Chapter 9
Digital Maps for Driving Assistance Systems and Autonomous Driving

Alexandre Armand, Javier Ibanez-Guzman, and Clément Zinoune

9.1 Introduction

The use of digital maps in the form of on board vehicle navigation systems or in mobile telephones has been widely deployed for in past years. Their use has become part of current driving tasks, particularly for drivers unfamiliar with the roads leading to their destination. Digital maps store information on the road and its attributes; further currently, vehicle connectivity means that geolocalized information can be cascaded from multiple sources to drivers enhancing the information in them. Digital maps allow drivers to anticipate what will encounter as they drive, to form better mental models with regard to their expectations and ultimate shape their intentions. For example, it is possible to anticipate the arrival to a complex intersection or the crossing of a sharp bend or a sudden change in vehicle speed limits.

After a slow start like with the adoption of radars for adaptive cruise control (ACC) systems, the deployment of exteroceptive sensors on board passenger vehicles has rapidly expanded over the past years. Vision-based ADAS systems form part of the current offer in particular for the detection of lane markings, vehicles and pedestrians. In Europe, the independent assessment of the safety level available in passenger cars, Euro NCAP, has encouraged safety improvements in new car designs and thus propelled the use of sensor-based safety systems on board of European vehicles as proactive safety systems. Most of the current offer relies on the use of video cameras, whose outputs are processed by purpose built processors achieving results that were only found in research laboratories years back. The output of perception systems are classified objects with their position and time orientation is estimated with respect to the vehicle navigation frame. Whilst this
is a major progress, due to the complexities found in using machine vision systems in real time, challenges remain.

Perception systems are far from perfect. The presence of false positives and false negatives leading to hazardous situations remain. This is being addressed through the use of multi-sensor approaches and the fusion of their acquired information to provide a better perceived world as well as increased interest on the use of a new generation of active sensors like laser rangers or the application of convolution network techniques for the classification of video data. Nevertheless, once the perceived world is classified and registered, the machines using this information for decision-making (e.g. stop the vehicle or perform a collision avoidance manoeuvre) need to make sense of this information.

Situation understanding is a major function prior to decision-making and the lateral/longitudinal motion control of the vehicle. It is necessary to understand the spatio-temporal relationship between the perceived objects with respect to the road network and the subject vehicle. However, making sense of the perceived object without context is a difficult task for a machine. For example, as the subject vehicle arrives to an intersection, it needs to scan for vehicles likely to arrive perpendicular to its path or those in front that might turn. Without contextual information, the presence of vehicles around the intersection could be confusing to the machine and likely lead to hazardous situations. By projecting the perceived objects on top of a digital map, it can be inferred, e.g. how far a vehicle is from an intersection, or who has priority at such an intersection by reading the road attributes associated with it. Similarly, the presence of a pedestrian next to a road intersection or next to a pedestrian crossing would lead to an interpretation that the pedestrian is likely to cross in front of the vehicle, whilst without such information, the machine will only know that there is a pedestrian close to its path.

The perceived objects are projected into segments of the stored maps in order to facilitate the understanding of spatio-temporal relationships between these objects, road infrastructure and the subject vehicle. This information is then used to provide a situation understanding and subsequently to infer decisions. Therefore, a local world model is first constructed by mapping the vehicle position on the digital navigation map. This results in the extraction of a segment where the perceived objects can be projected on. This provides an instantaneous representation of the world.

The first part of this chapter introduces a methodology to infer situation understanding and thus to identify the likely interactions between perceived road users and road features in order to estimate their impact on vehicle navigation. Therefore, a framework for the structured representation of the data in terms of a purpose built ontology is proposed. This is then used to assess risk as applied to road intersections, information that is used to infer decisions on the action to take at a road intersection. The principle is shown in Fig. 9.1. A major feature of this approach is that the application is inferred from existing production sensors and that the output can be either used to provide anticipatory information to drivers in the case of ADAS systems or to the decision-making mechanisms in the case of highly automated vehicles. The chapter includes results of the application of the approach in an experimental vehicle evolving in public roads.
The digital map is the underlying component in this process. It provides the stored information that is exploited for contextualization and to infer situation understanding. The source for these maps originates from map suppliers, the most known being HERE (previously known as Navteq) and TOM TOM. Digital maps have been built over the years from different sources including the use of purpose built surveying cars that will collect road information from laser rangers, video cameras and vehicle position estimates. That is, digital maps can be regarded as a patchwork of maps originating from various sources. Their precision and resolution varies and thus not only road attribute errors but also geometric errors exist. This can represent several metres in relative terms between the links and nodes but larger errors in absolute terms, that is with respect to the world reference frame like the WGS84. Another error could reside on the road structure itself, that is maps are not up-to-date as the road structure would have changed, for example, the incorporation of a roundabout replacing a road intersection. If the maps are wrong, the approach proposed in the first part of this chapter will not be applicable, the inference of situations will be made based on the wrong assumptions which shall lead to errors, and eventually hazardous situations. The second part of this chapter proposes a framework that enables the detection of faults in the geometry of navigation maps through the use of standard vehicle proprioceptive sensors plus GNSS receivers. A particular feature of the described approach is that it can solve the ambiguity that could arise due to the presence of errors in the localization systems. That is when an spacial error occurs, this can be either from errors in the map geometry but it can also originate due to errors in the position estimates. Whilst in the first run, the framework can indicate that an error exists, its origins on whether this originates at the map or the position estimate is not know, however, once the vehicle crosses again the same area, the framework can discern that the error is a map geometric error or a position estimate. This approach can be possible as most driving journeys are repetitive, and thus vehicles will drive over similar paths often as when commuting from home and the work place in a daily manner. The different algorithms applied to implement the fault detection are included as well as results of trials performed using data collected from public roads.
9.2 Ontology-Based Situation Understanding

9.2.1 Ontologies

Ontologies are an essential concept for semantic-based approaches and are therefore introduced first.

The term ontology was first used by philosophers to designate the study of being of existence. It has been adopted by researchers in artificial intelligence from the beginning to design computational models for automated reasoning [11]. The most famous definition describes an ontology as ‘an explicit specification of conceptualization’ [9]. The conceptualization of a domain is the manner how a domain is perceived and understood, and the specification of this conceptualization is actually a formal description of this conceptualization.

More concretely, an ontology is a description of the concepts and relationships that are relevant to model a domain of interest. It specifies the vocabulary that is necessary to make assertions, and which may be inputs/outputs of knowledge agents (e.g. software, etc.). Moreover, it provides the language for communication between agents [10]. Figure 9.2a illustrates this definition.

![Ontology Diagram](image)

**Fig. 9.2** Ontology. (a) Ontology used by agents (e.g. software) for communication. (b) Structure of an ontology-based on Description Logic
Ontologies are based on Description Logics (DL) which is a formal language for knowledge representation [3]. A DL enables to model Concepts, Roles and Individuals through its two functional parts, the Terminological Box (TBox) and the Assertional Box (ABox). Figure 9.2b illustrates this structure.

The TBox consists of the definition of all the concepts that the ontology aims to describe. An analogy can be done between the TBox and the knowledge that human have. The knowledge that humans acquire along their life is used to understand and to interpret the world. The ontology TBox represents prior knowledge, and the definition of it is performed through the definition of Concepts, Roles and Relations. The following definitions were established after [13]:

- **Concepts** (or classes) are concrete representations of the concepts of the domain that the ontology aims to describe. These concepts can be organized into a superclass–subclass hierarchy, which is generally called **Taxonomy**.
- **Roles** are properties which can be defined and assigned to concepts. Roles can be classified into two groups:
  - **Object Properties** aim to define axioms in the form of Triples. In other words, they are binary relationships between two concepts in the form Concept1—Object Property—Concept2. Characteristics may be attributed to object properties, such as symmetry or transitivity with respect to other object properties.
  - **Data Properties** are used to assign properties to single classes or instances of classes in the form Concept1—Data Property—Property Value.
- **Relations** between concepts are defined with taxonomic relations (hierarchical relations), axioms (classes linked by object properties) and rules. The definition of rules can be done using basic description logic axioms which only enables the definition of basic class equivalence. More sophisticated languages enable to define more complex and expressive rules. Among these languages, the Semantic Web Rule Language (SWRL) is one of the most common [14].

The ABox consists of the definition of instances of classes previously defined in the TBox. These instances, commonly called **Individuals**, represent real life data that the ontology aims to interpret. Again, an analogy may be done with humans as the ABox can represent objects that humans observe and understand because of prior knowledge (TBox). Further, in the same way as properties can be attributed to concepts defined in the TBox, Object and Data Properties can be attributed to individuals defined in the ABox.

The main advantage of ontologies is the possibility to reason on knowledge using **Reasoners**. Reasoners are pieces of software able to infer logical consequences from a set of asserted facts or axioms [1]. In other words, they aim to exploit information stored in the TBox in order to infer new information and knowledge which is not specifically expressed about **Individuals** defined in the ABox.

The reasoning methodology refers to the algorithm that is used for the reasoning task (e.g. tableau-based or hypertableau algorithms [23, 35]). The methodology has a significant influence on computational efforts required for reasoning.
of a reasoner also includes the soundness and completeness which is the capability to correctly perform all inferences which should be done in theory.

### 9.2.2 Situation Understanding

Three terms are generally used when talking about situation understanding: *Scene*, *Situation* and *Scenario*. The meaning of each of these terms in the context of intelligent vehicles is rather vague and definitions are often contradicting. Geyer et al. formally defined the meaning of these terms in the context of assisted and automated driving guidance [7] as shown in Fig. 9.3. The definitions are rather abstract and have been interpreted with respect to the problems addressed in this article:

- A **scene** is a snapshot of a collection of cohabiting road entities, including the subject vehicle and the surrounding static and dynamic entities. Each entity is defined by its type and state. A *scene* can therefore be represented by all data returned by the data sources.

- A **situation** is the *scene* as it has to be understood by a particular dynamic entity of this scene (i.e. the subject vehicle). This consists in understanding how the interactions between all entities of the Scene are propagated to this entity and constrain it in its navigation. Therefore, Situation understanding consists in giving sense to a Scene.

- A **scenario** is understood as a sequence of Scenes which is the consequence of the Situation of all interacting road entities present in a primary Scene. In the context of this thesis, predicting a Scenario consists in predicting the future state of the participating entities in order to estimate the risks lead by the Situation.

