Electronic Communications of the EASST
Volume 11 (2008)

Proceedings of the
First International DisCoTec Workshop on
Context-aware Adaptation Mechanisms for
Pervasive and Ubiquitous Services
(CAMPUS 2008)

Integration of Ontological Scene Representation and Logic-Based Reasoning for Context-Aware Driver Assistance Systems

Simone Fuchs, Stefan Rass, Kyandoghere Kyamakya

12 pages

Guest Editors: Romain Rouvoy, Mauro Caporuscio, Michael Wagner
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Integration of Ontological Scene Representation and Logic-Based Reasoning for Context-Aware Driver Assistance Systems

Simone Fuchs[1], **Stefan Rass**[2], **Kyandoghere Kyamakya**[3]

[1]simone.fuchs@uni-klu.ac.at
[2]stefan.rass@uni-klu.ac.at
[3]kyandoghere.kyamakya@uni-klu.ac.at
Department of Smart System-Technologies
Alpen-Adria-Universität Klagenfurt, Austria

Abstract: Co-operative driver assistance systems share information about their surrounding with each other, thus enhancing their knowledge and their performance. For successful information exchange and interpretation, a common domain understanding is needed. This paper first presents an ontology-based context-model for driving scene description, including next to spatio-temporal components also additional context information like traffic signs, state of the driver and the own-vehicle. For traffic rules, we integrate the ontological scene description with a logic programming environment, to enable complex and powerful reasoning on the given information. The proposed ontology is discussed with respect to a set of validation criteria. For integration with logic programming a prototypical development of an overtaking assistant is shown to demonstrate the feasibility of the approach.

Keywords: ontology, context, logic programming, reasoning, driver assistance

1 Introduction

Context-aware collaborative driver assistance systems (DAS) need a common domain description for information exchange. Context for a DAS refers to the driving situation, consisting of the environment and all objects and traffic participants within it, which are currently relevant to the own vehicle. The driver's state and experience as well as the technical state of the vehicle with the mounted DAS are also influencing a driving situation. One additional, often neglected factor is the traffic law. The main task of an intelligent DAS is driver support, in contrast to autonomous vehicles. To make correct decisions, the system must be aware of the driving surrounding. Thus, a context-model is needed for representing knowledge about driving scenes. Collaboration with other vehicles and infrastructure enhances a single DAS' knowledge with additional information. Here a context-model is the basis for knowledge exchange between participants. Present-day DAS (e.g. the adaptive cruise control - ACC) are mostly stand-alone solutions, focusing on a highly specialized subtask of driving, with limited context-awareness. Current trends in DAS indicate that for future systems integration of stand-alone solutions is going to take place, thus resulting in smarter DASs. Those will be able to free the driver from difficult and tedious tasks. The overall driving context will become important for correctly recognizing and interpreting complex driving situations. DAS will become increasingly knowledge-based and methods will

be needed for modeling, handling and exchanging the vast amount of context information.
This paper presents an ontology-based context-model intended for scene-representation and information exchange in intelligent DAS. The model is presented and discussed with respect to a variety of pre-defined ontology-engineering criteria. Integration of the context-model with traffic rules in a logic programming environment is outlined, using the prototypical implementation of an overtaking assistant.

