Context Modelling applied to the Intelligent Vehicle Navigation

Federico Faruffini, Hugo Pousseur, Alessandro Corrêa Victorino, Marie-Hélène Abel

Context Modelling applied to the Intelligent Vehicle Navigation

Federico Faruffini, Hugo Pousseur, Alessandro Correa Victorino, Marie-Hélène Abel

***Abstract*— This paper faces the problem of intelligent vehicles in interaction with their occupants and the environment, by modelling the semantic context associated to the navigation. With a semantically modelled context, an intelligent vehicle will not only drive itself safely, but it will also be able to reason on the situation and act accordingly. To do so, it is necessary to first define the Context of Navigation, and then to set the inference rules for it, in order to enrich the robot's comprehension of the situation. In this paper we propose our definition of the Context of Navigation, based on the information that could be needed by the vehicle's controller. We split it into two components: the Dynamic Context and the Static Context. In this paper we will focus on the latter. We then model the Static Context of Navigation of the autonomous driving - for instance the passengers' driving preferences - and to make the robotic car adapt its behaviour to this new information in real time. Finally, a short practical example of our proposition is shown and discussed.**

***Index Terms*— Human Factors, Assistive Technology, Human-Machine Cooperation and Systems**

I. Introduction

The problem of autonomous navigation has been studied for years now, and since the DARPA's autonomous vehicles challenges [1] many functioning solutions for it have been proposed. State of the art techniques allow vehicles to reach their goal while performing obstacle avoidance and respecting the traffic laws. A field related to the one of autonomous navigation is that of Advanced Driving Assistance Systems (ADAS), electronic systems built to help the carman in tasks as driving and parking. Their scope is to improve safety for the passengers and the other road users by the use of different sensors and/or cameras.

Even if many intelligent vehicle models and ADAS systems were proven to solve well the problem of navigation, most of them don't take into account the surrounding context when taking decisions, or consider just a fraction of it. Instead, we could be also interested in modelling the static context, comprehending for instance the driving preferences of the passengers, the presence of fragile or sick people on board, or an emergency scenario. In these cases, the vehicle should be able to modify its behaviour to adapt to the current situation. We can use semantic representations, called ontologies, to model the context of the navigation, in order to let the car apply inference rules on it.

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex federico.faruffini@etu.utc.fr, alessandro.victorino@hds.utc.fr, marie-helene.abel@hds.utc.fr

Research study funded by Idex Sorbonne Université - Labex MS2T.

II. Problem statement

Until now, just a few studies tried to model the parts of the Context of Navigation which aren't closely related to path planning, vehicle control or obstacle avoidance. This way, unfortunately, the vehicle has a huge lack of contextual information that are instead usually taken into account by the human drivers. This leads the vehicle to behave in the same way in different contexts, resulting in a less comfortable and customizable ride.

We will hereby introduce a simple example to show the limits of current solutions and to introduce the problem we faced. In our scenario, we consider to have a pregnant woman on board as a passenger, while the vehicle is driving fully autonomously. The passenger may want it to drive in a different way to feel safer, maybe avoiding sharp breakings or taking smoother turns. However, current solutions would not make any differences in the way the car behaves during the trip. In this study we are going to propose a simple way to add to the intelligent vehicle's model new contextual information, as in our example, through the use of semantic structures. Before to illustrate our proposition, in the next section we will see some of the most interesting works on semantic modelling of the Context of Navigation.

III. Related Work

Some ontology-based solutions were proposed in ADAS, for example the ones from Zhao et al. to prevent the driver from exceeding the speed limit [2] or to help him in uncontrolled intersections [3]. The authors proposed to use 3 ontologies: the Map Ontology, the Control Ontology and the Car Ontology. The first one is used to model the routes of transportation, with their types (HighWay, OrdinaryRoad, PrivateRoad etc) and their parts (as Roundabout, OneWayLane and BusLane). The Control Ontology is used to represent the structure of the road, seen from the route planning point of view, for example with the Classes LaneSegment and IntersectionSegment. Finally, for the Car ontology they proposed to model the different vehicles (for instance Bus, SpecialVehicle and PassengerCar) and the components of a car (e.g. the Engine and the different Sensors). In Fig. 1 we can see the structure of Zhao's Car Ontology.

