

# An Ontology to Decide on Respect or Relaxation of Traffic Rules in Unusual Situations for Autonomous Vehicle Assistance

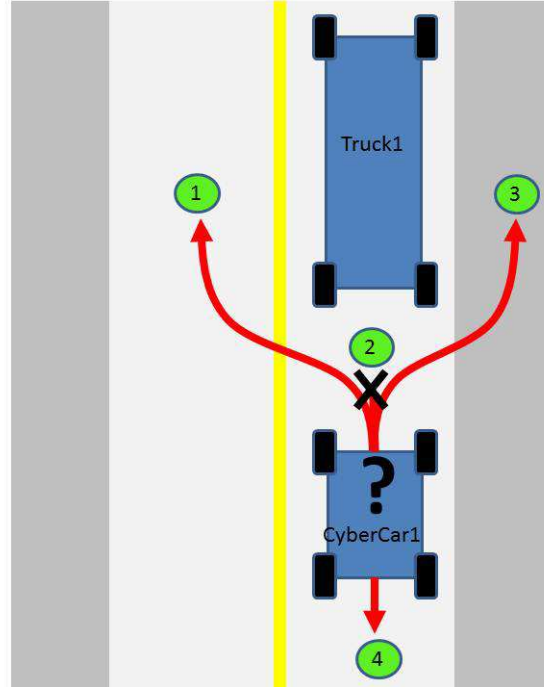
**Abstract.** Traffic may exhibit practical situations in which relaxation of traffic regulation, though illegal, might be tolerated, e.g., if a truck unloads furniture along a continuous line in front of an automated vehicle. In this paper, we propose a discrete, symbolic, high level representation of automated vehicles, other vehicles and infrastructure, in order to decide on respect/relaxation of traffic regulation in unusual situations (i.e., situations not planned for by the traffic regulation). This model, aimed at being embedded into an automated vehicle, is based on an ontology and uses inference rules to induce the next discrete motion of an automated vehicle, often respecting traffic rules, and sometimes relaxing them. Experiments in simulation, using OWL and SWRL, exhibit performances compatible with adding a reasoning component into the perception / planning / control cycle.

**Keywords:** Knowledge representation, Ontology, Intelligent Transportation Systems, Traffic regulation

## 1 Introduction

There is a large body of research dedicated to assisting in vehicle's driver (Advanced Driver Assistance Systems, or ADAS), or to autonomously driving vehicles (e.g., for the DARPA challenge [18]). The final purpose of this work is to integrate computer-driven vehicles into regularly human-driven vehicles flow, as a first step towards replacing human-driven vehicles by computer-driven ones. The advantages of this automated approach would be more efficient surface transportation and increased safety [15]. However, one century of transportation via human-driven vehicles has led to traffic regulation, aiming at both circulation enablement and safety [6]. As a consequence, vehicles must respect traffic regulations, whether the vehicle is driven by a human or by a computer. For example, vehicles are supposed to circulate on roadways and not on sidewalks [8]. Or, continuous lines must not be crossed (article 26.1 [6]).

However, unusual traffic situations might occur, in which traffic regulation can paradoxically be relaxed for practical concerns. For example, if a truck is unexpectedly stopped on the roadway unloading furniture, an acceptable behavior of cars arriving behind it is to safely overtake after some reasonable time spent waiting, despite a continuous line (see Fig. 1); Or, if a vehicle is stopped on the roadway with an engine problem, the same safe overtaking behavior might be acceptable for arriving cars as well. Many similar examples of traffic rules relaxation can be taken from every driver's experience, in which not respecting traffic rules might be tolerated because of unusual traffic situations (e.g., because of a blocked roadway).



**Fig. 1.** Situation in which a truck *Truck1* is stopped on the roadway unloading furniture, in front of an arriving automated vehicle *CyberCar1* along a continuous line. Will the automated vehicle overtake the truck (case 1), stop and wait (case 2), use the sidewalk (case 3), or back up (case 4)?

