# The Automotive Ontology: Managing Knowledge Inside the Vehicle and Sharing it Between Cars

Michael Feld
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
michael.feld@dfki.de

Christian Müller
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany and
EIT ICT Labs Intelligent Transportation Systems
christian.mueller@dfki.de

## ABSTRACT
Cars have been increasingly equipped with technology, meeting the demand of people for safety, connectivity, and comfort. Upcoming HMIs provide access to in-car systems and web services in a personalized manner that facilitates a large array of functionality even while driving, with other passengers also benefiting from an enhanced experience. Such intelligent applications however depend on a solid basis to be effective: Personalization, adaptive HMI, situation-aware intelligent systems – either of these require semantic knowledge about the user, the vehicle, the current driving situation. Since advanced functions coexist with sensors, other functions, and even other vehicles, where collaboration can be highly beneficial, a common understanding of knowledge and a platform to exchange it are also essential to reach the next level of intelligent in-car systems. This work describes the Automotive Ontology, which is located at the core of such an open platform. We give an overview of design areas relevant to automotive applications, as well as meta aspects that facilitate inference and reasoning.

## Categories and Subject Descriptors
H.3.4 [**Systems and Software**]: Information networks, User profiles and alert services; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages, Semantic networks

## General Terms
Design, Theory, Human Factors

## Keywords
Context and Situation Model, User Modeling, Personalization, Domain ontology

## 1. INTRODUCTION
The HMI that makes an appearance in new cars is becoming increasingly sophisticated. Long gone is the time when its purpose was restricted to basic assistance in the primary (steering-related) and secondary (safety-related) driving tasks as defined by [4]. Nowadays, large color screens with (more or less) intuitive interaction concepts provide an interface to Advanced Driver Assistance Systems (ADAS), offer views on different aspects of driving, but also on information, entertainment and comfort functions. In addition, the underlying functions themselves are becoming more complex. For example, a navigation system could give destination suggestions based on driving history and calendar, or it could determine when the driver is looking for a parking space, suggesting only sheltered parking sites in consideration of the weather forecast. Moreover, the HMI is not limited to the driver anymore: a consistent level of service and appearance throughout the car is a requirement for new innovations. We even go one step further by asserting that the experience should not only remain intact when you change seats, but also when you change the car – at least on the level that reflects personal characteristics and preferences.

With such a vast variety of services provided to the passengers, a lot of functions are actively running on the vehicle's on-board systems. Data is what all of these functions are based on, be it sensor data, user profiles, traffic broadcasts or car-to-car messages from peer vehicles. The key to success in making such an ecosystem work however lies in the organized and efficient access to the data: The heterogeneity and distributedness of data sources makes it essential to collect everything in one place, making it available to all functions that depend on it. As the data becomes enriched with structure and semantic meaning, we are also making the transition to knowledge management. By specifying meta information such as time, confidence, and privacy, functions can employ their own reasoning and knowledge can be shared across the boundaries of a vehicle without being dependent on a strict low-level protocol or a particular manufacturer.

The idea of having an "Automotive Ontology" at the core of the car's information systems was introduced in [8]. The present work evolves both the concepts and the ontology design, giving a concrete description of the knowledge representation aspect. Hence, its main contributions are a reference ontology design highlighting vital areas of automotive application domain knowledge, and a collection of meta properties that situation-aware in-car functions often have to deal with in conjunction with reasoning, including a recipe for modeling them. After fortifying our claim that advanced

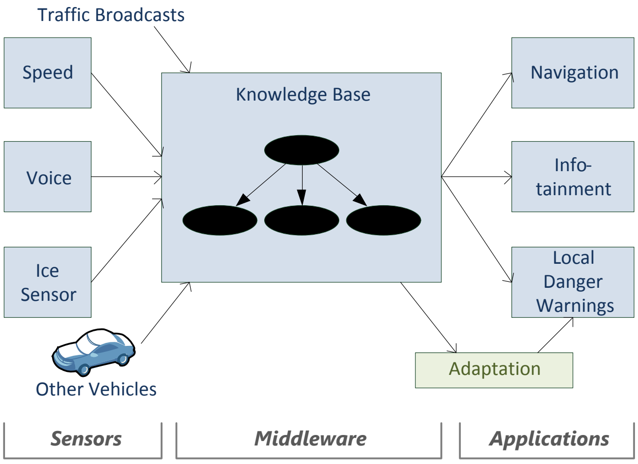


**Figure 1: The vehicular application platform: Relations between sensors, middleware (knowledge base), and applications. Depicted items are examples.**

knowledge is indeed a key ingredient of modern systems by looking at related work from the automotive field, we introduce the main areas of the ontology in Section 3. In Section 4, the meta attribute concepts are explained and specified. Section 5 summarizes our work.