![Fig. 9.3 Relation between scene, situation and scenario](image)
9.2.2.1 Methods for Situation Understanding

Situation understanding consists of reasoning on the scene representation, i.e. adding meaning to the scene information. From the subject vehicles point of view it means to infer how it is constrained by the surrounding environment. Two main categories for situation understanding were identified: probabilistic approaches and semantic-based approaches.

Probabilistic Approach

Road situations are often highly complex due to the variability of situations, especially in urban environments. Vehicles are able to perceive plenty of surrounding entities, however, most of these entities are not relevant. Platho et al. proposed to decompose situations into ‘parts of situations’ or ‘configurations’ [25]. This simplifies the scene by considering only relevant entities. The notion of relationship between road entities is introduced by, for example, considering that the behaviour of a perceived vehicle can be affected by a red traffic light or by another vehicle which has stopped. The recognition of configurations is performed through a Bayesian Network. The approach was tested in simulated environments, and was used to predict the velocity profiles of other road users in intersection situations [26]. However, the approach only allows to consider direct relationships between entities but no chain reactions (e.g. if the subject vehicle follows a vehicle that approaches to a pedestrian, the interaction between the lead vehicle and the pedestrian is not taken into account).

Probabilities allow to take into account uncertainties on perception data, but so far they can only be used to model basic situations based on scenes represented with conventional methods. The amount of possible situations which may occur makes it difficult to define a generic probabilistic model which would be well suited for all situations and which would be capable of considering interactions between road entities. This may explain why literature does not propose other probabilistic frameworks for situation understanding than the one presented above.

Semantic-Based Approaches

Semantic-based approaches do not only use semantics for description, but also apply reasoning. Most of these approaches either use First-order Logics [36] or Description Logics [3] to describe concepts in ontologies. First, semantic-based approaches were used to model and understand the road network from the point of view of the subject vehicle. Later, they were also used to model and understand the whole interaction between road network and dynamic entities.

One of the first works exploiting DL for situation understanding was done by Hummel et al. [17]. The proposed ontology introduces the concepts of road networks (roads, lanes, dividers, road markings and junctions) and is used as a
complement of vision sensors and digital maps to retrieve relevant information about intersections. For example, if the precision of the localization sensors is not able to determine the current lane the vehicle is navigating on, this information can be inferred by the ontology from map and camera data. Whilst this formalism does not take into account cohabiting road entities (vehicles, pedestrians, etc.), it shows that ontologies can be used to reason, at least partially, on road situations.

The representation of road intersection networks through ontologies was introduced by Regele [32]. It was used to solve the traffic coordination problem of autonomous vehicles, i.e. to handle conflicts between vehicles reaching the same intersection or cohabiting in the same area. This work inspired Hülsen et al. who proposed a generic description of road intersections for situation understanding [16]. This ontology enables to infer conflicts and thus potential risk situations for vehicles reaching the same intersection. Figure 9.4 illustrates a sample representation of a relationships. The framework was tested on several intersections and its efficiency was proved even for very complex intersections. A real time implementation of the framework in simulated environments was successfully performed [15]. In these ontologies, vehicles and other road entities are not formally represented.

Vacek et al. [37] introduced an ontology-based representation of other road entities. In that way, semantic information about road entities (i.e. types, etc.) were defined in an ontology which is used within a case-based reasoning framework to perform scene understanding. The tenet was to recover similar or resembling situations in order to infer the behaviour which seems to be the most appropriate to negotiate the situation. Whilst the ontology allows to represent different types of

---

Fig. 9.4 Example of semantic representation of an intersection. After [16]
First-order probabilistic languages (FOPL) was used by Schamm et al. [34] to perform situation assessment. A FOPL knowledge base was used to model driving situations and interactions. This a priori knowledge is thus used with sensor information to automatically create a probabilistic network and then to infer the situation in a structured manner. Since all entities are conceptually of the same type, interactions between entities of different types are not considered. Moreover, first-order logic suffers from poor expressivity, which therefore prevents from editing complex rules in the knowledge base. Further, it seems difficult to extend such a system for more complex situations than those presented in [34].

Zhao et al. built a knowledge base which contains information about maps and traffic regulation. It is used within safety ADAS to take decision at road intersections in case of over speed [39]. Three ontologies were defined for this purpose. The first one aims to describe information which may be stored in a digital map, the second aims to describe control strategies and the last one aims to describe vehicles. Interactions between road entities are considered only between vehicles reaching the same intersection, for the generation of collision warning.

An ontology that models traffic scenes in order to establish the state space of the subject vehicle with respect to other vehicles and the road network was proposed by Kohlhaas et al. [19]. Two categories of objects are considered, namely the environment objects (related to the road network) and the dynamic objects (related to vehicles). The interactions between the vehicles and the road network as well as the lateral and longitudinal interactions between vehicles are formally stated. Further, the ontology contains information about traffic rules through defined conditions.

Finally, Pollard et al. proposed an ontology that represents features of the road network, environmental conditions, sensors states, subject vehicle state and presence of moving obstacles [27]. This ontology enables vehicles to perform self-assessment on their automated driving capabilities, with the aim to decide on the most appropriate automation level (from fully manual to fully automated).

Semantic-based approaches enable to model road scenes and interactions between entities as well as to reason in a straightforward manner on situations. Their main limitation is their inability to take data uncertainties into account. FOPL enables to fill this gap, however, it does not enable to model complex situations because of the low expressivity of the language.

### 9.2.3 Framework for Ontology-Based Situation Understanding

The developed ontology is part of an overall framework for situation understanding shown in Fig. 9.5.
The ontology presented here is not exhaustive and must be considered as a draft that is used to confirm the coherence of the approach. Further extensions and optimizations are required for a real-world validation.

### 9.2.3.1 Observations

The first prerequisite for the framework is information about surrounding entities. This step is represented by the Observations box in Fig. 9.5. Two types of data sources are considered for the awareness of the environment. Modern vehicle sensors such as smart cameras, radars or lasers allow for the real-time perception of moving entities. Most of them are able to perform classification on the perceived entities and provide an estimation of their state with respect to the subject vehicle on which they are embedded. Further, digital maps can store and provide contextual information about the road network features. For instance, this a priori information can contain information about the coming road intersections, about coming pedestrian crossing, etc. Figure 9.6 shows a sample situation as it may be perceived by a vehicle.

### 9.2.3.2 World Model Principles

Information is provided in a piecemeal manner from different mostly independent data sources. It is therefore necessary to organize this data as a list of surrounding entities. This structured and organized list is called the World Model. Figure 9.7 provides a sample world model for the situation shown in Fig. 9.6.

The creation of this world model may require some preliminary processing on perception data. One entity may be perceived by several sensors at the same time or sensors may not be synchronized. For these purposes, data and sensor fusion techniques have to be employed [20]. These problems are complex to solve and today they remain a meaningful challenge for the data fusion community. Here,
perception is considered as a black box performing sensor and data fusion on the data returned by a set of perception sensors.

9.2.3.3 Situation Understanding

Situation understanding is based on the use of an ontology (see Fig. 9.5) consisting of two fundamental parts, the TBox and the ABox.

The TBox consists of a conceptual description of the entities and contextual objects which can be met by a vehicle in a navigable space. In other words, it enables to define the types of entities which can be met and the relationships and interactions which are likely to exist between them. An analogy can be done with the knowledge that drivers acquire when they learn driving at driving school, which is fundamental
### Type Information

| #  | Type                          | Information                                           |
|----|-------------------------------|-------------------------------------------------------|
| 1  | Car                           | Subject Vehicle                                       |
| 2  | Car                           | Coming from the left Turning right                    |
| 3  | Motor Bike                    | Same lane x meters ahead                              |
| 4  | Truck                         | Coming from the right x meters before intersection    |
| 5  | Pedestrian Crossing           | x meters ahead                                        |
| 6  | Stop Intersection             | x meters ahead                                        |
| 7  | Pedestrian                    | x meters ahead On left pavement                       |
| 8  | Pedestrian                    | x meters ahead On the road                            |
| 9  | Pedestrian                    | x meters ahead On right pavement                      |

Fig. 9.7 World presented in Fig. 9.6 in the form of World Model table

to makes them able to understand situations. The TBox is the permanent part of the ontology and was developed with respect to the Description Logic specifications. The focus of this ontology is only on situations which can be represented in one-dimensional space (i.e. road entities on the same navigation lane).

The ABox can be considered as the conversion of the World Model into the ontology language. For each entity in the World Model an instance of the corresponding concept is created. The ABox is the changing part of the ontology and is updated at each update of the World Model. After each update of the ABox, reasoning can be performed on the whole ontology in order to give more sense to the data. More precisely, it means to take into consideration the interactions which are likely to exist between the entities and also chain reactions which may happen as a consequence of these interactions. The purpose is to infer a high level interpretation of the perceived situation in order to select the risk assessment algorithms which suit the situation best.

**TBox**

Figure 9.8 shows the taxonomy which defines the ontology. It is described in the following:

- The *Context entity* aims to list and classify the road entities which may be met in a driving space. Road entities were classified into two sub-concepts, *Mobile Entity* and *Static Entity*. Information about a mobile entity (i.e. pedestrians and vehicles)
cannot be a-priori known. This information has to be obtained in real time from perception sensors. Static entities are assumed to be part of the road network. Their presence is perfectly predictable and can be stored in digital maps. The presented ontology represents two types of static entities: *Road Infrastructures* which effect the behaviour of vehicles such as speed bumpers and pedestrian crossings and *Road Intersections* (classified into three categories: “Stop”, “Right of Way” and “GiveWay Intersections”).

The *Context Parameter* aims to define spatio-temporal thresholds which allow to decide whether interactions between two entities are likely to exist. To illustrate the *IsFollowing Parameter*, lets imagine two vehicles (the leader and the follower) navigating at the same speed, on the same road, and in the same direction. If the two vehicles are separated by 90 m, the interaction between them depends on their speed. If they are moving at 30 km/h, the leader is 6 s ahead of the follower, so it may be considered that there is no interaction between them. However, if they are moving at 90 km/h, the leader is only 2 s ahead. It can therefore be considered that interaction between the two vehicles is established. The *IsFollowing Parameter* allows to set the threshold in the form of time duration which enables to consider if a vehicle is following another one. Following the same logic, the *IsClose Parameter* and the *IsToReach*
Parameter are also defined. Numerical values are given to these concepts through Data Properties.