Armand et al. in [4] modelled the inference rules for an ADAS system with the objective to make it reason as human-likely as possible. As an example, they propose three scenarios, as in Fig. 2. If the person is far from a pedestrian crossing as in Situation 1, the car should infer that he's not likely to cross, while in the second Situation the car should infer the opposite. The authors showed how their method

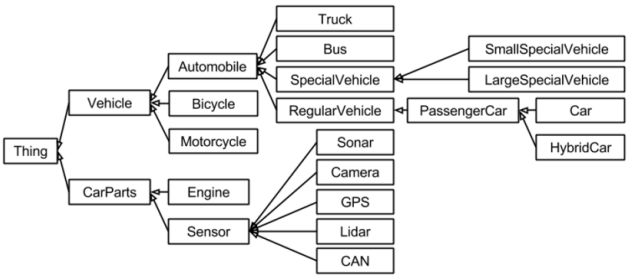

Fig. 1. The Car Ontology by Zhao et al. [2], [3]

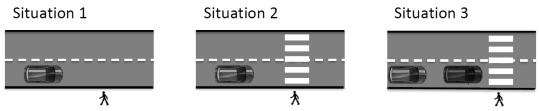

Fig. 2. The 3 situations proposed by Armand [4]

can be used to model such inference and be applied to more complex scenarios. One example is the one of Situation 3: the intelligent car perceives the leading car, the crossing and the pedestrian, and infers that the car in front is going to stop at the crossing.

This leads it to slow down in advance, resulting in a safer autonomous driving. Their ontology is easily scalable to add more features.

As for intelligent vehicle navigation, Regele in [5] proposed a simple way to create a high-level model to solve standard traffic situations as overtaking or managing complex intersections and roundabouts. At first, each lane is modelled by an arrow, leading to the creation of a graph. Then, each couple of close lanes is assigned a label: opposing, conflicting, neighbouring or nearby. This way it is possible to model simple inference rules to make the car follow right-of-way rules. For example, the "conflicting" label is assigned to merging lanes, for which the risk of an accident is high. A car entering a lane which is in conflict with another will look for cars in the second lane, and will give them right-of-way if necessary. As stated by the author, it is important to realize that, this way, the vehicle doesn't need to know the exact topology of an intersection, as from its perspective it is just a merge of conflicting lanes.

In the field of autonomous driving, different solutions have been proposed to solve the well-known problems due to GPS sensor fails. One of these solutions was studied by Lima and Victorino in [6] and was based on Visual Servoing (VS). In particular, the authors proposed to use 2 controllers, one deliberative (VS) and one reactive (IDWA - Image based Dynamic Window Approach). With that controller they showed how they were able to let a physical robotic car drive autonomously: in normal conditions the control was computed through VS, while in precence of obstacles the IDWA took over, letting the vehicle avoid them and proceed safely. With their new controller, called VS+IDWA, they managed to solve the issues related to the GPS desyncronization in urban environments. The vehicle is able to drive staying the center of the correct lane.

Finally, many studies were performed on the perception the human passengers have of the acceleration/deceleration rates and how these can affect the perceived safety of the driving, which is part of our objectives for this study. A common approach to enhance the passenger comfort consists in the minimization of the jerk (the derivative of the acceleration) acting on the passenger itself, as in [7]. Another study by Werling [8] adapted the optimal planning of autonomous vehicles in order to execute human-like trajectories, in order to obtain a more familiar driving style. This would help the passengers to feel less the presence of a robotic driver, a factor that could instead affect their comfort for the drive.

All of the studies we just presented either didn't consider the static context of navigation, or considered just a small portion of it, resulting in a loss of information to feed the model. In this paper we are going to illustrate our proposition on how to shape a semantic model of the context related to autonomous navigation and we will show a small example to verify its effectiveness. Since we too are going to use ontologies to build the context and reasoning on it, we will introduce them in the next section, as well as the reasons for their choice as modelling technology.

IV. Ontologies

According to Studer et al. in [9], an ontology is "a formal, explicit specification of a shared conceptualisation", meaning it is a way to represent a domain of interest in a way that is both understandable by humans and computers.