Reasoning on traffic regulations themselves and their applicability to practical traffic situations is needed indeed. We propose to represent the knowledge necessary to handle such unusual situations, at the border between general traffic legality and temporary traffic illegality. Our long term goal is to embed this model and reasoning into automated vehicles (e.g., CyberCars [7]) in order for them not to stay stuck in such unusual situations (previous experiments with a CyBus in the city of La Rochelle on the CityMobil project in Nov. 2011 led to such blockages), but to take the lead in such unusual situations, escape from them and finally keep circulating, e.g., despite blocked roadways.

## 2 Model

### 2.1 Vehicle and traffic representations

An interesting approach towards representing traffic is vehicular ad hoc networks (VANET), e.g., [16] [17].

In this model, each road is represented as a queue and a vehicle is represented as a packet in one queue. Then control laws on the volume of this queue are set in order to

observe less global congestion of the traffic distributed over this network [17]. Other models in this macroscopic view represent the vehicle level by a state automaton [16], hence enabling to describe some compiled form of reasoning inside an automated vehicle.

Other models, less global than those above and more focused on the automated vehicle, explicitly introduce reasoning inside each automated vehicle [3] [14] [20]. In these models, each automated vehicle embeds (in simulation) an ontology, to describe static knowledge, and inference rules, to represent dynamic knowledge (reasoning) over this static representation. With such additional representational and reasoning capabilities, automated vehicles are able (for example) to infer to yield right of way at complex intersections for every vehicle to pass safely [14] [20], or to infer to free the leftmost lane on highways for emergency vehicles to pass safely [3]. Potentially problematic situations can then be “solved” by reasoning.

We have to make the following assumptions for our approach to hold: (i) there is a device in each vehicle which is capable of communicating with the other vehicles, (ii) each vehicle reasons on self-perceived and received information before taking decision on the action to perform next, and (iii) reasoning is readily performed by an automated vehicle, i.e., without dooming its own perception, path-planning and control algorithms. The former includes the whole domain of vehicle/infrastructure communication (V2V, V2I, V2X). The latter probably is the stronger assumption, since the larger the representation, the slower the reasoning (the complexity of these algorithms is exponential in the worst case, due to the general combinatorial explosion entailed by using such symbolic representations).

Software architectures, organizing software components along their response time, have been proposed in order to integrate such computational costly component into other components connected to the outside environment of the robot/vehicle [1] [5] [9] [12].

In order for the automated vehicle to relax traffic regulations in specific local unusual situations, we keep the principle of the approaches [3] [14] [20] (and therefore their assumptions) and define dedicated ontology and inference rules.

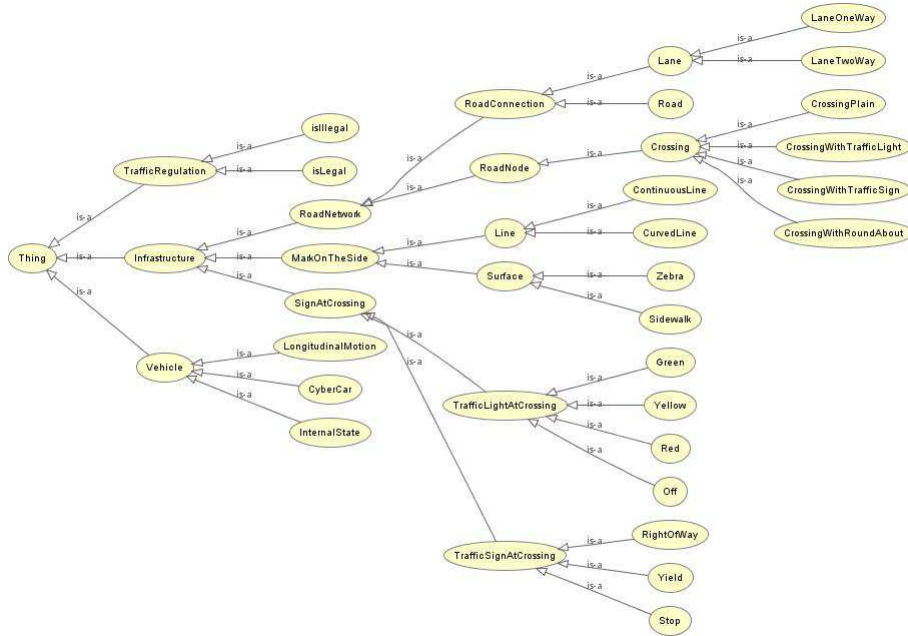
## **2.2 Ontologies**

An ontology is a specification of a formalization of a domain of knowledge [11], i.e., a representation in a formal language of concepts completely describing a specific knowledge domain (e.g., traffic regulation, traffic infrastructure, automated vehicle). It is composed on one side of concepts, roles/properties and axioms (together forming a terminological box, or TBox) and, on the other side, of individuals and relations among them (together forming assertional box, or ABox).