The Inputs for ADAS is presented in the red shaded area in Fig. 9.8. It aims to store concepts which describe the situation of vehicles. Further, these concepts are guidelines for embedded risk assessment systems as they state which entities and which associations of entities are pertinent to be monitored to ensure safety. The purpose is to infer class equivalences on the subject vehicle in order to choose the most suitable risk assessment algorithm for the current situation.

- **Object properties** aim to define the relationships and interactions which may happen between two concepts of Context Entities. The state of a mobile entity with respect to another one can be described through the goesTowards, isCloseTo, isToReach and isFollowing properties. Further, near future behaviours are defined through the isToReach, willDecelerate and willReach properties. Finally, expected behaviours are defined through the hasToStop and hasToDecelerate properties. These object properties will be used within inferred triples such as Car - goesTowards - Stop Intersection, or Pedestrian - isCloseTo -Pedestrian Crossing.

- **Data Properties** aim to assign properties to individuals which will be defined in the ontology ABox.

  All individuals which will be defined in the ontology ABox have to be defined with their position in the scene. For this purpose, a reference frame had to be chosen. As most observations are performed with respect to the subject vehicle, it was chosen as the reference frame. Further, since the world is represented in one dimension in the ontology, the positions of entities with respect to the subject vehicle are defined as curvilinear abscissas along the road (in the same manner as the position of static entities are defined in the Electronic Horizon). In that way, the property distanceToSubjectVehicle was created, which expects arguments as numerical values.

  Further, some entities such as pedestrians can be either on the road or on the pavement. For a vehicle this information is important in order to decide if the entity has to be considered. Therefore, the boolean parameter isOnRoad enables to define in the ontology whether a pedestrian is on the road.

  Finally, the Context Parameter concepts require to be set. For this purpose the numerical data parameter hasValue was created.

- **Relations** aim to provide a priori knowledge about road entity concepts and their potential interactions and to extract the most relevant features of the situation. Relations consist of axioms aiming to affect object properties to the individuals which are stored in the ontology ABox.

  Relations were created in two steps. In the first step, axioms which enable to infer the likely interactions between the road entities stored in the ABox are defined. Most of this axioms require an expressiveness which cannot be provided by a Description Logic language. Therefore SWRL rules have been chosen for this purpose. The second step adds additional axioms to exploit the interactions which were inferred during the first step and thus to extract for all vehicles the most relevant features
Table 9.1 Example of 3 SWRL rules edited in the ontology

| # | SWRL rules                                                                 | Meaning                                                                                                                                 |
|---|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 1 | `vehicle(?v1) ∧ vehicle(?v2) ∧ distanceToSubjectVehicle(?v1,?d1) ∧ distanceToSubjectVehicle(?v2,?d2) ∧ subtract(?sub,?d2,?d1) ∧ isFollowingParameter(?fParam) ∧ hasValue(?f,?fParam) ∧ lessThan(?sub,?f) → isFollowing(?v2,?v1)` | The position \(d1\) and \(d2\) of the vehicles \(v1\) and \(v2\) are known thanks to the `distanceToSubjectVehicle` parameter. By performing a subtraction (line 4), it is possible to determine the distance \(sub\) between both vehicles. By comparing this distance with the threshold of the `isFollowingParameter` (line 7), it is determined whether one vehicle is following the other one (line 8). |
| 2 | `vehicle(?v1) ∧ StopIntersection(?stop1) ∧ willReach(?v1,?stop1) → willStop(?v1,?stop1)` | The vehicle \(v1\) will reach the stop intersection \(stop1\). This condition means that \(v1\) will probably stop at \(stop1\) (line 4). |
| 3 | `vehicle(?v1) ∧ StopIntersection(?stop1) ∧ isToReach(?v1,?stop1) → hasToStop(?v1,?stop1)` | The vehicle \(v1\) is about to reach the stop intersection \(stop1\). This condition means that \(v1\) has to stop at \(stop1\) (line 4). |

of the situation. For this purpose, it was possible to use the DL language as the corresponding axioms are simple. Note that SWRL could have been used, however, reasoning on SWRL rules is more expensive than reasoning on DL axioms.

For the first part, 14 SWRL rules were defined. Table 9.1 presents 3 of these rules. These rules aim to make it possible to infer spatio-temporal relationships between entities, near future behaviours of mobile entities and expectation about mobile entities manoeuvres. Rule 1 in Table 9.1 is one of five rules dealing with spatio-temporal relationships. Rule 2 is one of three rules dealing with near future behaviours of the mobile entities. Finally, rule 3 is one of six rules dealing with expected manoeuvres of the mobile entities. Some of these rules were defined to take into consideration possible chain reactions (e.g. vehicle that is following another vehicle that has to stop has also to stop in order to avoid a collision).

For the second part, one basic DL axiom was created for each `Output For ADAS` concept. In total, six axioms were edited for the ontology, two of them are presented in Table 9.2 in order to help the reader understand the principles. Axiom 1 aims to define that if a single vehicle is expected to stop at a stop intersection, it is pertinent to run an ADAS that makes sure that the driver is aware of the stop intersection. Further, axiom 2 aims to define that if a vehicle following another vehicle expected to stop at a stop intersection, it is pertinent to run an ADAS that makes sure that the driver is aware that the lead vehicle will stop soon.
Table 9.2 Example of 2 Description Logic Axioms edited in the ontology.

| #  | DL axioms                                                                 | Meaning                                                                                           |
|----|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1  | StopIntersection \( \equiv \) Vehicle \( \sqcap \) \( \exists \) hasToStop \cdot StopIntersection | If an instance of concept Vehicle is linked to an instance of concept StopIntersection through the object property hasToStop, then the instance of concept Vehicle is also an instance of the StopIntersectionAhead concept. |
| 2  | StopIntersectionBefore1Leader \( \equiv \) Vehicle \( \sqcap \) isFollowing \cdot StopIntersectionAhead | If an instance of concept Vehicle is linked to an instance of concept StopIntersectionAhead through the object property isFollowing, then the instance of concept Vehicle is also an instance of the StopIntersectionBefore1Leader concept. |

The ABox

The ontology ABox contains two types of individuals. There are those which are mandatory and created independently from the World Model, and those which are created according to the World Model.

- Mandatory Individuals
  
  Even if the World Model does not include any information about surrounding road entities, the ontology ABox requires four individuals to be defined. These individuals enable the ontology to work properly and are defined as follows.
  
  The World Model will always contain information about the subject vehicle that perceives its surrounding environment. Therefore, the ontology ABox has to store one instance of the Vehicle concept, representing the subject vehicle. This entity is taken as the origin of the frame, so the distanceToSubjectVehicle data property is affected to this individual and is set at 0.
  
  The three other individuals refer to the three concepts included in the Context Parameter major concept. These individual aim to activate the context parameters in the ontology and thus to assign a value to the three of them. In that way, one instance of the isCloseParameter concept has to be created. This individual is given the hasValue data property which sets the maximum distance between a pedestrian and a static entity to consider them close enough to interact. Further, one instance of the isFollowingParameter concept has to be created with the hasValue data property. The value of this property sets the distance between two vehicles from which it is considered that the following vehicle is no longer following the leader. This value depends on the vehicle speeds in stabilized conditions. Finally, one instance of the isToReachParameter concept has to be created, again with the hasValue data property. This parameter sets the distance of a vehicle to a static entity from which it is considered that the vehicle is about to reach the static entity. This parameter also depends on the vehicle speed in stabilized conditions.
• World Model Dependent Individuals
  These individuals can be considered as the conversion of the World Model contents into the ontology language. Thus, each entity stored in the World Model has its equivalent in the ontology ABox. For each entity, one instance of the corresponding concept is created and is affected the distanceToSubjectVehicle data property. The value of this property is the position of the concerned entity, with respect to the subject vehicle. Note that uncertainties on the position of entities are not considered by the ontology. Finally, the isOnRoad data property has to be attributed to all instances of the Pedestrian concept to declare whether the corresponding pedestrians are on the road or on the pavement.

9.2.4 Implementation and Experimental Evaluation

The ontology’s ability to infer pertinent information about a road situation has been tested using test data and real data. It was edited in the Protégé software (TBox and ABox), version 4.3, developed by the Stanford Center for Biomedical Informatics Research [29, 30]. This ontology editor enables the edition of SWRL rules.

9.2.4.1 Case Study Using Manual Data

This first case study was performed with an ontology whose ABox was filled manually, Fig. 9.9a describes the situation that was manually edited in the ontology. This situation consists of three vehicles (called Subject Vehicle, Vehicle 2 and Vehicle 3) going towards a stop intersection (called Stop 1). Vehicle 3 is the closest to the intersection and just passed a pedestrian crossing (called Pedestrian Crossing 1). Vehicle 2 goes towards Pedestrian Crossing 1, and Subject Vehicle follows Vehicle 2. Finally, a pedestrian (called Pedestrian 1) is walking next to Pedestrian Crossing 1.

As mentioned before, the ABox contains four mandatory individuals, one for the subject vehicle and three others for context parameters. In this case study, the highest allowed speed is 50 km/h, therefore the context parameters are set according to this speed. In that way, it was set that a vehicle is following another one if the following time is lower than 3 s. Therefore, an individual of the isFollowingParameter concept was created with the hasValue data property set to 42 m (distance travelled in 3 s at 50 km/h). Further, it was set that a mobile entity is about to reach a static entity if it is reaching it within 5 s at constant speed. Therefore an individual of the isToReach concept was created with the hasValue data property set to 70 m. Finally, an instance of the isCloseParameter concept was created with the hasValue property set to 3 m.
Fig. 9.9 Case study. (a) On the left, an illustrative picture of the case study (scale is not respected). In the boxes on the right, the World Model Dependant Individuals stored in the ABox. (b) Object properties and concept equivalence assertions after reasoning.
Results

Reasoning was performed through the Protégé software and the Pellet reasoner as it is compatible with SWRL and offers good performances [6]. Figure 9.9b shows the object properties and concept equivalence assertions performed by Pellet for the chosen case study. Some of these inferences are detailed below.