But why the use of an ontology to model the context of navigation? First things first, ontologies allow for reasoning on themselfs, i.e. using rules to infer new information over the asserted ones. This can be done through the use of a piece of software called a Reasoner. Secondly, as said before ontologies provide an easy-to-understand way to structure the data, shared by humans and machines. A third reason could be the advantages of ontologies over relational database management systems (DBMS) to our scope. In fact, as explained in [10], ontologies don't require many of the restrictions that databases impose, and also operate with the open world assumption, which avoids the semantic loss of using the close world assumption. Ontologies have also two other big advantages over relational DBMS: their structure can be updated in an easier way if data are already present, and there is no waste of storage space in presence of null data. In this case, in fact, a tuple having many null attributes would still use all its required storage space, while ontology triples with null values wouldn't be created in the first place, obtaining a structure in which no null information waste space. Finally, since ontologies are a part of the Semantic Web Technologies, they can also be easily shared online for later reuse.

Ontologies are expressed as a Description Logic (DL) knowledge-base, with two components: a terminological component (TBox) and an assertional one (ABox).

A. The ontology TBox

The TBox contains the definitions of the concepts in the domain of discourse, known as Classes. Classes have a hierarchical order and can have different properties. These can be of two types, object properties or data properties, and allow a Class to be linked to other classes or data attributes. Specific restrictions can be applied to properties, or relations, in the scope of the definition of the context. Classes may also be connected through rules or axioms. Rules are statements in the form if-then to be used in inference on the ontology. The Semantic Web Rule Language (SWRL) [11] allows us to define rules based on the OWL language. Axioms are assertions that, together, shape the theory of the modelled domain, and comprehend rules.

In our problem example we could use Classes for the types of person, the driving preferences and styles, the state of the passengers and so on. Regarding the properties, we could have for instance an object property relating a passenger object with her passenger state, or a data property linking her with her name. Finally, we could add a SWRL rule that sets the driving style of the vehicle as the one preferred by the passenger.

B. The ontology ABox

The ABox contains the individuals, that are the instances of the TBox Classes. In the ABox it is possible to store the real world data by linking to an individual other ones through object properties, or to simpler data through data properties.

In our problem example, we could have an instance *Emily* of class Person, which is linked to *EmilyPregnantState*, which is another individual of Class PregnantState. This is a subclass of PassengerState. The link just mentioned is expressed by an object property, let's call it *hasPassengerState*. In this case we obtain the following triple:

Emily hasPassengerState EmilyPregnantState

We will populate the TBox with real world data that are considered in the Context of Navigation.

V. The context of navigation

Many definitions for the Context can be found in literature, but most are not really useful to our scope, being too vague or too specific. Others are built for distinct use cases and cannot be applied in ours. In [12] it was eventually decided to base our definition of the Context of navigation on the following definition of Context proposed by Dey in [13]:

> Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Using this definition as a guide line, we decided that the Context of Navigation should be able to give information about the the road the vehicle is driving on, with its speed limits and regulations, as well as the right-of-way and traffic rules, that may vary based on the country and time of day. The path chosen to get to the destination should be part of the context too, as the different kinds of obstacle our vehicle may find in its trip. Finally, there is the need for information on the intelligent vehicle itself, as the maximum number of passengers, the type of carried load and information on the passengers. Examples of the latter could be the passengers' driving preferences, their current health state, the presence of fragile passengers.

We give hereby our definition of the Context of Navigation:

> The navigation context is any information that can be used to characterize the situation of navigation over a given period of time. Here, navigation is a movement considered relevant to the interaction between a driver and an application, including the driver and the applications themselves.

We distinguish between 2 components of the Context of Navigation: the Static Context and the Dynamic Context. The Dynamic context contains all the information that can change with respect to the vehicle during the navigation. The Static Context contains all the information that don't change with respect to the vehicle during the navigation.

VI. Context modelling

As previously stated, many interesting propositions on how to shape the Dynamic Context exist in literature, but few studies exist for the modelling of the Static Context of Navigation. We will therefore focus just on this part and on how to use it to obtain an autonomous driving behaviour more suited for the various preferences the passengers could have. For the sake of completeness, however, we will briefly state the other components of the Context.