Importantly, when the truth value of a term is not specified in a given ontology, it is assumed to be unknown (open world assumption), as opposed to other scientific domains where the same term would be assumed to be known and even false (closed world assumption, e.g., in task planning [10]).

Tools are available for creating/editing an ontology in OWL (Ontology Web Language), e.g., PROTÉGÉ [19], SWOOP [21] among others.

As a first step, the proposed ontology (see Fig. 2) is built not by analyzing a corpus of texts, but by interviewing experienced drivers (“experienced” meaning here “used to unusual situations”), the subtree rooted at the RoadNetwork class being inspired by [20] (a directed graph, with vertices being road intersections and edges being lanes).



**Fig. 2.** All classes of the ontology, sorted along the “is-a” relation, as shown by GraphWiz

### 2.3 Perception / path-planning / control

The main assumption for our approach to hold is discreteness: We assume that the motion of an automated vehicle is discrete, i.e., decomposable into separated elementary motions; And we assume that time is discretized by time steps, i.e., a decision is made every time tick, and stays the same between two successive time ticks (separated time segments). Under the two previous assumptions, we propose a symbolic deci-

sion model to compute which next motion the automated vehicle should execute, and this with regards to traffic rules.

The traditional computing cycle for ITS is composed of perception, then path planning, then control [15]. Perception is the operation of mapping signals from the vehicle's sensors to atomic percepts, e.g., the automated vehicle perceives a car in front of it. Path planning is the operation of computing a collision-free trajectory for a more or less far future given known obstacles, e.g., finding a trajectory through an intersection that leads to the opposite roadway while avoiding all other vehicles. And control is the operation of sending the right voltage to the vehicle's actuators, for the vehicle to move while following a planned trajectory, e.g., overtaking using a Bezier curve.

We propose to add an operation to this cycle, between the perception and the path planning module: reasoning, considered as the operation of deciding which symbolic action to perform next among a set of symbolic actions which might be taken by the automated vehicle. That is, reasoning is based on recognized percepts from the perception operation and produces goal locations, for which the path planning operation will produce a path. For example, facing the situation described in Section I the automated vehicle could decide to stop behind the unloading truck (case 2 in Fig. 1), overtake it (case 1), run on the sidewalk (case 3), or back up (case 4). Choosing among these "stop", "overtake", "run on sidewalk" or "back up" actions is the result of the reasoning operation.

However, a major problem with adding this reasoning operation to this cycle is complexity issues: in the worst case this reasoning operation produces a solution in an amount of time which is an exponential function of one of the problem's data dimensions (unless  $P = NP$ ). In the worst case, the proposed perception / reasoning / planning / control cycle could be stopped in practice inside the reasoning operation, the vehicle itself being stuck, hence leading to unacceptable safety problems. Therefore, we limit the reasoning operation to problems with fast solutions: in our case making inferences in a Description Logic (DL).

## **2.4 Representation of illegality**

Traffic rules (see [8] for France and [6] for an international convention) may be represented by several means. For example, the rule stating that vehicles must circulate on roadways except in case of absolute emergency (see section R412-7 in [8]), might entail to represent roadways as a graph: vertices are intersections, edges are roadways [20]. In this case, the above rule is implicitly encoded, since there is no way to represent a vehicle's location except into this graph --- therefore vehicles circulate on roadways, as dictated by this traffic rule.