2 Related Work

The authors of [TFK08] present SCORE - the Spatial Context Ontology Reasoning Environment. The system is made of modular components that distribute the ontological knowledge and reason about the context's low-level spatial properties. SCORE understands queries like "Is car X overtaking car Y?". SCORE uses a description-logic based reasoner to derive information about a driving situation. However, it remains unclear which context objects are contained in the ontology and how the spatial information and relationships are actually represented. Also, the reasoning mechanism or the content of the rule base are not explained in detail. The authors seems to consider neither temporal concepts nor uncertain information.
[LGTH05] demonstrates a spatio-temporal solution that exploits qualitative motion descriptions. Movement parameters (e.g. speed) are mapped from raw sensory data to qualitative abstract classes. Production rules are used for reasoning on the qualitative scene descriptions. The approach is feasible for the spatio-temporal representation of moving objects. The question remains, if for some driving decisions numerical movement parameters are the better choice. Speed and distance values can be obtained easily and seem to be the better choice for time/speed calculations, especially if the vehicle is supposed to yield better estimations than a human driver. The presented rule base solely focuses on the spatio-temporal reasoning, further influence factors on the driving task are not taken into account.
Ontologies with context information for DAS have been developed in the RENA project [WBSS]. However, the focus of this project is on context-aware navigation systems, with a seamless handover between different in- and outdoor positioning systems, not on driver assistance.
Traffic rules, static traffic objects (e.g. signs) and environmental conditions are, to the best of our knowledge, not dealt with in current approaches, although they have major influence on recommended driving behavior. With the ongoing technical progress of sensing systems and GIS, information about those conditions will be soon available and should be included in both context-representation and reasoning. Our context-model extends spatio-temporal data with additional context information, necessary for deducing context-aware driving recommendations.

3 Ontology-based Driving Scene Representation

The context model has been developed in OWL, which has been chosen as suitable language for context representation and sharing, based on the results of the survey in [SL04]. An overview of the ontology's content is shown in Figure 1. There are three main superclasses in the hierarchy: *ContextObject, ObjectRelationship* and *MetaInformation*. *ContextObjects* includes both static and dynamic context objects of a driving situation. Examples are the driver, the own-vehicle, the

