Semantic Traffic Data Analysis for a Local Leader Election Algorithm (LLEA)

Roua Elchamaa *,†, Rima Kilany Chamoun†, Baudouin Dafflon‡, and Yacine Ouzrout*
*Univ. Lyon, Lumière University Lyon 2, DISP, F-69676, Bron, France
Email: {roua.elchamaa, yacine.ouzrout}@univ-lyon2.fr
†Saint-Joseph University (ESIB) , Lebanon
Email: rima.kilany@usj.edu.lb
‡ Univ. Lyon, University Claude Bernard Lyon 1, DISP, F-69676, Bron, France
Email: baudouin.dafflon@univ-lyon1.fr

***Abstract*—MASCAT is a research-based road traffic simulator. In this paper, we propose a plugin for MASCAT. The aim of this proposition is to provide a semantic data analysis for the simulators Multi-Agent System (MAS). The plugin introduce an interpretation phase during vehicle to vehicle communication (V2V). It will allow connected vehicles to make the most efficient behavioral decisions based on the state of surrounding environment. We designed a semantic web ontology to describe traffic data and ameliorate behavioral decisions. Our semantic plugin can link, structure, analyze MASCATs traffic data and can also optimize the Local Leader Election Protocol. Indeed, we demonstrate that a small percentage of connected cars can ensure traffic regulation specially in a shifting environment.**

***Index Terms*—Ontologies, Semantic Rules, Multi-agent System, Connected Vehicles, Local Leader Election Algorithm (LLEA)**

I. Introduction

The progressive growth of the number of vehicles in our cities is considered as a cause of traffic congestion, but it is not the only reason for traffic jam. In order to study this problem thoroughly, it is necessary to simulate real traffic conditions by using an appropriate traffic simulator that would help us study the behavior of the vehicles.

Connected vehicles are able to sense and adapt their behavior according to their environment. How these vehicles will change the way we deal with traffic regulation? What percentage of connected vehicles is enough to ensure traffic regulation? In this paper, we will answer these questions, by describing the details of the framework we implemented as a semantic plugin for the MASCAT simulator. Semantic description provides the means to store information while giving it a logical meaning and a richer context.

This also means that we can construct advanced and intelligent queries over an ontology. The objective is to obtain the information inferred (by a reasoner) from the ontologys set of pre-defined relations and rules. Semantic rules are simply a set of IF-THEN statements for structuring complex axioms about a specific domain. We will start in section II with exploring the related work. In section IIIwe will describe our proposed solution. The solution design and implementation are detailed in section IV, while results are shown and analyzed in section V. Finally, conclusions and future work are exposed in section VI .

II. Related Work

V2V (Vehicle-to-Vehicle) communications need to be tested through intensive experiments. Simulation models should include mobility models for providing accurate simulation of real time vehicular networking environment. The simulation tools should be selected based on their compatibility with application requirements and similarity to real-time traffic. We analyzed some existing traffic simulators. For the needs of our work, the chosen platform will not necessarily have to offer a very detailed definition of a mesh network.

Our affinities with free and community software push us to retain two main candidates: SUMO and MovSim, which offer in addition a lot of functionalities by their stage of maturity compared to the previous platforms. SUMO is widely used and well represented in research, and MovSim is a newer and less represented platform at the moment.

Simulation of Urban MObility (SUMO) is a road traffic simulator [12]. With SUMO, traffic demand consists of single vehicles moving though a given road network. Real-world networks are modeled as graphs, where the roads and intersections are respectively represented as a graphs. In SUMO, each vehicle's speed is computed using a Car-Following Model. This model usually compute a targeted vehicle's speed by looking at its own speed, its distance to the group leader, and the leader's speed. SUMO is widely used and well represented in research, but MovSim was recently developed based on the main recent concepts in traffic theory [28] [30].

Regarding V2V inter-vehicle communication, an extension of SUMO is under development and aims to study the effects of on-board applications on driver behavior [13]. In addition, VEINS project aims to offer a set of models dedicated to inter-vehicle communications (IVC) in SUMO. But in MovSim side, the developers imagined the integration of these forms of communication from the start of the project [23] [26]. MASCAT [8] is a research-based road traffic regulation simulator developed using the already existing Movsim simulator [29] by transforming it into a Multi-Agent System where each vehicle is modeled as an intelligent separate entity which can run according to an algorithm of its own. Each instance of the Vehicle entity is simulated independently either by respecting the well-known IDM mobility model, thus representing the unconnected vehicles or by respecting a Local Leader Election Algorithm (LLEA) to represent connected vehicles (CVs). Multi-Agent System (MAS) is a loosely coupled network of problem-solving entities called agents that collaborate in resolving problems that are beyond the single individual's scope of capabilities and knowledge [7]. In the traffic simulator MASCAT, connected vehicles are modeled as agents to find the best behavioral decision for the CVs that are moving in a constantly shifting environment. So, to ensure vehicle to vehicle communication, the vehicles must plan their actions jointly to allow better cooperation between them. So the key issue in a MAS is to formalize the coordination between agents [24]. Many works have been proposed based on MAS and ontologies. The intelligent levels of the system (Knowledge Base and Multi-Agent System) in [6], use the knowledge provided by the IOT devices and its semantic environment in order to reason and react to set theses devices. In [2] a model was implemented based on a multi-agent approach for urban freight transportation, a knowledge data model was used to represent the urban environment. A city logistics ontology was proposed. A method for the representation of knowledge and reasoning in Agile Worker-Cobot manufacturing has also been proposed in [21]. A similar system based on ontology and multi-agent in [4] has been proposed for the Construction and Cooperation Mechanism of Logistics Vehicle. Inspired by these related works, we decided to increment the MASCAT Multi-Agent-System with semantic knowledge in order to have more realistic simulations, and auto-adapting agents that are able to infer new traffic data such as changing weather conditions, accidents, roadworks, A Relational Database may have been the answer to store such knowledge, but a relational database [15] fails if provided with fragmentary or incomplete knowledge because it works upon closed-world assumptions. The traditional relational database stores concepts in the form of tables without containing information about the meaning of these stored concepts, and how they are related to other concepts. Ontologies [10] have also been adopted as a possible representation for complex concepts and domains. In fact, it is a powerful model for mapping and describing information related to real-world knowledge areas. [25]. Ontologies have advanced ways to provide automated classification for different types of data. However, the processing of the ontology's individuals is one of the costliest computational operations within ontology reasoning. To query this data model, we can use two query languages SPARQL [3], [9] and SQWRL [18]. Furthermore, the reasoning ability of ontologies can also lead to problems in processing time. Several types of research have been done [14] to analyze the performance of the available reasoners. It shows that the structure and size of the ontology and the complex queries submitted to the reasoner have a high influence on the performance of these ontologies.

