

# An Ontology-based Model to Determine the Automation Level of an Automated Vehicle for Co-Driving

Evangeline Pollard  
Team IMARA  
INRIA, Domaine de Voluceau  
78150 Le Chesnay, France  
Email: Evangeline.Pollard@inria.fr

Philippe Morignot  
Team IMARA  
INRIA, Domaine de Voluceau  
78150 Le Chesnay, France  
Email: Philippe.Morignot@inria.fr

Fawzi Nashashibi  
Team IMARA  
INRIA, Domaine de Voluceau  
78150 Le Chesnay, France  
Email: Fawzi.Nashashibi@inria.fr

**Abstract**—Full autonomy of ground vehicles is a major goal of the ITS (Intelligent Transportation Systems) community. However, reaching such highest autonomy level in all situations (weather, traffic, ...) may seem difficult in practice, despite recent results regarding driverless cars (e.g., Google Cars). In addition, an automated vehicle should also self-assess its own perception abilities, and not only perceive its environment. In this paper, we propose an intermediate approach towards full automation, by defining a spectrum of automation layers, from fully manual (the car is driven by a driver) to fully automated (the car is driven by a computer), based on an ontological model for representing knowledge. We also propose a second ontology for situation assessment (what does the automated car perceive?), including the sensors/actuators state, environmental conditions and driver's state. Finally, we also define inference rules to link the situation assessment ontology to the automation level one. Both ontological models have been built and first results are presented.

## I. INTRODUCTION

An autopilot is a system used to guide a vehicle without human assistance. It offers a high level of automation, on 11-level scale for aircraft automation levels [1], [2]. If this kind of technology is well mastered for aircraft since the 70s, and if total automation is even offered with drones since the 2000s, full automation still is a challenging issue for ground vehicles. Research in this area has to deal with many difficulties related to the perception that a vehicle has of its environment and to its ability to deal with unknown situations.

Since 94% of the road accidents in France are due to human errors, it clearly appears that the car of the future, from a safety point of view, should offer automation options. From a socio-economical point of view, driving automation offers new mobility solutions for traffic regulation and gas-emission limitation.

Following this idea, cybercars were designed as fully automated vehicles [3], thought since its inception as a new transportation system, for passengers or goods, on a network of roads with on-demand and door-to-door capability. This concept emerged in Europe in the early 90s and was introduced for the first time in December 1997 at Schipol airport (NL) for passenger transport.

Since this initiative, large efforts have been performed towards full automation with the first Darpa Grand Challenge

in 2004 [4] and its urban version in 2007 (Darpa Urban Challenge [5]). The Grand Cooperative Driving Challenge (GCDC) is the European counterpart and was extended to the test of communication abilities [6]. In 2010, the VisLab Intercontinental Autonomous Challenge, consisted in riding four vehicles from Parma, Italy, to Shanghai, China, mainly in an automated way [7]. The same year, Google announced having created an autopilot system for cars which already drove more than 200,000 km in a fully automated way [8].

The Google driverless car is the first case of a vehicle supposed to autonomously drive in an urban context. Even if Nevada was the first state to issue driverless vehicle licenses on public roadways, total automation remains a future goal, even if vehicles become more and more autonomous in the automotive industry. Indeed, many Advanced Driver Assistance Systems (ADAS) are now embedded into industrial cars in order to help the driver in his driving process. Part of these systems, such as speed alert or blind spot detection, is only providing advice or warning to the driver, but others, such as the Adaptive Cruise Control (ACC) or the Anti-lock braking system (ABS), can be considered as elements of a partial automation since they act on the control part of the vehicle. This increasing automation is possible because sensors become more and more effective and reliable.

However, sensors have predefined operating range and are not free of breakdowns. For example, it is common knowledge that GPS devices are less efficient in urban areas, because of multipath effects and temporary signal loss. Or that, for object detection, lasers can be affected by bad weather conditions such as rain or snow.

If it has been proven that fully automation is possible, even in urban scenarios, we argue in this paper that, given the current progress of this scientific area, safety for fully automated driving cannot be ensured yet at once in all situations (weather, traffic, ...). Therefore, even if full automation remains the final goal, in practice we must take an intermediate approach adapted to the situation at hand. In other words, an adequate level of automation has to be estimated and adapted on-line, depending on sensor limitations and road conditions. At each moment, extending the data process and bringing intelligence to the driving process are mandatory, in order to guaranty safety in all conditions and to cope with unknown situations.

Many efforts have been done in that direction and the whole scientific domain of Intelligent Transportation Systems (ITS) precisely aims at bringing intelligence to automated vehicles, in order to replace the driver with a computer to a small or (hopefully) large extent. This looks particularly essential in the case of intersections, for instance, where the situation is hard to assess and the driving rules difficult to interpret. Hulsen *et al.* [9] build a symbolic representation (an ontology) to describe the situation at intersections in order to reason on it using traffic rules (*e.g.*, the automated car should yield the right-of-way or not?). Bermejo *et al.* [10] also embed a symbolic representation (an ontology too) inside each vehicle in order to deal with emergency situations (*e.g.*, quitting the leftmost lane on a highway when an emergency vehicle is quickly approaching). In a complementary way, the automation level of the vehicle can be adapted to the observed state of the driver (from fully aware through drowsiness to fully asleep), by using a camera detecting eye opening level, blink frequency and blink duration (see the European project HAVEit [11]).

