Towards a Semantic Model of the Context of Navigation

Federico Faruffini[1,2(✉)], Alessandro Correa-Victorino[1], and Marie-Hélène Abel[1]

[1] Université de Technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203, Compiègne Cedex, France
federico.faruffini@etu.utc.fr,
{alessandro.victorino,marie-helene.abel}@hds.utc.fr
https://www.hds.utc.fr/

[2] Università degli Studi di Genova, Via Balbi, 5, 16126 Genoa, Italy
https://unige.it/en

Abstract. Many studies faced the problem of robot autonomous navigation in different fields, but nowadays just a few of them uses all the implicit information coming from the context in which such navigation is occurring. This results in a huge potential information loss that prevents us from adapting the robot's behaviour to each different situation it may be in. In this paper we therefore define the Context of Navigation using a semantic model built with ontologies. Then, we show what kind of inference is useful in such application. Finally, we provide a method to exploit the information coming from the context together with the control loop of the robot to better navigate it. Two application examples are provided to illustrate these concepts: a service robot and an intelligent autonomous vehicle.

Keywords: Decision support system · Navigation context modelling · Ontology · Robotics

1 Introduction

The field of robot autonomous navigation has met a great research interest since the Grand DARPA Challenges, held since 2004. Current state-of-the-art mathematical models are able to conduct global navigation autonomously, while performing a safe obstacle detection. Most of these robotic models are based on exteroceptive sensors to the robot, for instance cameras, lidars and other laser scanners, GPS systems and so on. A closely related field of research is the one of ADAS - Advanced Driving Assistance Systems - which are electronic systems built to help the driver while performing tasks as driving, overtaking, or parking.

Even if many of the existing solutions for autonomous navigation and ADAS systems show good results, a very few of them takes into account all the information coming from the semantic context in which the navigation is happening, resulting in a driving style which is not fully adaptable to the situation.

I. Saad et al. (Eds.): ICIKS 2021, LNBIP 425, pp. 168–183, 2021.
https://doi.org/10.1007/978-3-030-85977-0_13

Without taking into account the Context of Navigation, the autonomous vehicle misses many of the implicit information that usually a human driver knows, and to which adapts the way he drives. It is then clearly necessary to define a Context of Navigation and then to use the information it contains in order to change dynamically the way the robot/vehicle behaves in different situations. The application of such definition to different cases is complex as many information have to be taken into account, and also because we could have differences in the desired output of model: we can build a model more focused on reasoning or a more reactive one.

In a previous work [16] this problem was introduced, and a first theoretical approach was proposed for autonomous cars. In this work we introduce the modelling of the Context of Navigation, giving its definition and examples of its implementation. We also provide a second example to the one of the previous works, a service robot, to show how this method is applicable to any case of robotic navigation, through proper context modelling. Finally, we present the results of some tests we performed on the system.

In this work we study more in detail the modelling of the Context of Navigation, with a more complete result. We also provide a second example to the one of the previous works, a service robot, to show how this method is applicable to any case of robotic navigation, through proper context modelling.

This paper will proceed as follows: we are first going to propose some examples of problems we could solve with a semantic modelling of the context in Sect. 2, and then to give an overview of existing solutions both for autonomous navigation and driving assistance systems in Sect. 3. We then proceed to Sect. 4, which contains some of the reasons for the choice of such a model, while in Sect. 5 we provide with our definition of the Context and the different kinds of information we have to fit in it. Afterwards, in Sect. 6 we go through some of the different inference rules we can apply over our model. Our proposals for the interaction with the robot's control loop can be found in Sect. 7.

2 Problem Statement

In order to obtain a better adaptation of the robot's behaviour to the situation, we have to build a system to make it understand and exploit the contextual information during the navigation. In this paper we face the problem of enabling a robot to perform context-aware autonomous navigation. To do so, it is at first necessary to define the context in which such navigation occurs. Then, we have to define the way to reason over it and infer new information from it. Finally, we have to study a solution to interact with the robot's control loop in order to take into account the new information. We hereby present two situations of application of our work, to be used as example during this paper.

In the first example we have a service robot operating in an indoor home scenario. Its tasks are to operate many different home appliances in the house, as done by Nakamura et al. in [10], or to help the resident with other operations. This is a difficult task due to the environment and the many functions home

appliances have. The navigation of the robot in the house must be safe for the humans and itself, and must preserve the integrity of the environment.

The second example is the following: we have an autonomous robotic vehicle, which may have one or more passengers on board. Such passengers may have preferences on the way the vehicle drives: for instance, they could like it to drive at a slower pace than the maximum one allowed by the traffic rules, or to avoid roads with speed bumps to protect the integrity of a fragile load. We want to build a system which allows the car to adapt its own behaviour to these preferences, without causing traffic issues in a real application.