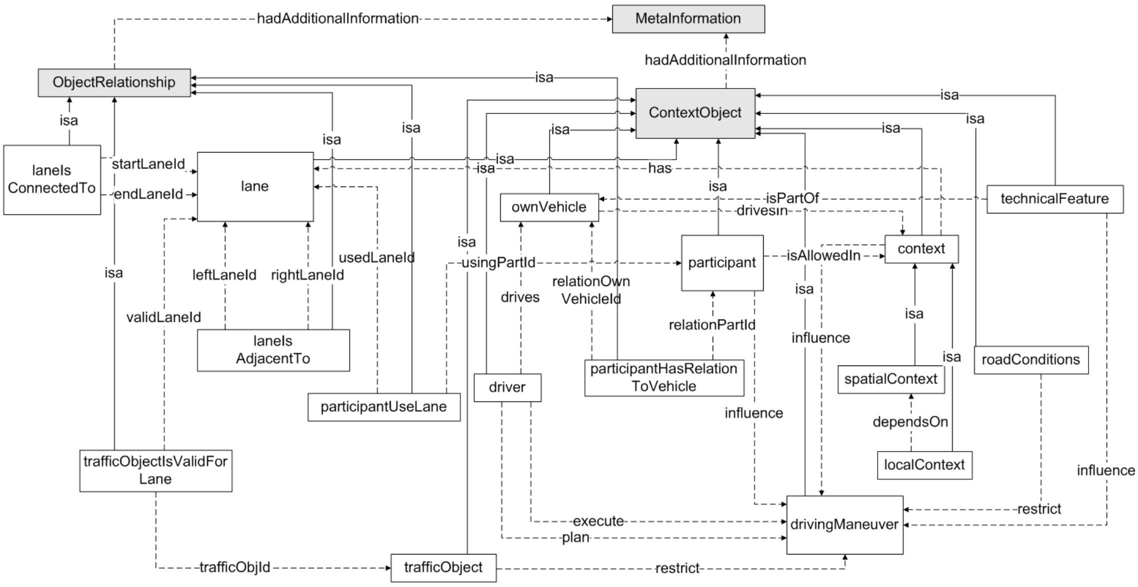

Figure 1: A context model for abstract driving scene description

driving context, participants, traffic signs etc. The spatial context is the current road type (highway, urban, ...) and is valid for a longer time-span. Local contexts are located within a spatial context and represent a sub-environment with special rules (e.g. intersection). Traffic objects are included from four major context categories: driver, own-vehicle, traffic regulations and driving environment with respect to the own-vehicle. Every object is annotated with datatype properties for further description. Traffic objects have relationships to each other, which are either of type 1:1 (represented with object properties) or n:m. In the latter case, the relationship is represented as a subclass of *ObjectRelationship*. For example, every other participant has a certain relationship to the own vehicle, the own vehicle itself has a relationship to an oncoming local context, traffic signs are valid for certain lanes and so on. Recognition of traffic objects using sensing devices has made substantial progress over the past decade. A number of projects have been conducted in the fields of pedestrian recognition [SGH04], traffic sign detection [MKS00], driver state detection [QY02] and lane recognition [MWKS04]. We therefore think it safe to assume that the traffic objects needed in our model can technically be provided.

3.1 Representing Uncertain Information

The input of a DAS is highly unlikely to be precise and reliable, especially if derived from sensing systems or provided by GIS systems. Therefore, uncertainty information has to be included in the context-model. The special class *MetaInformation* contains information about an object's source and it's reliability, the object's estimated quality (provided by the source) and the expected time-span of an object's validity, derived from distance or time-to-contact measurements, which provides information about when a certain object is becoming valid within the knowledge base and must be included in the reasoning process. One or more instances of the meta-information class are assigned to every object and relationship, because object information is gathered from different sources. At the moment the list includes on-board sensing systems, foreign sensing systems (e.g. from other vehicles) and static sources like geographic information systems (GIS), which can augment traffic object information (e.g. number of lanes, position of traffic signs, road type). The meta-information can be exploited during the decision process, using methodologies from the field of reasoning under uncertainty.

3.2 Representing Spatio-Temporal Information

Representing moving objects with a single time-span is sufficient for high-level motion description (cf. [LGTH05]). Spatial information between the own-vehicle and other participants is represented from an ego-centric perspective with the own vehicle at the center. Qualitative attribute values are used for the direction (front, rear, left, right and combinations) and the relative direction movement (towards, away, parallel). For the movement parameters speed, distance and line of sight our model uses numerical values, for reliable calculations associated with driving maneuvers (e.g. overtaking). Those rely on time-frames and speed-difference value calculations, which can be obtained from the given parameters with reasonable computational effort. Qualitative mapping is rather important when presenting results to driver. For spatio-temporal calculations, we expect better results from numerical calculations here in comparison to purely qualitative values. Also, numerical speed and distance values can be easily obtained from sensing systems.
The ontology is published on http://vi.uni-klu.ac.at/ontology/DrivingContext.owl. The ontology is intended for high-level scene representation of driving scenes, as it is used on a tactical level in DAS for driver decision support.

3.3 Representing Traffic Rules

For rule-representation, OWL currently supports the semantic web rule language (SWRL), a proposal for extending OWL with Horn-clause-like rules. Representation of complex rules is not efficient [Hor05] and until today, SWRL has not been improved and made part of the standard yet[1]. In [MBKL05], where OWL and SWRL are used to represent domain knowledge in logistics, some of the encountered problems, like lack of negation-as-failure, are discussed. For some years it has been rather quiet around SWRL now and not much progress has been made. A logic-based approach is more suitable and provides sophisticated reasoning mechanisms on the available knowledge. We used the *constraint satisfaction* paradigm. A constraint satisfaction problem (CSP) is defined as a triple $\langle X,D,C\rangle$ where X is a finite set of variables $X = \langle x_1,x_2,...,x_n\rangle$, D is a corresponding n-tuple of domains $D = \langle D_1,D_2,...,D_n\rangle$ such that $x_i \in D_i$, meaning a variable x_i can be assigned values from its corresponding domain $D_i = \langle v_1,v_2,...,v_n\rangle$. C is a finite set of constraints $C = \langle C_1,C_2,...,C_t\rangle$. A constraint $c \in C$ involving variables $x_i,...,x_j$ is a subset of the Cartesian Product $D_i \times ... \times D_j$ of compatible variable assignments. A constraint c is satisfied by a tuple of values $v = (v_i,...,v_j)$ assigned to variables $x_i,...,x_j$ if $v \in c$. An assignment is *complete* if a every variable is assigned a value. A complete assignment is a *solution* to a CSP if it satisfies all constraints in C. In a typical CSP the programmer defines the decision variables $x_i,...,x_j$ and