However, there is no definition of what this spectrum of automation/autonomy for ITS should be [12], not even speaking about intelligence. Inspired by [13], we propose in this paper an approach in the same spirit: A high-level symbolic model is proposed to determine the maximum autonomy level which the vehicle should adopt, in order to cope with the current state of the environment as perceived by the vehicle and with the current state of the vehicle itself, and to guarantee a safe driving.

We propose two ontologies: The first one is designed in order to define the relationship between the automation levels and the algorithmic needs; The second one is related to the situation assessment level of the JDL (Joint Director of Laboratories) [14] applied to road driving situation. Our goal is to provide a symbolic representation and a set of rules on it, generic enough to be generalized to any system, in order to estimate the maximum autonomy level according to the current situation and its uncertainty. In contrast to what is presented in the literature, we argue in this paper that an intelligent vehicle, in addition to assess the road situation, should also assess its own abilities in term of automation: To guaranty safety, it is absolutely needed to assess what the vehicle “knows” and to adapt the driving behavior to the perception uncertainties.

The paper is organized as follows: we first remind the reader what ontologies are, present our ontological models about autonomy levels and situation assessment for ITS, and present inference rules to use them (section II). Then, we present a case study in order to explain how further inference rules can be built, and we show results on an implementation in Description Logic (section III). Finally, we relate our approach to existing ones and sum up our contribution.

## II. AUTOMATION SPECTRUM

Before explaining the reasoning process to propose a maximum automation level, it is convenient to define what automation is. Indeed, by automation, we mean all the modes which imply actions which are done by the system through the actuators, including actions against the driver’s intention (*e.g.*, emergency braking). Providing advice and warning to the driver (*e.g.*, blind spot detection) is not considered as an automation mode.

### A. Ontologies

Ontologies are a way to represent knowledge inside a computer. An ontology may be defined as “*a specification of a conceptualization of a knowledge domain*” [15]. For example, the diagnosis of infectious disease constitutes knowledge of the medical domain. Then, its conceptualization might be its representation inside the brain of physicians. Then, its specification might be its representation in a formal computer language.

Alternatively, an ontology may be considered as a complete semantic network, emphasizing that the hierarchy of concepts has to be complete, *i.e.*, not missing concepts involved in the conceptualization. Therefore, ways for building ontologies may rely on analyzing a corpus of texts [16]. Due to lack of such texts in the ITS domain, we rely on interviews with automation experts (members of the team) to attempt at ensuring the completeness of our ontological model.

In an ontology, concepts, role definitions and axioms lay within a terminological box (TBox), while instances of concepts, roles among such instances lay in assertional boxes (ABox). A language for representing knowledge in an ontology is description logic (DL), a subset of first order predicate logic — a subset only due to complexity issues.

Inferences are possible in an ontology using DL-reasoners. Several implementations of such reasoners exist, such as FACT++, PELLET, RacerPro [17] and others.

As opposed to other computer science areas, this language makes the open world assumption, *i.e.*, assumes that when an assertion is not explicitly stated, it is uncertain. This is to be contrasted with the closed world assumption (*e.g.*, in task planning [18], in first order predicate logic) in which a fluent/predicate is assumed to be false when not explicitly specified.

In first order predicate logic, a predicate is the smallest compound syntactic element, *e.g.*,  $P(v1, v2, v3)$ ; in task planning [18], a fluent also is the smallest compound syntactic element, *e.g.*,  $on(blockA, blockB)$ ; The main difference between the two is that a predicate can eventually be negated, while a fluent is always expressed as positive, even if operators of an action plan can use its negation.

Ontology editors are available, *e.g.*, PROTEGE, SWOOPS among others. Due to its ability to include rule languages, we will use PROTEGE in the implementation section.

### B. Ontology description

In Fig.1, automation modes are separated into five layers which are detailed in the following subsections. These five layers correspond to the different control level of a car.

1) *Layer 1: Longitudinal control:* In this layer, the driver does not has to care about the velocity. According to the perception abilities, the system can adopt the *cruise control* mode, where the vehicle is in charge of maintaining a steady speed. With the *Dynamic Set Speed Type* mode, the maximum allowed velocity must be estimated. This can be done by combining a precise map of the environment with the GPS positioning or by perception of the road signs, or with a combination of all the available information, as done in [19].

