicles on the road. The system assesses the situation-dependent benefit of information and decentrally controls the communication with its distributed components.

III. Tracking and Genera System Architecture

To reason about the current traffic situation, cars need a long-ranging detailed overview of cars in their surroundings. Many sensors, such as GPS, radar, infrared, range and ultrasound sensors, are integrated into next generations' cars [4]. Thus valuable information about the immediate surroundings becomes available to on-board automatic analysis and reasoning. Yet, we expect such tracking data not to be sufficient since directly adjacent cars may be occluding further cars that contribute significantly to the total traffic conditions.

In our approach, we use a hybrid concept towards gathering and maintaining up-to-date traffic models. The approach involves continuous communication between car-based (”inside-out”) mobile tracking equipment in cars and road-based (”outside-in”) stationary tracking equipment along the road.

To provide ubiquitous support along the road, the street network is separated into road sections, with each of them containing a number of vehicles at a certain point in time (see Figure 8). For each road section, a logically related tracking system observes the explicit spatial context states of all vehicles within that area. When a car crosses an areal border, it becomes associated (based on its GPS-based position) with the next road section. These components serve as *spatial context providers*.

Bearing the prerequisites of such ubiquitous information management [9], [10] and high dynamics of spatial data in associated scenarios in mind, SCORE was designed to support applications on two separate layers that set up on the observed reality. In the *federation* layer, all data of a certain area is federated in a spatial context model. Components of the federation layer intentionally are associated to certain road sections. From the next layer above, ontology-based *reasoning* components can access the spatial context model. The reasoning components then intentionally reside in each car so that each car can deduce its own view. Figure 1 shows a sketch of this architecture. This approach is based on work by [11] and supports the combination of spatial context models for efficiently processing low-level spatial data with ontological context representations for information deduction as top layer.

The two layers clearly separate highly dynamic assertional knowledge in traditional coordinate-based models

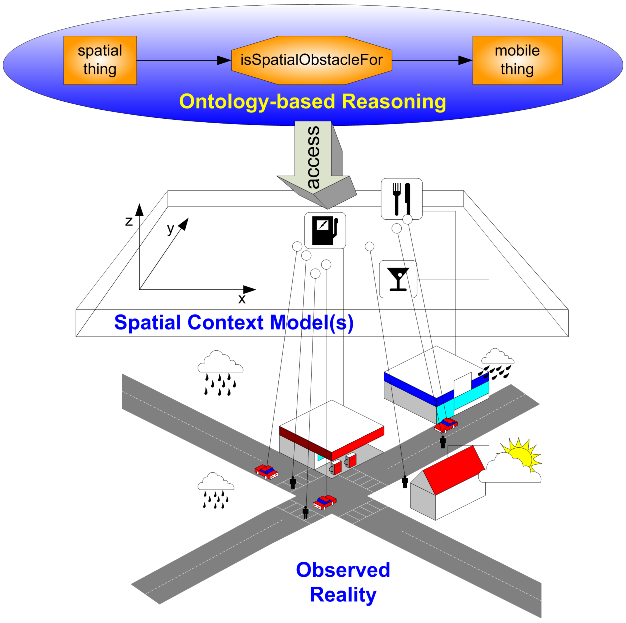


Figure 1. Combining Spatial Context Models and Contextual Ontologies

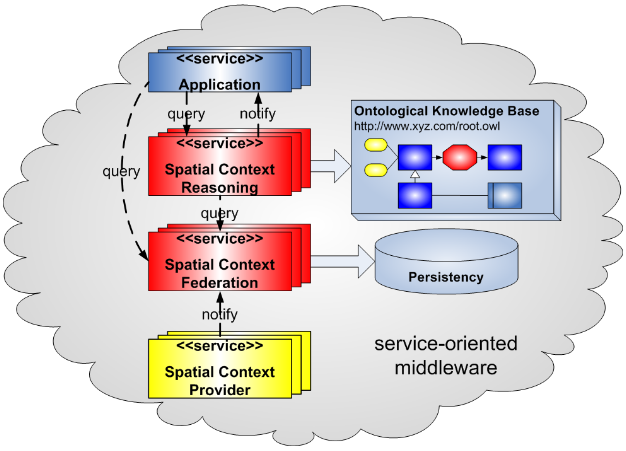


Figure 2. SCORE's Architectural Overview: Trackers define Spatial Context Providers that update Spatial Context Federation components. These are queried by components for Spatial Context Reasoning. ADAS Applications on top specify the user interface for the driver

and terminological information of real world concepts within interpretable ontology spaces [12]. Multiple instances of software components are allowed to run on each layer to warrant high availability of the system and avoid single points of failure.

In the federation layer explicit context data is aggregated in spatial context models as attributes of entities. Implicit information is deduced in the reasoning layer by ontology-based interpretation modules. Here high-level spatial context is represented by binary relations between the entity classes of the relation's domain and range. The decision whether such a relation is executed in particular situations is based on information observed at this moment. We strictly distinguish between the continuous acquisition of context and its usage by aware applications. This enables facilitated access to context data and also its sharing between applications [13].

Applications, such as advanced driver assistance systems, can access the components of the reasoning layer by querying for certain information (pull) or by being automatically notified (push). All components of the architecture act as autonomous services. Figure 2 illustrates the general architecture.

Our approach combines advantages of coordinate-based models with those of ontological knowledge representation. Therefore SCORE profits on the one hand from the flexibility regarding spatial scope, an extensible spatial detail. On the other hand, SCORE profits from scalability with respect to dynamics of context acquisition [14]. Additionally, it provides ontology-based reasoning capability, information sharing and reuse.

IV. FEDERATION

In SCORE's layered architecture a federation component mediates between various information sources and reasoning components and beyond that, the ADAS systems. Regarding the volume of communication traffic, several stationary computers – each logically related to one road section – run federation components that collect spatial data within the respective areas. Trackers act as *spatial context providers* (see figure 2). These inform the federation layer about spatial data with frequencies high enough to permit real time use in safety-relevant applications in ubiquitous computing and in Augmented Reality-based applications in general [15] and for visual support in cars in particular.