## 2. RELATED WORK

Most traditional car applications are either contained in a proprietary architecture by the car manufacturer or need to obtain their data directly from sensors and other sources. The ontology that we propose in this paper creates a link between sensors and auxiliary data stores on the one side and applications on the other (see Figure 1). According to this schema, there are two types of related work we consider in this section. In order to estimate the potential of vehicle, user and context knowledge that is available to any application, we look at selected state-of-the-art research studies that deal with such information, which are the producers (sensors) and consumers (applications, functions) of knowledge. The more complex the connections, the larger the benefit that a middleware can achieve. The second type of related work deals with other knowledge-based middleware solutions.

A concrete application is the Intelligent Navigation System [9]. The purpose of this system is to infer the most likely destinations for a user at any time, e.g. right after entering the car, in order to simplify the navigation system setup. A fuzzy logic reasoning component is used internally to combine knowledge from various sources, which may also be subject to uncertainty. The knowledge considered as of now are emails, which may be obtained from an external device but also from an email account linked to the car, as well as appointments stored in the user's calendar. This is complemented by knowledge on which individuals are present in the car. Further information sources that have not been evaluated yet could be the driving history (GPS sensor logs) and the contents of conversations.

In the $sim^{TD}$[1] project, multiple advanced safety functions are created and evaluated in a large test fleet of 200 cars in the Frankfurt area in Germany. An integrated HMI and car-to-infrastructure communication units are part of the system. By looking at the functions and their specifications, a whole array of information pieces can be identified for each of them. For example, the function "Internet Access and Local Information Services", which displays additional traffic-relevant information to the driver (including webcam imagery), utilizes the vehicle position, traffic status information such as level of service and average speed, traffic event information, and traffic camera data. Functions in $sim^{TD}$ do not use a shared knowledge base (except for certain shared APIs such as the Vehicle API). This means that for instance if one function receives weather or traffic congestion information from a traffic operator, another function cannot use it. This may not be a problem for a fixed function bundle, but it would limit the options when the repository was to be extended by new functions.

Warning functions such as those used in $sim^{TD}$ deliver a safety improvement on their own, but there are various studies that point out how additional information can improve the result even further. For instance, [16] present a Collision Warning System (CWS) that can be adapted to the driver's experience or reaction time in order to optimize certain parameters of the system. When experience and reaction time are not known explicitly, it is possible to make assumptions about them from the user's age. Age is a key attribute in the user model and prerequisite of many personalization strategies. In [12], the impact of the age of a voice used to give instructions (which would more generally be the voice of the TTS) on driving behavior and satisfaction was measured for elderly drivers. The findings were a clear positive effect on either for younger speakers. [6] examined the factors that influence the decision-making process of older drivers and sketched a framework breaking down age into individual factors, which were categorized into domains such as health, driving abilities, experience, attitudes, and behaviors. Fewer studies than are *using* age deal with the *acquisition* of age information in a non-intrusive way. One possibility is classification based on speech [7].

There are also more dynamic driver attributes that can change several times a minute. Cognitive load and – closely related – stress are particularly useful in the automotive HMI context. [11] demonstrates an in-vehicle obstacle warning system which may show more helpful hints (so-called action suggestions), e.g. "brake" or "change to the left lane", in addition to a mere obstacle warning, when stress is detected. A prototypic system was implemented in a driving simulator environment. Stress level information is based on biosensors and driving performance assessment. Affective state and emotion are dynamic aspects as well. Since negative emotion is known to have a deteriorating effect on driving performance, it seems like a good idea to counter such trends if possible. [17] performed an experimental study to confirm exactly that, classifying emotions based on biosensor readings, and making the driver aware of it. In addition, some counter strategies (e.g. suggesting breaks or breathing exercises) could be taken. Apart from that, the authors point

[1] Safe Intelligent Mobility Test Area Germany, `www.simtd.de`

out the lack of established user models also supporting properties such as emotion and personality, hence they created a custom model. There are many more ADAS functions which can benefit from enhanced knowledge. [14] gives an overview of several current and possible future systems in the light of the AIDE project, which is itself very knowledge-dependent, trying to create an adaptive HMI for certain assistance systems.

Many of these surveyed examples are still in prototype stage, so they do not have to take into account how they could fit into existing platforms, collaborate with other applications, and ensure they have access to the information they need. They also might miss out opportunities for exploiting additional data that may be available. Although not as numerous, some work however is available on structuring, collecting, and processing knowledge, from which a few examples are presented in the following.

Most work can be found for specific areas, such as local danger warnings [18]. [5] come up with a classification scheme for situations that a car may record using its on-board sensors. The authors emphasize the need for a common understanding for such a scheme of situations in order to maximize the safety benefit in particular when the information is exchanged between vehicles. Obviously, the same applies to on-board applications, which may adjust their behavior to fit the type of situation the car is currently involved in.