A. Dynamic Context

The first part of the context contains all the external information that are not static with respect to the navigation. It consists of georeferenced data that could have an impact on the navigation. These can be of two types: the ones obtained by the Web and the ones obtained by the car's sensors. Examples of the first group could be the presence of long queues, accidents or road works. Examples of the second group may include temporary obstacles as other vehicles or pedestrians. A way to handle obstacle avoidance at control-level is the one by Lima and Victorino [6], which works by computing the free space around the obstacle and the best linear and angular speeds that allow the car to avoid it, if any. Another way, which is ontology-based, was proposed by Schlenoff et al. in [14]. After the detection of the obstacle, its type and estimated speed are computed. These information are fed to a software that simulates the impact with the vehicle and the damage dealt to it, its passengers and its load. The output of the simulator is used by the planner to decide if to avoid the obstacle or not.

B. Static Context

This part of the context contains all its components which are static with respect to the navigation. These comprehend

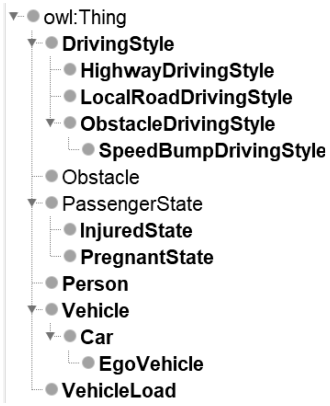

Fig. 3. The Ontology Classes in our Context

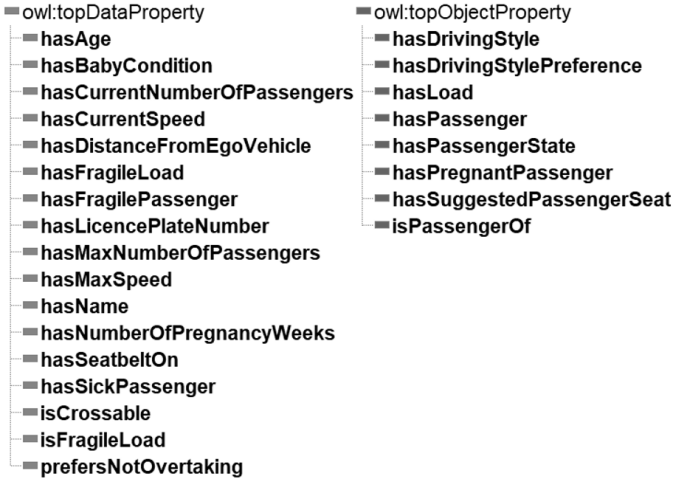

Fig. 4. The object and data properties in our ontology

external information and information regarding the car and the passengers. Examples of external information could be the max speeds allowed on the different kinds of road. The Map Ontology presented in [2] could be taken as a reference, considering that specific country regulations can be linked to these information as done by Buetchel et al. in [15]. The data could be shaped as proposed and taken from open source websites as OpenStreetMap [16], as done in many past studies as [2], [4], [17]. Another component of the Static Context comes from the car itself, its load and its passengers. It contains all the information that don't change during the current section of the trip, for instance the number of passengers, their driving preferences and the kind of load carried. It is important to notice that a vehicle's ride could be composed of many smaller trips, and the Static Context may vary between them, after a stop. For example, after some time a new passenger may enter the car. This modifies the Static Context, that however won't change until a future stop, being constant during the driving.

In our example, the pregnant passenger affects the Static Context. Given the state of her pregnancy, she could be allowed to take the car only for smaller trips or could not have any restrictions at all. Also, in this last case she could have her own driving preferences, for example a smoother and slower driving style. Our intelligent vehicle should be able to adapt to these situations for a better and safer experience of the passenger.

