Modeling and Using the Context of Navigation: Towards Context-Aware Navigation of Autonomous Vehicles

Sélim Chefchaouni Moussaoui[1], Alessandro Corrêa Victorino[1,2] and Marie-Hélène Abel[1,2]

***Abstract*— The problem of the autonomous navigation of intelligent vehicles has been studied for years now, and nowadays efficient sensing-based controllers have been developed and are even included by autonomous car manufacturers. However, these controllers do not take into account the contextual information from the vehicle's environment. Recently, a few studies tried to consider the possibility of considering this information in control laws, in order to adapt the vehicle's behavior to each different situation it may be in. In this paper, we develop a method to model a context of navigation which provides information on both local and global navigation, using ontologies and reasoners. Then we explain how this information can be handled in local and global control laws. Finally, we illustrate our method with a small-sized ontology of the context of navigation, and its integration to be taken into account by a local driving controller.**

I. Introduction

The problem of autonomous navigation has been studied for years now. Nowadays many functioning solutions for it have been proposed, and efficient sensor-based control can be achieved. Efficient automated driving stacks can now perform both local driving tasks (e.g. going through an intersection or avoiding an obstacle) using sensor information from cameras and lidars, and global tasks (e.g. following a desired itinerary) using global information sources like maps and localization sensors. These automated driving solutions can perform an efficient autonomous navigation in many scenarios, and some of them are even included by autonomous car manufacturers. In this paper, we will refer to the components of these driving stacks as controllers – even though it is generally used to describe only components sending commands to actuators – to insist on the fact that our methodology can be applied regardless of the kind of driving task we are handling.

These controllers use essentially *explicit* information from the sensors, in the sense that they use a information extracted from the sensor's raw data, e.g. features of an image or estimated position from a GNSS receiver, then consider this information directly to provide a control output. They no not use the *implicit* contextual information from their perception, as they do not *reason* over this information to obtain implicit contextual information. It follows that the vehicle's behavior does not adapt to the context, could it be the state of the passengers, the state of the road or the the vehicle's load.

We believe that taking this information into account in the control law could lead to a smarter navigation, meaning that the vehicle behavior could adapt better to the its environment.

Previous works [1] [2] have shown that ontologies are an efficient tool to model contextual information, because they can be used with reasoners that are able to deduce the implicit information from the perception to form the context, and provide driving suggestion to the controller. The same works have shown the possibility to take these suggestions into account in a visual-servoing based controller. More recently, these works have been extended to allow the driving suggestions to be updated in real-time [3].

In this paper, we want to propose a general methodology for modeling the context of navigation and considering in the control law, based on these previous works. The methodology aims to be generic enough to be used to add context awareness when solving both local and global navigation problems.

The paper will proceed as follows. First, we propose a brief state of the art on context aware navigation in Sec. II. Then, we propose a generic method to model the context of navigation of intelligent vehicles, and propose a small-sized ontology-based model in Sec. III. We then develop a methodology for handling the modeled context in local and global navigation systems in Sec. IV. Finally, Sec. V is dedicated to the tests and results obtained in simulation and is followed by a conclusion in Sec. VI.

II. Related Work

In the past, some solutions have been proposed for representing the context of navigation with ontologies, and were mainly focused on the use of the context in ADAS systems. Zhao et al. proposed an ontology-based system to prevent the driver from exceeding the speed limit [4] or to help him in intersections [5]. Their model was based on three different ontologies taking into account information on different levels: the Map Ontology, the Control Ontology and the Car Ontology.

More recently, Armand et al. in [6] modeled the inference rules for reasoners programs reasoning over the ontology-based model and thus propose driving suggestions, in the context of ADAS systems. This principle was reused by Faruffini et al. to use it in autonomous navigation systems, not only ADAS. They proposed a solution to take into account these reasoner suggestions directly in the control law [1] [2], in the case of a visual-servoing based controller.