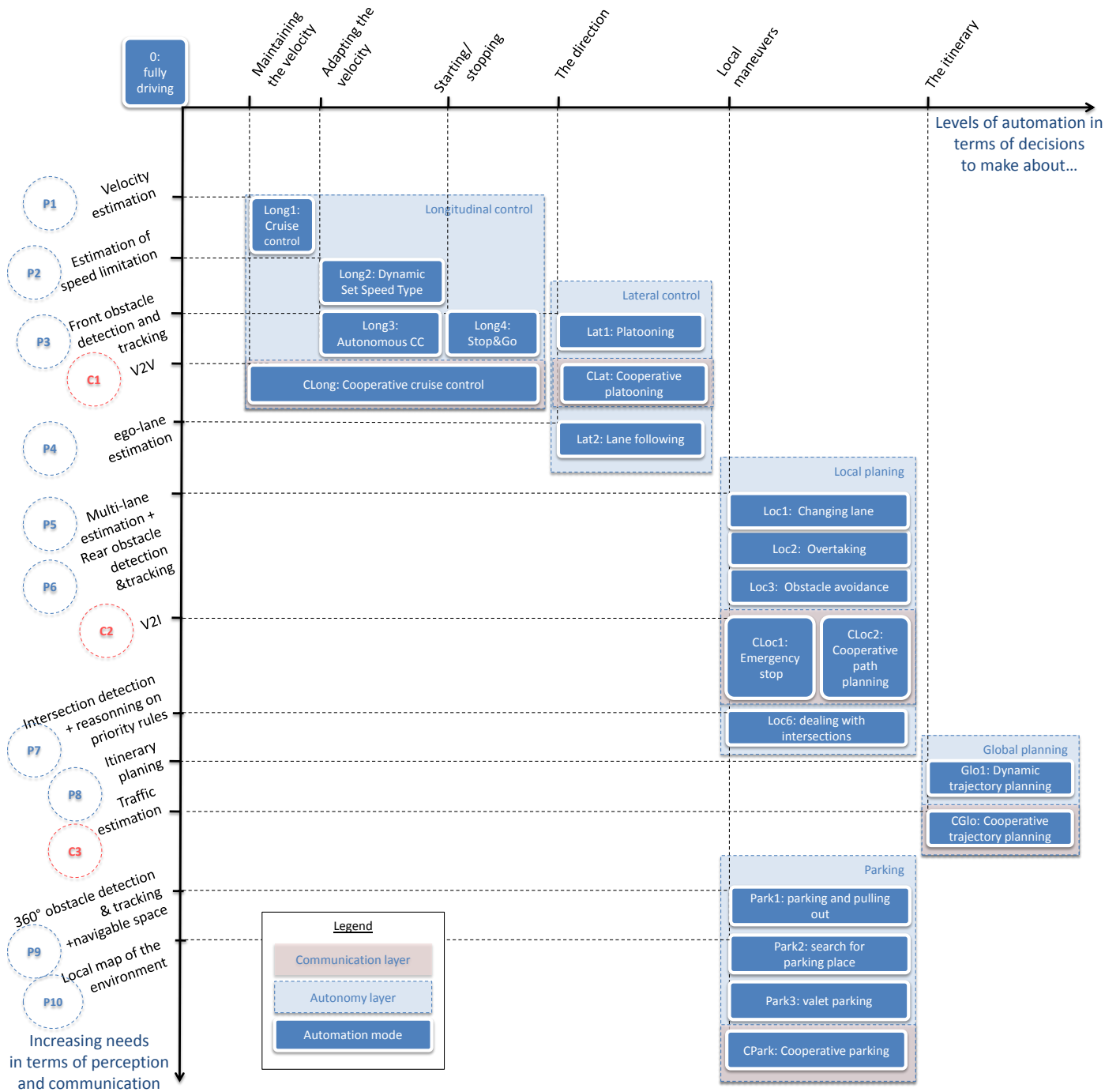


Figure 1. Ontology of autonomy layers

Authorizing the use of the ACC implies to detect and track front vehicles in order to estimate their position, velocity and acceleration over time. In this mode, the system automatically adjusts the vehicle speed to maintain a safe distance between the automated vehicle and the vehicle ahead. When the system can decide to start itself, then the *Stop&Go* mode is available. Finally, if the vehicle has some communication ability, the ACC can become *cooperative*. The vehicle in front broadcasts his own information in order to maintain a safe distance.

2) *Layer 2: Lateral control*: The second layer means that the driver can drive without hand, the system takes control of the steering wheel. Following Fig 1, no additional perception abilities are required to access to the *platooning mode*; Front obstacle detection and tracking is enough (if the accuracy is sufficient) for a vehicle to follow the same trajectory as the one before. Similarly to the longitudinal control, communication ability allows to activate the *cooperative platooning mode*. Finally, if the system is able to estimate the shape of the ego-lane, then the *lane following* mode can be activated. This can be performed by vision [20], but also by using a precise map and a precise ego-localization, or by a combination of both [21].