[1] http://www.w3.org/Submission/SWRL/, accessed on 6th May 2008

states the constraints as well as an (optional) optimization function. A standard solver tries to find assignments for the decision variables that satisfy all constraints, while at the same time minimizing (or maximizing) the objective function (*constraint optimization problem*).
Within a driving situation, the traffic rules represent the constraints that must be fulfilled. We have a *mixed CSP*: variables containing pre-determined values that cannot be changed, but must be included in the reasoning process. Examples are speed and distance values of other participants, provided by the context-model. The decision variables we want to find a value for are our own speed (integer value) and driving maneuver (set of finite values). We try to find a variable assignment that does not violate any traffic rules. If no solution can be found, one or more traffic rules are violated. There are hard and soft constraints. A *hard constraint* must not be violated in any case, e.g. a double white line or a given speed limit. A *soft constraint* can be gradually fulfilled, until it becomes a hard constraint. An example within a DAS would be an oncoming vehicle during an overtaking maneuver. The meeting point with the oncoming vehicle depends on the speed and distance of the oncoming vehicle and related to the overtaking duration. The hard constraint must hold that the overtaking duration is smaller then the time to contact, otherwise a collision will occur. If the constraint is fulfilled, it is so with a certain risk. The time to contact can be long after completion of overtaking (low risk) or very short (high risk).

3.4 Integrating Context-Information with the Reasoning Component

Contextual information of the present driving scene is represented with class instances of the provided context ontology, using OWL syntax. In this form, information is machine-readable and thus easily exchangeable between collaborating vehicles and infrastructure. Since a logic-based programming environment is typically not able to read OWL, the context information must be transformed, to be of use to the reasoning component. We developed a set of transformation rules (cf. [FRLK08]) that translates a scene description (given as OWL class instances) to the dynamic knowledge base of the reasoning component. First, the static framework (structures, enumerations etc.) is created out of the context-ontology. Every class, together with its datatype and object properties, is automatically transformed to a *struct*, representing the class description in logic programming syntax. This only has to be done once for the initial ontology and every time the ontology changes (making migration of the reasoning component necessary). Once the structures are available, every class instance of the form

```
<ownVehicle rdf:ID="ownVehicle_7">
   <speed rdf:datatype="&xsd;int">120</speed>
   <ownMaximumSpeed rdf:datatype="&xsd;int">180</ownMaximumSpeed>
   <lineOfSight rdf:datatype="&xsd;float">305</lineOfSight>
   <brakeIntensity rdf:datatype="&xsd;string">light</brakeIntensity>
   <steeringWheelAngle rdf:datatype="&xsd;int">2</steeringWheelAngle>
   <throttleIntensity rdf:datatype="&xsd;string">low</throttleIntensity>
   <length rdf:datatype="&xsd;float">3.09</length>
   <gear rdf:datatype="&xsd;int">5</gear>
   <width rdf:datatype="&xsd;float">1.65</width>
   <hasAdditionalInformation rdf:resource="#MetaInf_ownVehicle_7"/>
   <drivesIn rdf:resource="#spatialContext_1"/>
 </ownVehicle>
```