[1]Computer Science Department, University of Technology of Compiègne, Compiègne, France

[2]Heudiasyc (Heuristic and Diagnostic of Complex Systems) Laboratory, UMR CNRS 7253, Compiègne, France

E-mails:
selim.chefchaouni-moussaoui@etu.utc.fr,
alessandro.victorino@hds.utc.fr,
marie-helene.abel@hds.utc.fr

More recently, these works have been extended to update the context dynamically [3]. It has been demonstrated on driving simulator and showed good results, but is limited to this controller to handle local navigation problems.

In the scope of global navigation systems, several approaches have been discussed too. It was proposed to use ontologies to model contextual information that can infer on the global navigation, like information on road entities (like type, condition, etc…) [7], on the geometry of the road intersections, [8], or on traffic [9]. The inference between the information modeled in the ontology and the navigation is not part of all these papers, however M. Hülsen et al. demonstrated the consideration of their ontology – which models traffic situations at road intersections – on driving simulation [10] [11] and provided interesting reasoning results.

We also note that some works have been done regarding the filling of the ontological model from the vehicle's perception. On the particular case of visual information, R. Drouilly et al. proposed a semantic representation for geometric contextual information [12] [13]. This kind of representation can be used to fill an ontology-based model from sensor data. This part is out of the scope of this paper, but is an important point in context-aware navigation, and thus should be mentioned.

From this state of the art, we will propose a generic framework for context-aware autonomous driving, able to infer in either global and local navigation law, and a related ontology for the context model. In this paper, the developed framework will focus on local navigation by taking into account the dynamic evolution of the navigation context.

III. Modeling the Context

In this section, we detail how ontologies can be used to model the context of navigation, and give the example of our small-sized model developed for this work.

A. Ontologies

First, we need to recall what are ontologies and how they can be useful to model the context of navigation.

In computer science, an ontology can be defined as a representation, formal naming, and definition of the categories, properties, and relations between the concepts, data, and entities that substantiate a domain. This kind of representation has been much developed with semantic web technologies, and many tools and standards have been developed for this purpose. Ontologies have also been used in other fields, mainly related to AI, and more recently, their use for robotic applications has been considered.

The main advantage of ontologies, compared to other data representations, is the possibility to use them with formal reasoners, that are software able to operate logical deductions from a given ontology, and thus provide more information. In our work, this is what will be exploited to understand the implicit, contextual information, by providing driving suggestions to our controller.

A ontology-based model is composed of classes corresponding to the semantic concepts, and properties that are the links between the classes (object properties) or between classes and numerical data (data properties). Together, these abstract definitions are forming the *Terminological Box* (TBox). From these definitions, we can define individuals that are equivalent to the class instances. These individuals constitute what we call the *Assertional Box* (ABox). Together, TBox and ABox form a *Knowledge Base* (KB).

B. The context of navigation model

The context of navigation can be modeled with ontologies as stated before. For this, several tools – originally developed for semantic web technologies – can be used to model and manage the ontologies. Among them, we will use the Protege [14] as a front-end ontology editor, the *Ontology Web Language* (OWL) as a back-end language for describing the ontology, the Pellet reasoner [15] for logical reasoning over the ontology, and the owlready2 Python library [16] to use the ontology and the reasoner in our programs.

Since we have this tools, it is possible to use them for creating the model of our context of navigation. This model should include elements that describe both local information, i.e. the vehicle and its local environment, and global information like itineraries. As local and global information are not strictly separated, we believe it is better to include them in a single knowledge base. The following sections detail the process of modeling local and global information.

C. Modeling the local context

The local context of navigation could be defined as the contextual information related to the vehicle itself and its local environment. In this definition, we include information on:

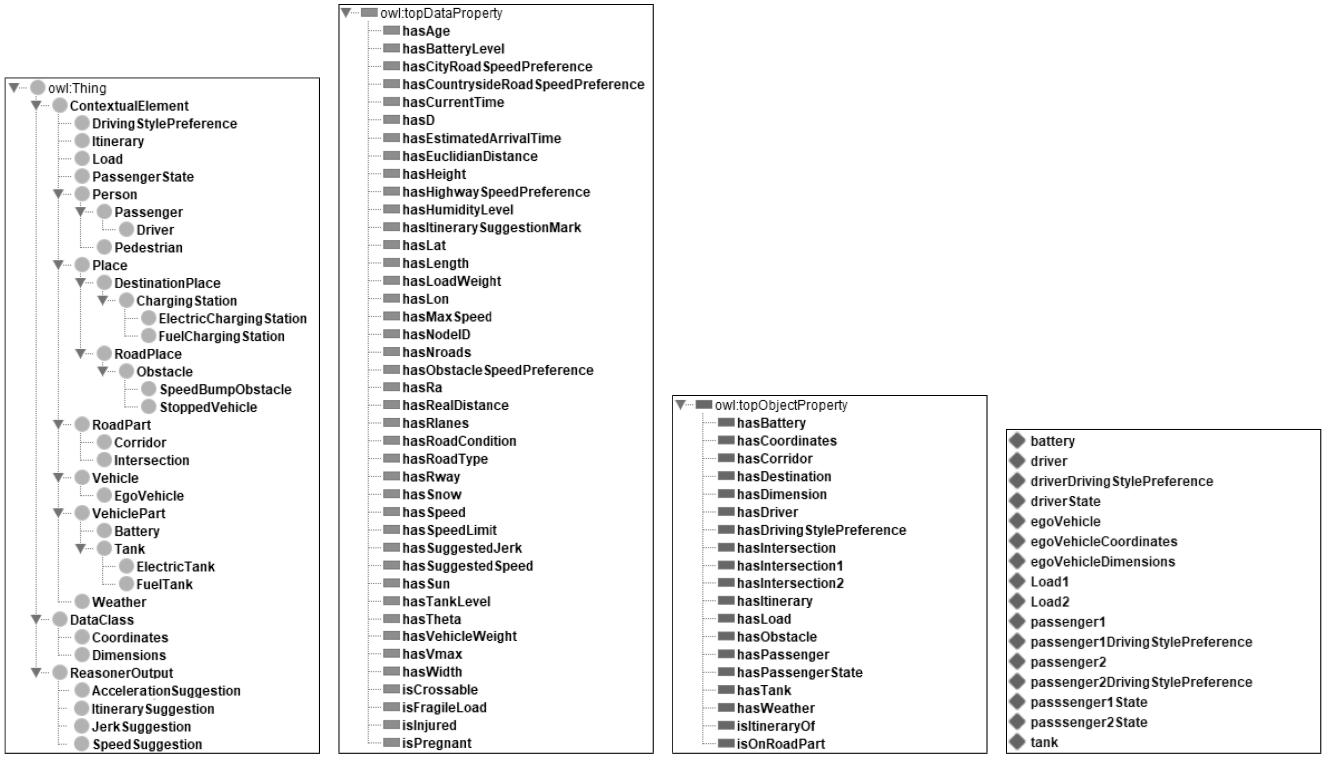
- the *vehicle* itself: its load, its fuel, its battery level, etc…
- the *passengers*: the number of passengers, their state (if a passenger is injured for example), their driving preferences, etc…
- the *road*: information on the road given by its perception (from sensors like camera or lidar), for example the presence of an obstacle or a pedestrian, the presence of a stop, the state of the road, etc…

We can see that the information on the vehicle and passengers are mainly static properties, that do not change when the car is driving or drived. However, information from the perception changes over the time. This implies to take real-time considerations into account when implementing a context-aware driving controller.

Together, the information listed above form what we call the *local context*. We have chosen to consider it together as they all have consequences on the same variables for controlling the vehicle: its speed, acceleration, and jerk. We will see after that information coming from the local context can be used to provide suggested speed, acceleration and jerk for the vehicle.

D. Modeling the global context

The *global context* can be defined by opposition with the local context, as a model of the information that is not in the local vehicle environment. It includes information on:



(a) Classes (b) Data properties (c) Object properties (d) Individuals

Fig. 1: The OWL ontology of the navigation context shown in Protege. The OWL file created in Protege handles the classes (a), data properties (b), object properties (c), and individuals at the initialization (d).

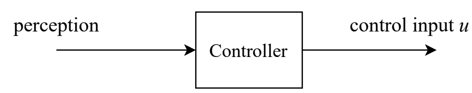
- the vehicle's *localization*: global localization given from the vehicle's GNSS receiver.
- the vehicle's *itinerary*: the point the vehicle needs to reach and the itineraries the vehicle can follow to reach it.