3) *Layer 3: Local planning*: The third layer, local planning, consists of several local maneuvers which an automated system can deal with. It requires to have a deeper view of the environment. For instance, to activate the *changing lane* mode, the system must provide an estimation of the number of lanes on the road, as well as their sense of driving, to be sure that there is another lane where the vehicle is allowed to move. As for overtaking and obstacle avoidance modes, it is mandatory to estimate the state of the moving obstacles behind and on the side. It means that the vehicle shall be equipped with rear obstacle detection sensors. Vehicle-to-Infrastructure (V2I) communication allows to access to the *emergency stop* mode. If a driver loses control of a vehicle, *e.g.* due to a health emergency, the emergency stop assistant can detect the situation and autonomously take control of the car and bring it to a safe stop. The system activates the emergency flashers, carefully monitors traffic, guides the vehicle to the right shoulder of the road and alerts emergency services by communication. At the *cooperative path planning* level, the autonomous vehicle takes into account information coming from a supervisor to optimize the traffic. The *dealing with intersections* mode is, to our knowledge, out of the state of the art. It is one of the most complicated problem in ITS for a mixed traffic (no intersection supervisor). It is sometimes hard to detect intersection type if it not provided by a map. There are many elements to assess which are specific to intersections (lane marking type, painted arrows, traffic lights,...). A 360° horizontal field of view is required for obstacle detection in order to reason about priority rules (as in [9]). Even when the situation is perfectly assessed, it is sometimes necessary to duplicate human behavior to cross an intersection: go down slowly to have a better view, wait for a car crossing and then go.

4) *Layer 4: Parking*: The *Parking* level contains four different modes. The first one, *parking and pulling out*, just requires to have a 360° horizontal field of view to detect obstacles all around the vehicle at low speed and to detect the navigable space to avoid curbs and holes. For the search for parking place mode, the autonomous vehicle needs to build a local map of its

environment. The *valet de parking* mode is a compression of the two previous ones, and is already a fully automated mode for low velocity. And finally, the *cooperative parking* mode implies communication with a parking supervisor, which could indicate a parking space and a local map of the parking place.

5) *Layer 5: Global planning*: The *dynamic trajectory planning* mode is also a fully automated mode: The driver only has to provide a destination and the vehicle chooses the best itinerary. In *cooperative trajectory planning* mode, the autopilot receives information about the traffic and can change its itinerary, *e.g.* in order to avoid traffic jams for instance

Reference system following (see [22] for a reference system for highway or wire guidance) is not considered in this paper. Requirements for an automated vehicle are exposed here without considering any change in the infrastructure except for V2I communication.

### C. Reasoning on automation modes

There are two levels of reasoning (see the ontology in Fig. 1): reasoning by layer and reasoning by perception abilities. To activate any mode of the lateral control layer, the longitudinal control layer must be activated. In other words, the *longitudinal control* layer is exclusive of upper layers. This can be written as:

$$\neg Long \Rightarrow \neg Lat \ \& \ \neg Loc \ \& \ \neg Glo \ \& \ \neg Park \quad (1)$$

where  $\neg Long$  represents the event "the Longitudinal control layer is not activated" and  $Lat$ ,  $Loc$ ,  $Glo$  and  $Park$  designates the lateral control, the local planning, the global planning and the parking layer. This rule can be applied recursively on upper layers:

$$\neg Lat \Rightarrow \neg Loc \ \& \ \neg Glo \ \& \ \neg Park \quad (2)$$

$$\neg Loc \Rightarrow \neg Glo \quad (3)$$

The second level of reasoning solves the problem of activating a layer and how to link perception abilities to autonomy modes. Perception abilities allow to active automation modes. In the following equation  $P1$  means that this perception skill is available, and  $\neg P1$  means that this perception skill is not available or not done with enough accuracy.

Following Fig. 1, we can write the following rules for the longitudinal control layer:

$$\begin{aligned} P1 &\Rightarrow Long1 \\ Long1 \ \& \ P2 &\Rightarrow Long2 \\ Long2 \ \& \ P3 &\Rightarrow Long3 \ \& \ Long4 \end{aligned} \quad (4)$$

But as for layers (see eq. (1) to (3)), the non activation of a mode is exclusive of the upper mode:

$$\begin{aligned} \neg Long1 &\Rightarrow \neg Long2 \ \& \ \neg Long3 \ \& \ \neg Long4 \\ \neg Long2 &\Rightarrow \neg Long3 \ \& \ \neg Long4 \\ \neg Long3 &\Rightarrow \neg Long4 \end{aligned} \quad (5)$$

Only the validation of the lower to the upper automation mode of the layer implies its activation:

$$Long1 \ \& \ Long2 \ \& \ Long3 \ \& \ Long4 \Rightarrow Long \quad (6)$$

Concerning communication layers, their communication modes need that the lower perception modes be activated.

For example, the *cooperative cruise control* mode cannot be activated if the *Cruise Control* mode is activated or if no front obstacle detection and tracking abilities are provided by the system (for safety reason, communication must be merged with perception). But this communication layer will not be activated to allow the activation of the current and upper layer.

$$C1 \ \& \ Long1 \ \& \ Long2 \ \& \ Long3 \ \& \ Long4 \Rightarrow CLong \quad (7)$$

Following Fig. 1, the logics for reasoning about the activation of modes and layers can be extended.

Several of the terms above require representing an uncertainty level (*e.g.*, a probability). Although float numbers (*e.g.*, a float number between 0 and 1 representing a probability) can be encoded into OWL (Ontology Web Language), we prefer to discretize this uncertainty on a scale for easiness of interpretation, *e.g.*, representing the lateral position as three mutually exclusive subclasses “high”, “low” and “bad” (see Fig. 2).