When the so-called *broker* of the federation subsystem (see Fig. 3) is notified, it instantly accesses the *warehouse* to aggregate attributed data items. Such items are each linked to a unique entity within the real world.

Each data item describes exactly one entity and contains information about its primary and secondary spatial context. An entity's primary context covers its unique name and its positional and orientational values at a particular time. The federation component uses this information with the entity's class membership to identify each entity. The secondary context depends on this identification and – among other things – includes a number of valid spatial context attributes such as position, orientation, velocity, acceleration, distances and trajectory angles. Since a federation component may serve multiple applications at a time, namespace statements are supported to uniquely name entities and their class memberships.

The *warehouse* organizes an efficient volatile data structure to continuously index and cache those data items to optimize query response times and to provide quick information retrieval. Among other things it also computes distance values for efficient range-based queries.

An additional persistency mechanism is represented by connections to relational database management systems which store snapshots of the contextual configuration in discrete intervals. The federation component also provides query mechanisms for applications requesting mere

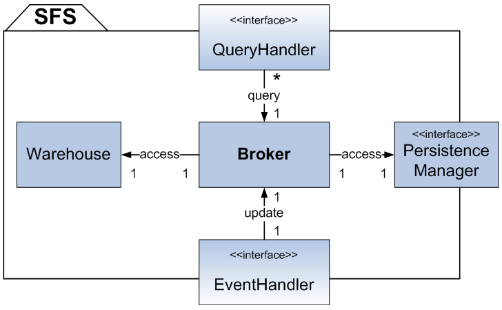


Figure 3. Design of SCORE's federation component

explicit information such as reasoning components used for deduction of context. The querying component can request current and past contextual information about a number of specified entities. Applications and reasoning components also issue range-based queries to obtain data of entities that are in a specified range to a special entity.

V. Reasoning

To deduce relevant information from federated data, the reasoning system uses an ontology rule-based approach. In each car a reasoning component dynamically connects to one or more federation servers in range to access spatial data necessary for interpreting the vehicle's situational information. Therefore the principal task of SCORE's reasoning component is context interpretation for queries from applications running at this moment. Similar to the federation component's *broker* the reasoning system's central *controller* manages the control flow. The controller mediates between the component's interfaces, the subscription component, the query parser, the reasoner subsystem and the persistence manager (see figure 4).

Using the *query handler* interface, applications can directly request deduced spatial context from a reasoning component or can subscribe to notifications regarding static queries. These queries are then executed in defined intervals. The subscriber is notified only if contextual changes regarding the observed entities occur.

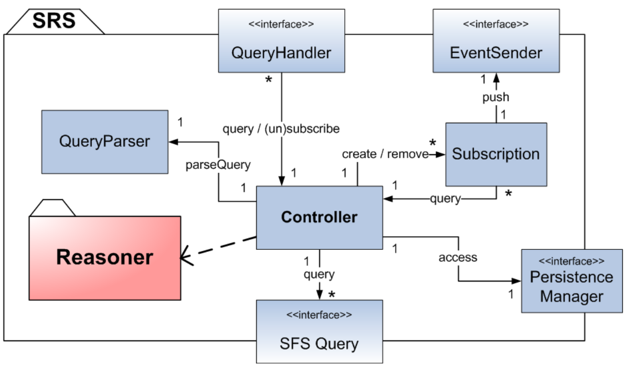


Figure 4. Design of SCORE's reasoning component

The query language is declared to be easy to use, so that non programmers can define their own queries for use in additional ADAS systems. Each query consists of a triplet comprising an ontological object property. Later, its domain and range are dynamically mapped to entities by the reasoner. Hence, the important prerequisite for SCORE's functionality is the automated matching of entity classes to ontological classes, entity class relations to ontological binary relations between classes, and entity class attributes to ontological properties, that assign data types to ontological classes. The following two examples show the request of contextual information for queries whether vehicle "car007" overtakes "car001', and which lorry represents a spatial obstacle for the motorcycle "bike27". To support multiple applications at the same time optional namespace declarations can be stated in a query.

```
Vehicle=car007 overtakes Car=car001
Lorry=* isSpatialObstacleFor
                             Motorcycle=bike27
```

The controller invokes the *query parser* only upon the arrival of a new query. At that point in time, the query is parsed, transformed into a processable abstract syntax and afterwards cached due to efficiency purposes.

Next the abstract query is handed to the reasoner subsystem, where it is processed by the description logic-based reasoner (*DL-reasoner*). T-box information contained in ontologies describe terminological entity relationships that are used by the reasoning component to understand how to interpret assertional spatial data that is collected in the entities' environment [12].

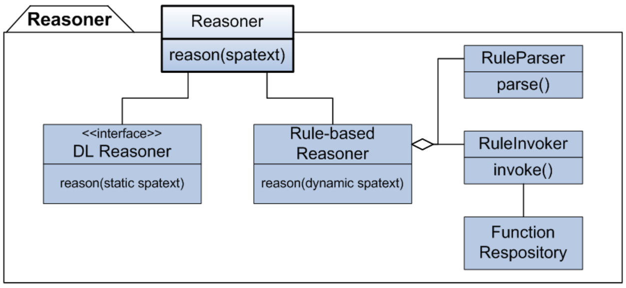


Figure 5. Reasoner subsystem within SCORE's reasoning component ("spatext" denotes "spatial context")

The terminological interpretation process is based on ontologies [16] known from the *Semantic Web* [17], [18]. For defining static knowledge, a global *ontology knowledge base* plays a central role in SCORE's architecture. It comprises extensible ontologies for a homogeneous application domain terminology as well as rules that describe the validity of binary relations between ontology classes. Ontologies are well-suited for logic-based reasoning, especially when description logic ([19], [20]) is applied. This is due to the similarity between the language constructs of description logics and those of ontologies. Additional information can be deduced automatically from pre-defined ontologies with properties such as equality, symmetry, transitivity, inversion, disjointness and others (see figure 6). For this purpose, an off-the-shelf ontology-based reasoning framework is accessed via the *DL-reasoner* interface.