3 Related Works

Many mathematical models exist to handle the safe global navigation of a robot, may it be a humanoid, a service one or an autonomous car. We now go through some existing solutions for navigation and for other robotic tasks which may benefit from the use of the semantic information given by the context of navigation.

In the field of mobile robots, Reinaldo et al. in [12] proposed a solution to model the robot navigation in unknown environments. In the study, the authors developed a system the robot can use to predict the future behaviour of the recognized objects - for instance a person - to make decisions on the navigation path. In the field of service robots, Sato et al. [13] proposed a solution to let an indoors robot pick up objects that are being pointed by a human. This task is made easier by the use of some contextual information. Luperto et al. [8] developed and tested a system to lead a service robot to navigate the home of an elderly resident to provide help. The tested tasks comprehend navigating the house to locate the user to assess his health state. One of the key problems of service robots is related to the correct object detection, since in a home scenario very often the target object is partially occluded by other objects. Lim et al. [6] proposed a structure to help the robot in this task by letting it consider the contextual information.

In the field of vehicle navigation, the autonomous model usually has a component called controller, which receives as input the desired position of the car and gives as output the actual controls (acceleration, braking, rotating the wheel etc.) to reduce as much as possible the error between the desired position and the actual one. There exist different approaches to this problem, for instance some are sensor-based, relying on sensors as GPS ones to obtain the current position, while others exploit the image given by the frontal camera. One of the latter solutions, proposed by Lima and Victorino in [7], exploits the images coming from the frontal camera and the laser scans to keep the vehicle in the center of the lane while proceeding forward. In the case an obstacle is perceived, the controller computes the best couple of angular and longitudinal velocities to overtake it. Even if this model is capable of performing the global navigation fully autonomously, it doesn't take into account any information besides the ones obtained by the endoceptive and exteroceptive sensors and a few others, as the speed limit, which is set to a static constant.

Other solutions tried to incorporate a few more external information on the car's behaviour, as in the study done by Regele in [11]. He proposed to decompose the topological structure of the road into simple graph arcs, which can be assigned labels for an easier decision-making by the intelligent vehicle. Such labels allow it to understand which are the points in which different lanes converge - which are positions with a higher risk of accidents - or which lanes are allowed or forbidden, and so on. This way it is possible to reduce the computational power required for the local path planning, by feeding the model with just the correct and useful input data, deleting the excess ones. Another solution, proposed by Schlenoff et al. in [14] aimed at building a navigation planner for robotic vehicles to take into account surrounding obstacles in order to make precise predictions on the outcomes of a collision with them, regarding damage to the passengers, load and the vehicle itself. With this predictions, the planner decides if it is necessary to avoid the obstacle or not.

In the field of Advanced Driving Assistance Systems, we can see some solutions which addressed the problem of using external information to aid the performance of the system. Armand et al. in [1] proposed a method to let a car exploit the perception of a pedestrian on the side of the road to predict its behaviour in order to have a quicker reaction time in the case breaking is needed. The authors explain how, with their model, a car may actively reason over the situation: in the case of a person close to a pedestrian crossing, for instance, the car understands she is likely to cross, and decreases its own speed. Some ADAS solutions aim to make the human driving safer, as in the case of Zhao et al.: the authors proposed semantic models to help the driver navigate uncontrolled intersections [19] and to prevent him from exceeding the speed limit [18].

Even if the studies hereby presented provide good solutions for the indoor navigation, automatic driving or ADAS systems, only in [1] the context in which the navigation is happening plays an important role. Some of the other studies presented do this just partly [8,11,13,18]. Of course, this leads to the loss of many potential information that could be used to better adapt the robot's behaviour to the situation. In this paper we present the Context of Navigation and make examples on its application to service robots and autonomous navigation. Before the definition of the Context, however, it is necessary to motivate the choice of technology we made, which is represented by ontologies.

4 Why to Use a Semantic Model for This Application

An ontology is a semantic structure to store real world data with a formal representation. The most important benefit in using an ontology structure is the possibility to exploit reasoners, which are pieces of software which can operate logical deductions on the asserted knowledge and data, and provide new information over it. For this reason, an ontology-based structure fits well in our problem, as in fact many of the proposed related works in Sect. 3 already used them [1,6,11,13,14,18,19]. Also, the context can be used to obtain a better interaction with a system of systems (in this case, an example could be that of

many different vehicles on the same road): in [5] a context-aware recommender system for such a system was proposed.

An ontology-based structure works, in this situation, better than a DBMS-based one for different reasons besides the possibility of reasoning over them. For instance, they have a modular nature which allows the designer to update their structure after the detection. Also, they can be uploaded on the Web to be easily reused in other ontologies, helping in the development phase.