Now traffic rules may be represented in other ways. For example, one general way to represent traffic rules is to say that a vehicle must not execute any illegal motion. This would be written in SWRL as:

$$\begin{array}{l} \text{Motion}(?m), \text{isIllegal}(?m) \\ \rightarrow \text{not}(\text{isExecutable}(?m)) \end{array} \quad (1)$$

That is, if motion  $?m$  is illegal, then  $?m$  cannot be executed (the automated vehicle has to select one specific motion among the set of executable ones, excluding the unexecutable ones). Unfortunately, there are several problems with this representation:

- A. The class *isIllegal* represents the result of the application of traffic rules to a specific motion  $?m$ . This entails that (i) the vehicle's motions are represented in a more detailed way, e.g., motion  $?m$  could be "changing from lane  $?l1$  to lane  $?l2$ "; and (ii) the whole body of traffic rules is represented as concluding onto the mutually excluding classes *isIllegal* or *isLegal*.
- B. Negation cannot be used in the right-hand side of an SWRL inference rule (the only way in SWRL to encode the negation of a property in the right-hand side of a SWRL rule is to state a negated property "*isNotX*( $?x$ )" and to restrict the cardinality of variable  $?x$  in the left-hand side). So the legality aspect of a motion should only conclude on executable motions, not on unexecutable ones. Taking the contrapositive of the previous inference rule makes no computational sense either.

The representation that we propose is to use inference rules which conclude on the executability of a motion of the automated vehicle. Without including legal aspects, the inference rule concluding on the motion to execute is the following:

$$\begin{array}{l} \text{Car}(?c), \text{CurrentSituation}(?c, ?s) \\ \rightarrow \text{hasNextMotion}(?c, ?m) \end{array} \quad (2)$$

That is, given the current situation  $?s$  of vehicle  $?c$ , the next motion of the vehicle is  $?m$ . But now, we must introduce the legal/illegal aspect of this motion  $?m$ , which leads to the following inference rule.

$$\begin{array}{l} \text{Car}(?c), \text{CurrentSituation}(?c, ?s), \text{isLegal}(?m) \\ \rightarrow \text{hasNextMotion}(?c, ?m) \end{array} \quad (3)$$

That is, we added an extra term to the left-hand side of the inference rule, forcing the considered motion  $?m$  to be legal. As a consequence, all inferred motions  $?m$  respect traffic rules --- this is the regular case. Now, we aim at representing unusual situations, in which traffic rules may be relaxed under some circumstances, as described in section 1. This leads to the following additional inference rule.

$$\begin{array}{l} \text{Car}(?c), \text{CurrentSituation}(?c, ?s), \text{isIllegal}(?m), C \\ \rightarrow \text{hasNextMotion}(?c, ?m) \end{array} \quad (4)$$

That is, in addition to inference rule 3: even if the motion  $?m$  is illegal, but if generic condition  $C$  holds, then the motion  $?m$  can be executed anyway, i.e., rule 3 cannot conclude on motion  $?m$  because of its illegality. Therefore an automated vehicle can indeed relax traffic rules (i.e., perform illegal motions) under some generic condition (noted  $C$ ). Inference rule 4 may be considered as an exception of rule 3 under generic condition  $C$  --- rule 3 being the regular case.

Generic condition  $C$  states under which conditions an illegal motion can be performed by the automated vehicle and thus is of major importance. For drivers, the domain of cognitive psychology brings mechanisms with which drivers are aware of a traffic situation [2] but to our knowledge there is no result in that domain regarding relaxation of traffic rules by drivers in unusual situations. For example, in the situation of Fig. 1, we believe that every driver needs some time to solve the conflict between respecting traffic rules (i.e., staying blocked behind the truck unloading furniture on the roadway because of a continuous line, see case 2 of Fig. 1) and overtaking when the opposite lane is clear from other vehicles (i.e., safely crossing the continuous line, see case 1 of Fig. 1). This time probably also depends on many other factors such as the level of tiredness of the driver, the level of alcohol in his blood, etc. Facing this complexity, we propose to model generic condition  $C$  by a time out: when the time out is still running, we assume that the vehicle's state is labeled "Waiting" and the automated vehicle keeps being stopped; And when the time out has expired, we assume that the vehicle's state is labeled "Passing" and the automated vehicle can move again despite interdictions --- the threshold itself on elapsed time needing more experiments to be set.