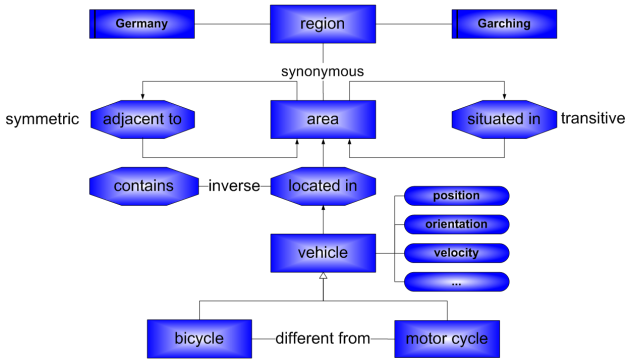


Figure 6. Example Ontology

A successful interpretation leads to a pre-defined instance of the ontology class ”*rule*” (see figure 7). To identify the correct rule definitions for an object property, the query relation's domain and range classes are mapped to the ontology relation's domain – i.e., to the range classes or to the proper sub-classes, if either the domain, the range or both classes in the query inherit from those declared by the relation in the corresponding ontology. As mentioned before, the rules themselves are instances of the ontology class *rule*. They are defined by application developers in a generic way, such that they can be reused in various application domains. Rules are specified in the custom-defined RDFS-based language SRL (**S**patext **R**ule **L**anguage, *Spatext* abbreviates *Spatial Context*) [21].

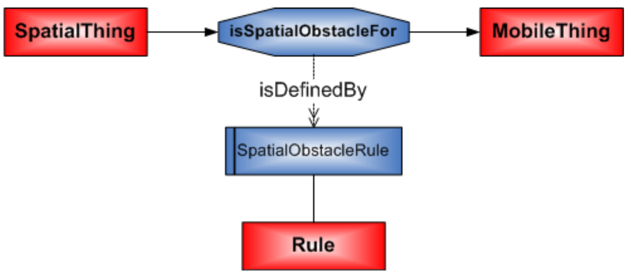


Figure 7. Reasoning about ontological object properties

For each rule derivations are defined similarly to ”if” statements well-known from general programming languages. Conditions can be expressed using sub-conditions, boolean operators, comparison operators and *reasoning functions*, where the latter ones are defined within an extensible *function repository*. These functions can also be nested, and are used to compute spatial relations between entities. For instance there are functions returning angles, distances or the time to collision (TTC) between entities.

The interpretation process results in a set of rules that just define the binary relation referenced in the query. Similar to the query that is parsed by the *query parser* the rule set is transformed by the *rule parser* of the *rule-based reasoner* into an abstract syntax and cached internally to speed up succeeding operations.

To execute the rules, the *rule invoker* invokes those functions declared in the rule conditions together with the explicit spatial context of entities matching the query. Here the federation layer plays an important role as it is accessed for obtaining this information.

Ontological knowledge bases are not well-suited for maintaining a detailed history of highly dynamic assertional data due to the continuous growth of information resulting in inefficient data management. In spite of that and to cope with the increasing amount of data without giving up the benefits of an ontological representation of concept knowledge in application domains dynamic spatial context data is managed on the federation tier and only requested for interpreting situational information. The reasoning process is rule-based, since it addresses the objectivity of spatial relations, e.g., traffic rules, that in its own sense disregard a scope of varying interpretation [12].

Each successful rule invocation of the *rule-based reasoner* results in interpreted spatial context, returned to the central *controller*, where it is cached as well and forwarded to the *persistence manager*. The *persistence manager* of the reasoning component can be compared to the one of the federation subsystem. It handles the storage and retrieval of spatial context. In this case it only manages implicit data, not low-level context. Hence the reasoning component's *persistence manager* only enables storage of those location-based information, that it can deduce. Alike the *persistence manager* of the federation layer this one does not implement persistent storage on its own. Instead persistent storage is dependent on an additional component providing persistence management of interpreted spatial context.

Finally the interpretation result is provided to those applications, that have subscribed to the corresponding query or actively issued the query. Besides the implicit information deduced from a binary relation between certain entities the result also comprises descriptive information specified by application developers as well as corresponding explicit context data requested from the federation layer during the reasoning process.

VI. AD-HOC NETWORKING

Spatial context providers and federation components can be placed in the environment and therefore can be interconnected statically. Similarly, instances from the reasoning layer can have hard-coded interconnections to applications inside the car. Interconnections between the federation and the reasoning components require dynamic bindings, when a car leaves one federation sector and moves to another. Then once more the car built-in reasoning system must get connected to the forthcoming sector's federation component and disconnect from the previous one.

For this kind of ubiquitous networking we used the DWARF [22] architecture to describe and interconnect all components. DWARF specifies a peer to peer framework that manages services to dynamically interconnect them depending on their contextual attributes [23]. For instance localization and tracking distances of services are specified by such attributes. The DWARF middleware supervises these attributes and interconnects corresponding connectors, therefore enabling the data flow. When the system is started, it establishes the interconnections between federation and reasoning components as well as between base data providers or applications that declare driver assistance systems.

For the case, a car moves through the environment, it updates the contextual requirements of the reasoning service that specify its global position. When a car is going to leave the sector of a specific federation component, the DWARF middleware determines, that another environmental federation service has contextual attributes that now fulfill the changed need of the reasoning service. The new connection between the entered sector's federation component and the car's reasoning system is established and the present connection between the two services is removed. Figure 8 shows in a sketch how interconnection is executed while a car drives along the X-axis. While this operation is performed, some of the reasoning service's attributes are automatically passed to the federation service, adjusting and configuring it for addressed queries and notification frequency [24].

To provide additional tracking data, among other various information sources a car built in GPS system can also act as a spatial context provider and therefore propagate the own car's position to the federation service that is connected at this moment.

Inconsistencies that may arise during a hand-over from one federation service to another, which are similar to hand-overs of WiFi systems, are not dealt with at the moment. At the moment our work focuses on applied federation and reasoning, which we will investigate further on in the future.

VII. Implementation

This section describes the current implementation state of the SCORE framework as well as the procedure for developing new applications on top of the SCORE system. Some missing functionality implementations correspond to the integration of an off-the shelf persistence management system on the federation layer as well as the implementation of the persistency manager on the reasoning layer. However the possibility of an easy extension of SCORE has been carefully considered during its design to meet these functionalities in future development.