[21] introduce an approach to context-awareness, which tries to collect "context atoms", which may be individual observations from sensor data, and infer "context situations" from them, such as left turn or acceleration. They also define an ontology for the context atoms, which contains the three main concepts of environment, car, and driver. The car in turn is split into the three systems power, security, and comfort. For the driver, the most critical knowledge seems to be related to the driver's current state, which is made up of atoms such as heart beat, blood pressure, diameter of pupils, etc. While the ontology was not designed to comply with a particular standard, it still appears to capture many common scenarios. One drawback of the approach is the lack of support for meta information, which makes it more difficult e.g. to take uncertainties into account for the reasoning.

A similar recent attempt to create an ontology for the automotive field, in particular what was called Intelligent Driver Assistant Systems (I-DAS), can be found in [13]. Developed in conjunction with InVANET, which investigates alerts based on context and driver parameters, it features a middleware component for reasoning tasks supported by an ontology that models concepts of various common driver assistance systems, e.g. adaptive cruise control and lane departure warning. It is however not clear whether the system is extensible beyond this pre-selection of systems. Another possible disadvantage is the outsourcing of history data into a different store, thereby reducing the possibilities for reasoning. Apart from that, the authors emphasize the lack of existing ontologies that model the automotive systems field in a broader sense, and not specific to individual functions such as multimedia or navigation systems.

A current EU-funded project also dealing with knowledge management in the vehicle is OVERSEE[2], although with a strong focus on security. Overall, work in the area is limited and the outcome is often not very concrete yet. The next sections address those challenges by presenting several ontology design concepts.

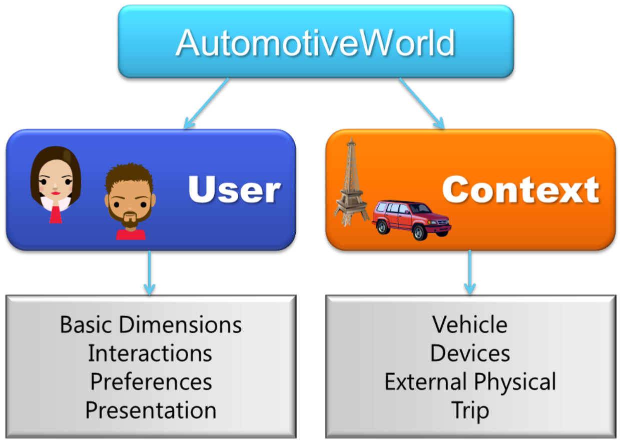


**Figure 2: Areas of the ontology, which map to first and second level concepts.**

## 3. ONTOLOGY DESIGN: CONCEPTS AND RELATIONS

Ontologies formally describe how knowledge for a specific domain looks like and what relations exist between concepts. The ontology, inter alia, enables the sharing of information between different sources and applications, because it defines a common basis on where to find and how to interpret data. Ontologies range from simple and flat (i.e. containing few hierarchical relations) to extremely detailed and complex. While a simple ontology is easier to learn and understand for a human working with it, adding structure increases the knowledge that instances implicitly express. A common disadvantage of large ontologies is the increased inefficiency in querying and reasoning tasks. Even powerful machines (which CPUs in cars are generally not) can reach their limit when the structure is getting very complex or when the number of instances is very high. Hence, a design goal for the Automotive Ontology was to avoid modeling an unnecessarily high level of detail. This also coincides with another design goal, namely to simplify access to information for applications. Apart from the argument of ease of handling, storing raw data such as sensor data in the ontology and keeping a history of old data could otherwise cause issues with respect to performance. As an additional advantage, a simple model also simplifies data exchange with other applications. The ontology design makes use only of several ontological constructs, which are:

- Concepts. Concepts fulfill the same purpose as in other typical ontologies by defining entities. They are a prototype of a knowledge instance.
- Is-a relation. A relation between concepts through which they can inherit from other concepts using this

[2]Open VEhiculaR SEcurE platform, `www.oversee-project.com`

relation. Example: A microphone is a device. In general, inheritance does not play a major role in the Automotive Ontology design, as in most cases, it would not add sufficient value to justify the introduction of a new concept.

- Has-part relations. This is the predominant type of relation between concepts. The semantics are that one concept is a child of another concept. One common case is a collection, which has one or more child concepts. Has-part relations can be annotated with number restrictions that express how many parts can be specified. Example: The car (interior) has four seats. Each seat "has" a passenger.
- Properties. Pointers to plain values that do not have an identity, but are only evaluated with respect to the concepts to which they belong. They are sometimes also called attributes, slots, or fields. Example: A seat heating has a property *level* with a value of *50%*.
- Has-property relations. These relations have concepts as the domain and properties as the range. They define the properties of an object.
- Meta-data. Meta data is a special relation from properties to meta values. There is a fixed set of meta properties that can either be present or not. They are described in Section 4.