With a semantic model it is possible to store contextual information, as the age and preferences of each passenger of an intelligent car, the kind and fragility of the carried load, the conditions of the road, the environmental conditions and so on. However, it can still be used for storing essential physical data of our robot, as its max longitudinal and angular velocities, its suspensions performances, the kind of equipped tires, its length and so on. In order to define the context and the relative ontology we took inspiration from the methodology proposed in [14]. In this study, the authors give some suggestions on how to shape an ontology for autonomous navigation by asking some questions on the context and the extent to which we want our information to be precise. We then proceeded expanding the Context to fit all the information that are relevant to the navigation.

Some vehicle-related ontology-based standards already exist, and they are worth mentioning before moving on with the Context definition. The Vehicle Ontology[1] by Katsumi is proposed as a way to capture concepts related to vehicles. It contains many useful information related just to the vehicle and some of its data and object properties could be used in our scopes. For instance, for the first group we cite *accomodates bicycle*, *accomodates wheelchair*, *number of doors*, while for the second we have properties as *fuel consumption*, *fuel efficiency*, *drive wheel configuration*. Similar solutions can also be found in the Automotive Ontology Community Group[2] and in the Used Cars Ontology[3].

Our approach is different from the previous studies since it considers more information which are not taken into account in them. In fact, many of the ontologies proposed in the state of the art to solve the problem are somewhat similar to the ones just discussed, containing useful technical information, but just related to the vehicle, the topology of the road and/or the traffic rules. Our solution for the Context of Navigation aims at integrating this useful data with others which are missing in the previous works, as instance the ones related to the human users and their preferences.

[1] http://ontology.eil.utoronto.ca/icity/Vehicle/1.2/.
[2] https://www.w3.org/community/gao/.
[3] http://ontologies.makolab.com/uco/ns.html.

5 The Context of Navigation

In this section we illustrate our definition of the Context of Navigation and its structure. On the literature many definitions for Context have been developed, but most are too vague or too case-specific to our scopes. We found on Dey's one from [3] the most interesting:

> 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.

Following this definition, we then had to define what are the relevant entities to our application. In particular, we wanted to store information about the car, its passengers and load, the trip, the road and the other road users. Then, we gave our definition of 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.

The Context of Navigation has two components: the Dynamic Context and the Static Context. The first contains all the information that vary during the trip with respect to the Navigation. For instance, for a service robot we could have the position of people or pets in the house, while in the example of vehicle navigation we could have traffic and right-of-way rules, the position of obstacles and their nature. The Static Context of Navigation contains information which don't change with respect to the navigation. In the case of a service robots we can make examples such as the shape of the rooms in the house or the different functions of the home appliances. In the case of intelligent navigation we can add to the Static Context information on the number of passengers, their driving preferences, the type of carried load, the model and performances of the car. Since the Dynamic Context of Navigation is more related to the path planning and obstacle avoidance, and many solutions for this already exist, we currently addressed mostly the Static Context.

Having decided to model our ontology in the OWL language with the Protégé editor [9], now we go through its main classes, data properties and object properties. To be noticed, to this day only a part of the full Context of Navigation has been implemented, but future works will complete it.

5.1 Context of Navigation in Service Robots

We now take the first example we defined in Sect. 2, about the service robot, and list some of the information about the Context of its navigation it may benefit from during its navigation. At the end of this subsection, we propose an example of the result of a first ontology with the information we discussed.

Person-Related Information. Since the robot will operate inside a home scenario, the people who live there have to be correctly encoded in the context. Of course, we need a class for *Person*. The residents have many attributes which are important to shape the context, as their age, their possible illnesses, the room they sleep in etc.etera. We then propose to model object and data properties as *hasAge*, *hasName*, *hasIllness*. We may also be interested in storing a state for each dweller, for example the one for sleeping, cooking and watching tv. This could be helpful in order, for example, not to wake up with noise an elderly person who is sleeping in the afternoon. Such a behaviour could be obtained by changing the speed of the displacement in order to reduce the noise, or by postponing non-urgent tasks. Also, when the information on the state of a person isn't available, we could shape a set of reasoning rules to let the robot infer in what state she could be, as we discuss in Sect. 6.

Object-Related Information. Of course, in order to obtain a well-defined Context of Navigation for a service robot it is of great importance to have a deep and precise structure for the surrounding objects. First of all, we have to define if a perceived object is recognize as static or not, for instance a chair will be and a person won't. This can be handled with a boolean property *isStatic*. Many static objects can be obstacles to the navigation, so this information is vital to the safety of the robot and the environment. Pets represent a kind of dynamic obstacle, which could be avoided implementing some kind of behaviour prediction as proposed in [12].