A. Federation Subsystem

Due to high performance requirements the federation service is implemented in portable ANSI/ISO C++. Thus efficient main memory management speeds up the service's *warehouse* through fast memory (de)allocation. To meet fast response time requirements of ubiquitous computing and Augmented Reality applications, the service's warehouse (see Fig. 3) itself is based on an efficient and flexible indexing data structure combining *minimum binary heaps* and *AVL-trees*. Following a pointer from an item within an array of ontology classes to the corresponding heap, range-based queries can be executed quickly since the heaps are arranged by distances between tracked objects. For general queries also internal binary search trees pointing to items within heaps can be traversed. Indeed, for both indexing methods the worst case runtime for finding or updating an entity's contextual history refers to logarithmic time complexity on the total number of items in the corresponding tree.

At this moment the federation service provides two interfaces for communicating with other system components. Spatial context providers use DWARF's asynchronous CORBA event notification connector matching the federation service's receiver connector to enable events regarding low-level location-based information about entities. Multiple spatial context provider services can simultaneously send notification events to a single service.

The decision for using an event-driven communication with the federation service's need interface is founded upon the asynchronous property of this communication mode. In contrast to synchronous method calls we integrated context providers by use of continuously sent structured events of spatial information, therefore not enforcing the federation component to take care of multiple communication sessions, but delegating this issue to the DWARF middleware. The asynchronous communication mechanism significantly increases the performance of context providers, especially when the frequency of context acquisition is high. Though there is no absolute guarantee for delivering events (they might get lost on the network if netload is too high) the benefits in the flexibility of asynchronous communication clearly surpasses this limitation.

The federation service also has a common interface for querying basic spatial context histories about contextual entities using synchronous communication via CORBA method calls. Here the service is capable of handling multiple queries at the same time. By use of the synchronous communication mode the calling system is blocked during query processing. In contrast to the need interface, where the federation service only receives events, here the communication is based on a client-server approach, where the client such as a reasoning service specifies additional attributes for constraining the possible results of its query. Events are far too inflexible and inefficient for the transmission of both the dynamically declared queries and the varying responses.

B. Reasoning Subsystem

As a concrete ontology reasoner the reasoning subsystem uses the open-source *Jena2 Semantic Web framework*[1] at this moment. This Java-based framework is the second version of the Jena reasoner. It follows the *HP Labs Semantic Web Research Program*[2], and is now available under a BSD open source software license. Jena2

[1] http://jena.sourceforge.net/
[2] http://www.hpl.hp.com/semweb/

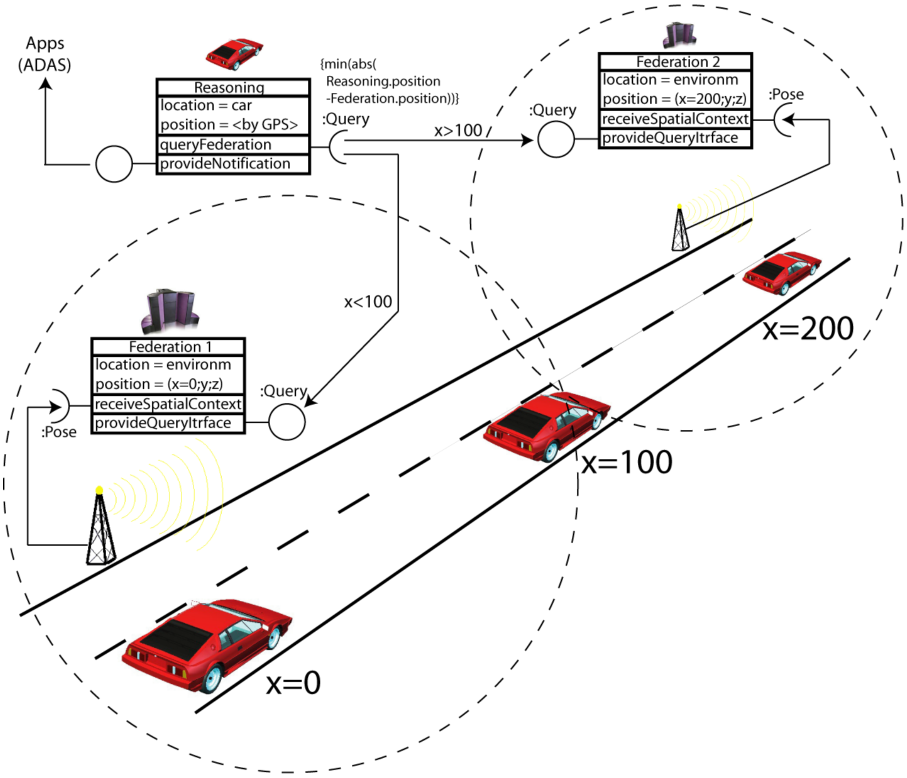


Figure 8. Sketch of dynamic reconnection of car internal reasoning service between two federation service sectors as car moves along X-axis

supports reasoning with respect to RDFS and subsets of DAML and OWL. The framework provides several Java APIs for accessing RDF models and ontologies. Besides an implementation of RDQL, Jena2 also allows for the persistence of ontology models in relational databases. Though Jena2 restricts the programming language of SCORE's reasoning service to Java, it nevertheless has been selected, because it provides an easy to use OWL API and both a volatile and persistent storage mechanism. Above all the OWL API facilitates the retrieval and reasoning about terminological information, that is contained in SCORE's ontologies.

For describing this information in SCORE's central ontological knowledge base the *Web Ontology Language (OWL)* was chosen. OWL is a powerful XML-based language, that has been recommended by the W3C in February 2004[3]. It is based on RDF's features of data and meta data modeling, and RDFS's capabilities of defining the corresponding vocabulary and constraints. OWL extends this first approach of knowledge modeling by specifying formal semantics with the help of additional restrictions in the usage of RDF. The decision for using OWL as the description language for ontologies not only allows for an uniform modeling of common knowledge about application domains. Furthermore it enables sharing of these information among cooperating applications and also other context management systems. Also the extensible structure of an OWL knowledge representation enables developers and system architects to understand ontology evolution.