The ontology graph has another property: It can be transfered into an object model, like they are used with object-oriented programming languages. An object model is very similar to an ontology in many regards [20]. The main difference is that ontologies can cope with arbitrary relationships between concepts, thus allowing advanced reasoning. The isomorphic relationship between both allows an object model to be generated from the ontology description, which in turn makes it more accessible to programming languages. Furthermore, it suggests that object modeling techniques such as UML [3] can be applied to the automotive technology as well, which is consequently applied herein for visualization.

The basic layout of the ontology is shown in Figure 2. The root of the ontology is the *AutomotiveWorld* (often also simply called *Root*). A few concepts are top-level collection concepts. They are *Users*, *Vehicles*, and *Trips*. The ontology generally has a user-centered view on the world, although the largest part of the remaining knowledge is unaffected by which user is currently logged in and could be shared. For this reason, we differentiate between *user* and *context* model. It is possible to divide the ontology into several sections. These sections represent particular contexts or sub-domains. The following pages describe these individual parts of the ontology.

## 3.1 User Model

The user-sensitive aspects mostly reside in the special **Users** branch and *User* concept, since the user is the subject of applications following the User-Centered Design (UCD) principle. These are most notably user characteristics and state. All remaining aspects are part of the context model discussed later.

An excerpt of the knowledge contained in the node is shown in Figure 3. The most personal information is located in the *Basic Dimensions* concept. This concept was created largely after the General User Model Ontology (GUMO) [10], which collects user dimensions such as physiological and emotional state, skills, and characteristics. The native GUMO part is complemented by contact information, which is one of the less detailed aspects of GUMO. Here, the extensible XML Contacts Schema [15] was chosen for the representation of contact information, such as names, addresses, and phone numbers.

**Preferences** are an important aspect of adaptive systems. Many preferences are application-specific, but there are also more general ones, which should be included in the ontology. We distinguish between preferences and interests, the latter being more abstract and less related to HMI. Preferences could be navigation preferences (e.g. types of road, driving at night), visual preferences (e.g. font, colors), interaction settings, privacy settings, and others. Interests include music, movie, literature, and other interests using an extensible taxonomy of categories, which would be used for instance in the recommendation of points of interests for navigation.

**Interactions** seen from the perspective of a particular user are part of the user model. It contains discourse information on human-machine interaction, but also on inter-human interaction such as conversations. Dialog systems are heavily based on knowledge in this branch of the ontology. Knowing who is talking to whom also helps with resolving the topic of a discussion and creating an interaction graph. Another application is to determine whether a certain user is currently "occupied", i.e. spending most of his cognitive resources on some resource-intense task.

Similarly, the **Presentation Model** is specific to one user. It is the primary source of information when the system autonomously adapts the user interface, a process that benefits if every adaptable entity has an instance representing it in the knowledge base. The model describes concepts of the user interface at a high abstraction level, so it does not take the role of a user interface manager or support rendering. The granularity is just sufficient to model the entities that are subject to adaptation rules. Examples are informational or warning messages, segments of speech output, or different display regions of the screen.

Additional concepts are the **Authentication** node, which facilitates the identification of a user based on IDs or biometric data, and the **Services** node for storage of messages, appointments, contacts, etc.

## 3.2 Context Model

Most aspects related to the vehicle can be found in the **Vehicle Model**. The vehicle in which the system is running plays a special role, but the structure is intended to host knowledge on any vehicle, including those in the vicinity of the user or other cars owned by one of the passengers. To make the knowledge accessible, large parts of the vehicle model is driven by a physical view: For example, the root level distinguishes between interior (cabin) and exterior. The exterior includes the engine compartment, which in turn contains the engine with properties such as the max-