Therefore, we studied yet other solutions to find a technology that can provide similar capabilities with better performance. In our research, we are interested in the GDBMS that support RDF triple stores and inferences. So based on the mentioned ranking and on a comparison of different GDBMS done by the university of Leipzig [11].

We came up with a list of the most popular and free to use GDBMS that support RDF/SPARQL as their data model: Virtuoso, allegroGraph, StarDog, GraphDB, BlazeGraph.

And for rule-based inference [1], there are two principle reasoning strategies: Forward-chaining and Backward chaining. The first one is the Forward-chaining, this type of reasoning strategy involves applying the inference rules to explicit statements in order to produce new facts. The second one is the Backward-chaining strategy that which require to start with a fact illustrating or a query answering.

In our case, we are interested in forward-chaining reasoning because query time is an essential factor that guarantees the required performance to keep up with the constantly "shifting environment" of the connected vehicle in the traffic simulator MASCAT. Therefore, we need to dynamically create the Environment individuals and obtain the possible inferences before querying the semantic knowledge. Virtuoso and StarDog GDBMS only support Backward-Chaining Reasoning, which is not suitable for our case, so we are now only down to three Graph Database Management Systems: AllegroGraph, GraphDB and BlazeGraph. AllegroGraph [34] leads the largest deployment with loading and querying 1 Trillion triples. However AllegroGraph is available on a Windows platform through a Linux Virtual Machine, which could degrade the performance factor as directly declared on AllegroGraphs website. Therefore, we considered the use of Ontotexts GraphDB since it is more popular than BlazeGraph.

Graph database [31] makes the application of operations possible based on a graph of data and metadata. Such operations can improve performance and greatly increase the speed of recovery operations while maintaining the same precision of pure ontology-based approaches and reasoning. Since with Graph database there is no schema, each entity can contain different data attributes, and there is no need to perform join operations on multiple tables in order to obtain the needed information. In this way, we will eliminate the need for saving redundant data and sending complex queries to retrieve woven datasets. But, what Graph Database would we adopt? GraphDB [17], [22] is built on OWL. It uses ontologies that allow the repository to automatically reason about the data [19]. It offers OWL inference allowing to create new semantic facts from existing facts. Massive loads, queries and inferencing can be handled in real time. And to augment the expressiveness of our designed ontology, we defined a set of Semantic Rules. These rules for GraphDB are called "Entailment Rules". In conclusion, we decided to use an approach based on the GraphDB Graph database to provide our vehicles with data interpretation capability in case of disturbance without having a high consumption of time. We managed to bypass our exclusive need for the standard OWL Ontology and benefited from a Graph Database Structure that takes an Ontology file as its input.

III. Proposed Solution

A. Global Overview

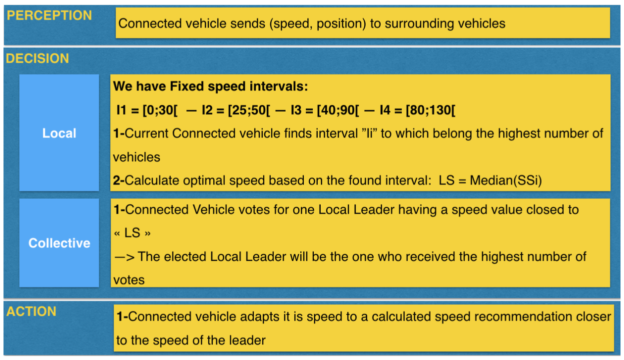

Fig. 1. Vehicle behavior in the LLEA

Based on our studies, we know that the vehicle driving behavior is one of the main factors that can disturb traffic flow. Each strange condition on the road can affect the surrounding vehicles. In case of an accident for example because of snowy weather, near surround- ing vehicles will be aware of the situation and will try to control their behavior accordingly. However, this reaction can have diverse effects on the flow, because other vehicles on the road might not be aware of what is happening. They only know that the road is blocked and they cannot react appropriately. This is not the case of connected vehicles which can react to regulate congested traffic in normal traffic conditions as we demonstrated in our last work [5], where we succeeded to regulate traffic using our proposed Local Leader Election Algorithm (LLEA) (Fig.1) with only 10% of connected vehicles on the road. This algorithm is computed for each CV every 10 seconds. The CVs speed is programmed to vary in a speed interval I = [0;130[(KM/h). We try in this case to mimic realistic traffic state variations, this interval was originally decomposed into 4 smaller, static and fixed intervals based on a certain logical traffic state:

- I1 = [0;30[: I1 includes vehicles in traffic congestion
- I2 = [25;50[: I2 includes vehicles in critical state
- I3 = [40;90[: I3 includes vehicles in normal traffic
- I4 = [80;130[: I4 includes vehicles in fluid traffic

These intervals will be modified in our actual solution and will be dynamically change based on each CVs Environment. This LLEA consists also of two consecutive decisions. The First Decision D1 is a local decision that each CV has to make. It consists of three steps:

- The CV must determine the interval Ii whose speed limits are respected by the highest number of surrounding CVs
- The CV must compute the optimal speed of the interval Ii which happens to be the median of the speed values of the surround- ing CVs in Ii
- The CV must choose the neighbor CV that has the closest speed to the computed median and send this decision to the surrounding CVs

The Second Decision D2 is a collective decision that happens in a certain zone. It consists of three steps:

- Each CV receives the D1 Decisions of the surrounding CVs
- The votes are counted for each CV in this zone
- The CV that received the most votes is elected to be the local leader for this zone and all CVs must adapt their speed to follow the speed of the local leader.

In our previous work, the implementation of this election protocol improved traffic flow in an optimal environment with no disturbances, by providing speed recommendations to the driver. This kind of recommendation will be offered to a connected vehicle after a negotiation with the surrounding vehicles based on a V2V communication. It allows a reduction of traffic congestion through an election of a local leader selected from the neighboring vehicles. These will adapt their behavior to follow the elected local leaders speed recommendation. Since this protocol is based on the concept of multi-agent system, each vehicle follows the cycle of Perception- Decision-Action However, many disturbances can affect the flow in real life like weather conditions, roadworks, unsafe and unexpected events, obstacles on the road, vehicle stopping, stuck vehicle, no protected accident area, emergency braking, unexpected queue end, ...

In order to be able to achieve traffic regulation in any context, it is strongly needed to provide a common framework that allows Traffic data to be shared and reused between entities. Therefore, it is necessary to Link and structure road traffic data exchanged between vehicles (agents) to simply access to the knowledge that it already contains and express accurately the real road traffic situation.

In other terms, we need to give our connected vehicles modeled as agents in the LLEA a particular knowledge of the environment to get them to: first understand the situation, then inform other connected vehicles on the road and finally emerge with a correct behavior whatever the conditions, without causing massive traffic congestion. As we showed in the state of art section, an ontology can provide a complete description of traffic data, and thus can allow an accurate interpretation of traffic context, which will give the connected vehicles adequate reactions. This ontology will represent the real road traffic road environment with the definition of all entities and properties that can affect the Connected Vehicles behavior. By augmenting the ontology with pre-defined traffic rules which respect to the European Traffic Norms and Regulations, the connected vehicles will adapt their behavior according to the knowledge inferred, at the interpretation level.

We choose to begin with inferring the adequate speed intervals needed for different situations on the roads, in order to provide CVs a with flexible intervals as we see in table (Fig.2). This will enhance the local leader election algorithm, since the local leader speed will be calculated based on dynamically inferred speed intervals. The framework we implemented is extensible, so in order to cover any environmental and contextual knowledge inference, we only need to add related concepts to the ontology and the corresponding rules.

Road Type	Road Properties							France Law	Ontology			
	Weather				Visibility V			Max S (Km/h)	Intervals provided to choose à LL			
	Sunny	Rainy	Snowy	Foggy	High	Average	Low (V< 50 m)		I 1	I2	I3	I4
HighWay	x				x			130	[0;30[	[15;50[	[40;90[	[80;130[
		x				x		110	[0;30[	[15;50[	[40;90[	[80;110[
							x	50	[0;30[	[12;28[	[25;38[	[35;50[
			x					50	[0;30[	[15;50[	[17;30[	[26;40[
Dual CarriageWay	x				x			110	[0;30[	[15;50[	[40;90[	[80;110[
		x				x		100	[0;30[	[15;50[	[40;90[	[80;100[
							x	50	[0;30[	[15;50[		
			x				x	50	[0;30[	[15;50[		
Two-Way Street	x				x			90	[0;30[	[15;50[	[40;90[	
		x				x		80	[0;30[	[15;50[	[40;80[	
							x	50	[0;30[	[15;50[		
			x				x	50	[0;30[	[15;50[		
Agglo	x				x			50	[0;30[	[15;50[		
		x				x		50	[0;30[	[15;50[		
							x	50	[0;30[	[15;50[		
			x				x	50	[0;30[	[15;50[		