Interpreting dynamic context is done by the *rule-based reasoner*. On the basis of globally defined interpretation rules, it transparently invokes functions, that refer to the function repository - a special Java class designed for easy extension. These functions are identified at runtime and dynamically invoked via Java reflection by passing the explicit context that matches a certain application query and is subsequently requested from the federation layer. Java reflection is used due to the convenience for application developers, who just have to add new functions to the repository in the case they do not yet exist, and afterwards can instantly use them in new rules. Since functions can be nested within conditional statements, more complex interpretation behavior is also achieved by piping spatial data through a set of functions. Independent from whether functions are added to the repository, composed during interpretation or nested in complex conditional statements, because of the reflection mechanism, application builders do not have to think about how and when interpretation functions are executed by the reasoning service.

As mentioned before reasoning services query basic spatial information from the federation layer by calling methods in the corresponding interface represented by the federation service's ability. Similar to the federation service also SCORE's reasoning subsystem enables other services such as applications to query for spatial information about entities. In further development the reasoning service will also provide a subscription mechanism via event notification to send contextual events to its sub-

[3]http://www.w3.org/TR/owl-features/

scribers. Therefore aware applications are able to actively request or be informed about implicit context processed on the basis of a terminological context representation and the dynamically changing location-based context of those entities, an application is interested in.

C. Developing Applications on top of SCORE

Figure 9 shows that SCORE's usage in new applications is facilitated. In the requirements analysis phase application builders first identify entities, e.g., individual persons, locations and objects, that must be supported by the application. After unique names are provided for each entity they are mapped to common ontology classes, e.g., 'person', 'vehicle' or 'service station', for which binary relations are analyzed. If necessary corresponding ontology rule instances are defined properly for each relation using the XML-based *Spatext Rule Language*.

Yet missing terminological information (concept, relation, property data) is added to the ontology knowledge base (OKB) [12].

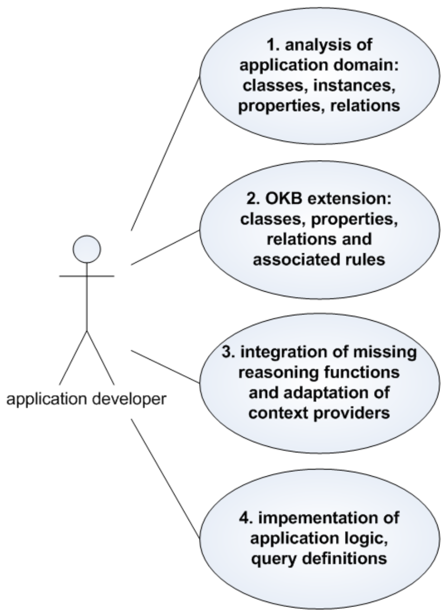


Figure 9. Use cases for building new SCORE applications

If a required reasoning function is still missing in the function repository, then it is integrated additionally. Then existing generic spatial context provider services are configured or new ones are built depending on the application's requirements.

In the last step the application is built. During the development, queries that were identified in the requirements analysis are declared using SCORE's query language. It might be necessary to define classes and properties with fully qualified URIs if their short names raise ambiguities.

Now the complete system can be started up. SCORE will automatically detect any errors related to misstated declarations in the terminologies, rules or in queries, for instance. Here it informs the application developer where and why errors arose.

VIII. Performance

To test the performance of our system, we used a driving simulator to simulate additional cars, whose amount reflected an averagely frequented road. We developed an ADAS application that observes the *time to collision (TTC)* with various additional cars. To support this functionality it registers with respective events that are released if the TTC falls below certain thresholds. The foreign cars' positional data is propagated by stationary spatial context providers to the federation layer running on another computer. Two federation services managed two areas (left and right of a horizontal X axis dividing the world into two areas), see figure 10.

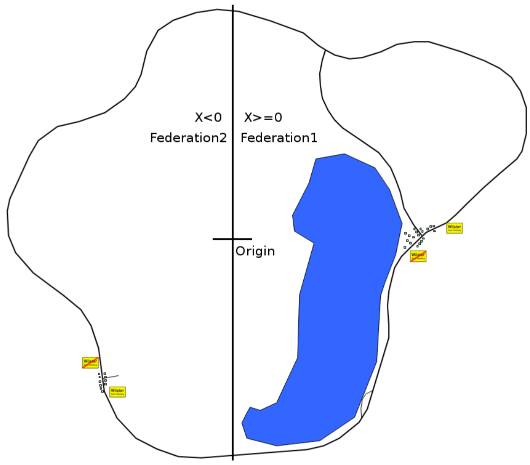


Figure 10. The road course of the driving simulator. *Federation1* manages the positive side of the X-Axis, *Federation2* the negative

The own car's position in the virtual world was computed by a single-lane driving dynamics model – we simulated a GPS tracker. Depending on the own car's position in the virtual world, the car's reasoning service (running on a Linux PC) was connected to one federation service, querying for spatial data. Depending on its location in the world, the reasoner updated its predicate for the selection of a suitable federation.

The upper part of figure 11 illustrates a snapshot of the SCORE system's service structure. Within this snapshot, the car is on the positive side of the simulated world and therefore is connected to the service named *Federation1*. The lower part of figure 11 shows the *predicate* that is adjusted in correspondence to the current position of the own car.