VII. Context building

Since in this paper we are mostly interested in the Static Context, we will assume to get reliable information from the sensors of our car, i.e. GPS, stereo cameras and lidars. We assume to have access to the precise position of car and to be able to locate precisely any other object on the path, with a $100m$ range. At first, we had to declare a Class for the reasoning autonomous vehicle, since it will have all the standard attributes and axioms of a Vehicle and some more. We call it EgoVehicle. Since the Toyota car ontology (in Fig. 1) already contains most of the information we need, we will make our EgoVehicle a kind of Car, including their ontology in our work. Then, we added a Class of the driving styles: it has many subclasses to adapt to the different roads and obstacles. For the sake of simplicity, we show just a few of them in Fig. 3. Also, classes for Person, Obstacle and VehicleLoad are created. The class PassengerState is used to add information to a Passenger: in our example, Emily will have her own Instance of PregnantState, we will call it EmilyPregnantState.

In Fig. 4 we show some of the data and object properties in our ontology. In our example, we set *EmilyPregnantState hasNumberOfPregnancyWeeks 20* and *EmilyPregnantState hasBabyCondition "Healthy"*. Of course, all the other properties were set as well, before starting the reasoning. The other Individuals in our ontology are an EgoVehicle, a SpeedBump (of class Obstacle), a Crystal Glasses Set (of class VehicleLoad), and the Standard and Custom driving styles for different kinds of road and for the fragile carried loads.

We then proceeded by building our SWRL rules and we will here briefly illustrate a couple of them. The first one will let the EgoVehicle infer that it is currently carrying a fragile object:

$$hasLoad(?v,?l) \wedge VehicleLoad(?l) \wedge EgoVehicle(?v) \wedge isFragileLoad(?l,true) \rightarrow hasFragileLoad(?v,true)$$

while the next one will let the EgoVehicle use the preferred LocalRoadDrivingStyle by the passenger:

$$EgoVehicle(?v) \wedge hasPassenger(?v,?p) \wedge Person(?p) \wedge hasLocalRoadDrivingPreference(?p,?pref) \rightarrow hasLocalRoadDrivingStyle(?v,?pref)$$

Finally, SPARQL (SPARQL Protocol and RDF Query Language [18]) queries were built: in the following example we have the update query to set a new maximum speed in real time for the ego vehicle. As we can see, the triple storing the maximum speed is firstly deleted, than reintroduced with a different value (in this case, 30), which can be set via code. In this query we can see that the information defined in the custom ontology has to be addressed with its ontology prefix, in our case *contextOntology*.

DELETE {
?egoVehicle contextOntology:hasMaxSpeed ?oldMaxSpeed}
INSERT {
?egoVehicle contextOntology:hasMaxSpeed 30}
WHERE {
?egoVehicle a contextOntology:EgoVehicle.
?egoVehicle contextOntology:hasMaxSpeed ?oldMaxSpeed.
}

VIII. Validation Test

A. Test setup

We modelled our static context in the Protégé 5.5.0 software [19] as shown in the previous section. We chose the Pellet [20] reasoner, as it is able to infer new data from SWRL rules too. The third-party software used in the simulation was SCANeR Studio 1.8[1], a professional car simulator for automotive and research. We can see in Fig. 5 a screen capture from the simulator for our test. The ontology inference was run in a Python environment using the owlready2 library [21]. The output of the interaction was then simulated too in order to obtain the behaviour of the car, through the use of scripts internal to SCANeR Studio. Anyway, in the future developements of this work we will code the missing connection.

In the simulated scenario we have an intelligent car, using a control algorithm as [6], which is driving on a straight road at $50km/h$ speed, and that perceives the presence of a speed bump, forward, located at $100m$ distance in its lane. This obstacle is the only object which isn't part of the static context of navigation to be present in the example. An automatic gearbox car was chosen as subject of the test. In the first test we chose to simulate the case in which

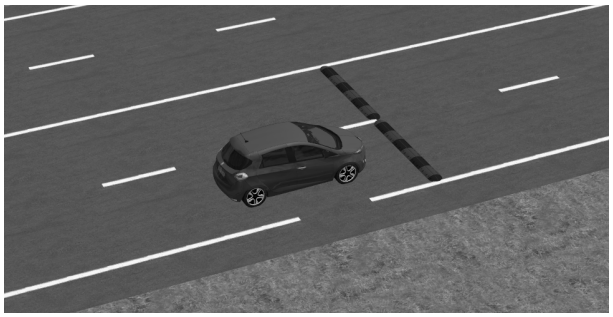

Fig. 5. The vehicle approaching the speed bump in SCANeR Studio

the car doesn't break after perceiving the bump, in order to make the proper comparisons with the following tests. In the second one we have a pregnant woman on board, whose *DrivingPreference* with respect to speed bumps is to start breaking a lot in advance, in order to have a smoother deceleration and overpassing.