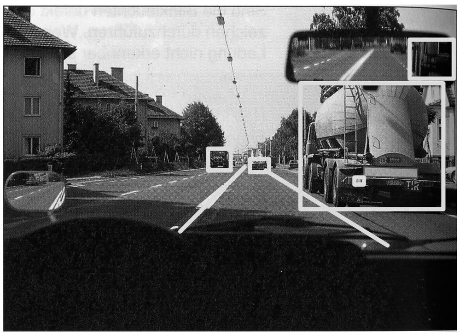

Figure 2: Relevant objects within a driving scene

```
<MetaInformation rdf:ID="MetaInf_ownVehicle_7">
   <source rdf:datatype="&xsd;string">staticallyProvided</source>
   <source rdf:datatype="&xsd;string">onBoardSensing</source>
</MetaInformation>
```

is translated to a dynamic fact

```
ownVehicle{
 objId:ownVehicle_7,
 speed:120, ownMaximumSpeed:180, lineOfSight:305, brakeIntensity:light,
 steeringWheelAngle:2, throttleIntensity:low, length:3.09, gear:5, width:1.65,
 source:[staticallyProvided, onBoardSensing],
 drivesIn: spatialContext_1}.
```

and asserted to the reasoning component's dynamic knowledgebase. Now the information is available and can be used as input for the decision process. When an object is changing or no longer valid, the dynamic fact is updated resp. retracted from the knowledgebase.

4 Discussion and Implementation of the Proposed Approach

4.1 The Context-Model

In addition to the standard tests provided by the ontology development tool, a representative variety of driving scenario snapshots was taken from real-world video-streams and from an Austrian driving school's teaching book. Approximately 120 scenarios were chosen, representing intersection crossing, overtaking and various situations from urban, highway and rural road driving. Within every scenario, the driving relevant objects have been manually tagged first (see Figure 2) and modeled with the ontology afterwards. Information, which would be present in a real world system but was not derivable from a scenario image, was given a plausible imaginary value (e.g. state of the driver). When looking at the driving scenarios, we found that most are similar, with only minor differences, e.g. different participants, number of lanes and speed/distance combinations, presence/abscence of traffic signs, local contexts etc. Therefore, the seemingly small number of scenarios is sufficient to show the feasibility of the approach for a first demonstration.

Based on the scenario mapping, we compared our ontology against a set of modeling and engineering criteria [KS07], with respect to suitability for the task of representing tactical driving decisions.

- **Applicability:** The model is useful to applications in need of abstract traffic scene representations, but not to completely foreign domains, like e.g. intelligent meeting rooms.

- **Comparability:** For our intended task, we consider this criterion less important. The ordering of qualitative classes for representation of spatio-temporal representation within driving scenes is the same world-wide. Mapping from quantitative sensory data to qualitative classes should not be done within the context model. Rather, the model should abstract from this details. For numerical speed/distance values different interpretations are possible: the SI or the English system. Since only three states worldwide are using the latter one (U.S., Liberia and Myanmar), the SI units can be assumed per default. Changing between system can solved using a system configuration entry outside the model.

- **Traceability:** The source, it's reliability and a quality assertion are recorded for every object in the meta-information class. Mapping of sensory data to a qualitative value (e.g. direction is front_left) should be done by a mapping component, because the input data and consequently the processing algorithm differs for various sensor systems. Since the source of the abstract object (containing qualitative values) is recorded, the mapping can be made available, either outside or inside the context-model with reasonable effort. A DAS operating on a tactical level will usually only be interested in the abstract object representations, not in the quantitative sensor data. Wherever the numerical values are important, they are represented explicitly within the context-model (e.g. speed of a vehicle).

- **History, logging:** In the current version, historization and logging is not yet included.

- **Quality:** For object quality information, the meta-information class should be used.

- **Satisfiability:** For qualitative values, the allowed range is listed in the model, using OWL-enumerations. For standard data types, we used the *xsp:minInclusive* and *xsp:maxInclusive* properties for providing range interval values. Multiplicity is modeled using the "Functional" attribute of a property.

- **Inference:** Inference for DAS, even if done on an abstract level is too complex to be modeled with current OWL capabilities. Traffic rules are therefore not included into the ontology, but out-sourced to a logic based reasoning component. Tools for further abstraction of the model are also not included, since it is already a high-level model. Refinement to higher levels of detail is possible with reasonable effort, without affecting current model semantics, exploiting OWL's class hierarchy.

Beside the context modeling criteria, [KS07] defined a criteria set for evaluation of ontology engineering.