#### D. Situation assessment representation

Now the question we have to answer is how to link the different automation modes to the situation assessment made by the autonomous vehicle. Let us start with a general definition of the situation assessment for a road vehicle.

Following [23] for military applications, situation assessment for a road vehicle can be defined through six levels (see Fig. 2).

These levels contain sub-levels (subclasses) and each leaf has to be assessed (*i.e.*, instances have to be defined for these classes of the ontology). For numerical value (boxes in sky blue in Fig. 2), such as the ego-state, not only the numerical value must be calculated, but also an estimation of the error. Then threshold must be used to establish if the accuracy is enough to validate the corresponding perception or communication task. For symbolic values (speed limit for instance, boxes in pink in Fig. 2) a confidence rate must be provided.

- The driver: the goal here is to assess the ability to drive of the driver by using interior camera or stereo-camera to monitor the driver. His ability to drive can be studied through several factors: position/orientation of the head, gaze direction, blinking, position of the hands, ... A confidence rate about the ability of the driver to take the control of a part of the driving process must be provided.
- The ego-vehicle: Beyond the absolute positioning of the vehicle which can be provided by a GPS-RTK, what is interesting is to estimate the position regarding the lateral and longitudinal error. Threshold must be established to classify the position estimations as high, low or bad precision. In addition, nominal velocity, orientation of the vehicle and acceleration must be assessed.
- Communication: Quality of service is the ability of broadcasting data in good conditions. It combines several aspect for a communication channel: transmission rate, lost rate, delay,...

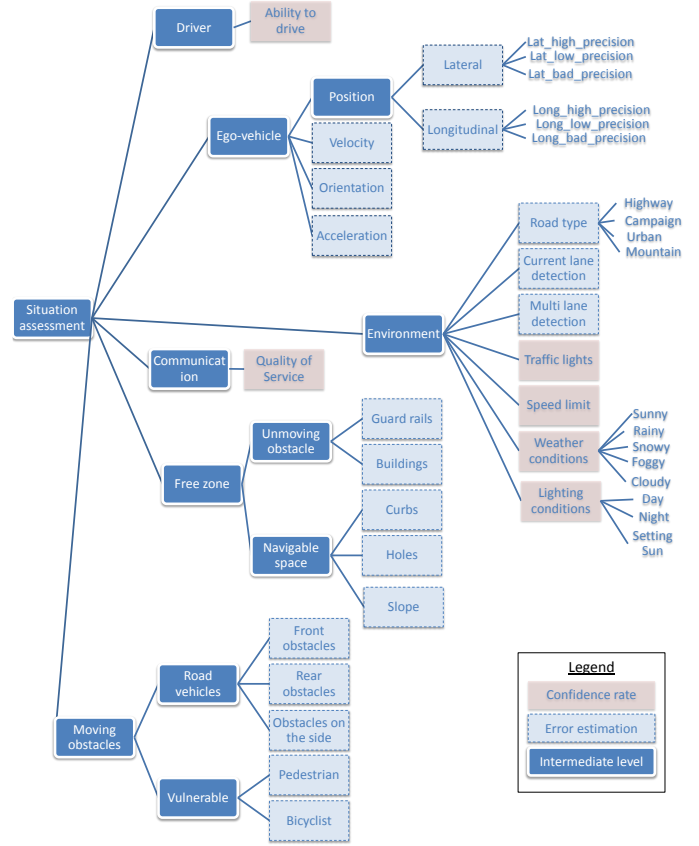


Figure 2. Ontology for situation assessment

- Free zone: The free zone combines the state estimation of both unmoving obstacles and navigable space, which designates the area where the autonomous vehicles can move (for example it cannot go through a curb). For this part the goal is to detect the presence of some elements, such as guard rails or buildings, as well as curbs, holes and slope.
- Moving obstacles: Moving obstacles can be classified in two categories: *road vehicles* such as car, bus and truck and *vulnerable people*, such as pedestrians and bicyclists. The goal is to provide the best estimation of their state as possible, including, position, velocity, acceleration and orientation.
- Environment: The environment contains many element which can be provided by a map, such as the traffic light position, the lane shape and the speed limit, as well as elements which have to be assessed or broadcast such as the weather conditions or the lighting conditions.

Representing all these elements on the same picture enables the vehicle to have the most accurate image of the ground reality.

### III. CASE STUDY

The goal is to provide an estimation of the maximum allowed automation mode  $\tilde{M}_k$  at each iteration  $k$  in the perception / path planning / control cycle.

The operators  $<$  and  $>$  designate the fact that an automation mode is said upper or lower than another one in term of automation. For example,  $\mathcal{Loc}1 < \mathcal{Loc}2$  or  $\mathcal{Lat}1 > \mathcal{Long}4$  according to Fig. 1.

The perception system allows to estimate the state of all, or part of, the variables of the situation assessment presented in Fig. 2. One issue is to evaluate if the used information are reliable or not.