A. Performance of Single Services

The federation service gives a response time below $10\,ms$ for up to five spatial context providers with up to 250 entities per provider. Compared to an average amount of cars on an road interval, these response times are acceptable. The reasoning service responds to n-to-n queries in $37.1\,ms$ seconds for 100 entities. This result is a worst case result, because general reasoning systems are intended to be integrated in each car and therefore only have to manage 1-to-n queries. Going down to 50

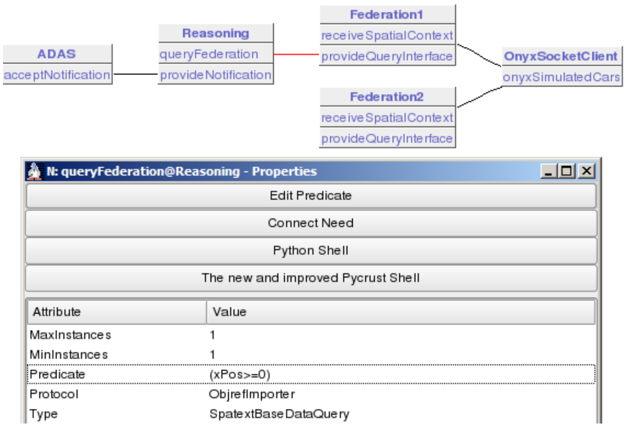


Figure 11. A snapshot of the services' interconnections: The own car is in the positive side of the federation sector „border"

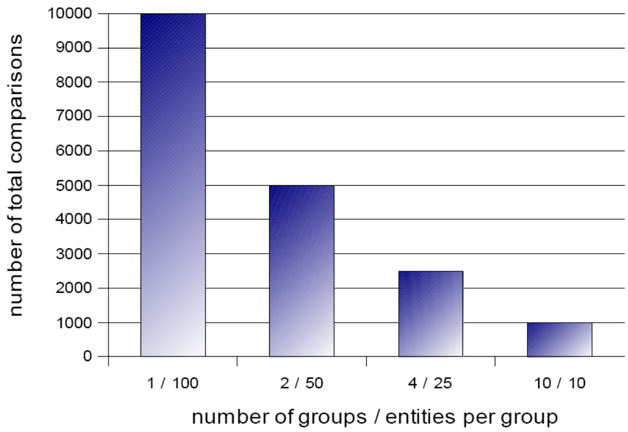


Figure 13. Reducing the complexity of queries by distributing entities over particular numbers of groups

entities, which still is a lot on open fast traveled roads like highways, the response time sinks to values around $10\,ms$, as shown in Figure 12. Taking network delays of cabled installations into account, the cumulated response or notification time of the system is below $30\,ms$. These measurements do not match any wireless communication and also do not take any computation time of tracking systems into account.

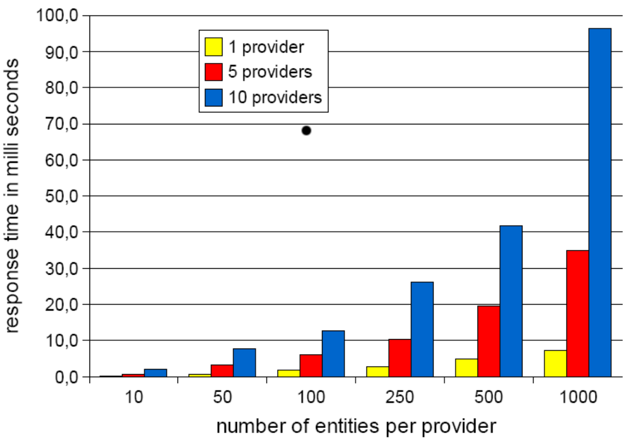


Figure 12. Overall response times of the federation service's entity histories access interface method (n-to-n query)

B. *Performance in Distributed Setups*

When dividing up entities a single query will only span one subset of entities. Therefore entities in different subsets are not related to each other with respect to the relation stated in the query.

Such an approach is depicted in Fig. 13 where 100 entities from the previous performance analysis are split into particular groups of entities so that the number of all entities in all groups equals 100 again. The diagram shows the decreasing number of comparisons while the number of groups is increased.

IX. Embedding and Applying in ADAS Systems

An ontology-based reasoning system alone is no assistance system for use in automotive environments. It only provides data of spatial interdependencies but does not apply them to the car or the driver.

Data from such a system can get applied in autonomous systems, as for instance, collision mitigation systems or automatic course correction systems, but also allows for driver information. At the moment driver information in assisting manner state the focused approach of in-car assistance systems as fully automated driving will be a topic of research for some more years. Assisted driving offers many opportunities for increased safety and environmental awareness.

Besides development of our context deduction system, our group also investigates top level ADAS system. The top level systems define the human-centered computer interface of the in-car system. In this section, different applications that relate on spatially deduced sensor data are illustrated.

A. *Direction Indication of Dangers*

One application that has to rely on such data is founded by an ADAS system that is intended to guide a car driver's attention to the direction on an imminent danger in the car's near environment [25]. Figure 14 shows a 3D arrow floating in the Head-up Display (HUD) of a car pointing into the certain direction of that danger. Since obstacles change their relative position quickly, the system requires very high update rates to give an immersive and accurate presentation. Our experiments indicate reduced reaction times with respect to mini-map presentations.

B. *Following Distance Assistance*

Queries from the reasoning system to the federation system, especially queries in a forward direction can be used to determine possible forward collisions. An ADAS system (see Figure 15), at the moment under development in our group uses information of this kind. In general, the system shows a bar floating in the HUD to extend a driver's anticipation about the car's physical behavior [26]. The bar indicates the position, the car would come to a stop when fully pushing the breaking pedal and shows,

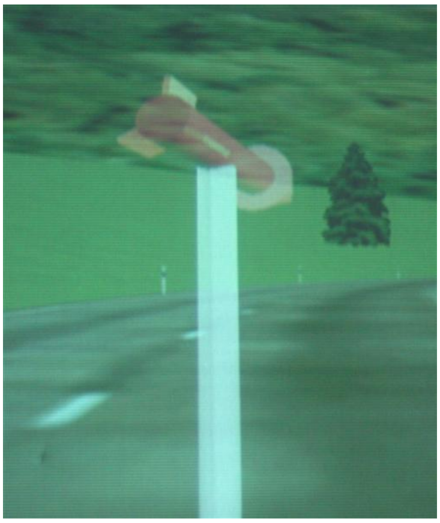
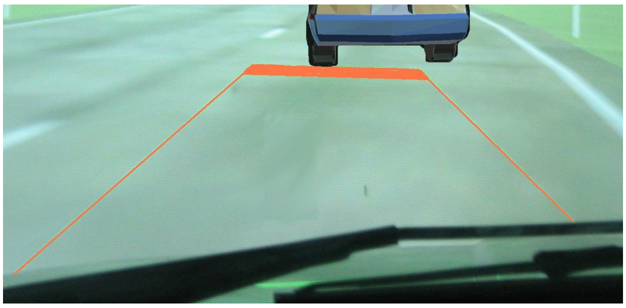


Figure 14. An indicator for the direction of an imminent danger in the car's environment, shown in the Head-up Display

which area the car will cover on the actual setting of the steering wheel. First results point to superior capability to stay in the own lane without increasing overall cognitive workload. Current extensions change the color of the bar, when approaching a leading car and therefore give an implicit follow-up distance warning.