- **Reusability, standardization:** Within the domain of machine-readable driving scene description on a tactical level, the model can be used for all tasks in need of such descriptions, without restriction.

- **Flexibility, extensibility:** New definitions can be added with reasonable effort without affecting existing dependencies. Particularly stepwise refinement with OWL class hierarchies can be done easily. This enables different applications to enhance the existing class-definitions to their necessary level of detail.
- **Genericity:** Our model does not provide a domain-independent upper-ontology.
- **Granularity:** Our model consists of abstract objects representing a high-level description for the tactical level of the driving domain. Refinement to finer levels of detail is possible (compare with *flexibility, extensibility*).
- **Scalability:** Cognitive and engineering scalability of our model is unproblematic, since it contains a comparatively small number of classes and properties. Reasoning scalability is not applicable, because it is entirely done outside the model.
- **Language, formalism:** Our model uses the Web Ontology Language (OWL), for scene representation resp. context-modeling. The reasoning process is outsourced to a logic-based approach and uses the OWL-descriptions as input.

For the scenario modeling, we found that the relevant information for representation of traffic driving scenes, including both traffic objects and their relationships, can be represented with our model. The model is not optimal for all criteria, which is mainly due to the fact, that it has been developed for a very specific domain. This is especially true for the context-modeling criteria; the ontology engineering criteria, the model fulfills to a great extent and is thus a suitable basis for context-representation within intelligence components for DAS on a tactical level. Traffic and reasoning rules are not directly represented in the context-model, due to the lack of complex rule-support within the web ontology language. Rules are implemented with a logic-based approach and the context-information is integrated as described above.

4.2 Implementation of the Rule Base

To test the feasibility of integrating OWL with constraint programming, we developed a prototype for overtaking assistance. The prototype translates and analyzes a given traffic scene (in OWL format) and uses the rule base to decide whether overtaking is currently wise or not. If not, the violated traffic rule(s) is (are) shown. Manually tagged driving scenes descriptions (cf. Section 4.1) were used as input. The automated collection of context information with computer-vision is an ongoing research topic in our group, but will not be discussed here. In the final system, the gathered context information will be dynamically retracted and inserted into the knowledge base, as new information about objects is obtained from the sensing systems. In the present version, transformation is always done for a complete traffic scene.
ECLiPSe was chosen as constraint programming environment for the reasoning component [Krz07]. A translation module has been developed, that automatically analyzes and transfers the OWL scene descriptions to ECLiPSe dynamic facts. Based on the resulting dynamic knowledge base, a set of constraints was specified to represent the hard and soft constraints for overtaking. A small graphical front-end was also created for presenting results of the deduction process. As programming language, the interpreted script language TCL/TK was chosen because it has an

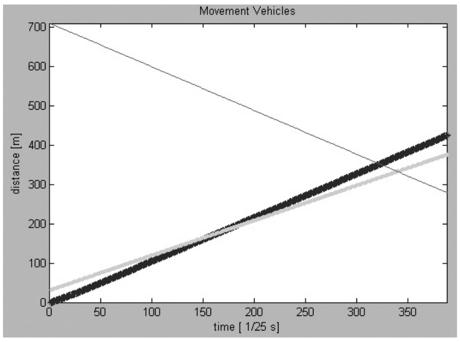

Figure 3: Speed/Distance curve of involved vehicles during overtaking

interface to the ECLiPSe environment. C++ would have been the alternative, also providing a tightly coupled interface to the knowledge base.