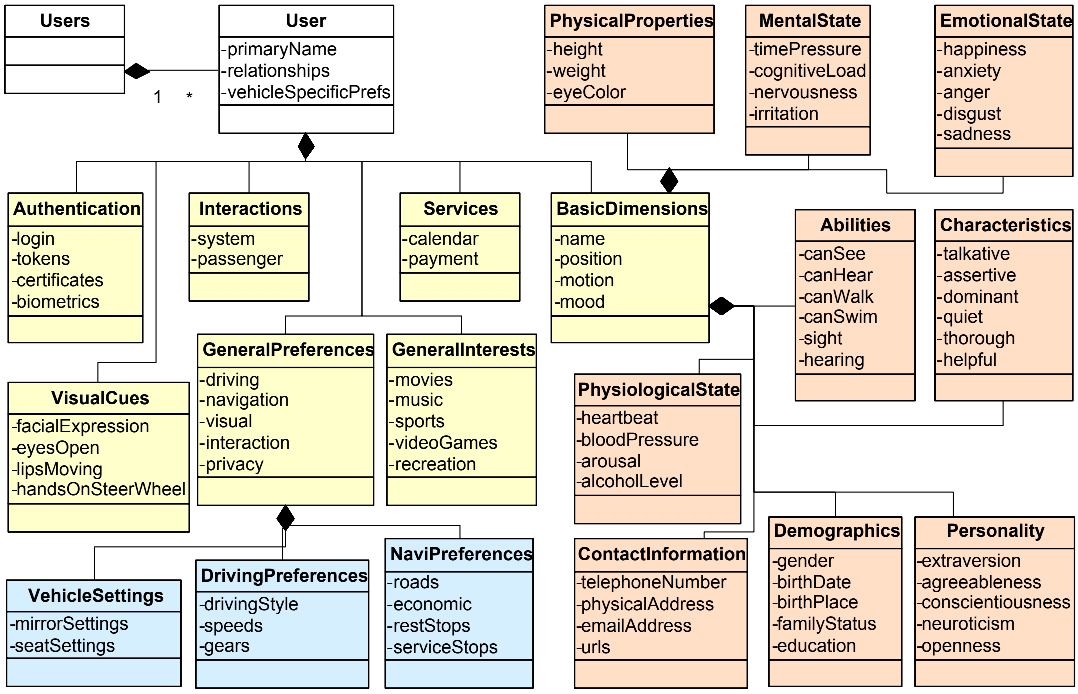


**Figure 3: The Automotive Ontology – User Model (excerpt). The relations are not labeled for better legibility. Color groups topically related aspects.**

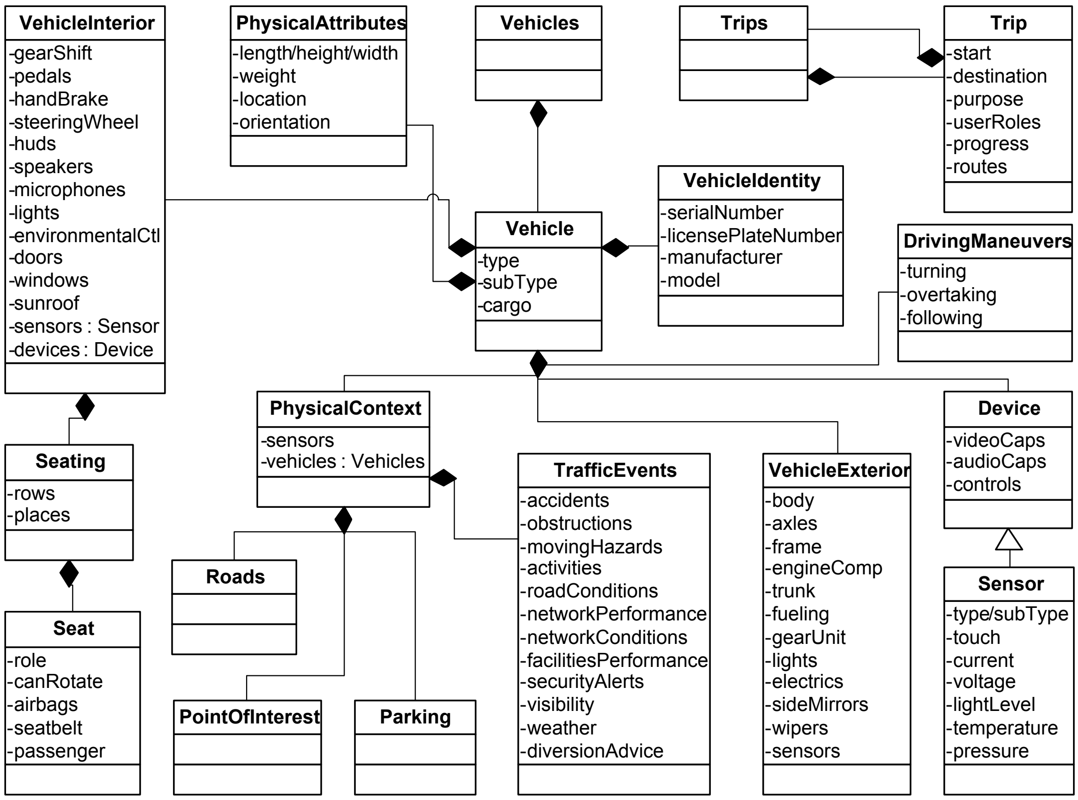


**Figure 4: The Automotive Ontology – Context Model (excerpt).**

imum rounds per minute. Additionally, the vehicle model contains information about the identity of the vehicle (license plate number etc.), current maneuvers, and all known cargo loaded into the cabin or trunk. The vehicle model connects to various other models: to the user model (each seat can be associated with a user), the external physical context, and the devices and sensors.