In the third test, we add the information that a set of fragile crystal glasses is is being carried by the car.

[1]AVSimulation, SCANeR Studio, https://www.avsimulation.com/catalog/

B. Test results and discussion

Fig. 6 represents the speed profile of each of the three tests, and the dotted vertical line represents the position of the speed bump ($100m$ from the starting position of the car). The dark continuous line represents our first test: the car keeps a constant speed approaching the speed bump, and has a little interference when crossing it. Of course, this behaviour

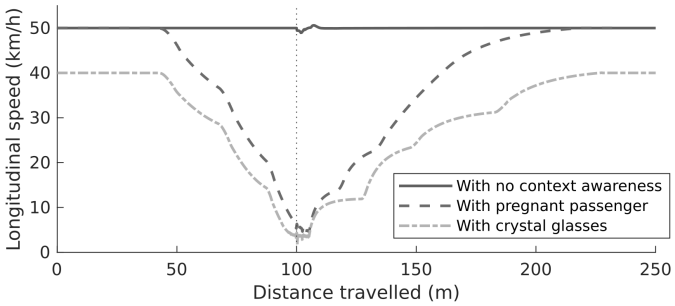

Fig. 6. The different speed profiles in the 3 tests

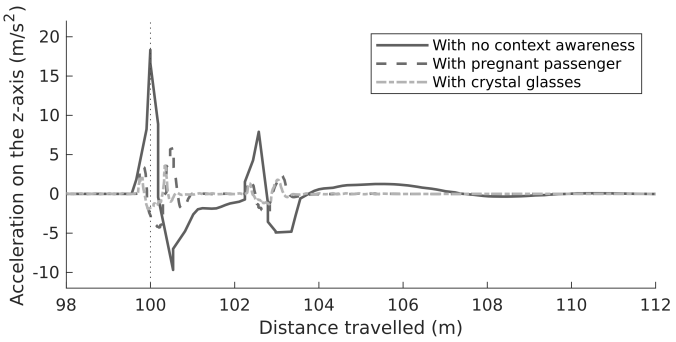

Fig. 7. The z-axis acceleration at the impact with the speed bump

isn't acceptable, as we can see from Fig. 7, representing the acceleration on the z-axis at the instant of the impact and in the following seconds. Not only the impact without context awareness isn't comfortable, but it could also be dangerous for the pregnant passenger's child. In our second test we have set the DrivingStylePreferences for Emily such that the car starts decelerating in the moment the bump is detected, with the objective to cross it at a speed close to $5km/h$. This is shown by the dashed line in Fig. 6. In the third test the SWRL rule *hasFragileLoad* is triggered, setting an even lower maximum speed in order to need a less sharp breaking in the case of the location of an obstacle. The rule also sets a specific behaviour for the car, clearly visible in Fig. 6. Firstly, its maximum speed is reduced to $40km/h$ to reduce the possibility of damaging the load with sharp breakings during the trip. For the same reason, the speed to overcome the bump is set close to $3.5m/s$. Finally, after the obstacle is surpassed, the acceleration to return to $v = 40km/h$ is slower than the one of the second case.

IX. Discussion

Our tests show how easily new contextual information can be added to the Context of Navigation to modify the behaviour of the vehicle. In this paper we showed just simple ontologies and rules, but it is possible to extend them to much more complete models with the proposed approach. For instance, we could be interested in modelling the preferred paths for the different passengers, or their preferences in overtaking. Also, emergency cases could be modelled with different levels of severity, letting the car, for instance, use the emergency lane. Finally, the most interesting addition would be to define a better and more complete interaction with the car's control loop, in order to let it implement rules more complex than if-then ones: an example could be the setting of the preferred maximum steering speed, to implement smoother turns for some passengers. This could also be applied to the case in which there is a nauseated passenger, to avoid worsening her sickness. Really different kinds of static information could be taken into account, for example during the day the roads close to schools could be semantically marked in order to increase the overall security of the road users, including pedestrians crossing the road far from a proper crossing. Also, the passenger degree of comfort could be enhanced with proper tuning of the maximum jerk, and with the introduction of it in the DrivingStylePreference. It is also to be said that the kind of car should be taken into account by the reasoner, as it could have a great impact in the final result: a bigger car normally handles impacts as the one of our example better than a littler one.