Depending on the spatial context, the system checks a different set of constraints. There are three hard constraints: 1) there must be a lane on the left for overtaking, 2) the legal speed limit must be reachable with a speed difference of at least 20 km/h and 3) there must not be a double white line. Soft constraints for overtaking are those that depend on the overtaking speed and the current speed and distance values in some way. Examples are the check for oncoming vehicles, for vehicles approaching from behind, sufficiency of line of sight, possibility of reaching a ban on passing while overtaking, sufficient side distance etc. Depending on the duration of the overtaking time of the front vehicle, these constraints are either fulfilled, but with a certain risk, or completely violated (see section 3.3). Figure 3 shows the speed/distance curve of a scenario with an oncoming vehicle, where the thick black line indicates the overtaking vehicle, the thick grey line is the front vehicle and the thin grey line represents the oncoming vehicle. The intersection of the lower two lines shows when the overtaking vehicle has reached the front vehicle. Realigning to the original lane (completion of the overtaking maneuver) takes place with one second safety distance. The meeting point with the oncoming vehicle is given by the intersection of the upper line and must take place after realignment of the overtaking vehicle, else the two vehicles would collide. Depending on the time difference between realignment and meeting point, a numerical risk value is determined. A small time difference indicates a high risk - the oncoming vehicle reaches the overtaking vehicle soon after realignment. The risk value decreases with increase of the time difference. The numerical risk value is mapped to a qualitative value using fuzzy classification, before presenting the result to the driver. The decision component searches for a speed value for overtaking fulfilling all constraints, using the minimal necessary speed difference (dictated by law) and the maximal possible speed difference, depending on the current speed limit as starting interval. The result of the search is either a single speed value or a narrowed down speed interval. In the latter case, the highest possible value is always communicated as a result, to minimize the overtaking time.

Next to spatial context information, traffic objects and participants, the decision component also includes the environmental conditions into the reasoning process. The values for maximum speed limit, acceleration/deceleration, safety distance and line of sight are adjusted to current visibility and road surface conditions. Furthermore, information about the state and risk-willingness of

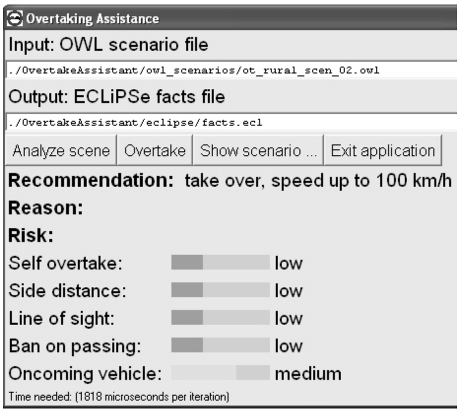

Figure 4: Prototype of Overtake Assistant

the driver is taken into account. If the driver is e.g. tired and the overtaking maneuver involves a high risk, overtaking is not recommended. A screen-shot of the prototype's graphical frontend is shown in Figure 4. Of course the presentation of results is not suitable for use in a real car while driving on a street. A discussion of how to best present information to a driver without overloading him/her, is beyond the scope of this paper.
Analyzing and translating the scene description to the dynamic knowledge base takes an average of approximately 100 milliseconds on an IBM Laptop with a 2 GHz Intel Pentium Processor and 2 GB memory, although TCL/TK is an interpreted language and thus slower. The file size of an average scene description is between 5 and 15 KB (unoptimized), depending on the complexity of the scene. Deducing a decision based on the contents of the dynamic knowledge base and presenting the result, is done with an average of approximately 1 to 2 milliseconds. This execution times show that it is possible without performance loss to integrate an OWL-scene description with a logic based reasoning system and exploit the power of deductive reasoning together with the ease and machine-readability of using OWL for context-representation.

5 Future Work

At the moment, the decision component does not take into account meta-information about traffic objects for the reasoning process. How to include uncertainty information and deal with it during deduction will be one of the major future steps.
We are currently also working on the design of a *learning component for self-improvement* of the DAS. Typically, drivers do not act one-hundred percent conform to driving regulations and with increasing experience develop a more efficient but also more risky style of driving. Decision parameters of the system should be automatically fine-tuned over time, using the decisions and behavior of experienced drivers as input. For this, historization and logging have to be added to the model. Instances of driving scene descriptions in which the driver acted oppositional to the proposed behavior are analyzed and archived together with the driver's state and behavior. If found necessary, rules are adapted accordingly: boundaries of risk mapping are shifted, tol-

erance values for speed differences are adjusted or additional maneuvers are allowed, always with respect to safe and legal driving. For hitherto unknown situations, where the system is not able to reach any decision, the driver's decision is validated and added to the knowledge base permanently as a new rule. The textual scene descriptions are a suitable mechanism for use in a pattern-matching process that compares driving situations with respect to their object and relationship instance value. If results pass a certain similarity threshold value, archived recommendations and driver behavior are retrieved and reused in the reasoning process.