Pedestrian 1 is inferred to be close to Pedestrian Crossing 1. The reasoner computes the distance between these two entities according to the distanceToSubjectVehicle data parameter set on the two corresponding individuals. This distance is of 1 m and satisfies the condition (that depends on the isClose context parameter) that was set in the ontology to claim that a pedestrian is close to a pedestrian crossing. It implies that the pedestrian is likely to have the intention to cross the road, therefore it means that a vehicle that would approach to the pedestrian would have to take care of the pedestrian. No concept equivalence is asserted on Pedestrian 1 because there is no axiom for concept equivalence defined in the TBox for the Pedestrian concept.

Three object property assertions are inferred for Vehicle 3. These assertions concern interactions between this vehicle and the stop intersection Stop 1. Thanks to the position of these two entities, is was inferred that Vehicle 3 passed all the static entities except Stop 1. Therefore, it is inferred that Vehicle 3 goes towards Stop 1. Further, the distance between these two entities is low enough to consider that Vehicle 3 is about to reach Stop 1. Moreover, since it was defined in the ontology that all vehicles about to reach a stop intersection have to stop at the intersection, it is inferred that Vehicle 3 has to stop at Stop 1. Finally, it is inferred that Vehicle 3 is an instance of the Stop Intersection Ahead concept. This is performed using the first DL axiom presented in Table 9.2.

Ten object property assertions are inferred for Vehicle 2. The ontology infers that this vehicle passed neither Pedestrian Crossing 1 and Stop 1, therefore it is inferred that it goes towards these two static entities. Moreover, Vehicle 2 is close enough to Pedestrian Crossing 1 and Stop 1 to say that it is about to reach them. Since all vehicles have to stop at stop intersections, it is inferred that Vehicle 2 has to stop at Stop 1. This assertion implies Vehicle 2 to be an instance of concept Stop Intersection Ahead. Moreover, all vehicle have to decelerate before reaching a pedestrian crossing, therefore Vehicle 2 has to decelerate for Pedestrian Crossing 1. Further, as it was inferred that Pedestrian 1 is close to Pedestrian Crossing 1, and since Vehicle 2 is about to reach Pedestrian 1, it is inferred that it has to decelerate for the pedestrian. This assertion implies Vehicle 2 to be an instance of concept Pedestrian Ahead. Finally, Vehicle 2 is close enough to Vehicle 3 to claim that it is following this latter. However, it was inferred that Vehicle 3 has to stop at Stop 1, and since Vehicle 2 is following Vehicle 3, Vehicle 2 has to stop behind Vehicle 3. This chain reaction implies Vehicle 2 to be an instance of concept Stop Intersection before 1 leader.

Eleven object property assertions are inferred for Subject Vehicle. Like Vehicle 2, Subject Vehicle passed neither Pedestrian Crossing 1 and Stop 1. It is therefore inferred that it goes towards these two entities. Moreover, Subject Vehicle is too
far from Stop 1 to consider that it is about to reach it. However it is inferred that Subject Vehicle will reach Stop 1, and therefore that it will stop at Stop 1. Further, as it is close enough to Pedestrian Crossing 1, Subject Vehicle is about to reach it, and thus has to decelerate. In addition, as it is for Vehicle 2, the chain reaction with Pedestrian 1 and Pedestrian Crossing 1 implies that Subject Vehicle has to decelerate for Pedestrian 1. This implies Subject Vehicle to be an instance of concept Pedestrian Ahead. Further, Subject Vehicle is close enough to Vehicle 2 to claim that it is following it. This implies several chain reactions with the other context entities. First, Subject Vehicle is following Vehicle 2 that is an instance of concept Stop Intersection ahead. This implies Subject Vehicle to be an instance of concept Stop Intersection before 1 leader. In addition, Vehicle 2 is also an instance of concept Stop Intersection before 1 leader, therefore is also implies that Subject Vehicle is an instance of concept Stop Intersection before several leaders. Finally, Vehicle 2 is an instance of concept Pedestrian Ahead, it therefore implies that Subject Vehicle is an instance of concept Pedestrian before 1 leader.

These results show that the proposed ontology enables to perform coherent reasoning on global road situations. It shows that interactions between road entities can be understood and considered to anticipate the behaviours of the mobile entities.

9.2.4.2 Case Study Using Recorded Data

The next step is to test the ontology with data recorded from sensors embedded on an experimental vehicle. Figure 9.10 shows a representation of the case study that was chosen for the evaluation of the ontology in real time conditions. It consists of the subject vehicle that is following a lead vehicle. Both vehicles are navigating towards a pedestrian crossing that precedes a stop intersection. Ten meters separate the pedestrian crossing and the intersection. Additionally, a pedestrian is located next to the pedestrian crossing.

Figure 9.11 presents the framework that was used for the real time exploitation of the ontology.

The framework requires several data sources. A priori information about the position of the pedestrian crossing and of the stop intersection were stored in a digital map in the Open Street Map format. In addition to this map, the localization data returned by the GPS receiver was used by the Navigation System in order to generate the Electronic Horizon in real time. Real time information about leading vehicles and pedestrians are provided by the Lidar sensor. Finally, the ontology TBox was stored in a Ontology Web Language (OWL) file [22]. This file format is the reference for the storage of ontologies.

Three pieces of software were necessary to exploit the ontology in real time. The first one is the navigation system that exploits the OSM digital map and that returns the Electronic Horizon at each new measurement of vehicle location. That is, it provides information about the static entities, i.e. the distance of the subject vehicle to the pedestrian crossing and to the stop intersection.
The second piece of software was developed in the C++ programming language for the RTMaps 4 middle-ware. This software allows to get information about the mobile and static entities as they are returned by the data sources. An RTMaps component was developed to feed the World Model structure according to the information about road entities.

The last piece of software was developed in the Java programming language. It enables to exploit the ontology and thus to reason about the World Model. Even if the Java language is not the best language for real time functions, it was chosen to use it as it was the only programming language that proposes accessible libraries for
ontology handling. For this purpose, the OWL API library was used [12]. Moreover, the software was developed as an ontology server, that is it communicates with clients which need to reason about World Models. The communication between the server and RTMaps is performed through the TCP protocol. The World Model structure is exchanged after having been serialized using the Protobuf library [31]. After reception of the World Model structure by the server, ontology individuals are created, completing the core ontology that was preliminary loaded from the OWL file. Reasoning is then performed through the Pellet reasoner and inferences are sent back to the TCP client. The inferences can therefore be exploited by an ADAS, which is, in this Chapter, an HMI that displays the ontology inferences.

Results

Figure 9.12 presents the results of the experimental evaluation of the ontology. Figure 9.12a shows the state of the lead vehicle and the inferred class equivalences over time for the corresponding ontology individual. Further, Fig. 9.12b shows the state of the subject vehicle and the inferred class equivalences over time for the corresponding ontology individual. From the point of view of the subject vehicle, the situation evolves over time through eight main events happening at times $t_1–t_8$.

These events are detailed hereafter.

From the beginning of the experiment, the distance between the subject vehicle and the lead vehicle is lower than the isFollowing threshold (see Fig. 9.12b). The ontology therefore considers that the subject vehicle is following the lead vehicle. It means that as soon as the lead vehicle interacts with at least one other road entity, this interaction is propagated to the subject vehicle.

At time $t_1$, the distance between the lead vehicle and the pedestrian becomes lower than the isToReach threshold (see Fig. 9.12a). Therefore, the ontology considers that there is interaction between the lead vehicle and the pedestrian and that the lead vehicle is about to reach the pedestrian. However, the pedestrian is close to the pedestrian crossing, therefore it is inferred that the lead vehicle individual becomes an instance of the Pedestrian Ahead concept (see Fig. 9.12a). Moreover, since the subject vehicle is following the lead vehicle, the interaction between the lead vehicle and the pedestrian is propagated to it. The subject vehicle individual therefore becomes an instance of the Pedestrian Before 1 Leader concept (see Fig. 9.12b).

At time $t_2$, the distance between the lead vehicle and the stop intersection becomes lower than the isToReach threshold (see Fig. 9.12a). Thus, the ontology considers that the lead vehicle is about to reach the stop intersection. The lead vehicle individual therefore becomes an instance of the Stop Intersection Ahead concept. Further, since the subject vehicle is still following the lead vehicle, the subject vehicle individual becomes an instance of the Stop Intersection Before 1 Leader concept (see Fig. 9.12b).

At time $t_3$, the distance between the subject vehicle and the pedestrian becomes lower than the isToReach threshold (see Fig. 9.12b). Since the pedestrian is still
close to the pedestrian crossing, the subject vehicle starts to interact with him and therefore the subject vehicle individual becomes an instance of the Pedestrian Ahead concept. Note that at this time the lead vehicle did not pass the pedestrian, therefore the subject vehicle individual is still an instance of the Pedestrian Before 1 Leader concept.

At time $t_4$, the distance between the subject vehicle and the stop intersection becomes lower than the isToReach threshold (see Fig. 9.12b). Therefore, the ontology considers that the subject vehicle starts to interact with the intersection and thus the subject vehicle individual becomes an instance of the Stop Intersection Ahead concept. Note that at this time the lead vehicle did not pass the stop intersection, therefore the subject vehicle individual is still an instance of the Stop Intersection Before 1 Leader concept.

At time $t_5$, the lead vehicle passes the pedestrian (see Fig. 9.12b). As a consequence, the lead vehicle individual is no longer an instance of the Pedestrian
Ahead concept. Further, that implies that the subject vehicle is no longer following a vehicle that is about to reach a pedestrian. Therefore, the subject vehicle individual is no longer an instance of the Pedestrian Before 1 Leader concept (see Fig. 9.12b). It means that the subject vehicle no longer indirectly interacts with the pedestrian.

At time $t_6$, the lead vehicle passes the stop intersection. As a consequence, the lead vehicle individual is no longer an instance of the Stop Intersection Ahead concept (see Fig. 9.12b). Therefore, the subject vehicle is no longer an instance of the Stop Intersection Before 1 Leader concept (see Fig. 9.12b).

At time $t_7$, the subject vehicle passes the pedestrian. Therefore, the subject vehicle individual is no longer an instance of the Pedestrian Ahead concept (see Fig. 9.12b). The stop intersection therefore becomes the only pertinent road entity for the subject vehicle.