## 2.5 Application

In order to represent case 1 of the situation depicted in Fig. 1, the inference rule is the following.

1. *Lane(?l1), Lane(?l2), CrossableZone(?s),*
  2. *Car(?a), Car(?b),*
  3. *isAfter(?a, ?b),*
  4. *hasBesides(?l1, ?s), hasBesides(?l2, ?s),*
  5. *hasMotion(?a, Stopped),*
  6. *isOn(?a, ?l1), isOn(?b, ?l1), DifferentFrom (?l1, ?l2)*
  7. *isClear(?l2),*
  8. *hasInternalState(?a, Passing),*
  9. *-> isNextOn(?a, ?l2),*
  10. *hasNextMotion(?a, Forward)*
- (5)

That is, the automated vehicle  $?a$  is stopped on lane  $?l1$  behind vehicle  $?b$  which is stopped too on the same lane; There is a continuous line  $?s$  besides lanes  $?l1$  and  $?l2$ ; If lane  $?l2$  is clear from other vehicles and the time out has expired (i.e., the internal

state of vehicle *?a* is labeled “*Passing*”), then vehicle *?a* passes onto the opposite lane *?l2* and overtakes vehicle *?b*.

The previous inference rule 4 can be read from the inference rule 5 with the following equivalences:

- Lines 1 and 4 represent the illegality of the motion from lane *?l1* to opposite lane *?l2* because of the continuous line *?s* in between. This can be formulated in inference rule 4 terms as:

*Lane(?l1), Lane(?l2), CrossableZone(?s),  
hasBesides(?l1, ?s), hasBesides(?l2, ?s)  
-> isIllegal(?l1, ?l2)*

- The second term of line 2, line 3 and line 6 represent the current situation of vehicle *?a* --- hence are equivalent to the term *CurrentSituation(?a, ?s)* of inference rule 4.
- Lines 7 and 8 (elapsed time out and clearness of the opposite lane) represent the generic condition C of inference rule 4.

### 3 Experiments

The ontology and inference rules are tested on simulated cases, we describe 2 of them as illustration. The examples are computed together by the reasoner PELLET in 499ms on a 4-core CPU at 2GHz using 4Gb RAM.

#### 3.1 Case 1 of Fig. 1

Here are the properties inferred for the individual *CyberCar1* before individual *Truck1*, which is stopped on individual *Lane1* separated by opposite individual *Lane2* by a continuous line:

*isNextOn(Lane2)  
hasNextMotion(Forward)  
hasMotion(Stopped)  
iaAfter(Truck1)*

The latter two properties correspond to the current time step, inference rules (other than 5 and 6) make completion of the current situation. They state that, since the individual *Truck1* is stopped on individual *Lane1*, and since the individual *CyberCar1* is before individual *Truck1*, then the inverse property holds (the property *isAfter* is the inverse of the property *isBefore*) and the current motion of the individual *CyberCar1* is *Stopped* (avoiding collision with individual *Truck1*). The former two properties are inferred by the rule 5 above, and state that the individual *CyberCar1* will break the



traffic regulation at the next time step and overtake the individual *Truck1* by passing onto individual *Lane2*.

### 3.2 Case 3 of Fig. 1

As above, an individual *CyberCar2* is before an individual *Truck2*, on an individual *Lane3*, which current motion is individual *Stopped*. But this time, there is no opposite individual *Lane4* adjacent to *Lane3*, but an individual *Sidewalk3* along it. Here are the properties inferred for the individual *CyberCar2*:

```
isNextOn(Sidewalk3)
hasNextMotion(Forward)
hasMotion(Stopped)
isAfter(Truck2)
```

The latter two properties result from inference rules for completion of the current situation and inverse properties, as above. But this time there is no continuous line along individual *Lane3* but a sidewalk: the individual *CyberCar2* will move on it next, due to the inference rule 6 above, hence will perform an illegal motion because of the unusual aspect of the situation at hand. As previously, changing the property *has-InternalState* of individual *CyberCar2* to individual *Waiting* instead of individual *Passing* would have cancelled the former two inferred properties (time out not expired yet, leading the individual *CyberCar2* to keep waiting behind individual *Truck2* at the next time step).