6 Conclusions

Co-operative driver assistance systems (DAS) need a common domain understanding and a need for information exchange, with regard to driving scene description. In this paper, we presented an ontology-based context-model for traffic scene representation, which can serve as a foundation for domain-understanding, information-exchange and context-aware reasoning. We discussed the proposed ontology with respect to a set of both domain-specific and domain-independent modeling and engineering criteria. The model was found sufficiently expressive for the intended use and it has a variety of different applications. For traffic rule-representation we showed that it's feasible to integrate OWL and constraint logic programming, to exploit the advantages of both powerful information representation and reasoning, with feasible effort. The system is able to analyze the scene description and to deduce and present a recommendation near to real-time.

Bibliography

[FRLK08] S. Fuchs, S. Rass, B. Lamprecht, K. Kyamakya. A Model for Ontology-Based Scene Descriptions for Context-Aware Driver Assistance Systems. In *Proceedings of the 1st International Conference on Ambient Media and Systems (Ambi-Sys'08)*. Pp. 1–8. Quebec, Canada, February 2008.

[Hor05] I. Horrocks. OWL Rules, OK? In *W3C Workshop on Rule Languages for Interoperability*. 2005.

[Krz07] Krzysztof R. Apt and Mark Wallace. *Constraint Logic Programming using ECLiPSe*. Cambridge University Press, 2007.

[KS07] R. Krummenacher, T. Strang. Ontology-Based Context-Modeling. In *Third Workshop on Context Awareness for Proactive Systems (CAPS'07)*. 2007.

[LGTH05] A. Lattner, J. Gehrke, I. Timm, O. Herzog. A Knowledge-based Approach to Behavior Decision in Intelligent Vehicles. In *Proceedings of the Intelligent Vehicles Symposium*. Pp. 466–471. 2005.

[MBKL05] C. J. Matheus, K. Baclawski, M. M. Kokar, J. J. Letkowski. Using SWRL and OWL to Capture Domain Knowledge for a Situation Awareness Application Applied to a Supply Logistics Scenario. In *Rules and Rule Markup Languages for the Semantic Web*. Volume 3791, pp. 130–144. Springer Berlin/Heidelberg, 2005.

[MKS00] J. Miura, T. Kanda, Y. Shirai. An Active Vision-System for Real-Time Traffic Sign Recognition. In *Proceedings of the 2000 IEEE International Conference on Intelligent Transportation Systems*. Pp. 52–57. 2000.

[MWKS04] K. Macek, B. Williams, S. Kolski, R. Siegwart. A Lane Detection Vision Module for Driver Assistance. In *Proceedings of the IEEE Mechatronics & Robotics Conference (MechRob '04)*. 2004.

[QY02] Q.Ji, X. Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging* 8(5):357–377, 2002.

[SGH04] A. Shashua, Y. Gdalyahu, G. Hayun. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *IEEE Intelligent Vehicles Symposium*. Pp. 1–6. 2004.

[SL04] T. Strang, C. Linnhoff-Popien. A Context-Modeling Survey. In *First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004*. 2004.

[TFK08] M. Toennis, J.-G. Fischer, G. Klinker. From Sensors to Assisted Driving - Bridging the Gap. *Journal of Software* 3(3):71–82, 2008.

[WBSS] W. Wahlster, J. Baus, T. Schwartz, C. Stahl. RENA: Resource-adaptive Navigation. URL: http://w5.cs.uni-sb.de/rena/, last access on 03/03/2008.