A representation for the itineraries have been proposed by I-F. Kenmogne et al. [17] and can be easily included in an ontology-based model, as shown in [3]. We will keep this representation for our global context model.

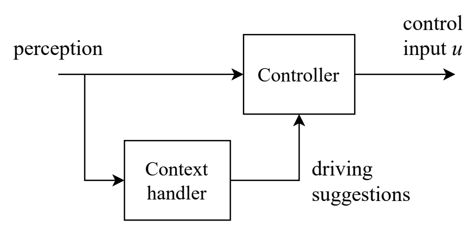
As the local context was grouped to match the same suggestion variables, the information from the global context infer on the same variables too. In that case, the suggested variable is the itinerary, that is handled by a high-level itinerary planner, that provides new objectives for the low-level controller.

E. Final model

We defined a small-sized ontology model using the elements from the local and global context described before. For the local context, we decided to stuck to mainly static elements for our example, for demonstration purposes. The ontology has been written with Protege, and is displayed in Fig. 1. In the list of classes, we can see that in addition to our context, some classes are defined to represent the reasoner's suggestions described before: speed, acceleration, jerk, and itinerary.



(a) Standard approach



(b) Context-aware approach

Fig. 2: Difference between the standard approach and the context-aware approach. The context handler module adds context awareness to the controller by providing suggestions based on the context model, populated with the vehicle's perception.

IV. Using the Context

After we modeled both local and global information in our ontology, we will explain how this information can be used to improve a controller.

A. Principle

As said in the previous section, contextual information is provided to the controller using suggestions on input variables for the controller: speed, acceleration, jerk, and itinerary. The difference between the standard approach and the context-aware approach is displayed in Fig. 2.

From this figure, we can distinct two steps for using the contextual information in our controller, detailed in the following paragraphs:

1) Using the reasoner to provide suggestions
2) Using the suggestions in the control law

B. Providing suggestions

The context modeled in the previous section is a logical semantic structure, thus it is possible to have a logical reasoning on it. For this purpose, we need to define a set of rules, using the *Semantic Web Rule Language* (SWRL) language. These rules will be used by the reasoner – Pellet for instance – to infer new information on our individuals. For example, the following rule:

$$\begin{aligned}
&\text{EgoVehicle}(?v) \wedge \\
&\text{SpeedSuggestion}(?sg) \wedge \\
&\text{hasPassenger}(?v, ?p) \wedge \\
&\text{hasDrivingStylePreference}(?p, ?dsp) \wedge \\
&\text{hasRoadType}(?rp, \text{"CityRoad"}) \wedge \\
&\text{hasCityRoadSpeedPreference}(?sp) \wedge \\
&\text{isOnRoadPart}(?v, ?rp) \\
&\rightarrow \text{hasSuggestedSpeed}(?sg, ?sp)
\end{aligned}$$

aims to provide suggested speed w.r.t. the passenger preferences.

Thus, we need to define a set of rules that have an impact on the suggested speed, acceleration, jerk or itinerary. These rules will be used by the reasoner to provide a suggestion only based on the contextual information. This last point is important, and means that the reasoner does not considers the vehicle's dynamic as the controller does. This is the reason why we need to modify the controller to make it consider both the dynamic and the contextual suggestions.

C. Considering the suggestions in the navigation task

Now, we will see how the suggestions can be considered in the control law. A control law can be generally represented as the solution to an optimization problem on one or several input variables, even if it is not its main formulation.

Let us denote $\mathbf{u}$ the input vector, which can represent for example the longitudinal and angular speed v and ω, but also other inputs as the torque or the steering angle. The control law can be expressed as an optimization problem under the form:

$$\mathbf{u}_{\text{opt}} = \arg\max_{\mathbf{u}} \; [\, f(\mathbf{u})\,] \tag{1}$$

To take the context into account in this problem, we need to add a function c representing the contextual suggestions to our contextual suggestion:

$$\mathbf{u}_{\text{opt}} = \arg\max_{\mathbf{u}} \; [\, \alpha f(\mathbf{u}) + (1-\alpha) c(\mathbf{u})\,] \tag{2}$$

where $\alpha \in [0, 1]$ allows to modulate how much the context is considered.