Fig. 2. Intervals according to the European Traffic Norms and Regulations

Fig.3 is a UML activity diagram that shows the control flow of the processing steps that enable a connected vehicle to understand its environment based on a realistic inferred knowledge. Individuals of our semantic data model, representing the environmental conditions, will be created dynamically on load time. Each connected vehicles can check its environment to know if there is a need to adapt its behavior. In our approach, the environment is controlled by three main criteria: the road type, weather, and visibility. As already explained, other criteria can be simply added further on. If the environment didn′t change, it would load the behavior of the connected vehicle in the LLEA directly without a level of interpretation. If it did change, a query is prepared with the Road Type and weather as well as the visibility detected by the corresponding sensor, as input parameters. The defined ontology is then queried with these aggregated values in order to infer new knowledge related to this correlation of values. If the query returns no result, this would mean that this is an occurrence of a new context, so new individuals will be created and added to the the ontology representing these detected conditions. If successful, the new inferred speed intervals will be returned by the query at the interpretation level in order to choose a Local Leader in the LLEA respecting the particular conditions at this particular time. The election protocol will be launched every 10 sec by every connected car, and based on the results of the election the Local Leader, vehicle adapt his speed in order to follow his Leader.

IV. Solution Design and Implementation

MASCAT is a research simulator based on a Multi-Agent System of vehicles, implemented in Java, over the MOVSIM simulator. In our previous research work on MASCAT we proposed and implemented a local leader election protocol. The purpose of this protocol was to ensure traffic fluidity and regulation on a highway. This solution has shown its effectiveness in tackling the congestion problem with a small percentage of connected vehicles. However, the MASCAT simulator does not include a very important feature which is the auto-adaption of connected vehicles under several disturbances in the environment. The connected vehicles in MASCAT do not take into account the environment in the ideal world under disturbances like the meteorological conditions such as the visibility, weather We will describe how we were able to alter the behavior of the MASCAT connected vehicles so that they would consider their surroundings, and this, without deteriorating the simulators performance while always ensuring traffic regulation.

A. Semantic Plugin

The fact that the interactions between agents are highly information-dense raises many problems. Because of that our research was oriented to use a semantic method. This method can achieve a step of data interpretation by each agent on the multi-agent system. This goal can be achieved after implementing the Semantic Traffic Data Analysis plugin. Its main objectives are to Link, structure and analyze Traffic Data in order to optimize the Local Leader Election Protocol implemented in MASCAT. MASCAT Connected Vehicules would elect a local leader in a certain radius around them, based on the computed median speed value in the speed interval having the highest number of vehicles. The vehicle with the closest speed to this median is selected to become the local leader, so its speed will be adopted by surrounding connected vehicles. The Plugin will provide each connected vehicle with the adequate ”Speed Intervals” relatively to a certain context. The LLEA will be then executed, based on the inferred intervals, in order to compute an adequate recommended speed. Doing so will make the simulation more realistic by ensuring that the CVs behavior would adapt to a continuously shifting environment. In the following sub-sections we will start by describing the ontology based approach and the connection between the plugin and the simulator, then we will describe the graph database approach which we adopted for its high performance compared to the ontology based approach.

1) Ontology Based Approach:

- Firstly, as shown in the Fig. 4, we started to implement our solution in Protégé with a preliminary prototype including an initial ontology representing the context of road traffic. This ontology includes main classes (Context, Weather) and properties (hasWeather, hasSpeedLimit, hasRecommendedSpeed)
- Secondly, in order to obtain a recommended speed based on knowledge about meteorological conditions, this solution was augmented with semantic rules using semantic web rule language (SWRL) to respond to each traffic context (for each individual). This ontology uses the rule engine Drools and the Pellet reasoner in order to generate the axioms of the defined rules, inferring facts and checking the consistency of the ontology.
- Thirdly, based on this solution, we succeeded in returning a value of ”the recommended speed” for a certain context by using a SPARQL query. [22]

2) Connecting a semantic module to the MASCAT simulator: In order to connect MASCAT to the semantic plugin, we implemented a Java module that uses the Jena API to query the OWL Ontology. The communication with the simulator