Besides being static or not, an object or obstacle could also represent a direct danger for the robot or the residents. For example, a robot must correctly perceive stairs as a really dangerous obstacle. We can do it with another object property, let's call it *isDangerous*. Other members of this category could be a pot on the stove or a newly washed floor, which could be temporary slippery.

An object could be marked as interactive, if such interaction is useful to the robot's tasks. For instance, all the house appliances are interactive and could be approached as proposed by studies like [10]. Also, obstacles like doors could be marked as interactive if the robot is allowed to use them to get into another room.

Action-Related Information. A service robot operating in a home scenario is supposed to perform many tasks different in their nature, as locating and reaching the user, bringing objects to her, operating the appliances and so on. It is therefore needed a way to encode the different actions into an ontology so that we can use it to reason and ameliorate the robot's behaviour. We need a class for *Action*, and the possibility to link its instances to form a plan, for instance with the object property *hasFollowingAction*. Even if the planning operations will be performed by a planner and not by our ontology, we can still use the latter to improve the final plan taking into account the context (Fig. 1).

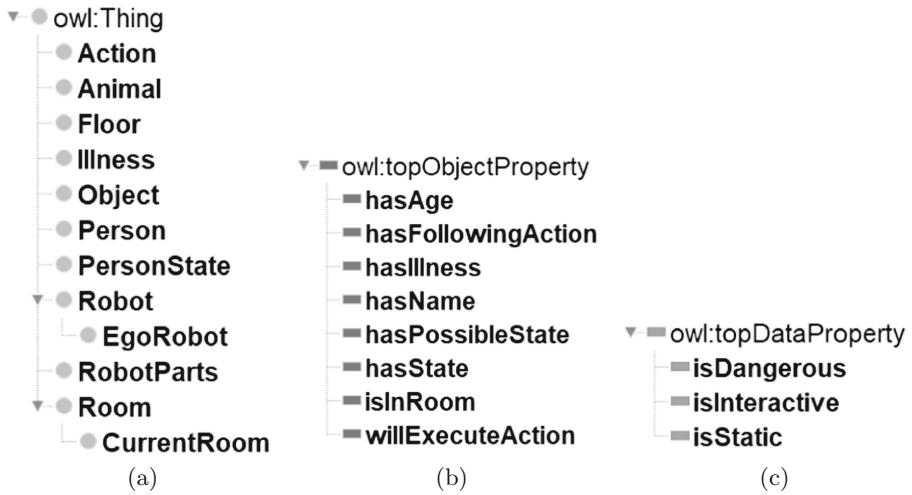

Fig. 1. The ontology for service robot navigation. (a) Classes; (b) Object Properties; (c) Data Properties

5.2 Context of Navigation in Autonomous Vehicles

We now see what are the different kinds of contextual information an intelligent vehicle can benefit from, along with our proposition on how to encode them into the Context.

Passenger-Related Information. There are many information related to the passengers that must be considered by the Context of Navigation. For sure, the passengers may have different preferences over the way the vehicle is driven, for instance some people may prefer a driving style that is slower and smoother than the standard one given by the automatic controllers, others may prefer to take specific paths even if they impact on the trip's duration. The comfort of a passenger is directly related to the derivative of the acceleration of the car, called the jerk, and this value could be different from person to person, while general suggestions and limit values already found [4]. With the definition of different preferences it will be possible to set a maximum (suggested) level of jerk for the car, for each trip. Another well-known problem in scientific literature is that of perceived safety in a robotic vehicle, since automatic models tend to drive in a different way from humans. Some studies propose to solve this issue instructing the model to execute more human-like paths, as in [17]. Again, this problem could be addressed with this type of Context. We can model OWL classes for the different situations in which we can have preferences, for instance for local roads, highways, interaction with specific objects and so on: each passenger will be able to set its own specific preference, that will be saved into an instance. We decided to have standard instances for such classes, so that if there is no preference we can drive the car normally, just following the traffic rules. An example for a

local road is the class *LocalRoadDrivingStyle*, for which we have the standard instance *Standard_LocalRoadDrivingStyle* and the one by the passenger Emily, *Emily_LocalRoadDrivingStyle*.

The state of the passenger has to be taken into account too: a pregnant passenger may have specific needs, and a badly injured passenger needs that the vehicle takes him to the hospital as quickly as possible. Also, we could go further with some of these scenarios, for example the one of the pregnant woman: we may encode in the Context not only this fact, but all the other useful information regarding such pregnancy. For instance, we could use the information on the week of the pregnancy in order to understand if it is a safe period for travelling, the information on the healthyness of the mother, the one of the child and so on. The passengers could have preferences on the way to handle specific actions, as obstacle avoidance and overtaking.