(a) Bar presentation under normal conditions



(b) Bar presentation as follow-up distance warning

Figure 15. In-HUD presentation of the current driving state of the own vehicle showing different opportunities for open roads or lading traffic

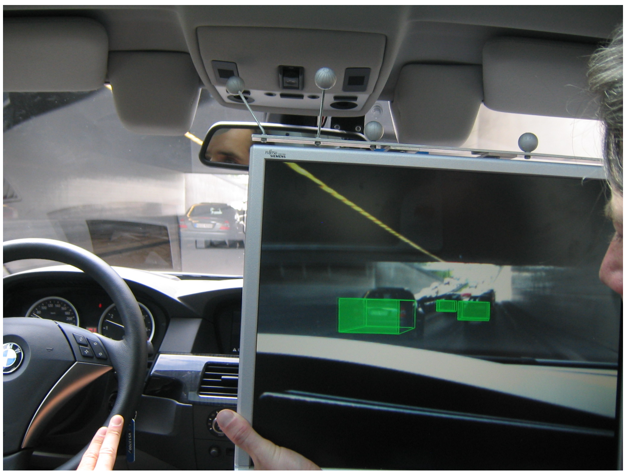
C. Designing ADAS Systems from Sensory Data

The modular approach of SCORE enables continuous integration and development of new ADAS systems. As ADAS systems must not be distractive, the design phase of visual schemes is a critical part of the whole development. Depending on the availability of sensor data and thus deducible knowledge, HCI developers have to find out how a top level ADAS presentation scheme can look like.

Such presentation schemes often depend on large scale HUDs. To enable parallelized research between HUD and presentation development, an in-car AR presentation system [27] has been developed.

The system incorporates alternatives for in-car spatial context providers, given by the laser-scanner. Feeding sensed tracking data into the federation services would allow for detailed knowledge of the spatial situation in front of the car and enable reasoning. The system itself enables discussion and evaluation of top level presentation schemes. For instance, Fig. 16(a) only shows the laser-scanner's sensor data directly, while Fig. 16(b) shows a boxed representation.

(a) One can see the laser-scanner data superimposed on the vehicles and the guardrail



(b) One can see the green boxes superimposed on detected vehicles

Figure 16. The TFT of the sensor data presentation system SensorVis held into the tracking volume while driving

X. Conclusion

In this paper, we presented our system that uses ontology-based reasoning, resting upon distributed federating components. As such a system requires a suitable middleware, we used DWARF for this purpose and showed, that, besides the tracking infrastructure, such a system is capable of performing spatially related contextual reasoning. We also showed that an overall highly dynamic traffic environment can be split into separate independent areas of stationary data acquisition, while traveling cars access this data and conclude information that is relevant for them. Dynamic short range network connections can get established to guarantee access to important data. The system shows how Semantic Web technologies and logic/rule-based systems are applicable to spatial context. Certainly additional focus must be laid on research in the field of networking when pursuing a system deployment in a wide area environment.

By using several examples we illustrated, that the functionality of the framework eases development of innovative ADAS systems.

However before deploying the framework for use with a real world application, we will extend the function repository of SCORE to serve in proactive assistance for local guidance in road traffic scenarios. Our further research on ADAS applications intends to simulate a perfect driver in order to give advises to the real car's driver. For instance, the proposed system recommends, which lane to choose in heavy traffic or, in other cases, for instance, just to accelerate to comfortably let another car get on the own lane. Identifying such a system's ontological structure will have a certain impact for ubiquitous computing technologies in the sector of intelligent ground transportation systems. Therefore the concept of Ubiquitous Augmented Reality [23] can focus new challenges in HUDs, enabling further distraction reduced driver assistance systems.

REFERENCES

[1] S. G. Klauer, V. L. Neale, T. A. Dingus, D. Ramsey, and J. Sudweeks, "Driver inattention: a contributing factor to crashes and near-crashes," in *Proc. of HFES 49th Annual Meeting*, 2005, pp. 1922–1926.

[2] C. Christen, "Intelligent Safety Systems - Sustainability Report in Economics and Mobility," 2002, Adam Opel AG.

[3] BMW, "Active-Cruise-Control Homepage," 2005, http://www.bmw.co.za/products/acc/default.asp.

[4] L. Walchshaeusl, R. Lindl, K. Vogel, and T. Tatschke, "Detection of road users in fused sensor data streams for collision mitigation," in *Proceedings of the 10th International Forum on Advanced Microsystems for Automotive Applications (AMAA'06)*. Berlin: VDI/VDE/IT, April 2006.

[5] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," in *1st International Symposium on Handheld and Ubiquitous Computing (HUC)*, 1999.

[6] C. Kray, "Spatial reasoning and agents: Exploring possibilities," 2001. [Online]. Available: citeseer.ist.psu.edu/594645.html

[7] Bundesministerium für Bildung und Forschung (BMBF), "INVENT - Driver Assistance Systems - Active Safety," 2005, http://www.invent-online.de/.

[8] T. Kosch, "Situation-Aware Communication in Vehicle Adhoc Networks," Ph.D. dissertation, Technische Universität München, 2005.

[9] A. Galloway, "Resonances and everyday life: Ubiquitous computing and the city," 2003. [Online]. Available: http://www.purselipsquarejaw.org/mobile/cult_studies_draft .pdf

[10] P. P. L. Siu, N. Belaramani, C. L. Wang, and F. C. M. Lau, "Context-Aware State Management for Ubiquitous Applications," Department of Computer Science, University of Hong Kong, Pokfulam, Hong Kong, Tech. Rep., 2004.