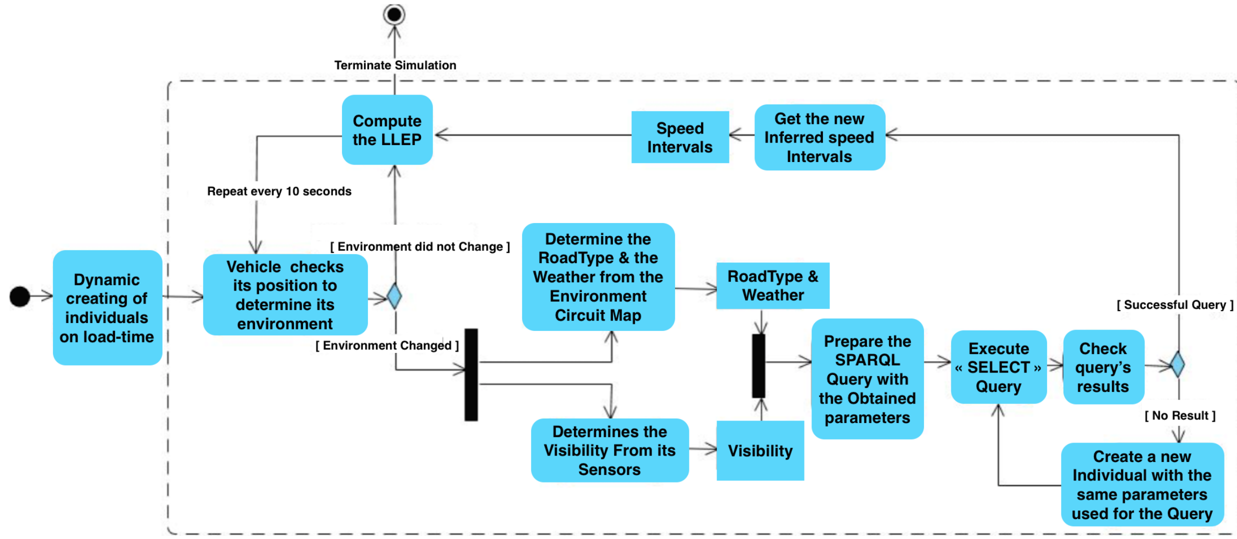

Fig. 3. Vehicle behavior in case of a semantic traffic data analysis

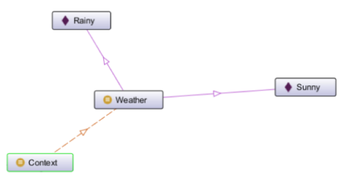

Fig. 4. First Prototype Ontology on Protégé

was done through a dedicated Singleton Java class that we named the SemanticDecision class. A simple variable was used to toggle the semantic behavior On and Off, which will consequently toggle the interpretation phase On and Off, thus granting the vehicles with two possible behaviors:

- The First behavior where the CVs are following the steps local leader election algorithm (LLEA) without a semantic data analysis (without interpretation level)
- The second one where the CVs have an altered behavior: the Semantic Decision Behavior in which they adapt according to the inferred semantic knowledge.

This approach was tested in order to communicate an inferred recommended speed to the connected vehicles. The test showed that the ontology lacked the performance needed to keep up with the response delay of the simulator. Therefore, we opted to consider as an alternative approach the use of a graph database, which according to ref can perform much better than the ontology in terms of response time.

3) Graph Database approach: As we found when we explored the related work, and after severals tests and comparisons, we concluded that the use of the GraphDB Graph database would be the best choice to enhance the performance of our semantic plugin. We used Eclipse RDF4J (formerly known as Sesame) to connect GraphDB to the MASCAT code, and to process and handle RDF data. RDF4J also supports creating, parsing, scalable storage, reasoning and querying with RDF and Linked Data (Section II).

GraphDB can process an input OWL Ontology file and has a built-in reasoner (TRREE) that automatically can make inferences at load time (Forward-Chaining Reasoning), with a very good performance level, enabling us to connect to MASCAT and run our simulations appropriately. After ensuring that the semantic plugin is highly functional at the technical level, we went through enhancing it at the semantic level, by extending the input ontology in order to be able to infer speed intervals according to the vehicule's context. We explored many proposed existing ontologies [16], [20], [27] which include many entities and concepts that can affect the behavior of the CV and their perception to discover their surroundings and respond to perturbations (entities such as the Weather property of the entity Environment). We used a combined subset of these concepts, nevertheless the ontology could be extended and refined furthermore in future works. As for the speed intervals used in our last work for the implementation of the local leader election algorithm, they were defined in the protocol in a static way. But if we want to give the connected vehicles the possibility to adapt their behavior according to their context, the speed intervals should now be inferred from the GraphDB's Ontology.

Thus we added the following classes to the ontology Fig.5 : "Environment, RoadType, Weather, Visibility" and the following object properties "hasRoadType, hasWeather, hasVisibility" and the 4 inferred speed Intervals which are represented by 8 Data Properties which are the minimum and maximum bounds of each of these four intervals (hasInterval1Min, hasInterval1Max, ...)

We also augmented our Ontology with semantic rules to obtain the speed intervals when needed. In GraphDB these rules are called entailment rules and can be added in a custom ruleset .pie file. The main goal is to offer a connected vehicle with speed intervals (Fig.2) that correspond to its specific

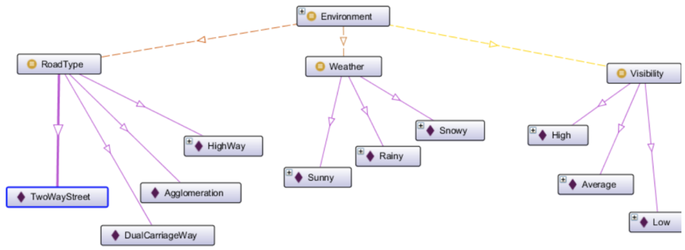

Fig. 5. Final Solution Ontology

Environment individual, and thus respect road type, weather conditions, visibility and regulations.

V. Results

As the simulations done in our latest work show, [5] a small percentage of connected vehicles can improve traffic flow, when adopting V2V (Vehicle-to-Vehicle) communication approach combined with a traffic regulation scheme based on a decentralized election protocol. In this model, the elected Local Leader plays a key role in the regulation of the traffic. What was missing in the simulations already undertaken, is the study of the behavior of the connected vehicles in critical weather conditions. This is exactly what the semantic plugin will help us test, since the LLEA will be based on the knowledge (inferred intervals corresponding to the specific weather condition) offered by our GraphDB semantic approach.