Road-Related Information. Of course, in order to have a safe and legal driving style, it is necessary to encode information on the kind of road the vehicle is currently on. This will let it know the maximum allowed speed value, the possibility to switch to other lanes, the existing right-of-way rules etc. This can be enhanced by using road sign recognition systems, which are easily translatable into a semantic context. Country-specific regulations can be addressed with an approach similar to the modular one proposed by [2].

Load-Related Information. The Context should also model the information related to the load of the car for different reasons. For instance, for really heavy loads, we may want to let the robot know the approximate weight, so that it is capable of taking it into account in its reasoning and come up with a correct behaviour. This way, the car knows it may take longer to accelerate if compared to a scenario without load. Another information we could have regarding the carried load is its fragility: if a human driver knows she's carrying one or more fragile objects, she will adjust the way she drives and the path she chooses to reach the destination, for example avoiding sharp breakings or roads with speed bumps on them. It is enough to set a data property for encoding that information such that a robotic driver can drive accordingly, we can call it *isFragileObject*.

Robot-Related Information. As previously said, in order for the robot to make proper reasoning it is necessary to add to the ontology all the data that are related to its physical model. The level of the gas tank or of the battery, the number and kind of tires equipped, the maximum number of passengers and the current number on board, these are all examples of this kind of information. Even if it is possible to reason without them and still be able to modify the robot's behaviour, many of these information improve the quality of reasoning of the car, making it more adaptable but most importantly safer. For instance, if the car doesn't know if winter tires are equipped, if it starts snowing it can't take fully conscious decisions, possibly leading to dangerous situations. Also, the

robot must be able to know if a spare tire is equipped, since it affects its stability at higher speeds, and set the maximum speed to a safer, lower value. Instead of using a simple boolean data property, in this case we could use an object property linking the car to a specific type of tire, with its own data.

Fig. 2. The ontology for autonomous driving. (a) Classes; (b) Object Properties; (c) Data Properties

Obstacle Information. In order to enhance the tasks of safe navigation and obstacle avoidance, we have to model the different kinds of obstacles the robot could encounter in its path. If the robot has equipped an obstacle detection and recognition system, it is possible to try to label the external object and perform inference over it. In fact, the previously discussed ontology-based collision simulator proposed in [14] requires at least an estimation of the weight and material of the object to function. With this information it is possible to decide which action has the least bad consequences and to take it. Going further, we can try to assign different labels to known classes of obstacle: for instance, a traffic sign, a road hole or a tree could be labelled as static obstacles, while a car in front could be labelled as moving. Another example can be that of the perception of a child: this could be marked as an unpredictable obstacle, since even he is on the sidewalk, so he's not currently an obstacle, he may suddenly change its position and become one. A similar reasoning could be applied to the case of pets and wild animals.

Other Information. Of course our list is not exhaustive, since to model the whole context many more kinds of data are necessary. These comprehend for instance the weather conditions, the cyclic traffic behaviours, the presence of temporary strikes that close some roads, the time of the day and more. More complex reasoning rules can be implemented, for example by letting the vehicle know that, during the day hours, it has to drive slower in the roads that are adjacent to a school.

In Fig. 2 we can see the first version of the ontology we developed for autonomous navigation, following the steps above. Future works will aim to improve it to integrate more information.

6 Reasoning over the Context of Navigation

As previously stated, the main reason for the choice of ontologies to model the Context of Navigation is the possibility to perform reasoning over it. We now discuss which kind of rules our robot may need depending on the application and how to encode them into the Context. The rule language chosen for this application is SWRL - the Semantic Web Rule Language - and the chosen reasoner is Pellet [15], which is able to reason over OWL ontologies and SWRL rules too. Some of the kinds of rules we used in our ontology, in the application to intelligent vehicle navigation, are presented now.

Regarding the passenger's preferences we have different rules, as the ones directly impacting the maximum speed value. Let's take one of the examples we defined in Sect. 2, of a passenger who prefers to displace in a car which moves at a significantly slower speed than the maximum one set by the traffic rules. With an instance *Micheal* as such passenger, and *MyCar* as our ego vehicle, we can store the following triples:

$$\begin{gathered} \textit{MyCar} \quad \textit{hasPassenger} \quad \textit{Micheal} \\ \textit{MyCar} \quad \textit{hasLocalRoadDrivingStyle} \quad \textit{Standard\_LocalRoadDrivingStyle} \\ \textit{Standard\_LocalRoadDrivingStyle} \quad \textit{hasMaxSpeed} \quad \textit{50} \end{gathered}$$

These three triples state that the car has *Micheal* as a passenger, and that the car is in an environment in which the maximum speed on local roads is 50 km/h.

$$\begin{gathered} \textit{Micheal} \quad \textit{hasDrivingStylePreference} \quad \textit{Micheal\_LocalRoadDrivingPreference} \\ \textit{Micheal\_LocalRoadDrivingPreference} \quad \textit{hasMaxSpeed} \quad \textit{30} \end{gathered}$$