The **Devices** sub-ontology is a part of the vehicle ontology. It provides an abstraction for devices in the car that can be used by applications. Sensors in the Automotive Ontology are special devices which serve as input source to the system, but provide only one or more plain, scalar input values. These values are stored directly in the knowledge base, as opposed to most other (high-bandwidth) devices, which need a separate communication channel for the actual data.

The concept of a **Trip** is important for applications which consider the course of driving. For example, various personalized assistance systems would act differently according to the type of travel – if it is a business trip, a shopping tour, a family vacation, or a chauffeuring of the kids to school. Such a classification also affects the roles of the passengers. There is also an upcoming category of in-car application dedicated to planning entire trips with stops for refueling, resting, and sightseeing. According to this, a trip is a logical concept. It cannot be mapped directly to phases where the engine is running or to a certain timespan. A long trip can span multiple days and can further consist of sub-trips, which results in a hierarchical structure.

**External Physical Context**. A car including its users always has a physical setting based on its current position. Its elements, which do neither relate to the user nor the user's car, are contained herein. The physical context is not constrained to a particular geographic region. Instead, the application determines which objects are available and relevant to be stored in the context. Instances in the physical context should make use of the meta data model to annotate the position or region to which they apply (see Section 4.3). This flexibility is needed because each application considers a different range as "close" or "far". The model consists of traffic events, parking facilities, and generic points of interest. The traffic event section is the most extensive of them and contains many safety-critical items of knowledge. It is designed after the ISO/TS 18234-4 norm for road traffic messages (RTM) [1], which allows many complex traffic situations to be accurately described. Its comprehensiveness was decisive for its selection over other models, such as the Local Danger Warning Ontology (LDWO) [18]. *Points of interest* are the most generic concept to refer to a certain location or region, and are not described by more than a set of meta attributes such as title and relevance. A selection of concepts from the context model can be found in Figure 4.

# 4. ONTOLOGY EXTENSIONS: ASPECTS OF META KNOWLEDGE

There are several aspects of the automotive domain that should be taken into account for knowledge representation. Some of these aspects are so universal that they affect the general design of the model. For instance, as a car is in motion most of the time, the context may change very often and also quite dramatically even in a short interval. Thus, the aspect of time plays a vital role for the design of an in-car knowledge management system. Being able to associate a point in time or some other recurring attribute with every instance and even property in the knowledge base allows advanced and efficient reasoning. In this section, we introduce the meta concepts time, confidence, location, and privacy that are relevant in our domain.

## 4.1 Modeling Temporal Aspects

Time is one of the factors that characterize a situation. The understanding of temporal aspects is important for a system to make qualified decisions, e.g. as part of reasoning or filtering approaches. This also affects both the physical context and the user's state, the latter which can also change from one instant to another when a difficult situation arises. There are two independent fields that can be set: A point in time and a duration. Specifying both results in an absolute interval.

The time information can be used to specify when a value was recorded, e.g. by a sensor; when it was computed using an algorithm, or when it was entered by the user. This information can be used to identify the situation to which data in the knowledge base refers. Based on this value, an application can determine whether the value is relevant for the decision at hand. Values that are considered to be outdated can be ignored. For example, a traffic jam broadcast from an hour ago might not be considered relevant for the collision warning function, but it would still be considered by the navigation system for choosing between two routes when time is an issue.

Certain types of knowledge in the Automotive Ontology are constantly updated. We speak of historical data when outdated knowledge is not deleted, but intentionally kept for reference. There are multiple reasons of why one would want to do so. Consider these examples: 1) To draw a map of the car's route, it is necessary to look the car's position over the last hour or so. 2) To generate a weather forecast, sensor information from a certain period of time can be aggregated. 3) To learn about the user's typical state and changes to it, the system has to monitor a set of variables over a period of time. In a concrete scenario, e.g. an intersection, it would be possible to look at cognitive load and reaction time at the previous intersections. Projection is in some sense the opposite of historical data. Sometimes, an application is interested in how a certain value would most likely develop in the near future, for example the distance to a certain car 10 seconds later. Predictions are rather dynamic based on both when they are computed, how they are computed, and how far the projection distance is in the future. As such, they are typically not included in knowledge bases. If the knowledge base however does support dynamic knowledge, their inclusion helps to make the design even more intuitive, as no external functions have to be invoked.