A. Set Up

In this section, we describe the common experimentation parameters to run our scenarios:
Since Highways suffer from an enormous daily amount of vehicles, the road type studied in our tests is the Highway type. The speed limitations are those defined by European Laws (Fig.2). The baseline scenario consists of a dense traffic state generated on a 3-lane straight highway. The input flow is maintained to 1800 vehicles per hour during the 1200 seconds simulation. Then, we gradually introduce CVs (from 0% to 30%) which will execute the modified LLEA which takes into consideration the inferred semantic knowledge. For now, this knowledge consists of the speed intervals, needed for the computation of the recommended speed, in each environment and thus the election of the local leader. These scenarios correspond to early CVs deployment phases that will occur soon in real-life scenarios. Traffic flow is expected to be gradually improved when we increase the number of CVs, but we need to verify vehicles behaviors in critical weather conditions based on the knowledge offered by our implemented semantic plugin. Simulations results presented in this paper were given by the mentioned Multi-Agent Simulator for Connected and Automated Traffic (MASCAT) [8], augmented with our semantic plugin.

B. Simulations

After the implementation of the GraphDB based Solution, it was time to test our overall solution to validate if the CVs in MASCAT were adapting their behaviors as a reaction to a specific disturbance (a specific weather condition) according to a deep interpretation realized by each agent in order to get correct intervals for this situation. In consequence, elect a local leader and respect the offered speed recommendation in the regulation strategy.

1) Particular Weather Condition Scenario: Initial traffic density in this scenario is set to 37 vehicles per kilometer. This setting tends to model a critical regime that can cause a network capacity drop due to the heterogeneities in the flow of numerous vehicles. We explained previously in Fig. **??** that connected vehicles should be able to adapt their speed after detecting the actual weather condition based on vehicle knowledge. For example, in case of snowy weather and low visibility, on a Highway Road, our semantic query will return the following inferred intervals: [0;30[[15;50[[17;30[[26;40[

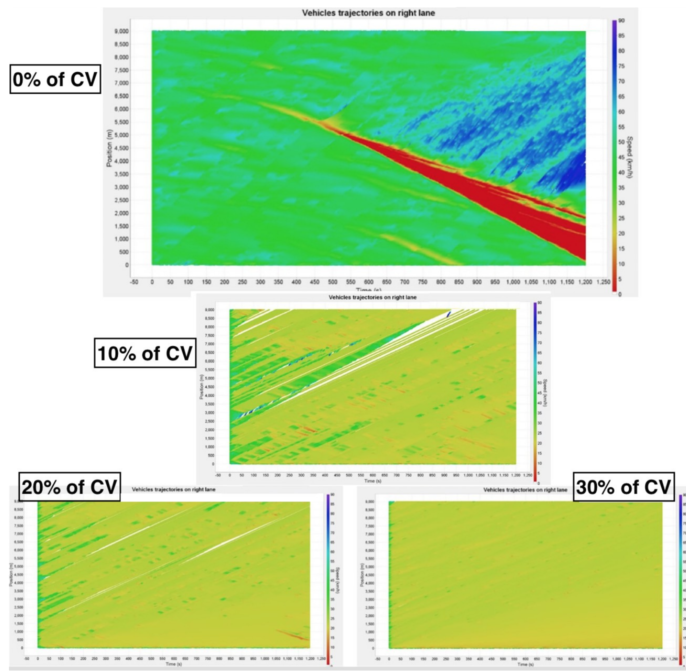

Fig. 6. Space-time diagram (vehicles trajectories and speed on the right-most lane) for the baseline simulation (with 0 to 30% of CV) in Snowy Scenario

Figure 6 depicts the trajectories (and speeds) of all the vehicles in the right-most lane, for the baseline scenario (0% of CVs). Then, we gradually introduce CVs (from 0% to 30%) executing the LLEA enabled with semantic knowledge. Traffic flow is expected to be gradually improved with the increasing number of CVs .
The baseline Scenario (0% CV) models a dense traffic state where congestion waves (in red color) appear spontaneously and grow, leading to the formation of massive traffic jams. The absence of connected vehicles in this baseline scenario will not allow the detection of environmental condition. The only possible way to control IDM vehicles speed in MASCAT simulator is to set their maximum speed to 40 km/h (equal to upper interval limit value). But, this maximum speed limit was

not useful. Vehicles didnt respect this maximum value and no regulation strategy of the LLEA was used. Traffic congestion was mainly due to the speed heterogeneities between individuals, and it is accentuated by lane-changes.

The purpose of our approach is the self-adaptation in any weather condition using our knowledge-based electoral protocol thus maintaining traffic fluidity even in case of low speed (snowy weather). We can observe on the same figure, the results for the 10% of the connected vehicle which show how connected vehicles were able to adapt their speed based on our proposed semantic approach in LLEA. The local leaders tend to stabilize the flow in very difficult weather conditions. We notice after simulating our scenario with an increased percentage of connected vehicles (up to 50% of CV), connected vehicles were able to behave correctly as we see in the space-time diagrams. We also check our approach with all weather conditions mentioned in Fig. **??** for the road type Highway. We were satisfied by the results, our knowledge-based LLEA shows its effectiveness and reactivity in case of a weather disturbance.

2) Shifting between several Weather Conditions Scenario: Initial traffic density in this scenario is set to 15 vehicles per kilometer in order to detect the connected vehicles behaviors changes from one weather condition to another on a Highway Road. The aim of this scenario is to validate the concept of auto-adaptation of our approach and analyze what happens in the step of shifting between two extreme weather conditions. We tested this scenario in semantically enabled MASCAT version, with a Road length of 9000 m. This Road is divided into 9 road segments 1000 m each. From position 0 to 4500, we choose to have the first Environment (HighWay - Sunny - High) while the second Environment (HighWay - Snowy-Low) will cover the position 4501 to 9000.In 7, we show how the vehicles started behaving at the beginning of the simulation.