Reasoners not only spend time on making inferences but also on checking class consistency (making sure that every class can actually have individuals, i.e., there are no inconsistent static constraints on classes). Other DL reasoners (FACT++, RACERPRO, etc) may be compared to PELLET to improve performances through software engineering (e.g., see benchmarks in [4]).

## 4 Related work

Introducing a reasoning operation into the perception / planning / control cycle of Robotics has led to several software architectures, e.g., [1] [5] [9] [12]. A notion of “levels” is proposed, but authors disagree on their roles: one (slower) level subsumes the lower (faster) one [5], 3 levels with specific roles [1] [9], or 2 levels with no intermediate control level [12]. However, in our practical cases, the reasoning operation provides a result quickly enough (see section 3) to be integrated into the perception / planning / control cycle, so none of these software architectures is needed so far. (In software architectures terms, our model requires a single level only.)

Vehicular ad hoc networks (VANET) are used for spatially distributing traffic in a network [16] [17]. Traffic density predictions can be computed using a state-automaton describing every vehicle’s decision [17]. Each road is represented as a queue (with a maximum value denoting congestion) and each vehicle is represented

as a packet in a queue; Then a law (e.g., adaptive proportional integral controller [16]) controls the traffic volume of each queue, for lesser congestion to be globally observed [16]. However, in these approaches, the decisional autonomy of each vehicle is limited, e.g., reduced to a state automaton in the best case [17]. We believe that embedding an ontology with inference rules inside each automated vehicle leads to a higher decisional power (at a higher computational cost, though), as exhibited by the inference rules presented above, which lead to a “smarter” behavior of the automated vehicle in the end (after all, even the most experienced human driver can be mathematically reduced to the instantaneous speed vector and location of his vehicle).

More interestingly, other approaches take advantage of ontologies to represent higher forms of reasoning inside automated vehicles [3] [14] [20]. Intersections and their surrounding infrastructure are represented in an ontology, and inference rules are defined in order to determine which vehicle can pass through the intersection according to traffic regulation (e.g., yielding right of way) [14] [20]. Or, an ontology is embedded into every vehicle in order to decide on its next motion, as a reaction to an incoming emergency vehicle (e.g., freeing the leftmost lane on highways) [3]. However, even if these approaches are based on an ontology and on inference rules (as is ours), their goal is to respect traffic regulation (even for [3]), which may lead in practice to blocked situations, examples of which are described in section 1. Our approach aims at sometimes relaxing them due to practical reasons (elucidating these reasons is of interest to us, this is what we call the unusual aspect of situations). A point common to these approaches, though, seems to be the reasonable performances in simulation, but only for limited representation of the environment of the automated vehicle (our approach confirms this point).

## 5 Conclusion

If safety dictates that traffic regulation must be respected by all vehicles, either human- or computer-driven, real traffic in real environments may exhibit practical cases in which breaking traffic rules might be tolerated by a vehicle, because these cases are indeed unusual (i.e., not planned for by traffic regulation). Facing this, we have presented a discrete, high-level, symbolic model (aimed at being embedded inside an automated vehicle) to decide on what to do next for an automated vehicle in such unusual but practical situations. This model is based on an ontology, the ABoxes of which represent a snapshot of the traffic environment, and on inference rules, representing the reasoning performed by the automated vehicle, to decide on breaking/respecting traffic regulation. These inference rules all derive from a generic rule for traffic regulation respect (noted 3 above) and from a generic exception rule for traffic regulation relaxation (noted 4 above). An implementation using the OWL language with the ontology editor PROTÉGÉ [19] and the inference rule model SWRL [13] with the PELLET reasoner exhibits attractive performances in simulation.

These performances (reasoning on a limited environment representation does not seem to doom perception and control) lead our future work towards porting this high-level symbolic model to the CyberCar platforms [5], by connecting percepts and

goal locations to our ontology's ABoxes, for testing on real roadways (unusual situations can be created in the private lanes of our research center) .