To summarize the prime aspects of time meta data:

- Specifies recording time and validity period
- Date/time and range are given in milliseconds
- Example: A speed of 80km/h was recorded at 9:22:15

## 4.2 Uncertain Knowledge and Reasoning

Reasoning describes the task of answering complex questions using the facts stored in a knowledge base, and possibly using a mechanism that describes how further facts can be automatically derived. Theoretically, the results can in many cases be unambiguously obtained by applying logical formulae. For example, when the speed of the car is known and the question is whether it surpasses the current maximum allowed by law, a comparison could be used as simple form of knowledge inference from *actual vehicle speed* and *allowed maximum speed* to *speed maintenance*. However, in practice, much of the collected information is not known for certain, therefore care must be taken when drawing conclusions from it. The reasons for this uncertainty are manifold and can be divided into two main categories: uncertain knowledge sources and uncertain reasoning methods. The first category includes the accuracy of information extraction systems, quality of pattern recognition models, average precision and reliability of hardware sensors, human errors, and more. The second results from probabilistic or heuristic models, design limitations, limitations of the experts specifying rules etc.

The existence of uncertainty is not a hindrance by itself. Instead, it requires a strategy to quantify and deal with it appropriately. It is known since the early days of AI that multiple uncertain sources can be combined to form a new, more reliable source (see e.g. [19]). To what degree uncertain information is acceptable obviously depends on the application. This is also related to the cost of a misclassification: If the cost of using the wrong information is low, then the impact of the confidence also decreases. It however also decreases if the cost of *not* using some vital information is high, for example, if it helps to avoid an imminent collision. Some the most significant sources information that contains uncertainty are emails (which contain a lot of information, but in a largely unstructured form that requires NLP techniques to understand), calendars, conversation in the vehicle (barely structured and including ASR errors), people present in the vehicle, driving history, and personal data (name, age, height, living and work place, etc., which may involve uncertainty when obtained using non-intrusive methods). An example of how such sources can be combined in the automotive domain was given in [9].

Confidence should not be confused with the trustworthiness of the information source. It also does not consider the age of the information. Reasoning under uncertainty, e.g. applying fuzzy logics, requires that confidence meta information can be specified for every fact in the knowledge base as follows:

- Expresses the likelihood that a value is accurate
- Specified as a probability value in the range 0..1
- Example: The confidence in fuel level = 30% is 0.9.

## 4.3 Modeling Location

Georeferencing is a concept where the location at which a physical entity is located or an area which it occupies is described in a formal way, e.g. using a 3D coordinate system or by referencing particular entities. Location is needed for all types of traffic events, both large-scale (traffic congestion near Frankfurt in 30km) or close-range (pedestrian crossing the street 10 meters ahead). Distance and dimensions indicate the impact and relevance to the driver. Georeferencing is also an essential component of the navigation task. Finally, the annotation of entities with location helps the vehicle to obtain a more detailed picture of the user's world, and therefore extends the options for personalization, such as by identifying favorite shopping areas. By specifying location meta-information, the location to which an entity refers can be described. It can also be used to document the location at which a fact was produced, e.g. the car's position at the time a sensor value was recorded.

There are several reasons for having multiple ways of expressing location information. For example, to highlight a location on a map, a precise coordinate system such as GPS can be used by the car internally. GPS would also be used to stamp data from environmental sensors on the car, since using the car's own positioning system provides the best accuracy. However, when exchanging information about a traffic jam using car-to-car and car-to-infrastructure communication, the location of a traffic jam or accident would more likely be specified by using a reference to the road, mile, and lane since maps have varying accuracies and can be offset from each other, in which case GPS coordinates could easily be mapped to the wrong road. The driver might have yet another preference for expressing a location. For her, it might be appropriate to refer to "the intersection behind the yellow building", i.e. to use descriptive wording in natural language and references in the visual range.

In short, location meta data consists of:

- Position where a physical entity is located or an area that it occupies
- Specified by one or more schemes: GPS coordinates, TPEG-LOC [2], references to instances in the ontology, or natural language description
- Example: The traffic jam is at 51.49L, -0.14F.

## 4.4 Privacy

As soon as profile information is shared with others, be it other users, cars, public or official sites, the question of who may read what gains significance. Allowing anyone access to the full knowledge base would most certainly include access to items not meant to be published, both most obviously from a security perspective (log-in data, credit card information...) or rather from a privacy point of view (user's age, weight, driving history...).

The Automotive Ontology introduces a privacy specification meta concept. It defines two intersecting "layers" or dimensions of access that can be used to describe the extent to which information can be distributed. The first dimension is the user relationship range, including contacts, friends, family, colleagues, and traffic management, while the second is the physical range, consisting of the vehicle, convoy, and several fixed-range geographic ranges. This concept does not replace a fine-grained access control that specifies IDs of users given or revoked access including inheritance.

- Describes the sensitivity of information
- Fixed access specifiers of type *relationship* and *range*
- Example: The vehicle's speed may be queried by family members or cars in 1km range only.