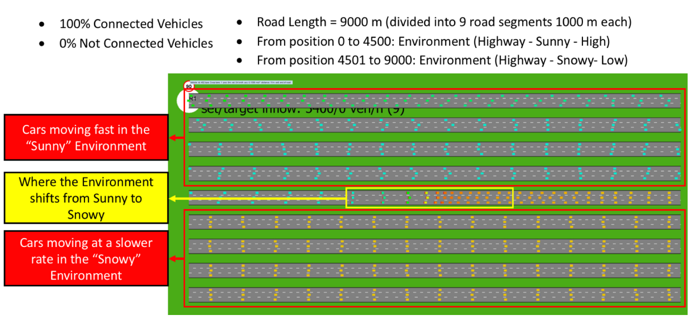

Fig. 7. Vehicles behaviors adaptation in Shifting scenario

As expected, we can see in Figure 7 that the highway is divided into two parts:

- The first part (between the positions 0 and 4500) in which the CVs are moving noticeably fast in a Sunny Environment (Notice the green and blue colors of the CVs).
- The second part (between the positions 4501 and 9000) in which the CVs are moving noticeably slower in a Snowy Environment (Notice the yellow and orange colors of the CVs)

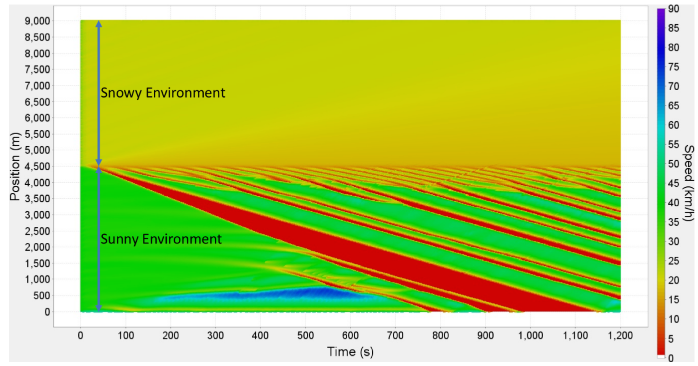

Fig. 8. Space time Diagram for 100 percent of CVs During Shifting Scenario

The space-time diagram shown in Fig.8 was obtained at the end of this scenario. Notice the middle road segment that is between the positions 4000 and 5000. As we can see, the cars are immediately slowing down when crossing from the Sunny to the Snowy Environment (Notice the red color of the CVs on the position 4500). This means that the CVs are actually abiding to the semantic LLEP result that was executed using the queried Speed Intervals. Each time a CV enters a new Environment, it queries the GraphDB for these Speed Intervals and will keep using them until this environment changes, therefore optimizing MASCATs performance. These results validates the overall behavior expected from our projects outcomes (the plugin and the ontology) that we detailed in the previous sections. Our solution was tested and validated after several simulations in MASCAT simulator. We can safely say that the Graph Database approach seems to be way more convenient than its semantic web ontology counterpart by effectively querying the Graph Database structure without having to rely on a less - performant exclusive ontology approach. We switched to the GraphDB GDBMS because we found out that the standard ontology could be easily inserted into a GraphDB repository. Doing so, we accomplished our projectsǵoals by implementing a plugin that would make the simulations more realistic. Being aware of the requirements and constraints, we ultimately did not deteriorate MASCATs pace while performing the necessary computations and tasks.

As we have seen in this paper, we have used many tools in order to develop this project and make it work. We design and build a working Semantic Web Ontology on Protégé and GraphDB, and successfully linked it to the MASCAT simulator by implementing a rigorous plugin. We also extended and improved our solution by dynamically creating the Ontologys individuals and activating the Visibility Sensor class in MASCAT. Our plugin helped achieve traffic regulation, by enabling the connected vehicles to sense and adapt to changing environmental conditions without deteriorating the simulator’s performance.

VI. Conclusions and Future Work

To summarize, results indicate that our approach can have a positive impact on the peoples lives where the driver of a CV could obtain a message in which the recommended speed of his vehicle is specified. He would then adapt to this speed knowing that its the most optimal, safe and law-abiding way to go. This way, he would make sure that the decision he takes would contribute in boosting the overall traffic conditions on the road and help in maintaining an efficient traffic situation that is conform to traffic regulations and laws. Additionally, our plugin made MASCATs simulations more realistic, and therefore more reliable by helping road traffic experts expand their studies and analyze the simulators results after introducing new factors that can now be taken into consideration thanks to the ever-expanding Semantic Web knowledge. The designed Ontology can be tailored to the experts needs in order to try to replicate any simulation environment thus contributing in finding solutions to the daily traffic congestions in shifting scenarios.

We propose now some of our recommendations for future works and implementations. We understand that this work is not fully complete in what concerns Semantic Traffic Data Analysis. The first noticeable thing that could be done is expanding the Ontologys designs in order to include more parameters and entities that can influence the behavior of the CVs in MASCAT such as the Drivers skills & abilities, traffic lights, road intersections, obstacles & road works, pedestrians, etc. Ultimately, the CVs sensors could be linked to a stream processing software platform like Apaches Kafka in order to simulate real-time environment detection in MASCAT.

References

[1] A. Al-Ajlan. The comparison between forward and backward chaining. *International Journal of Machine Learning and Computing*, 5(2):106, 2015.