With these triples we state that, when it comes to local roads, *Micheal* prefers to drive below 30 km/h. It is easy to create a very simple SWRL rule to adapt the maximum speed of the car to the one of the passenger, in the case there is only one person on board:

$$\begin{gathered} EgoVehicle(?v) \wedge hasPassenger(?v, ?p) \wedge Person(?p) \wedge \\ hasLocalRoadDrivingPreference(?p, ?pref) \rightarrow \\ hasLocalRoadDrivingStyle(?v, ?pref) \end{gathered}$$

In the case of multiple passengers with different preferences, it is possible to define rules to choose just the lowest one, or the one of the pregnant passenger, or to use other criteria. Of course, in the case a passenger sets a preference for a speed over the one defined by the traffic rules, this won't be taken into account. To do this we can create a rule that, when different *hasMaxSpeed* values are present, always chooses the smallest.

More complex rules, defined over others can be defined, resulting in a multi-layered rule structure, as for the next two SWRL rules:

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

$$EgoVehicle(?v) \wedge hasFragileLoad(?v, true) \rightarrow shouldAvoidRoadsWithObstacle(?v, SpeedBump)$$

This couple of rules will lead the car to choose, when possible, a path to its destination which doesn't present speed bumps in order not to damage the load. The safety of the trip may also depend on other information, as the presence of a mounted spare wheel, since these normally has a lower maximum speed than standard ones, a maximum number of kilometers to be used for, and other stats which are specific to the car and wheel's model and brand. We can model our ontology to store this information, and use SWRL rules to let the vehicle reason over it, by setting its own maximum speed with the one related to the kind of installed spare wheel.

Many other rules can be applied, for instance to modify the way the car overcomes speed bumps when a pregnant person is on board, in order not to hurt the child. Tests of this scenario were performed with good results.

Also, it is important to notice that many of these values for maximum speed must be intended as suggestions, as the ones of the preferences of the car. This means we don't expect the robotic vehicle to always follow them, but to try to adapt to these values when it is possible. However, there are cases in which it won't be possible or advisable for the vehicle to follow these suggestions. In Sect. 7 we give our solution to this problem.

7 Interaction with the Control Loop

In order to really adapt the behaviour of the car to the situation, just setting the maximum longitudinal speed as we proposed in Sect. 6 is not sufficient. To do so, we need to have a more complex interaction with the robot's control loop. For the sake of simplicity we consider for now just the longitudinal velocity of the robot and the case in which a passenger prefers the car to move at a slower speed than the standard one. A simple solution we propose is to add a block in the block diagram of the feedback control of the robot, which we called the Decision Block. The latter receives as inputs the desired velocities by the controller and the reasoner, and will output the actual one we want the robot to follow. In Fig. 3 we can see the new block diagram with the Decision Block. In this part of

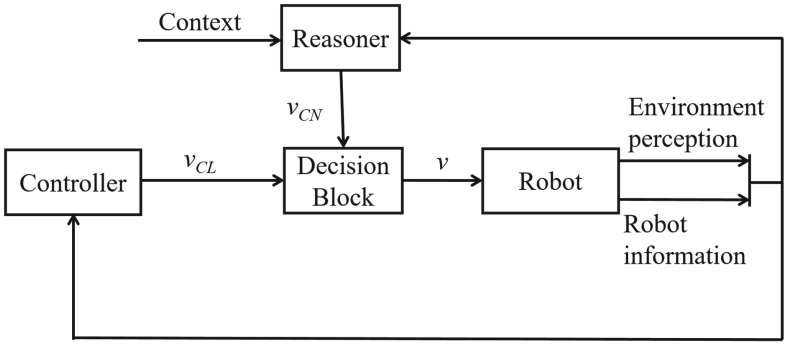

Fig. 3. The Decision Block outputs the reference speed for the vehicle to follow

the study, we proposed two ways the interaction between the ontology and the controller can be modelled. Since the first proposed solution was simply based on a modulation of the speed value given by the controller, we hereby present just the second one. We modelled our ontology, as we did in this paper, in such a way to output a *hasMaxSpeed* value for our vehicle, let's call it v_{CN} (as in velocity of Context of Navigation). The controller gives another reference output, let's call it v_{CL}. The mathematical rule for the Decision Block proposed is the following:

$$v = \begin{cases} v_{CL}, & \text{if } v_{CL} \leq v_{CN} \\ \gamma v_{CL} + (1-\gamma) v_{CN}, & \text{otherwise} \end{cases} \tag{1}$$