Finally, at time $t_8$, the subject vehicle reaches the stop intersection. Therefore, the subject vehicle individual is no longer an instance of the Stop Intersection Ahead concept (see Fig. 9.12b). The ontology no longer infers any concept equivalence, therefore there is no more pertinent perceived surrounding entity that have to be monitored by the subject vehicle.

### 9.2.5 Discussion

It was shown that the proposed ontology enables to reason on road environments as they can be perceived by a vehicle. Reasoning on road environments can be performed with respect to the types of the entities which are concerned, while considering the interactions which are likely to happen between entities. The ontology enables to consider chain reactions in a straightforward manner, that is, the interaction between two entities can have consequences on the behaviour of another entity. In comparison, most conventional ADAS would have considered each perceived entity independently from the others and would have monitored the closest entity only.

The proposed ontology cannot be exploited to reason on every road context. Only situations compatible with it can be understood, that is, situations which only meet entities which have been described in the ontology TBox. It means that if the World Model contains an entity that is not formally described in the ontology, the latter will not be able to reason about this entity. If in real life this entity has influence on other entities known by the ontology, a great part of the reasoning will not be representative of reality and thus will not be consistent. Further, for the experiments which were presented, the values of the Context parameters were set in an ad hoc manner. This was because no studies aiming to define conditions for which entities can be considered as interacting were found in the literature. It would therefore be pertinent to carry out studies to fill this gap.

In addition to the quality of the knowledge that is stored in the ontology, the consistency of the inferred information depends on the quality of the information
stored in the World Model. If a pertinent entity of the situation misses in the World Model, reasoning about the situation cannot be coherent. Moreover, if the World Model contains incoherent information about the situation, reasoning will neither be coherent. For instance, let’s consider a situation for which the World Model contains an intersection and a lead vehicle in addition to the subject vehicle. If the distance between the subject vehicle and the intersection is under-estimated, the ontology may understand that the lead vehicle already passed the intersection while it did not. The consequence would therefore be that no interaction between both entities is inferred, therefore the situation would be misunderstood by the subject vehicle.

In the current state of research, one weak point of ontologies is their inability to take uncertainties into account. Again, this means that the precision of the data stored in the World Model is of great importance. This implies that all perception and localization sensors must provide precise and accurate measurements and that navigation maps are precise and up to date. In addition, this lack of uncertainty implies that it has to be assumed that drivers comply with rules, and that it is not considered that rules can be violated. Finally, neither uncertainties on interactions between entities, neither uncertainties on concept equivalence assertions can be estimated. Such uncertainties could be of great interest, especially for the risk assessment systems which may have to exploit the ontology inferences.

Finally, the time necessary to reason on an ontology is significant and has to be considered. For the experimental evaluation presented in Sect. 9.2.4, the average processing time necessary for reasoning was 71 ms on a 4 GB RAM laptop with a dual core 1.9 GHz processor. This processing time depends both on the complexity of the ontology TBox (number of axioms and especially the number and complexity of the SWRL rules) and on the number and types of road entity individuals stored in the ontology ABox. If the ontology has to be extended, an effort would have to be made in order to limit the number of axioms and rules and thus to limit the complexity of the reasoning step. For real time applications, if the processing time is too high in comparison with the frequency at which the World Model returns data, it would be conceivable to reason on the ontology asynchronously with the rest of the system as it was done in [15].

9.3 Map Error Detection

Intelligent and autonomous vehicles applications require the information provided by the navigation function in the form of the Electronic Horizon (EH). Erroneous data in the EH may result in undesirable behaviour of client systems and generate hazardous situations. These can be the consequence of faults that arise at any step of the EH generation. Localization system used in combination with the map may be perturbed and provide a position estimate that contains large errors. This may result in large errors on the estimation of the vehicle position onto the map. A map is a complex entity that represents the environment that is constantly evolving, it then
necessarily contains Faults. The scope of this section is on faults originating in the navigation map.

This section first defines the concepts of fault and error which are essential here. Next the pathology of faults found in navigation maps and their effects on the vehicle applications is described.

### 9.3.1 Definitions

The terms of fault, error, failure and integrity may have different meanings according to the application domain. The following definitions are used in the context of this research and are based on those given in [28]:

- **Fault**: Error generative process. The presence of a fault may not lead to an error.
- **Error**: A discrepancy between a computed, observed or measured value and the true, specified or theoretically correct value.
- **Failure**: Instance in time when a required function exceeds the acceptable limits or is terminated.
- **Integrity**: Reliability of the confidence indicator associated with an information with respect to specifications of the client application.

There is a loss of integrity of EH data if the error is greater than the estimation.

Figure 9.13 shows the geometry of the segment of a road network. The shaded shape represents admissible error associated with it. In this illustration, a fault occurs if the EH data is outside this envelop. Figure 9.14 shows the loss of integrity for the position estimate. The true vehicle position is indeed outside the confidence domain of the error associated with the position estimate.

The accuracy of the vehicle position estimate provided by the navigation function in the EH is closely related to both the localization and the navigation map.
Figure 9.15 highlights this dependency and the consequences on the development of driving assistance functions, from the assisted navigation to autonomous driving [8].

Passenger vehicles are mainly used either for commuting or to travel on known roads. Except for occasional journeys like going on vacations, the vehicle is therefore driven on roads on which it has already been driven. It must be noticed that the dysfunctions due to map faults are mainly systematic. Whenever the vehicle crosses again the erroneous road, the EH provides the same false data to ADAS which, themselves, will have the same uncomfortable behaviour. The quality perceived by the driver decreases significantly due to the frustration of repetitive illogical warnings or reactions of the vehicle.

### 9.3.2 Pathology

The road network is constantly evolving. Map providers consider that 15% of the road network of a mature country change each year. This value increases in developing countries. With the exception of few countries like Japan, road changes are not monitored by a central administration. It is therefore difficult for cartographers to keep the geographic databases up-to-date.

Map creation is a complex process that takes a long time. When loaded in the vehicle, the navigation map data is therefore already several months old and partially outdated. Mapmakers adopt two approaches to pursue fast map updates. The first is to facilitate mapping surveys by deploying a large fleet of vehicles. These are equipped with a specific set of sensors (e.g. lidar, inertial measurement systems, differential GNSS) for rapid mapping of the road network. Priorities are given to main roads and areas where changes have been reported. This strategy allows precise mapping but is costly in material and human resources. The second is to rely on user data. The records of the journeys travelled with the navigation devices are
automatically uploaded to the map provider server. Data mining algorithms are then applied to update the central database. This approach is less expensive and allows to update the road geometry or the mean travel times. However other structural elements such as traffic signs, speed limits still require field surveys.

The typical faults are summarized in a pathology. It provides an overview of the issues involved by each fault and associated scientific challenges. The distinction is made here between fault in the navigation map structure, geometry and attributes. Structural faults are related to the manner in which the elements of the map are connected or are identified in the map. Geometric faults are related to the shape or the geographic placement of these entities. This distinction is made because they require different approaches to be detected and corrected. These are detailed in the following paragraphs.

9.3.2.1 Structural Faults

Road Connectivity

The correct representation of road junction in the navigation map is essential for optimal path planning. The missing connection between two road may cause the path planner choose a suboptimal path and bother the driver. Reciprocally, a connection between two roads in the navigation map that does not exist in reality may result in impossible path. In this case, the driver may be misled by the navigation assistant and cause hazardous driving situations.

Type of Intersection

The significant information at road junctions is the right way of one road with respect to the others. Hazardous situations could occur if this information is missing. In navigation maps, the type of intersection can be associated with junctions in order to describe implicitly the order of priorities. Over the past years, a particular type of intersection that is roundabouts (or traffic circles) is preferred to others types and is built in many places. They indeed reduce fatalities and increase traffic exchange between roads. Their construction means that the network structure is changed [33, 38]. The manner in which the vehicle crosses them is different to a classic crossroad and the reachability to the intersection branches is different. The inclusion of a roundabout must be registered in the navigation map in order to provide convenient guidance information to the driver and adapt the approaching manoeuvre (e.g. speed reduction, lane placement).
9.3.2.2 Geometric Fault

Road Shape and Road Offset

For optimization purposes, the on board navigation maps are compressed and compiled. In this process, some road shape points are removed. The road curvature estimated using the shape points is therefore perverted. In the GIS domain, absolute and relative accuracies are distinguished [5]. The former describes the accuracy of the geographic feature with respect to a global reference coordinate system whilst the later describes accuracy relative to other features. Roads with low relative accuracy result in poor shape definition. The curve warning related ADAS (i.e. Contextual ACC, LKA and PFL) are directly affected by low relative accuracy of the road. In case of low absolute and high relative accuracies (road offset), the map-matching algorithm may not choose the right road candidate, especially in dense road network area. Moreover, Fig. 9.16 shows that the offset on junctions induces malfunctions of intersection warning systems for vehicles running on the other roads [2].

Fig. 9.16 Consequences of road offset on intersection warning systems. The true roads are in black. The red polylines are the roads stored in the navigation map. The vehicle is the yellow triangle

Missing Roads

Road networks change over time and some roads are created or closed. Until the next cartographic survey, every contextual ADAS application cannot operate properly. The case in which the vehicle is driven on a new road is split into two situations. First, if the new road is far from the roads stored in the navigation map, the map-matching function will switch to failure mode and provide the output off road. The contextual ADAS applications could then adopt a suitable strategy and limit the
consequences. Second, if the new road goes along a road stored in the navigation map, the vehicle position is likely to be matched on the wrong road with a high confidence level. The contextual ADAS could operate based on inconsistent data and cause hazardous situations.

9.3.2.3 Attributes Faults

Speed Limits

The speed limit can be displayed to the driver in Over-Speed Prevention (OSP) system or used to set the vehicle cruise speed in Contextual ACC applications. The navigation map stores the speed limit as an attribute for each road. This attribute originates from direct surveys or is inferred based on the road class, the number of lanes and the area (inside or outside built-up area). Smart cameras are nowadays capable of detecting speed signs in real time [18, 24], however, the challenge resides in determining whether the detected sign is applicable to the vehicle. Speed limits indicated by traffic signs may be dedicated to one particular lane (e.g. for a motorway exit), to a class of vehicles (e.g. trucks, vehicles towing caravans, buses) or to special whether conditions. An inappropriate estimation of the speed limit by the vehicle would bother the driver and decrease the perceived vehicle quality.