#### A. Example of system description

Each intelligent vehicle, due to its perception abilities, has its own limitation and a default maximum automation mode must be first calculated. In our case study, we were inspired by previous experiments in La Rochelle in 2011 [24], in which automatic transportation vehicles (Cybercars) were deployed. Our automatic vehicle is equipped with acceleration/braking and steering actuators. Laser range finder sensors are mounted in the front and in the back of the vehicle for the front and rear obstacle detection. A frontal camera gives the road shape and the number of lanes, and recognizes road sign units. A low cost GPS device is used for ego-localization, in addition to an Inertial Measurement Unit (IMU) and odometers, to improve the ego-vehicle state estimation. The driver is monitored through a frontal camera. The vehicle is able to communicate with the infrastructure, which supervises the intersections and provides information about the weather conditions. Finally, a precise map of the area is available, with road position, type and speed limitation. Our vehicle is not empowered with building a local map of the environment and detecting the elements of the navigable space. It can thus access to all the automation layers except the *Parking* one.

#### B. Weather consideration

The automated vehicle manufacturer ensures that the vehicle cannot ride on snow and ice. At this moment, it is the driver responsibility to drive under these conditions. That can be written as:

$$snowy \Rightarrow \tilde{M}_k = 0 \quad (8)$$

where  $\tilde{M}_k$  is current maximum automation mode.

It is common knowledge that under rain conditions, laser sensor performances decrease. It means that the obstacle detections are not reliable, limiting the maximum automation mode (the more rain, the lesser the laser performances, hence encoding a form of graceful degradation, as for anytime algorithms).

$$raining \Rightarrow \neg P3 \Rightarrow \tilde{M}_k = \mathcal{Long}2 \quad (9)$$

Foggy conditions affects the visibility and computer vision algorithms (lane detection and speed limit unit detection). However, if the lateral and longitudinal ego-localization precision are high (*lat HP* means that the lateral positioning is performed with high precision, and similar considerations for *long HP*), lane and speed limit estimation can be still

obtained by using localization and mapping. But if the lateral or longitudinal localization is damaged then some functions are lost.

$$foggy \ \& \ \neg lat \ HP \ \& \ long \ HP \Rightarrow \neg P4 \ \& \ P2 \quad (10)$$

$$foggy \ \& \ lat \ HP \ \& \ \neg long \ HP \Rightarrow P4 \ \& \ \neg P2 \quad (11)$$

#### C. Combining all the information

As soon as a piece of information is coming, either on the automated vehicle's sensors/actuators or on the environment, the corresponding individuals must be updated in the ontology, in order to assess the perception abilities. According to these perception abilities, the current maximum automation mode  $\tilde{M}_k$  is computed through inference rules (see section III-E below).

#### D. Combining with the current driver state

In order to include the driver into the model, we distinguish between the automation level of the vehicle  $\tilde{M}_k$  and the same automation level, but including the driver  $\hat{M}_k$ . More precisely, if a piece of information, which does not concern the driver state, is used to calculate  $\tilde{M}_k$ , then this current maximum automation mode must be combined with the current driver state to provide the estimated maximum automation mode  $\hat{M}_k$ . In fact, if the current automation mode implies the driver intervention and if this last one is not ready to drive ( $\neg DS$ ), then the level of automation cannot decrease. This can be expressed as:

$$(\tilde{M}_k < \hat{M}_{k-1}) \ \& \ \neg DS \Rightarrow \hat{M}_k = \hat{M}_{k-1} \quad (12)$$

In this case, an alert is given to the driver to give him more control back.

This strategy, also coarse in the previous equation, models a minimal safety concern.

#### E. Implementation

As a first step, the ontologies of section II have been implemented in OWL (Ontology Web Language) using the ontology editor PROTEGE. The rule language SWRL (Semantic Web Rule Language) [25] is used to represent the inference rules, the DL-reasoner PELLET is used to make inferences over these rules.

For example, eq. (9) can be written in SWRL as:

$$Rainy(?c), FullyManual(?a) \quad - > \ DynamicSetSpeedType(?a) \quad (13)$$

That is, when there exists an individual  $?c$  of the class "Rainy", and when the automation level  $?a$  is the default one (*i.e.*, an individual of the class "FullyManual", which *de facto* satisfies equation 8), then  $?a$  is also an individual of the class "DynamicSetSpeedType", which was denoted by the proposition  $\mathcal{Long}2$  in Sec. III-B.

Since SWRL does not support negated terms (due to the monotonicity of Description Logics), the maximum aspect of an automation level (*i.e.*, the contrapositive of eq. (1), (2) and (3)) can be represented by 14 inference rules, the structure of which follows:

$$AutonomousCC(?a) \text{--} > DynamicSetSpeedType(?a) \quad (14)$$

$$DynamicSetSpeedType(?a) \text{--} > CruiseControl(?a) \quad (15)$$

That is, when there is an individual  $?a$  of class “AutonomousCC”, then  $?a$  is also an individual of the class “DynamicSetSpeedType”, the immediately lower automation level. A propagation towards classes of lower automation level can thus be encoded this way, denoting this maximum aspect.