[11] C. Becker and D. Nicklas, "Where do spatial context-models end and where do ontologies start? a proposal of a combined approach," in *Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management in conjunction with UbiComp 2004*, J. Indulska and D. De Roure, Eds., University of Stuttgart : Special Research Area SFB 627 (Nexus: World Models for Mobile Context-Based Systems). Nottingham, England: University of Southhampton, 2004, pp. 48–53.

[12] Fischer, J.-G., "Spatial Context Management for Augmented Reality Applications, SCORE: Spatial Context Ontology Reasoning Environment," Master's thesis, Technische Universität München, Chair for Computer Aided Medical Procedures and Augmented Reality, 2005.

[13] H.-W. Gellersen, "Ubiquitous Computing," Institute for Telematics, Telecooperation Office, Universität Karlsruhe, Tech. Rep., 2001.

[14] K. Rothermel, M. Bauer, and C. Becker, "Digitale Weltmodelle - Grundlage kontextbezogener Systeme," 2003.

[15] R. Azuma, "A survey of augmented reality," 1995. [Online]. Available: citeseer.ist.psu.edu/azuma95survey.html

[16] T. Gruber, "Introduction to Imaging in Biomedical Informatics," Department of Biomedical Informatics, Columbia University, Tech. Rep., 2005. [Online]. Available: http://www.dbmi.columbia.edu/bioimaging/G4032/ontology .htm

[17] T. Berners-Lee, J. Hendler, and O. Lassila, *The Semantic Web*. Scientific American, 2001.

[18] J. Davies, D. Fensel, and F. van Harmelen, "Towards the Semantic Web: Ontology-driven Knowledge Management," Wiley, UK, Tech. Rep., 2002.

[19] D. Calvanese, M. Lenzerini, and D. Nardi, "Description logics for conceptual data modeling," in *Logics for Databases and Information Systems*, 1998, pp. 229–263. [Online]. Available: citeseer.ist.psu.edu/article/calvanese98description.html

[20] F. Baader, D. Calvanese, D. L. McGuiness, D. Nardi, and P. F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge University Press, Cambridge, UK, 2003.

[21] D. Brickley and B. W. W. W. C. McBride, "RDF Vocabulary Description Language 1.0: RDF Schema," 2004, http://www.w3.org/TR/rdf-schema/index.html.

[22] M. Bauer, B. Brügge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor, and M. Wagner, "Design of a component-based augmented reality framework," in *Proceedings ISAR*, Oct. 2001. [Online]. Available: citeseer.nj.nec.com/bauer01design.html

[23] A. MacWilliams, "A Decentralized Adaptive Architecture for Ubiquitous Augmented Reality Systems," Ph.D. dissertation, Technische Universität München, Chair for Applied Software Engineering, 2005.

[24] A. MacWilliams, T. Reicher, and B. Brügge, "Decentralized coordination of distributed interdependent services," in *IEEE Distributed Systems Online – Middleware '03 Work in Progress Papers*, Rio de Janeiro, Brazil, June 2003. [Online]. Available: http://dsonline.computer.org/0306/f/rei.htm

[25] M. Tönnis and G. Klinker, "Effective Control of a Car Drivers Attention for Visual and Acoustic Guidance towards the Direction of Imminent Dangers," in *Proceed-*

ings of the 5th International Symposium on Mixed and Augmented Reality (ISMAR), Oct. 2006.

[26] M. Tönnis, C. Lange, G. Klinker, and H. Bubb, “Transfer von Flugschlauchanzeigen in das Head-Up Display von Kraftfahrzeugen (Transfer of Flight Tunnel Visualizations into the Head-Up Display of Cars),” in *Proceedings of the VDI-Gemeinschaftstagung ”Integrierte Sicherheit und Fahrerassistenzsysteme*, Oct. 2006.

[27] M. Tönnis, R. Lindl, L. Walchshausl, and G. Klinker, “Visualization of Spatial Sensor Data in the Context of Automotive Environment Perception Systems,” in *Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR)*, Nov. 2007.

Marcus Tönnis Dipl.-Inf. Marcus Tönnis has studied physics and computer science at the Technical University of Munich, focusing on software engineering and data management. The topic of his diploma thesis covered the development of a data management architecture for ubiquitous ad-hoc networks and Augmented Reality. He spent a year at the software company Beck et al. before joining Prof. Klinkers AR research group as a Ph.D. student in 2004. He is now involved in a research project to develop novel AR-supported driver assistance systems for head-up displays in cars.

Jan-Gregor Fischer Jan-Gregor Fischer holds a Master’s degree in Computer Science from the Technical University of Munich with a focus on information technology in the field of mechanical engineering as well as on technologies in the context of Ubiquitous Computing and Augmented Reality. His Master’s Thesis comprises the development of a generic ontology-based software platform for managing spatial context in a way suitable to meet real-time requirements of Augmented Reality applications. In 2001 he joined the development team of c.baX GmbH Munich as software engineer specializing on development of custom data acquisition software. Since 2006 he has been working with the Intelligent Autonomous Systems research group at Siemens Corporate Technology (CT) in Munich. His research focus lies on ambient intelligence and agent technologies, semantic data interoperability and reasoning.

Prof. Gudrun Klinker, Ph.D. studied computer science (informatics) at the Friedrich-Alexander Universität Erlangen, Universität Hamburg (Diplom) and Carnegie-Mellon University (Ph.D.) in Pittsburgh, PA, USA, focusing on research topics in computer vision. In 1989, she joined the Cambridge Research laboratory of Digital Equipment Corporation in Boston, MA, working in the visualization group on the development of a reusable tele-collaborative data exploration environment to analyze and visualize 3D and higher-dimensional data in medical and industrial applications. Since 1995, she has been researching various aspects of the newly emerging concept of Augmented Reality, first at the European Computer-industry Research Center, then at the Fraunhofer Institute for Computer Graphics, and since 2000 at the Technical University of Munich. Here, her research focus lies on developing approaches to ubiquitous augmented reality that lend themselves to realistic industrial applications.