with $\gamma \in [0, 1]$. (1) stands for the following: when the speed proposed by the controller is greater than the one suggested by the ontology, the final value of the reference velocity is computed through the convex combination of the two, with parameter γ. The value for this parameter has to be computed in real time, as it represents how much we follow the reasoner's suggestions. There could be cases, for instance one of little to no traffic, in which we can let the car proceed at a slow speed, so we can set $\gamma \cong 0$. On the other hand, in a really busy road this would create traffic issues and we can't let the car follow completely the suggestions. We then set $\gamma \cong 1$. Of course, values in-between may be used, for instance with $\gamma = 0.3$ we obtain a behaviour mostly based on the contextual suggestions, but with still a component related to the controller's output. In Fig. 4 we can see the velocity profile of a robotic car which longitudinal speed value was computed with Eq. 1. The example shows an overtaking scenario. The solid line represents the behaviour of the robot without context awareness, while the bottom one represents the behaviour when adapting to the speed preferences of the passenger. The lines in between represent the results we obtain with two different values of γ, as explained before.

Another solution is depicted in Fig. 5. Instead of using a Decision Block to merge the outputs of the controller and the reasoner, the reasoner output is fed to the controller as input. This means that the control law must be built or

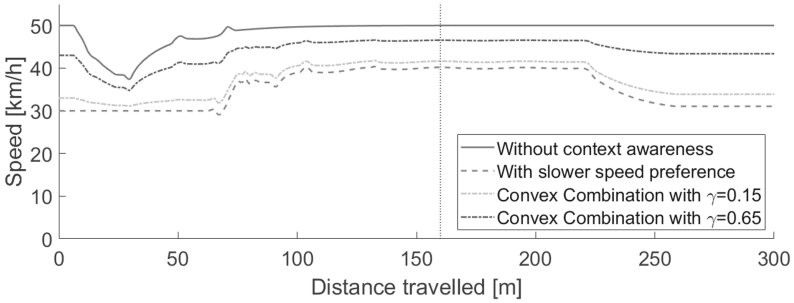

Fig. 4. Velocity profile of the robotic car applying (1)

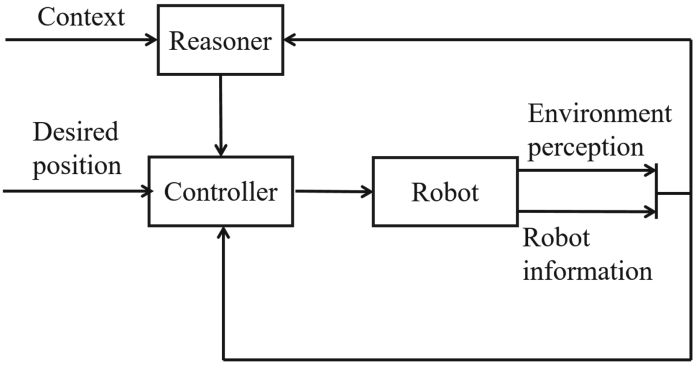

Fig. 5. The Reasoner's suggestions are taken into account directly in the robot's controller

modified accordingly to the way the ontology is shaped. We can also decide if to use the ontology as it is (for instance, suggesting a value of maximum speed as we did before) or a lighter version of it, maybe with simpler suggestions (*goSlower, goFaster* etc.). Also, since with this solution it will be possible to access the control loop, we can take into account values as the ones of the angular velocity, angular rate, jerk and others, and we can set specific rules on them.

8 Conclusions and Future Work

In this paper we showed how a semantic model of the Context of Navigation is required to let a robot adapt completely its behaviour to the contextual information at its disposal. In the use-cases of service robots and of autonomous vehicles, we showed how many contextual information can and must be considered to perform such a task. We have given our definition of the Context of Navigation and listed many of the areas of information that may be interesting in its scope. We have also already tested a first version of this solution in a

simulator in two autonomous navigation scenarios, showing promising results. Tests on the service robot use-case will have to be performed too.

However, many more aspects of this work have to be deepened, starting with the full implementation of the ontology and rules of the Context of Navigation. Also, a deeper interconnection between it and the robot's control loop must be studied, in order to adapt the actual robot's control to the situation, instead of its output in speed. The problem of the performance of the reasoning system is of great importance in vehicle-related systems, as all the inference must be done in a matter of milliseconds for safety reasons. Many optimized reasoners were studied in the literature, and the most promising ones will have to be tested for this specific application.

Many questions remain open, as the way to provide the physical vehicle with the information: for instance, it could be unpleasant for the passengers to input their preferences each time they start the vehicle, and to solve this problem we could study a way to create a profile for each passenger. Of course, this will open security and ethical issues related to the personal data, that will have to be addressed.

Acknowledgement. This research study was funded by Idex Sorbonne Université - Labex MS2T.