Due to the same constraint on negation absence in SWRL, eq. (10) and (11) have to be simplified in the following inference rules:

$$Foggy(?c), LongitudinalHighPrecision(?b), FullyManual(?a) \text{--} > DynamicSetSpeedType(?a) \quad (16)$$

$$Foggy(?c), LateralHighPrecision(?b), FullyManual(?a) \text{--} > LaneFollowing(?a) \quad (17)$$

That is, if there exists an individual  $?c$  of the class “Foggy”, subclass of the class “WeatherCondition” (see Fig. 2), and if there exists an individual  $?b$  of the class “LongitudinalHighPrecision”, subclass of “LongitudinalPosition”, then the autonomy level  $?a$  should also be an individual of the class “DynamicSetSpeedType”, specifying an autonomy level, in addition to the default one “FullyManual”. The second inference rule above follows the same scheme.

The temporal dimension of eq. (12) can be implemented the following way:

$$NotAbilityToDrive(?d), modeBefore(?a) \text{--} > mode(?a) \quad (18)$$

It uses (1) two mutually excluding classes, *AbilityToDrive* and *notAbilityToDrive* (due to negation absence in SWRL), both subclasses of the class *DriverState* in the ontology on situation assessment ; and (2) additional classes *modeBefore* (*mode* being each autonomy layer name, *i.e.*, *AutonomousCC*, *DynamicSetSpeedType*, ...) representing the existence of each previous automation mode. In other words, if the driver is not able to drive, each potential autonomy mode is propagated forward in time, therefore keeping the same maximum autonomy level at the next cycle, given the propagation mechanism above. (Lowering the automation level of the vehicle, as states in section III-D, can be encoded by adding a “DriverState” term in the inference rule above.)

Here is an example of assertions regarding the autonomy level, when a SLAM algorithm is present, inferred by the DL-reasoner by classifying the ontology, in 250 ms on a PC computer with 2 CPUs, a clock frequency of 2.5 GHz and 6 Gb RAM. It is obtained by filling the classes of the ontological

model with individuals asserting the capabilities of the vehicle, *e.g.*, SLAM present.

```
CruiseControl
DynamicSetSpeedType
AutonomousCC
Stop&Go
Platooning
LaneFollowing
ParkingAndPullingOut
```

That is, all the autonomy levels up to the maximum *ParkingAndPullingOut* (see Fig. 1) are activated by the propagation inference rules above (class “ParkinAndPullingOut” and below owns an individual).

#### IV. RELATED WORK

There is a growing interest in the ITS community towards ontological representation and use. For example, as said in section I, Regele [26] and Hulsen *et al.* [9] build a high level ontological representation of the environment (the road network, vehicles and traffic signs) in order to infer the next motion of vehicles at intersections, as prescribed by traffic regulation. In our framework, that work aims at reaching the *global planning* autonomy level in Fig. 1, and their ontology of the environment would fit as subclasses of the “environment” class in the situation assessment ontology (see Fig. 2).

Our work may appear as a way to choose the perception algorithm that the automated vehicle should use given the sensors and actuators available now in the robotic vehicle/system, given its environment and given the driver’s state. This concern is close to the one of robotic software architecture, seen as organizing the various software components (with various constraints and response time) and on-line selection of the “best” software component to activate (see for example the Subsumption Architecture [27], the ATLANTIS architecture [28], or 3-layer architectures [29] among others). There is a major difference with this body of work, though: Our point is to select an adapted level of autonomy for an automated vehicle, on a range from fully manual to fully automated, whereas these authors directly jump to the fully automated one. Once again, we advocate that current ITS do not reach this level yet in all situations, and that the autonomy level of the automated vehicle must be adapted to sensors/actuators state, environmental conditions and driver’s state.

Parasuraman *et al.* [30] propose four concepts (information acquisition, information analysis, decision and action selection, and action implementation) to define a framework of interaction between humans and a computer system, and a method to use it. In the view of these authors, our ontological model on situation assessment may fit into the information analysis class, while our inference rules may fit in the decision & action selection one (*i.e.*, one way to implement deciding on which module to activate). The algorithms represented by each class of our ontological model on automation layers may be considered as one way for performing the action implementation of these authors.

#### V. CONCLUSION

In this paper, we present a knowledge representation system for on-line configuration of perception algorithms used

to determine the automation level / autonomy layer of an automated vehicle. It can be considered as a self-assessment of the perception system to monitor co-driving. Two ontologies are proposed: one for representing the autonomy layers of the automated vehicle, on a range from fully manual to fully automated; And another one for situation assessment, integrating the vehicle perception, environmental conditions and the driver's ability. Inference rules are proposed, relating the latter to the former, hence computing the automation level each specific automated vehicle can reach.

Future work includes embedding these two ontologies into real platforms (CyberCars) by linking the classes' individuals to percepts (sensor/actuators state, environmental conditions, driver's state) and perception algorithms.

#### ACKNOWLEDGMENT

This work is part of the Link&Go project. The authors would like to thank the Agence Nationale pour la Recherche (ANR) and the Conseil Général des Yvelines for supporting the project.