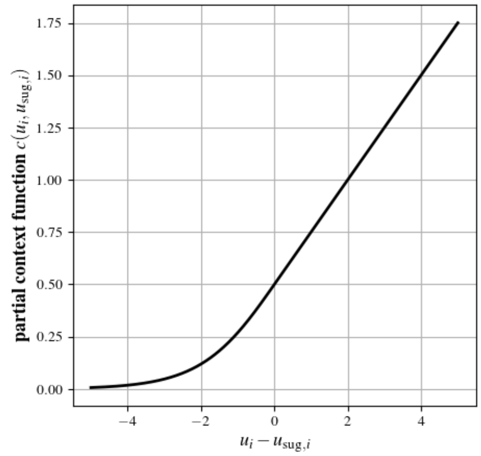


Fig. 3: The partial context function $c_i(u_i, u_{\text{sug},i})$ from (4).

The context function c can be defined using the input suggestion, denoted $\mathbf{u}_{\text{sug}}$ hereafter, and the real control input $\mathbf{u}$. From now, we will denote $\mathbf{u} = [u_1, u_2, \ldots u_n]^T$ and $\mathbf{u}_{\text{sug}} = [u_{\text{sug},1}, u_{\text{sug},2}, \ldots u_{\text{sug},n}]^T$. From this, we define the context function c as:

$$c(\mathbf{u}) = \sum_{i=0}^{n} w_i \, c_i(u_i, u_{\text{sug},i}) \tag{3}$$

where the w_i are weights applied to the partial context functions c_i.

The partial context functions c_1 to c_n are functions that should penalize violations of the suggestion. As they the idea is to solve an optimization problem with these functions, they should be continuous, derivable and convex, thus we can use a gradient descent algorithm. We propose the following function, but other functions with the same properties convene too, depending on the optimization problem.

$$c_i(u_i, u_{\text{sug},i}) = \begin{cases} \dfrac{1}{1 + e^{-(u_i - u_{\text{sug},i})}} & \text{if } u_i \leq u_{\text{sug},i} \\ \dfrac{1}{2} + \dfrac{1}{4}(u_i - u_{\text{sug},i}) & \text{otherwise} \end{cases} \tag{4}$$

The aspect of the proposed partial context function is displayed in Fig. 3.

The first controller we developed using this method is the *Image and Context based Dynamic Window Approach* (ICDWA), a visual-servoing based controller with context awareness. It uses $\mathbf{u} = [v, \omega]^T$ (longitudinal and angular speed) as input, and the following objective function:

$$\begin{aligned} f(\mathbf{u}) = \alpha \cdot \text{heading}(\mathbf{u}) + \beta \cdot \text{dist}(\mathbf{u}) \, + \\ \gamma \cdot \text{velocity}(\mathbf{u}) + \delta \cdot c(\mathbf{u}) \end{aligned} \tag{5}$$

where the heading function is based on the final orientation of the robot to lead it to the final goal position, the dist function prioritizes movement in areas free of obstacles, the velocity function prioritizes longitudinal speeds close to the desired constant set point, and c is the context function proposed before.

This approach we developed for using the context is not limited to ICDWA and local controllers, but could also be used to add context awareness in global navigation problems, if they can be expressed as optimization problems. For example, we could consider metrics on an itinerary as a control input: $\mathbf{u} = [\text{distance}, \text{time}, \text{price}]^T$, the same method can be used on to add context awareness on a minimization problem on these metrics.

V. Validation Tests and Results

This section aims to show the tests done to validate our method. Test focus on a specific application of our methodology: the ICDWA.

A. Test scenario

The tested scenario is the following: we consider an intelligent car, driving in a city (so the speed limitation is 50km/h), on a straight road, with an obstacle at 150m in front of it in its lane that should be avoided.

The vehicle is simulated in SCANeR Studio and is commanded by our ICDWA controller, which communicates with the simulator via ROS. This implementation of ICDWA uses the lane detector by A. Shour et al. [18] to detect the lanes and thus be able to compute its objective function.