References

1. Armand, A., Filliat, D., Ibanez-Guzman, J.: Ontology-based context awareness for driving assistance systems, pp. 227–233 (2014). https://doi.org/10.1109/IVS.2014.6856509
2. Buechel, M., Hinz, G., Ruehl, F., Schroth, H., Györi, C., Knoll, A.: Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles (2017). https://doi.org/10.1109/IVS.2017.7995917
3. Dey, A.: Understanding and using context. Pers. Ubiquitous Comput. **5**, 4–7 (2001). https://doi.org/10.1007/s007790170019
4. Elbanhawi, M., Simic, M., Jazar, R.: In the passenger seat: investigating ride comfort measures in autonomous cars. IEEE Intell. Transp. Syst. Mag. **7**(3), 4–17 (2015). https://doi.org/10.1109/MITS.2015.2405571
5. Li, S.: Context-aware recommender system for system of information systems. University of Technology of Compiègne (2021)
6. Lim, G.H., Suh, I.H., Suh, H.: Ontology-based unified robot knowledge for service robots in indoor environments. IEEE Trans. Syst. Man Cybern. - Part A: Syst. Humans **41**(3), 492–509 (2011). https://doi.org/10.1109/TSMCA.2010.2076404
7. Lima, D., Victorino, A.: A hybrid controller for vision-based navigation of autonomous vehicles in urban environments. IEEE Trans. Intell. Transp. Syst. 1–14 (2016). https://doi.org/10.1109/TITS.2016.2519329
8. Luperto, M., et al.: Towards long-term deployment of a mobile robot for at-home ambient assisted living of the elderly. In: 2019 European Conference on Mobile Robots (ECMR), pp. 1–6 (2019). https://doi.org/10.1109/ECMR.2019.8870924
9. Musen, M.A.: The protégé project: a look back and a look forward. AI Matters **1**(4), 4–12 (2015). https://doi.org/10.1145/2757001.2757003

10. Nakamura, T., Yuguchi, A., Maël, A., Garcia Ricardez, G.A., Takamatsu, J., Ogasawara, T.: Ontology generation using GUI and simulation for service robots to operate home appliances. In: 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 315–320 (2019). https://doi.org/10.1109/IRC.2019.00058
11. Regele, R.: Using ontology-based traffic models for more efficient decision making of autonomous vehicles, pp. 94–99 (2008). https://doi.org/10.1109//ICAS.2008.10
12. Reinaldo, J.O., Maia, R.S., Souza, A.A.: Adaptive navigation for mobile robots with object recognition and ontologies. In: 2015 Brazilian Conference on Intelligent Systems (BRACIS), pp. 210–215 (2015). https://doi.org/10.1109/BRACIS.2015.50
13. Sato, E., Sakurai, S., Nakajima, A., Yoshida, Y., Yamguchi, T.: Context-based interaction using pointing movements recognition for an intelligent home service robot. In: RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication, pp. 854–859 (2007). https://doi.org/10.1109/ROMAN.2007.4415204
14. Schlenoff, C., Balakirsky, S., Uschold, M., Provine, R., Smith, S.: Using ontologies to aid navigation planning in autonomous vehicles. Knowl. Eng. Rev. **18**(3), 243–255 (2003). https://doi.org/10.1017/S0269888904000050
15. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a practical OWL-DL reasoner. J. Web Semant. **5**(2), 51–53 (2007). https://doi.org/10.1016/j.websem.2007.03.004. https://www.sciencedirect.com/science/article/pii/S1570826807000169. Software Engineering and the Semantic Web
16. Victorino, A., Abel, M.H.: On the implementation of semantic model for intelligent vehicle navigation. In: Proceedings of the 2nd International Conference on Deep Learning, Artificial Intelligence and Robotics, ICDLAIR 2020 (2020)
17. Werling, M., Ziegler, J., Kammel, S., Thrun, S.: Optimal trajectory generation for dynamic street scenarios in a frenét frame. In: 2010 IEEE International Conference on Robotics and Automation, pp. 987–993 (2010). https://doi.org/10.1109/ROBOT.2010.5509799
18. Zhao, L., Ichise, R., Mita, S., Sasaki, Y.: An ontology-based intelligent speed adaptation system for autonomous cars. In: Supnithi, T., Yamaguchi, T., Pan, J.Z., Wuwongse, V., Buranarach, M. (eds.) JIST 2014. LNCS, vol. 8943, pp. 397–413. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15615-6_30
19. Zhao, L., Ichise, R., Yoshikawa, T., Naito, T., Kakinami, T., Sasaki, Y.: Ontology-based decision making on uncontrolled intersections and narrow roads. In: 2015 IEEE Intelligent Vehicles Symposium (2015). https://doi.org/10.1109/IVS.2015.7225667