Driving Directions and Vehicle Restrictions

In navigation maps, roads are assumed to be drivable in both direction unless a dedicated attribute is associated with the road. Similarly, attributes are defined in order to establish some traffic restrictions (e.g. trucks, pedestrian and maximum height). The path planner uses these attributes to exclude wrong-way roads and roads that does not comply with the vehicle type. Faults in these attribute may cause the path planner to choose a suboptimal route or ask the driver to take a forbidden route. This may have severe consequences especially for large goods vehicles that cannot manoeuvre easily.

As a conclusion on this fault pathology, it must be noticed that the variety of the information stored in the navigation map induces a variety in the faults in the map. The methods employed to address the detection and correction of these faults are necessarily diverse in terms of sensors and formalisms. Some of the formalisms that could permit to address this problem require an a priori knowledge on the correctness of the navigation map. However, to the author’s knowledge, no trustworthy study on the navigation map overall reliability has been done. Here, the navigation map is considered to be globally correct but potentially locally erroneous.
9.3.3 Page’s Trend Test

The quality of the road representation within the navigation map in terms of geometry has a direct impact on the performance of intelligent vehicles navigation systems. The knowledge of the road shape in front of the vehicle is used in existing intelligent vehicles to improve sensor tracking (e.g., lane markings for lane keeping functions or leading vehicle for adaptive cruise control applications) and anticipate hazardous situations by adapting the vehicle speed. The navigation map road geometric description is also essential for autonomous driving like for path planning, decision-making, and control functions. To avoid dysfunction of these systems, the quality of the geometric description of the road in the navigation map must be monitored.

In passenger vehicles, no access is permitted to internal variables or data of navigation function which is therefore considered as a black box in the approach presented here. When a failure occurs on the application functions it is difficult to identify their origin, to correct them or reduce their effect. A method to detect error in the map geometry is to compare the vehicle position estimate provided by the navigation function with the positioning information given by the available sensors as illustrated by Fig. 9.17. The challenge here is that the level of performance of vehicle sensors is limited considering measurement applications. The low accuracy and high level of noise introduced by vehicle sensors make probabilistic approach appropriate.

Statistical tests are appropriate to evaluate parameters of a probability law based on a set of outcomes. In our application, we aim at detecting a change of the mean of the probability density function (PDF) of a set of observed data while the standard deviation of this PDF is in the same order of magnitude than the expected mean gap. Page’s trend test works sequentially and is especially efficient for stream data. The problem is therefore formulated as the detection of a change of the mean of a

![Fig. 9.17 Page’s trend test for fault detection in navigation integrity monitoring context](image-url)
Fig. 9.18 The distance between the estimate from sensors $G$ and the estimate from navigation $N$ taken as random variables. $d$ and $e$ are the lateral and longitudinal offsets between the estimates from navigation and from sensor, respectively. $R_0$ is the East–North plane locally tangent to the Earth surface and $R_1$ is the frame aligned with the road segment on which the position is matched by the navigation function.

random variable that represents the distance between the estimates from sensors and from navigation as illustrated by Fig. 9.18.

9.3.3.1 Signal Generation

This section details how the distance signal is generated and described in terms of mean and standard deviation. Let us consider the estimate $N$ from the navigation as the result of a random process based on the true vehicle position $P$ in a frame $R_1$ aligned with the road:

$$N = P + \alpha$$

$$\Sigma_\alpha = \begin{bmatrix} \sigma^2_d & 0 \\ 0 & 0 \end{bmatrix}_{R_1}$$

where $\alpha$ is a noise supposed zero-mean with a diagonal covariance matrix $\Sigma_\alpha$. Indeed, since the roads are represented in the navigation map by zero width polylines, the variance of the navigation map-matched error normal to the road segment is by definition null. However, a map-matched position error along the road segment exists and $\sigma_d$ denotes the longitudinal standard deviation of the navigation estimate.

The estimate of the vehicle position from sensors $G$ can be encoded as a two-dimensional point $G = (x, y)^T$ in the East–North plane $R_0$ locally tangent to Earth
with the covariance matrix $\Sigma_\beta$ of the estimation error $\beta$:

$$G = P + \beta$$  \hfill (9.3)$$

$$\Sigma_\beta = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$  \hfill (9.4)$$

In order to make the distance signal independent of the road direction, an isotropic approach is chosen and it consists in using the outer circle of the ellipsoid. Its radius is $\eta = \max (\eta_i)$, $\eta_i$ being the eigenvalues of $\Sigma_\beta$. So, the covariance matrix expressed in $\mathbb{R}_1$ is $\eta \cdot 1$ (with $1$ being the identity matrix).

In $\mathbb{R}_1$, the difference between the map-matched and estimated positions is given by a vector $L$ which has two independent components.

$$L = \begin{bmatrix} d \\ e \end{bmatrix} = N - G = \alpha - \beta$$  \hfill (9.5)$$

Under the hypothesis of independent errors, the lateral $d$ and longitudinal $e$ signals have the following variances:

$$\sigma_d^2 = \eta$$

$$\sigma_e^2 = \eta + \sigma_a^2$$  \hfill (9.6)$$

The relevant information in terms of the application is the lateral position of the roads in the navigation map. The fault detection is therefore made by detecting mean changes of the signal $d$.

### 9.3.3.2 Formulation of the Test

Page’s test consists in statistically detecting a change in the mean of a random variable [4]. Let us consider $q$ samples $d_i$ of a random variable $D$. The likelihood of two hypotheses $H_0$ and $H_1$ are compared. The first hypothesis states that $D$ has a constant mean $\mu_0$ among the $q$ samples. The second one assumes that, given $0 < r \leq q$, the mean of $D$ is $\mu_0$ for the first $r - 1$ samples and $\mu_1$ for samples $r$ to $q$:

$$H_0 : \quad d_i = \mu_0 + b_i, \quad i = 1, \ldots, q$$

$$H_1 : \begin{cases} d_i = \mu_0 + b_i, & i = 1, \ldots, r - 1 \\ d_i = \mu_1 + b_i, & i = r, \ldots, q \end{cases}$$  \hfill (9.7)$$
where \( b \) is a zero-mean noise of standard deviation \( \sigma \). The generalized likelihood ratio of both hypotheses is given by (9.8).

\[
\Lambda(D) = \frac{\prod_{i=1}^{q} p(d_i, r|H_1)}{\prod_{i=1}^{q} p(d_i|H_0)} \tag{9.8}
\]

Since the likelihood of the alternative hypothesis \( H_1 \) depends on an unknown parameter \( r \), its maximum likelihood estimation is considered.

\[
\Lambda(D) = \sup_r \left( \frac{\prod_{i=1}^{r-1} p(d_i|H_1) \prod_{i=r}^{q} p(d_i|H_1)}{\prod_{i=1}^{q} p(d_i|H_0)} \right) \tag{9.9}
\]

As the likelihood of the null hypothesis \( H_0 \) does not depend on \( r \) and having \( \prod_{i=1}^{r-1} p(d_i|H_1) = \prod_{i=1}^{r-1} p(d_i|H_0) \), the likelihood ratio can be simplified as follows:

\[
\Lambda(D) = \sup_r \left( \prod_{i=r}^{q} \frac{p(d_i|H_1)}{p(d_i|H_0)} \right) \tag{9.10}
\]

Let \( \delta \) denote the mean gap \( (\delta = \mu_1 - \mu_0) \). Under Gaussian assumption, one can get [21]:

\[
\ln(\Lambda(D)) = \frac{\delta}{\sigma^2} \sup_r \left( \sum_{i=r}^{q} \left( d_i - \mu_0 - \frac{\delta}{2} \right) \right) \tag{9.11}
\]

The decision of choosing either \( H_0 \) or \( H_1 \) is made by comparing the likelihood ratio with a threshold \( \lambda_\Lambda \):

\[
\begin{cases} 
H_0 : \ln(\Lambda(D)) < \ln(\lambda_\Lambda) \\
H_1 : \ln(\Lambda(D)) > \ln(\lambda_\Lambda)
\end{cases} \tag{9.12}
\]

For a real time implementation, it is especially convenient to formulate the test sequentially. Let us then define the cumulative sum as in (9.13):

\[
S^q_r(\mu_0, \delta) = \delta \sum_{i=r}^{q} \left( d_i - \mu_0 - \frac{\delta}{2} \right) \tag{9.13}
\]
which can be re-written as:

\[ S_i^q (\mu_0, \delta) = S_i^{q-1} (\mu_0, \delta) + S_i^q (\mu_0, \delta) \]  

(9.14)

The decision rule then becomes

\[
\begin{align*}
H_0 & : S_i^q (\mu_0, \delta) - \inf_r (S_i^{q-1} (\mu_0, \delta)) < \gamma \\
H_1 & : S_i^q (\mu_0, \delta) - \inf_r (S_i^{q-1} (\mu_0, \delta)) > \gamma
\end{align*}
\]  

(9.15)

where \( \gamma = \sigma^2 \ln(\lambda_A) \).

Let \( \delta_m \) be the minimum value of \( \delta \) which must be detected. The test is split into two sub-tests running in parallel, the first aiming at detecting a mean growth and the other a decrease of the mean.

Finally, at the current time \( k \), a mean growth is detected as soon as (9.16) is true:

\[ U_k - m_k > \gamma \]  

(9.16)

where

\[ U_k = U_{k-1} + d_k - \mu_0 - \frac{\delta}{2} \]  

(9.17)

\[ m_k = \min (m_{k-1}, U_k) \]  

(9.18)

Conversely, a mean decrease is detected when:

\[ M_k - T_k < \gamma \]  

(9.19)

where

\[ T_k = T_{k-1} + d_k - \mu_0 + \frac{\delta}{2} \]  

(9.20)

\[ M_k = \max (M_{k-1}, T_k) \]  

(9.21)

As soon as threshold \( \gamma \) is reached, the cumulative sums are reset to zero. The actual mean change happened at the last time \( m \) (resp. \( M \)) has reached its minimum (resp. maximum) before crossing \( \gamma \). Another formulation for the localization of the mean change which is used in this work is to find the last time the decision variable \((U_k - m_k)\) for increase detection or \((M_k - T_k)\) for decrease detection) is null. This test is then very efficient in terms of required computational load. Indeed, at every new time step, it only requires to make additions and comparisons of scalar variables. The mean change cannot be detected at the samples prior to the last one at which
the decision variable is null. Those samples are useless and can be removed from the computer memory.