# 5. SUMMARY

The previous sections presented what we consider an important step to a comprehensive, open platform for knowledge management in the automotive context. The Automotive Ontology introduced above is the result of a survey of practical aspects that are recurring in many intelligent systems today and in the near future, and they are in several cases based on existing taxonomies. Researchers in the area of automotive HMI should consider basing the data infrastructure of applications on this scheme. For general knowledge-based approaches such as personalization frameworks, including recommendation and adaptation, or car-to-car communication, this ontology might be a good starting point for data representation. We have also pointed out a number of key challenges that are associated with the in-car domain and how they can be supported by the knowledge base: temporal aspects, fuzzy reasoning, location, and privacy considerations for car-to-car data exchange. In our view, the related ontology concepts make up the minimal tools an in-car middleware application should offer to support a large variety of application scenarios. While the current stage of the ontology design is still research that needs to prove itself in different practical scenarios, one of the goals of this work is to encourage discussion about the informed knowledge exchange between vehicular applications, thus enabling an open platform and rich assortment of applications to come.

# 6. REFERENCES


[1] ISO/TS 18234-4: Traffic and Travel Information (TTI) - TTI via TPEG data-streams – Part 4: Road Traffic Message (RTM) application, 2006.

[2] ISO/TS 18234-6: Traffic and Travel Information (TTI) - TTI via TPEG data-streams – Part 6: Location referencing applications, 2006.

[3] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Object Technology Series. Addison-Wesley Professional, 2nd edition edition, 2005.

[4] H. Bubb. *Der Fahrer im 21.Jahrhundert*, chapter Fahrerassistenz primär ein Beitrag zum Komfort oder zur Sicherheit?, pages 25–33. VDI-Verlag, Braunschweig, 2003.

[5] F. Dötzer, T. Kosch, and M. Strassberger. Classification for traffic related inter-vehicle messaging. In *5th IEEE International Conference on ITS Telecommunications*, Brest, France, 2005.

[6] D. W. Eby, L. J. Molnar, J. T. Shope, J. M. Vivoda, and T. A. Fordyce. Improving older driver knowledge and self-awareness through self-assessment: The driving decisions workbook. *Journal of Safety Research*, 34:371–381, 2003.

[7] M. Feld, F. Burkhardt, and C. Müller. Automatic speaker age and gender recognition in the car for tailoring dialog and mobile services. In *Proc. of Interspeech 2010*, pages 2834–2837. ISCA, 2010.

[8] M. Feld and C. Endres. Sharing User and Context Models in Automotive HMI. In *Adjunct Proceedings of the Second International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Pittsburgh, United States, November 2010.

[9] M. Feld, M. Theobald, C. Stahl, T. Meiser, and C. Müller. Generating personalized destination suggestions for automotive navigation systems under uncertainty. In *Adj. Proc. of the 19th International Conference on User Modeling, Adaptation and Personalization (UMAP)*, pages 22–24, Girona, 2011.

[10] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. GUMO - The General User Model Ontology. In *User Modeling*, pages 428–432, 2005.

[11] D. Ibrahimi. User-adaptive warning systems: Online assessment of driving performance and driver stress on the example of local danger alerts. Master's thesis, Computer Science, Saarland University, April 2011.

[12] I.-M. Jonsson, C. Nass, H. Harris, and L. Takayama. Matching in-car voice with driver state: Impact on attitude and driving performance. In *Proceedings of the Third International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pages 173–180, 2005.

[13] S. Kannan, A. Thangavelu, and R. Kalivaradhan. An intelligent driver assistance system (i-das) for vehicle safety modelling using ontology approach. *International Journal of Ubicomp*, 1:15–29, 2010.

[14] A. Lindgren and F. Chen. State of the art analysis: An overview of advanced driver assistance systems (adas) and possible human factors issues. In *Human factors and economics aspects on safety*, pages 38–50, 2006.

[15] Microsoft Corporation. The Windows contacts schema. Web page, 2009. http://msdn.microsoft.com/en-us/library/ms735869.aspx.

[16] R. Miller and Q. Huang. An adaptive peer-to-peer collision warning system. In *in IEEE Vehicular Technology Conference (VTC*, pages 317–321, 2002.

[17] F. Nasoz, C. L. Lisetti, and A. V. Vasilakos. Affectively intelligent and adaptive car interfaces. *Inf. Sci.*, 180:3817–3836, October 2010.

[18] F. Reolon. Kontextsensitive, P2P-basierte Erkennung und Darstellung von Gefahren in einer Motorradumgebung. Diploma thesis, Computer Science, Saarland University, 2007.

[19] R. C. Schank and K. M. Colby, editors. *Computer Models of Thought and Language*. W. H. Freeman & Co., New York, NY, USA, 1973.

[20] W. V. Siricharoen. Ontologies and object models in object oriented software engineering. In *IMECS*, pages 856–861, 2006.

[21] J. Sun, Z. hui Wu, and G. Pan. Context-aware smart car: from model to prototype. *Journal of Zhejiang University SCIENCE A*, 10(7):1049–1059, 2009.