The knowledge base of the context of navigation is initialized from a configuration file, which states there is a single passenger, with the following speed preferences:

- 110 km/h on highways
- 70 km/h on countryside roads
- 40 km/h on cities
- 5 km/h when crossing an obstacle

The example SWRL rule presented in Sec. IV will be used to consider the passenger's speed preferences.

As the scenario alternates between driving on a city and crossing an obstacle, we can see how the controller adapts to the context when crossing the obstacle.

B. Software implementation

For testing this scenario, we use a ROS implementation of the ICDWA controller, developed in [1]. The scenario is tested on the professional driving simulator SCANeR Studio (ver. 2022.1) that provides a bridge with ROS to control the simulated vehicles with ROS-based controllers.

The context handler module is implemented with the controller. It uses the owlready2 Python library to manage the ontology. The ontology used is the one developed in Sec. III, stored as an OWL file.

C. Results

The results for our scenario are displayed in Fig. 4. We can see that when driving in the city and avoiding the obstacle, the vehicle's speed is adapted from the passenger's preferences. We used varying values for the coefficient δ of the ICDWA function, corresponding to the coefficient $(1-\alpha)$ in (2).

The velocity profiles presented in Fig. 4 show that the passenger's speed preference is considered when the vehicle crosses the obstacle, more or less according to the coefficient δ. The higher δ is, the more consideration is given to the context. A compromise has to be found between a to high importance given to the contextual information, and no context consideration. This study is being performed in our going works.

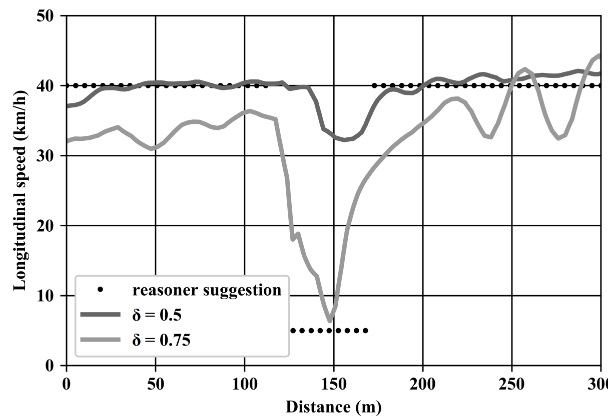


Fig. 4: Velocity profile obtained for the vehicle with our test scenario, with varying coefficient δ. The velocity is reduced by taking into account the passenger preference.

D. Discussion

This scenario shows how contextual information can inferred to the vehicle's control law in the case of the ICDWA controller, using a context model and a context-aware controller developed with our methodology. It demonstrates this methodology in the case of ICDWA and could be extended to larger context models and other controllers.

During the tests, we also verified the response time of the reasoner, and we found 25ms for our scenario, on our test computer. This far below the time needed to cross the obstacle in our scenario. However, we keep in mind that this response time is the one for a small context with a few SWRL rules, and that the response time of a reasoner like Pellet can heavily increase if the ontology is huge, and this point has to be considered when developing a complete model for the context of navigation.

VI. Conclusion and Perspectives

In this paper, we presented a methodology to model the context of navigation of an intelligent vehicle, and use it in autonomous navigation systems, can they be controllers or itinerary planners. This aims to lead to a smarter navigation of autonomous vehicles, as it takes into account implicit information in addition to the vehicle's perception. The method developed aims to be generic enough to be used on different controllers and navigation systems, and has been tested on a visual-servoing controller, the ICDWA, with a driving simulator.

In further works, a much wider ontology of the context of navigation could be developed, and a special attention should be given to consider the use of the reasoners in real-time. In particular, the use of other reasoners like HermiT could be considered, and the performances of the Pellet and HermiT reasoners could be compared in terms of response time and suggestion based on the contextual information. Also, the coefficients defined in our method should be fined-tuned, by choosing wisely at which point the contextual suggestions should be considered, keeping in mind the safety norms and considerations to the navigation integrity. These points are essential for the development of context-aware navigation systems in the future.