As many more examples could be thought of, it is clear how a semantic modelling of the Static Context could improve the autonomous vehicle future developments.

X. Conclusion and future work

In this paper we presented a simple way to add contextual information to the intelligent vehicle navigation, in order to improve it on the passenger side. We have run a little test on a simulator, combining the car's dynamics and an ontology-based inference system, and seen how promising the results are.

That being said, it is clear how a more complete Static Context has to be developed, and how finer interactions with the car's control loop rules must be studied. Since the navigation is a real-time task, the time performance of the system is vital to its safety, so this aspect will have to be investigated, in the case of a much bigger Static Context. Of course, this much data have to be stored in some way, and this opens to problems of ethics and security, that will have to be addressed in the future. Also, since the information coming from the context could modify the input given to the controller of the car, safety will be a key point of future research.

References

[1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[2] 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.

[3] 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.

[4] A. Armand, D. Filliat, and J. Ibañez-Guzman, "Ontology-based context awareness for driving assistance systems," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 227–233, June 2014.

[5] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles," in *Fourth International Conference on Autonomic and Autonomous Systems (ICAS'08)*, pp. 94–99, 2008.

[6] D. Alves de Lima and A. Corrêa Victorino, "A hybrid controller for vision-based navigation of autonomous vehicles in urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2310–2323, Aug 2016.

[7] M. Elbanhawi, M. Simic, and R. Jazar, "In the passenger seat: Investigating ride comfort measures in autonomous cars," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 4–17, 2015.

[8] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in *2010 IEEE International Conference on Robotics and Automation*, pp. 987–993, 2010.

[9] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: principles and methods. data knowl eng 25(1-2):161-197," *Data & Knowledge Engineering*, vol. 25, pp. 161–197, 03 1998.

[10] C. Martínez-Cruz, I. Blanco, and M. Vila, "Ontologies versus relational databases: Are they so different? a comparison," *Artificial Intelligence Review - AIR*, vol. 38, 12 2011.

[11] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "Swrl: A semantic web rule language combining owl and ruleml," w3c member submission, World Wide Web Consortium, 2004.

[12] F. Faruffini, A. Correa-Victorino, and M.-H. Abel, "Towards a semantic model of the context of navigation," in *The Fifth International Conference on Information and Knowledge Systems*, 06 2021.

[13] A. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, pp. 4–7, 02 2001.

[14] C. Schlenoff, S. Balakirsky, M. Uschold, R. Provine, and S. Smith, "Using ontologies to aid navigation planning in autonomous vehicles," *The Knowledge Engineering Review*, vol. 18, no. 3, p. 243–255, 2003.

[15] M. Buechel, G. Hinz, F. Ruehl, H. Schroth, C. Györi, and A. Knoll, "Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, 06 2017.

[16] M. Codescu, G. Horsinka, O. Kutz, T. Mossakowski, and R. Rau, "Osmonto - an ontology of openstreetmap tags," in *State of the map Europe (SOTM-EU) 2011*, 2011.

[17] E. C. M. Pereira, D. A. Lima, and A. C. Victorino, "Autonomous vehicle global navigation approach associating sensor based control and digital maps," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 2404–2409, Dec 2014.

[18] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF." W3C Recommendation, January 2008. http://www.w3.org/TR/rdf-sparql-query/.

[19] M. A. Musen, "The protégé project," *AI Matters*, vol. 1, pp. 4–12, June 2015.

[20] 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.

[21] J.-B. Lamy, "Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," *Artificial Intelligence in Medicine*, vol. 80, pp. 11–28, 2017.