#### REFERENCES

- [1] B. Clough, "Metrics, schmetrics! how the heck do you determine a UAV's autonomy anyway?" in *Performance Metrics for Intelligent Systems Workshop*, Gaitersburgh, MA, 2002.
- [2] T. Helldin and G. Falkman, "Human-centred automation of threat evaluation in future fighter aircraft," in *Proceedings of the 6th. Workshop in Sensor Data Fusion: Trends, Solutions, Applications*, 2011.
- [3] M. Parent, "Advanced urban transport: Automation is on the way," *Intelligent Systems*, IEEE, vol. 22, no. 2, pp. 9–11, march-april 2007.
- [4] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-e. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. V. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley : The Robot that Won the DARPA Grand Challenge," vol. 23, no. April, pp. 661–692, 2006.
- [5] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhne *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.
- [6] E. van Nunen, R. Kwakernaat, J. Ploeg, and B. Netten, "Cooperative competition for future mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1018–1025, sept. 2012.
- [7] M. Bertozzi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, S. Debatisti, R. Fedriga, M. Felisa *et al.*, "The vislab intercontinental autonomous challenge: 13,000 km, 3 months, no driver," in *Proc. 17th World Congress on ITS, Busan, South Korea*, 2010.
- [8] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV)*, 2011 IEEE, June, pp. 163–168.
- [9] M. Hülsen, J. M. Zöllner, and C. Weiss, "Traffic Intersection Situation Description Ontology for Advanced Driver Assistance," in *Intelligent Vehicles Symposium*, no. Iv, 2011, pp. 993–999.
- [10] A. J. Bermejo, J. Villadangos, and J. J. Astrain, "Ontology Based Road Traffic Management," *Intelligent distributed computing*, pp. 103–108, 2013.
- [11] F. Flemisch, F. Nashashibi, N. Rauch, A. Schieben, S. Glaser, H. Mosebach, J. Schomerus, S. Hima, and A. Kaussner, "Towards Highly Automated Driving : Intermediate report on the HAVEit-Joint System Introduction : From ADAS to highly automated vehicles," in *TRA*, 2010.
- [12] F. Flemisch, J. Kelsch, and C. Löper, "Automation spectrum, inner/outer compatibility and other potentially useful human factors concepts for assistance and automation," *Human Factors for assistance and automation*, no. 2008, pp. 1–16, 2008. [Online]. Available: [http://elib-v3.dlr.de/57625/1/08HumanFactorsForAssistanceAndAutomation\\\_FlemischEtAl\\\_AutomationSpectrum.pdf](http://elib-v3.dlr.de/57625/1/08HumanFactorsForAssistanceAndAutomation\_FlemischEtAl\_AutomationSpectrum.pdf)
- [13] C. Schlenoff, R. Washington, T. Barbera, and C. Manteuffel, "A Standard Intelligent System Ontology," in *Proceedings of the unmanned ground vehicle technology*, 2005.
- [14] D. L. Hall, S. Member, and J. Llinas, "An Introduction to Multisensor Data Fusion," vol. 85, no. 1, 1997.
- [15] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1992.
- [16] N. Hernandez, "Ontologies de domaine pour la modélisation du contexte en recherche d'information," Ph.D. dissertation, Université Paul Sabatier, 2005.
- [17] V. Haarslev, K. Hidde, R. Möller, and M. Wessel, "The racerpro knowledge representation and reasoning system," *Semantic Web Journal*, vol. 2, 2011.
- [18] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*. Morgan Kaufmann, Elsevier, 2004.
- [19] A. Puthon, F. Nashashibi, and B. Bradai, "Improvement of multisensor fusion in speed limit determination by quantifying navigation reliability," in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2010, pp. 855–860.
- [20] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.
- [21] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
- [22] W.-b. Zhang, R. E. Parsons, and T. West, "An intelligent roadway reference system for vehicle lateral guidance/control," in *American Control Conference*, 1990, pp. 281–286.
- [23] A.-C. Boury-Brisset, "Ontology-based Approach for Information Fusion," in *International Conference on Information Fusion*, vol. 1, no. 418, 2003, pp. 522–529.
- [24] L. Bouraoui and C. Boussard, "An on-demand personal automated transport system: The CityMobil demonstration in La Rochelle," in *Intelligent Vehicles Symposium*, no. Iv, 2011, pp. 1086–1091. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\\_all.jsp?arnumber=5940545](http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=5940545)
- [25] I. Horrock, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean, "Swrl: A semantic web rule language combining owl and ruleml," vol. Submission to W3C <http://www.w3.org/Submission/SWRL>, no. May, 2004.
- [26] R. Regele, "Using Ontology-based Traffic Models for more efficient Decision Making of Autonomous Vehicles," in *International Conference on Autonomic and Autonomous Systems*, no. 028062, 2008, pp. 94–99.
- [27] R. Brooks, "A robust layered control system for a mobile robot," *Journal of Robotics and Automation*, IEEE, vol. 2, no. 1, pp. 14–23, 1986.
- [28] E. Gat, "Three layer architectures," in *A.I. and Mobile Robots*, D. K. et al., Ed., 1998.
- [29] R. Alami, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *International Journal of Robotic Research*, vol. 17, no. 4, pp. 315–337, 1998.
- [30] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, 2000.