REFERENCES

[1] F. Faruffini, A. Corrêa Victorino, and M.-H. Abel, "Vehicle Autonomous Navigation with Context Awareness," in *2nd IEEE International Conference on Human-Machine Systems ((ICHMS 2021).* Magdeburg, Germany: IEEE, Sep. 2021, pp. 1–4. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03523227

[2] F. Faruffini, H. Pousseur, A. Corrêa Victorino, and M.-H. Abel, "Context Modelling applied to the Intelligent Vehicle Navigation," in *47th Annual Conference of the IEEE Industrial Electronics Society (IECON 2021).* Toronto, Canada: IEEE, Oct. 2021, pp. 1–6. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03523249

[3] S. Chefchaouni Moussaoui, H. Pousseur, A. Corrêa Victorino, and M.-H. Abel, "Dynamic Context Awareness in Autonomous Navigation," *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2022)*, Oct. 2022 (preprint).

[4] L. Zhao, R. Ichise, S. Mita, and Y. Sasaki, "An ontology-based intelligent speed adaptation system for autonomous cars," in *The 4th Joint International Semantic Technology Conference*, 11 2014.

[5] L. Zhao, R. Ichise, T. Yoshikawa, T. Naito, T. Kakinami, and Y. Sasaki, "Ontology-based decision making on uncontrolled intersections and narrow roads," in *2015 IEEE Intelligent Vehicles Symposium*, 06 2015.

[6] A. Armand, D. Filliat, and J. Ibanez-Guzman, "Ontology-based context awareness for driving assistance systems," 06 2014, pp. 227–233.

[7] S. Vacek, T. Gindele, J. Zollner, and R. Dillmann, "Situation classification for cognitive automobiles using case-based reasoning," in *Intelligent Vehicles Symposium.* IEEE, 2007, pp. 704–709.

[8] ——, "Situation classification for cognitive automobiles using case-based reasoning," in *Intelligent Vehicles Symposium.* IEEE, 2007, pp. 704–709.

[9] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles," 04 2008, pp. 94–99.

[10] S. Vacek, T. Gindele, J. Zollner, and R. Dillmann, "Situation classification for cognitive automobiles using case-based reasoning," in *Intelligent Vehicles Symposium.* IEEE, 2007, pp. 704–709.

[11] M. Hulsen, J. Zollner, N. Haeberlen, and C. Weiss, "Situation classification for cognitive automobiles using case-based reasoning," in *Intelligent Vehicles Symposium.* IEEE, 2007, pp. 704–709.

[12] R. Drouilly, P. Rives, and B. Morisset, "Fast Hybrid Relocation in Large Scale Metric-Topologic-Semantic Map," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'14*, Chicago, United States, Sep. 2014. [Online]. Available: https://hal.inria.fr/hal-01010231

[13] ——, "Semantic Representation For Navigation In Large-Scale Environments," in *IEEE Int. Conf. on Robotics and Automation, ICRA'15*, Seattle, United States, May 2015. [Online]. Available: https://hal.inria.fr/hal-01122196

[14] M. A. Musen, "The protégé project: a look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015. [Online]. Available: https://doi.org/10.1145/2757001.2757003

[15] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007, software Engineering and the Semantic Web. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570826807000169

[16] J.-B. Lamy, "Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," in *Artificial Intelligence In Medicine*, vol. 80, 2017, pp. 11–18. [Online]. Available: http://www.lesfleursdunormal.fr/static/_downloads/article_owlready_aim_2017.pdf

[17] I.-F. Kenmogne, D. Alves De Lima, and A. Corrêa Victorino, "A Global Navigation Management Architecture Applied to Autonomous Robots in Urban Environments," in *18th IEEE International Conference on Intelligent Transportation Systems (ITSC 2015)*, Las Palmas de Gran Canaria, Spain, Sep. 2015, pp. 2109–2114. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01305749

[18] A. Shour, H. Pousseur, A. Corrêa Victorino, and V. Cherfaoui, "Shared Decision-Making Forward an Autonomous Navigation for Intelligent Vehicles," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2021).* Melbourne, Australia: IEEE, Oct. 2021, pp. 1634–1640.