The choice of $\gamma$ has consequences on the false alarm probability. It is not possible to express it formally because the PDF of hypothesis $H_1$ depends on an unknown parameter $r$. However, it can be set based on the number $h$ of estimated parameters in the PDF and on the number $n_\sigma$ of standard deviations: $\gamma = 2 \cdot h \cdot n_\sigma \cdot \sigma / \delta_m$ [21]. Here the mean is the only estimated parameter of the PDF and $n_\sigma$ is set to 2 as nominal tuning which is a good compromise between false alarm rate and time to detection. Then $\gamma = 4 \cdot \sigma / \delta_m$.

Let us consider the example shown in Fig. 9.19. The signal plotted in the upper graph of the figure is generated using a Gaussian function with a constant standard deviation of 2. The mean of this signal is 0 from sample 1 to sample 40 and from sample 61 to sample 100. The mean is set to 5 for sample indexes 41–60. Page’s trend test is run using this signal as an input and the cumulative sum $U_k$ as well as the decision variable $U_k - m_k$ are plotted in order to illustrate the behaviour of the test. A larger threshold than indicated previously is chosen in order to make the explanation more understandable; here $n_\sigma = 6$ thus $\gamma = 4.8$. When a sample of the signal deviates from zero and get close to 5 (at samples 5, 18, 21, for instance) the cumulative sum increases and thus the decision variable increases. In those situations a mean change of the signal is expected but not confirmed. Since the actual

![Fig. 9.19](image.png)  
**Fig. 9.19** Illustrative example of Page’s test. The signal to consider is plotted on the top graph. The corresponding cumulative sum and decision variable are beyond. The red stars denote the sample indexes for which the test detected a mean change.
mean of the signal is 0 from sample 1 to 40, the cumulative sum keeps decreasing and the decision variable goes back to zero before crossing the threshold. The expectation of a mean change is therefore rejected. At the sample indexes 41–43, the cumulative sum and the decision variable increase like before but the decision variable finally crosses the threshold at index 44 which confirms the mean change. Samples 41–44 are therefore marked with red stars on the graph. The cumulative sum and the decision variable are set to zero which is indicated by discontinuities of the plots. The Page’s test starts again: it detects at the sample 46 that a mean change occurred since the sample 44, and so on. In this example the test requires 3 or 4 samples to detect a mean change and missed the sample 59 and 60 because the decision variable was close to the threshold but did not reach it.

9.3.3.3 Experimental Evaluation of Fault Detection

Analysis Methodology

The critical issues of the fault detection are the reactivity and the ability to detect the fault when it occurs. It was shown in the previous section that the sequential formulation of Page’s trend test resets depending on the value of the samples. It therefore has the particularity to constantly adapt the size of the sliding window. The reactivity of this test is not known beforehand. In order to measure this, the test is evaluated in terms of distance-to-alert, distance-to-recovery and accuracy of map error localization, as shown in Fig. 9.20. The distance-to-alert and the distance-to-recovery are derived from the usual time-to-alert and time-to-recovery and adapted in the context of this work in which the distance travelled by the vehicle is the reference. The distance-to-alert $\delta_{H_0 \rightarrow H_1}$ is the distance travelled by the vehicle before detecting a fault. Reciprocally, $\delta_{H_1 \rightarrow H_0}$ is the distance-to-recovery, that is, the distance after which the test detects the end of a fault. Since the test may detect a fault a few samples in the past, the a posteriori accuracy of the test is measured by $e_{H_1|H_0}$ and $e_{H_0|H_1}$. $e_{H_1|H_0}$ denotes the length of the road that has been identified as faulty while being actually fault-free. This can be named false alarm distance. Reciprocally, $e_{H_0|H_1}$ stands for the length of road that actually contains a fault that has not been detected by the algorithm which can be seen as a missed detection distance.

The performance of Page’s trend test is compared to two other usual methods based on fixed length sliding window. These both aim to discriminate between $H_0$ and $H_1$:

$$
\begin{align*}
H_0 &: d_i = b_i, \ i = 1, \ldots, q \\
H_1 &: d_i = \delta_m + b_i, \ i = 1, \ldots, q
\end{align*}
$$

(9.22)

where $b_i \sim \mathcal{N}(0, \sigma^2)$. 

On the one hand, a simple decision rule based on the empirical mean of the sliding window was implemented:

\[
\frac{1}{q} \sum_{i=1}^{q} d_i \overset{H_1}{\gtrless} \delta_m
\]  

(9.23)

On the other hand, the Neyman Pearson probabilistic decision rule was used for comparison. This is based on the generalized likelihood ratio of the hypotheses. Under Gaussian noise assumption. The choice follows the rule (9.24) [4, 21]:

\[
\sum_{i=1}^{q} d_i \overset{H_1}{\gtrless} \sigma \sqrt{2 \cdot q \cdot \log(\Phi)}
\]  

(9.24)

Where the threshold \(\Phi\) arises from a compromise between desired false alarm (type I error) and missed detection probabilities (type II error) of the decision rule. The false alarm probability has been set to its usual value (0.1\%) for an appropriate comparison with the proposed method.

The size \(q\) of the sliding window must be large enough to be statistically representative and short enough to detect map errors as fast as possible. Moreover,
the samples $D$ must be reinitialized as soon as the vehicle leaves one road for another which happens frequently in urban environment. It has then been set to $q = 20$ which is equivalent to approximately 200 m of travelled distance.

Experimental Evaluation

The tests were run on a set of roads that were recently modified due to the construction of a new motorway in Normandy, France (see Fig. 9.21). This area is representative of typical geometric errors that a map may hold and which may cause severe malfunctions in driving assistance systems. In area 1 and 2, sharp bends were added to the road which was previously straight. This would makes a curve warning system inefficient. In area 3 a new carriage way was added to the old single track road. This induces a constant lateral offset of the new road which would make obsolete intersection warning systems on crossing roads. On the fourth area, the lateral road offset decreases while the vehicle goes. This situation is very useful to highlight the distance to recovery of the tests.

![Global view of the test areas](image)

**Fig. 9.21** Global view of the test areas. The out-of-date map being assessed is shown by yellow lines. A map of the actual road network is at the background in grey. The vehicle trajectory is in blue (starting from the bottom of the figure). The four test zones are circled in red.
Since the experiment is done in a rural environment and also for explanation convenience, it is assumed here that a fault always originates from an error in the navigation map.

Tests have been run on the lateral euclidean distance $d$. The most restrictive constraint comes from intersection warning systems which require a longitudinal precision of 10 m for the placement of an intersection on a road link. Indeed, a lateral offset of a road link induces a longitudinal misplacement of intersection on crossing roads. This value has then been chosen as the mean change to detect $\delta_m = 10$ m. Finally, the cumulative sum threshold is set dynamically to be $\gamma = 4 \cdot \sigma / \delta_m$ which represents a good compromise between false alarm rate and time to alert.

The results summarized in Fig. 9.22 show that Page’s trend test is appropriate for detection and localization of faults. Indeed, faults are detected less than 20 m after the beginning of the fault and well localized. The two other methods provide less suitable false alarm and missed detection rates. This is mainly due to the fact that fixed length sliding windows are used. The fault detection has then undesirable collateral effects on the tail of the window. These methods show bad results when the road error is small with respect to the size of the sliding window.

Let us focus on area 1, to better understand the strength and weakness of the three methods. The upper part of Fig. 9.23 shows the lateral error between the vehicle’s estimated position and its map-matched position against the travelled distance. The lower part shows the sequential outcomes of each methods while the vehicle is driven. It can be seen on this figure that Page’s test is very efficient for detecting the fault since it chooses $H_1$ as soon as the road is actually erroneous. Moreover, in this example, it locates perfectly the fault (from abscissa 520–1520 m). The outcomes of the two other tests are less accurate. Indeed, the faults are detected later and locate it very poorly. This is due to the fact that is not possible to know where the change happened within the sliding window. The whole window is supposed to belong to $H_1$ as soon as the threshold is crossed. Correct road points are then declared faulty.

| Area 1 | Test | $\delta_{H_0 \rightarrow H_1}$ | $\delta_{H_1 \rightarrow H_0}$ | $\theta_{H_0 \rightarrow H_1}$ | $\theta_{H_1 \rightarrow H_0}$ |
|--------|------|------------------------------|------------------------------|-----------------|------------------|
| Page   | 0 m  | 0 m                          | 0 m                          | 0 m             | 0 m              |
| Mean   | 73 m | 150 m                        | 90 m                         | 177 m           |                  |
| N.P.   | 46 m | 255 m                        | 220 m                        | 0 m             |                  |

| Area 2 | Test | $\delta_{H_0 \rightarrow H_1}$ | $\delta_{H_1 \rightarrow H_0}$ | $\theta_{H_0 \rightarrow H_1}$ | $\theta_{H_1 \rightarrow H_0}$ |
|--------|------|------------------------------|------------------------------|-----------------|------------------|
| Page   | 20 m | n.a.                         | 20 m                         | 0 m             | 0 m              |
| Mean   | 80 m | n.a.                         | 35 m                         | 0 m             | 170 m            |
| N.P.   |      | n.a.                         | 0 m                          | 0 m             |                  |

| Area 3 | Test | $\delta_{H_0 \rightarrow H_1}$ | $\delta_{H_1 \rightarrow H_0}$ | $\theta_{H_0 \rightarrow H_1}$ | $\theta_{H_1 \rightarrow H_0}$ |
|--------|------|------------------------------|------------------------------|-----------------|------------------|
| Page   | 0 m  | 0 m                          | 0 m                          | 0 m             | 0 m              |
| Mean   | 50 m | 100 m                        | 150 m                        | 200 m           |                  |
| N.P.   | 0 m  | 300 m                        | 0 m                          | 300 m           |                  |

| Area 4 | Test | $\delta_{H_0 \rightarrow H_1}$ | $\delta_{H_1 \rightarrow H_0}$ | $\theta_{H_0 \rightarrow H_1}$ | $\theta_{H_1 \rightarrow H_0}$ |
|--------|------|------------------------------|------------------------------|-----------------|------------------|
| Page   | 0 m  | 20 m                         | 0 m                          | 20 m            | 0 m              |
| Mean   | 50 m | 0 m                          | 250 m                        | 40 m            | 0 m              |
| N.P.   | 0 m  | 250 m                        | 0 m                          | 210 m           |                  |

\(3\) not applicable
\(4\) not detected

Fig. 9.22 Comparative results with nominal tuning of each test
Fig. 9.23  Sequential outcomes of three trend tests on hypotheses \( H_0 \) (correct road segment) and \( H_1 \) (erroneous road segment). The lateral distance to the map-matched position is denoted on the upper part. The plot colour shows the true state of the road: correct in green, with fault in red while they are not and vice versa. This example illustrates why these methods induce false alarms, missed detections and inaccuracies in fault localization.