[2] N. Anand, R. van Duin, and L. Tavasszy. Ontology-based multi-agent system for urban freight transportation. *International Journal of Urban Sciences*, 18(2):133–153, 2014.

[3] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. Ontop: Answering sparql queries over relational databases. *Semantic Web*, 8(3):471–487, 2017.

[4] L. Y. Chu, S. Dang, G. P. Xie, and C. Meng. Construction and cooperation mechanism of logistics vehicle scheduling system based on ontology and multi-agent. In *Applied Mechanics and Materials*, volume 865, pages 681–688. Trans Tech Publ, 2017.

[5] R. Elchamaa, M. Guériau, B. Dafflon, R. K. Chamoun, and Y. Ouzrout. A local leader election protocol applied to decentralized traffic regulation. In *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*, pages 1013–1020. IEEE, 2017.

[6] B. Gâteau, Y. Naudet, and J. Rykowski. Ontology-based smart iot engine for personal comfort management. *2016 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pages 35–40, 2016.

[7] M. Glavic. Agents and multi-agent systems: a short introduction for power engineers. 2006.

[8] M. Guériau, R. Billot, N.-H. El Faouzi, J. Monteil, F. Armetta, and S. Hassas. How to assess the benefits of connected vehicles? a simulation framework for the design of cooperative traffic management strategies. *Transportation research part C: emerging technologies*, 67:266–279, 2016.

[9] O. Hartig and G. Pirrò. A context-based semantics for sparql property paths over the web. In *European Semantic Web Conference*, pages 71–87. Springer, 2015.

[10] S. Jain. Intelligent decision support for unconventional emergencies. In *Exploring Intelligent Decision Support Systems*, pages 199–219. Springer, 2018.

[11] M. Junghanns, A. Petermann, M. Neumann, and E. Rahm. Management and analysis of big graph data: current systems and open challenges. In *Handbook of Big Data Technologies*, pages 457–505. Springer, 2017.

[12] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of sumo-simulation of urban mobility. *International Journal On Advances in Systems and Measurements*, 5(3&4), 2012.

[13] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of sumo-simulation of urban mobility. *International Journal On Advances in Systems and Measurements*, 5(3&4), 2012.

[14] T. J. Lampoltshammer and S. Wiegand. Improving the computational performance of ontology-based classification using graph databases. *Remote Sensing*, 7(7):9473–9491, 2015.

[15] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila. Ontologies versus relational databases: are they so different? a comparison. *Artificial Intelligence Review*, 38(4):271–290, 2012.

[16] P. Morignot and F. Nashashibi. An ontology-based approach to relax traffic regulation for autonomous vehicle assistance. *arXiv preprint arXiv:1212.0768*, 2012.

[17] S. Mouzakitis, D. Papaspyros, M. Petychakis, S. Koussouris, A. Zafeiropoulos, E. Fotopoulou, L. Farid, F. Orlandi, J. Attard, and J. Psarras. Challenges and opportunities in renovating public sector information by enabling linked data and analytics. *Information Systems Frontiers*, 19(2):321–336, 2017.

[18] M. O'Connor and A. Das. Sqwrl: a query language for owl. In *Proceedings of the 6th International Conference on OWL: Experiences and Directions-Volume 529*, pages 208–215. CEUR-WS. org, 2009.

[19] J. Pokornỳ. Graph databases: their power and limitations. In *IFIP International Conference on Computer Information Systems and Industrial Management*, pages 58–69. Springer, 2015.

[20] E. Pollard, P. Morignot, and F. Nashashibi. An ontology-based model to determine the automation level of an automated vehicle for co-driving. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 596–603. IEEE, 2013.

[21] A. R. Sadik and B. Urban. An ontology-based approach to enable knowledge representation and reasoning in worker–cobot agile manufacturing. *Future Internet*, 9(4):90, 2017.

[22] D. R. Schlegel, J. P. Bona, and P. L. Elkin. Comparing small graph retrieval performance for ontology concepts in medical texts. In *Biomedical Data Management and Graph Online Querying*, pages 32–44. Springer, 2015.

[23] M. Schönhof, M. Treiber, A. Kesting, and D. Helbing. Autonomous detection and anticipation of jam fronts from messages propagated by intervehicle communication. *Transportation Research Record*, 1999(1):3–12, 2007.

[24] P.-O. Siebers and U. Aickelin. Introduction to multi-agent simulation. 2008.

[25] M. Sir, Z. Bradac, and P. Fiedler. Ontology versus database. *IFAC-PapersOnLine*, 48(4):220–225, 2015.

[26] C. Thiemann, M. Treiber, and A. Kesting. Longitudinal hopping in intervehicle communication: Theory and simulations on modeled and empirical trajectory data. *Physical Review E*, 78(3):036102, 2008.

[27] C. Thierry, M. D. Hina, A. Soukane, and A. Ramdane-Cherif. Cognition of driving context for driving assistance. *World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering*, 5(2), 2018.

[28] M. Treiber and A. Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[29] M. Treiber and A. Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[30] A. M. Uhrmacher and D. Weyns. *Multi-Agent systems: Simulation and applications*. CRC press, 2009.

[31] L. Zheng, L. Zhou, X. Zhao, L. Liao, and W. Liu. The spatio-temporal data modeling and application based on graph database. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pages 741–746. IEEE, 2017.