## 6 References

- [1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, F. Ingrand. An architecture for autonomy. *International Journal of Robotic Research*, 17(4), pp.315-337, 1998.
- [2] E. Bailly. Situation awareness of drivers : fondamental aspects, methods and application to driver training (« Conscience de la situation des conducteurs : aspects fondamentaux, méthodes et application pour la formation des conducteurs »). *Ph.D. thesis, Laboratoire d'Etude et d'Analyse de la Cognition et des Modèles*, Lyon 2 University, 361 pages, Dec. 2004.
- [3] A.J. Bermejo, J. Villadangos, J. J. Astrain, A. Cordoba. Ontology Based Road Traffic Management. *Intelligent Distributed Computing VI*, G. Fortino et al. eds., SCI 446, pp. 103-108.
- [4] J. Bock, P. Haase, Q. Ji, R. Volz. Benchmarking OWL reasoners. *Workshop on Advanced Reasoning on the Web: Scalability and Common Sense (Area'08)*, June 2008.
- [5] R. A. Brooks. *A Robust Layered Control System for a Mobile Robot*. In *IEEE Journal of Robotics and Automation*, 2(1) pp. 14–23, 1986.
- [6] Convention on Road Traffic. Vienna, Austria, Nov. 1968. [http://en.wikipedia.org/wiki/Vienna\\_Convention\\_on\\_Road\\_Traffic](http://en.wikipedia.org/wiki/Vienna_Convention_on_Road_Traffic)
- [7] Cybercars project website, <http://www.cybercars.org>
- [8] French Traffic Regulation, <http://www.legifrance.gouv.fr/affichCode.do?idArticle=LEGIARTI000023095926&idSectionTA=LEGISCTA000006177121&cidTexte=LEGITEXT000006074228&dateTexte=20120717>
- [9] E. Gat. Three layer architectures. In D. Kortenkamp et al. eds, *A.I. and Mobile Robots*, AAAI Press, 1998.
- [10] M. Ghallab, D. Nau, P. Traverso. *Automated Planning : Theory and Practice*. Morgan Kaufmann, Elsevier, San Francisco, CA, 2004, 635 pages.
- [11] T. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199-220, 1993.
- [12] B. Hayes-Roth, K. Pfleger, P. Lalanda, P. Morignot, M. Balabanovic. A domain specific software architecture for adaptive intelligent systems. In *IEEE Transactions on Software Engineering*, 4(21):288-301, 1995.
- [13] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, *Submission to W3C*, May 2004 <http://www.w3.org/Submission/SWRL/>
- [14] M. Hulsen, M. Zollner, C. Weiss. Traffic Intersection Situation Description Ontology for Advanced Driver Assistance. *IEEE Intelligent Vehicle Symposium (IV)*, Baden-Baden, Germany, June 5-9, 2011.
- [15] C. Laugeau. The Century of Intelligent Car ("Le siècle de la voiture intelligente"). *Mines Press*, Paris, France, Nov. 2009, 252 pages.
- [16] B. K. Mohandas, R. Liscano, O. W. W. Yang. Vehicle Traffic Congestion Management in Vehicular ad-hoc Networks. 3er IEEE LCN Workshop on User Mobility and Vehicular Networks (ON-MOVE 2009), Zürich, Switzerland, 20-23 October 2009.
- [17] G. H. Mohimani, F. Ashtiani, A. Javanmard, M. Hamdi. Mobility Modeling, Spatial Traffic Distribution, and Probability of Connectivity for Sparse and Dense Vehicular Ad Hoc Networks. *IEEE Trans. On Vehicular Technology*, (58):4, 1998-2007, May 2009.
- [18] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haenel, T. Hilden, G. Haffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowsky, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, S. Thrun. Junior: The Stanford Entry in the Urban Challenge. *Journal of Field Robotics*. 25(9):569-597, 2008.
- [19] Protégé project website, <http://protege.stanford.edu>
- [20] R. Regele. Using Ontology-based Traffic Models for more efficient Decision Making of Autonomous Vehicles. *Fourth International Conference on Autonomic and Autonomous Systems (ICAS)*, pp. 94-99, 2008.
- [21] SWOOP project website, Univ. of Maryland, MD. <http://code.google.com/p/swoop/>