### 9.3.4 Discussion

The mathematical formulation of this test and its application to the comparison of vehicle position estimates was detailed. Given this, its ability to detect a discrepancy between the estimate from sensors and the estimate from navigation was evaluated using real vehicle data. Since the metrics usually employed to measure the performance of this test are not relevant in the intelligent vehicle context, the evaluation was done using metrics introduced in this chapter. The test showed convincing performance since it detected quickly (short distance-to-detection and distance-to-recovery) and accurately (short false alarm and missed detection distances) the discrepancies between the estimates.

Page’s trend test allows to detect with robustness when the two estimates of the vehicle position are significantly different and to conclude that a fault affects at least one of them. However, it does not permit to determine which one is affected by the fault if it is assumed that vehicle sensors can also be responsible for the fault. In this
sense, this test performs fault detection but does not perform isolation. In order to solve this ambiguity, an approach consists in taking benefit of the repeated vehicle journeys on the same roads. The repeatability of the map geometrical faults permits to isolate and correct faults after a few number of journeys [40].

9.4 Summary

The chapter examined a fundamental issue in the decision-making process for vehicle navigation under computer control, namely situation understanding of the spatial-temporal relationship between entities sharing the same segment of a road network. A solution is proposed through the use of ontologies that establish this relationship in an ‘ordered’ manner to structure this relationship. During this process, a fundamental source of information and knowledge resides within navigation maps. They provide contextual information that it is used to facilitate the situation understanding process. Whilst much progress has been attained on the deployment of digital navigation maps for vehicle guidance applications, their geometric descriptions is far from perfect, there are modifications to maps structures, there are geometric errors, etc. Thus the second part of this chapter presented a novel approach for the detection of faults in the geometric descriptions of the roads, a feature of this being the use of close to production components that are used in other ADAS functions. The theoretical developments in this chapter have been implemented in passenger vehicles and different trials performed in standard road networks.

The overall results have shown that further work is needed, in particular on the information needed to infer understanding of the situation. That is, how to infer information from the close environment to improve the spatio-temporal relationships amongst the relevant entities. For this purpose an area to explore is semantic road segmentation. It should facilitate classification as well as the definition of the driving or navigation space. Maps remain a challenge, the concepts portrayed in this chapter are being extended to the notion of learning maps by leveraging on the repeated trajectories that often drivers, e.g. daily commuting. The success of these endeavours shall be demonstrated when applied to autonomous driving as the machine has to interpret data, detect errors and interact with its environment in a safe manner as it drives to its destination.

References

1. S. Abburu, A survey on ontology reasoners and comparison. Int. J. Comput. Appl. 57(17), 33–39 (2012)
2. A. Armand, D. Filliat, J. Ibañez-Guzmán, Modelling stop intersection approaches using gaussian processes, in Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems-ITSC (2013)
3. F. Baader, *The Description Logic Handbook: Theory, Implementation, and Applications* (Cambridge University Press, Cambridge, 2003)
4. M. Basseville, I.V. Nikiforov, *Detection of Abrupt Changes: Theory and Application* (Prentice-Hall, Englewood Cliffs, NJ, 1993)
5. D.J. Buckley, *The GIS primer an introduction to geographic information systems*. Technical Report, Innovative, 1997
6. K. Dentler, R. Cornet, A.T. Teije, N. De Keizer, Comparison of reasoners for large ontologies in the OWL 2 EL profile. Semantic Web 2(2), 71–87 (2011)
7. S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier, T. Weißgerber, K. Bengler, R. Bruder et al., Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. IET Intell. Transp. Syst. 8(3), 183–189 (2013)
8. P.-Y. Gilliéron, H. Gontran, B. Merminod, Cartographie routière précise pour les systèmes d’assistance à la conduite, in *Proceedings of the GIS-SIT Conference*. TOPO-CONF-2006-015 (2006)
9. T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum. Comput. Stud. 43(5), 907–928 (1995)
10. T. Gruber, Ontology, in *The Encyclopedia of Database Systems*, ed. by L. Liu, M.T. Özsu (Springer, Berlin, 2009)
11. P.J. Hayes, The second naive physics manifesto, *Formal Theories of the Commonsense World* (1985), pp. 1–36
12. M. Horridge, S. Bechhofer, The OWL API: a Java API for OWL ontologies. Semantic Web 2(1), 11–21 (2011)
13. M. Horridge, S. Jupp, G. Moulton, A. Rector, R. Stevens, C. Wroe, A practical guide to building OWL ontologies using protégé 4 and CO-ODE tools edition 1. 2. The University of Manchester (2009)
14. I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B Grosof, M. Dean et al., Swrl: a semantic web rule language combining OWL and RuleML. W3C Member Submission 21, 79 (2004)
15. M. Hulsen, J.M. Zollner, N. Haeberlen, C. Weiss, Asynchronous real-time framework for knowledge-based intersection assistance, in 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC) (IEEE, New York, 2011), pp. 1680–1685
16. M. Hulsen, J.M. Zollner, C. Weiss, Traffic intersection situation description ontology for advanced driver assistance, in *2011 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, New York, 2011), pp. 993–999
17. Hummel, W. Thiemann, I. Lulcheva, Scene understanding of urban road intersections with description logic, in *Dagstuhl Seminar Proceedings* (Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Dagstuhl, 2008)
18. C.G. Keller, C. Sprunk, C. Bahlmann, J. Giebel, G. Baratoff, Real-time recognition of U.S. speed signs, in *2008 IEEE Intelligent Vehicles Symposium* (2008), pp. 518–523
19. R. Kohlhaas, T. Bittner, T. Schammar, J.M. Zollner, Semantic state space for high-level maneuver planning in structured traffic scenes, in *2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)* (IEEE, New York, 2014), pp. 1060–1065
20. L. Lamard, R. Chapuis, J.-P. Boyer, Multi target tracking with CPHD filter based on asynchronous sensors, in *2013 16th International Conference on Information Fusion (FUSION)* (IEEE, New York, 2013), pp. 892–898
21. D. Maquin, J. Ragot, Diagnostique des systèmes linéaires (Lavoisier, Paris, 2000)
22. D.L McGuinness, F. Van Harmelen et al., Owl web ontology language overview. W3C Recommendation 10(10), 2004 (2004)
23. B. Motik, R. Shearer, I. Horrocks, Optimized reasoning in description logics using hypertableaux, in *Automated Deduction–CADE-21* (Springer, Berlin, 2007), pp. 67–83
24. F. Moutarde, A. Bargeton, A. Herbin, L. Chanussot, Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular traffic signs recognition system, in *2007 IEEE Intelligent Vehicles Symposium* (2007), pp. 1122–1126
25. M. Platho, J. Eggert, Deciding what to inspect first: incremental situation assessment based on information gain, in 2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC) (IEEE, New York, 2012), pp. 888–893
26. M. Platho, H.-M. Gros, J. Eggert, Predicting velocity profiles of road users at intersections using configurations, in 2013 IEEE Intelligent Vehicles Symposium (IV) (IEEE, New York, 2013), pp. 945–951
27. E. Pollard, P. Morignot, F. Nashashibi, An ontology-based model to determine the automation level of an automated vehicle for co-driving, in 2013 16th International Conference on Information Fusion (FUSION) (IEEE, New York, 2013), pp. 596–603
28. V. Popovic, B. Vasic, Review of hazard analysis methods and their basic characteristics. FME Trans. 36(4), 181–187 (2008)
29. Protégé website, http://protege.stanford.edu/. Accessed 29 April 2015
30. Protégé Wiki website, http://protegewiki.stanford.edu/wiki/webprotege. Accessed 29 April 2015
31. Protobuf Website, https://developers.google.com/protocol-buffers/. Accessed 29 April 2015
32. R. Regele, Using ontology-based traffic models for more efficient decision making of autonomous vehicles, in Fourth International Conference on Autonomic and Autonomous Systems, 2008. ICAS 2008 (IEEE, New York, 2008), pp. 94–99
33. R.A. Retting, B.N. Persaud, P.E. Garder, D. Lord, Crash and injury reduction following installation of roundabouts in the united states. Am. J. Public Health 91(4), 628–631 (2001)
34. T. Schamm, J.M. Zollner, A model-based approach to probabilistic situation assessment for driver assistance systems, in 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC) (IEEE, New York, 2011), pp. 1404–1409
35. L. Serafini, A. Tamilin, Local tableaux for reasoning in distributed description logics, in Proceedings of the International Workshop on Description Logics, DL, vol. 4 (2004), pp. 100–109
36. R.M. Smullyan, First-Order Logic (Courier Corporation, North Chelmsford, MA, 1995)
37. S. Vacek, T. Gindele, J.M. Zollner, R. Dillmann, Situation classification for cognitive automobiles using case-based reasoning, in 2007 IEEE Intelligent Vehicles Symposium (IEEE, New York, 2007), pp. 704–709
38. R. Wang, H.J. Ruskin, R. Wang, H.J. Ruskin, Modeling traffic flow at a single lane urban roundabout. Comput. Phys. Commun. 147, 570–576 (2002). Proceedings of the Europhysics Conference on Computational Physics Computational Modeling and Simulation of Complex Systems
39. L. Zhao, R. Ichise, S. Mita, Y. Sasaki, An ontology-based intelligent speed adaptation system for autonomous cars, in Semantic Technology (Springer, Berlin, 2014), pp. 397–413
40. C. Zinoune, P. Bonnifait, J. Ibanez-Guzman, Sequential FDIA for autonomous integrity monitoring of navigation maps on board vehicles. IEEE Trans. Intell. Transp. Syst. 17(1), 143–155 (2016)