2016 IEEE Intelligent Vehicles Symposium (IV)
Gothenburg, Sweden, June 19-22, 2016

Fast Decision Making using Ontology-based Knowledge Base

Lihua Zhao[1], Ryutaro Ichise[2], Yutaka Sasaki[3], Zheng Liu[4], and Tatsuya Yoshikawa[5]

Abstract— **Making fast driving decisions at intersections is a challenging problem for improving safety of autonomous vehicles. Furthermore, representing sensor data in a machine understandable format is essential to enable vehicles to understand traffic situations. Ontologies are used to represent knowledge of sensor data for autonomous vehicles to aware traffic situations. In this paper, we introduce a fast decision making system, which utilizes only related part of the ontology-based knowledge base to make decisions at intersections. The decision making system performs real-time reasoning using traffic regulations and a part of the map information from the knowledge base.**

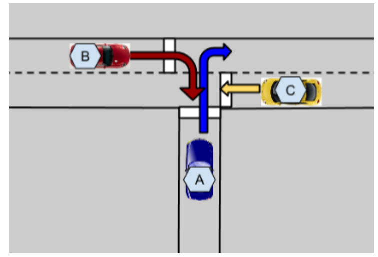


Fig. 1: Uncontrolled intersections and narrow roads.

I. INTRODUCTION

Driving environment perception has been noticed as one of the most challenging research topics for developing autonomous vehicles. The autonomous vehicles should be able to understand semantic meanings of sensor data to make safe driving decisions such as keep lane, give way or avoid obstacles. Although objects and lanes can be detected using various sensors, the vehicles cannot understand the meanings of driving environments without knowledge representation of the data. Therefore, a machine understandable knowledge representation method is required to fill the gap between environment perception and further knowledge processing.

We use ontologies to represent driving environment information in a machine understandable format. Ontologies are the structural frameworks for organizing information and are used in Artificial Intelligence, Semantic Web, and Biomedical Informatics as a form of knowledge representation about the world or some part of it. The Semantic Web is an extension of the World Wide Web that enables interoperation between systems by sharing data, most fundamentally in the format of Resource Description Framework (RDF) [1].

Resource Description Framework (RDF) is recommended by the *World Wide Web Consortium* (W3C) for constructing ontologies in data modeling [2]. An ontology mainly consists of concepts (classes) and the relationships (properties) among them. An instance is described by a collection of RDF triples in the form of <subject, property, object> [3]. To represent the sensor stream data from the sensors equipped on the autonomous vehicles, we use timestamp-based temporal RDF representation to construct RDF stream data [4]. Furthermore, we use a powerful RDF query language - *SPARQL Protocol and RDF Query Language* (SPARQL) to access RDF data [5], and use *Semantic Web Rule Language* (SWRL) to express traffic rules [6].

Currently, many autonomous vehicles can run on controlled intersections or on highways. However, running on urban streets such as uncontrolled intersections and narrow roads still remains as a challenging problem. In Japan, there are many narrow roads where even human drivers feel difficulty in driving. As shown in Fig. 1, when a car approaches an uncontrolled intersection that has no traffic lights, the driver has to carefully observe the other vehicles to decide whether to give way or not. In Fig. 1, the car C has the highest priority because it's running straight. Then the car A has higher priority than the car B, because it is going out from a narrow road to a wider road. Complicated intersection cases can be decomposed into several this kind of simple intersection cases to make appropriate decisions.

We have constructed ontology-based dataset for developing a decision making system by adding ontology-based traffic regulations [7]. The decision making system introduced in previuos work can detect dangerous situations and send warning signals to avoid overspeed or collision [8]. However, the decision making time depends on the size of the knowledge base, which increases rapidly as we extend the map information in it. Therefore, we developed a fast decision making system that can make driving decisions promptly by accessing to related knowledge only, rather than the whole knowledge base.

The remainder of this paper is organized as follows. In Section II, we list some related works, which also utilized ontologies for assisting vehicles. We introduce the improved fast decision making system that accesses to a part of the ontology-based knowledge base in Section III. We discuss real-world experimental results in Section IV and conclude this research work in Section V.

[1]Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST Tokyo Waterfront), Tokyo, Japan lihua.zhao@aist.go.jp
[2] Principles of Informatics Research Division, National Institute of Informatics, Tokyo, Japan ichise@nii.ac.jp
[3]Research Center for Smart Vehicles, Toyota Technological Institute, Nagoya, Aichi, Japan yutaka.sasaki@toyota-ti.ac.jp
[4] School of Engineering, Faculty of Applied Science, University of British Columbia (Okanagan Campus), Canada zheng.liu@ubc.ca
[5] Chassis & Vehicle Safety System Development Dept., AISIN SEIKI Co., Ltd., Kariya, Aichi, JAPAN yoshi@its.aisin.co.jp

978-1-5090-1821-5/16/$31.00 ©2016 IEEE

II. Related Work

An ontology-based traffic model is introduced to represent typical traffic scenarios such as intersections, multi-lane roads, opposing traffic, and bi-directional lanes [9]. Relations such as opposing, conflicting, and neighboring were introduced to represent the semantic context of the traffic scenarios for decision making. The traffic model was used as a basis for an autonomous vehicle simulator, which was proved to be beneficial for implementing and evaluating different driving behaviors without low-level trajectories.

An ontology for automated driving was constructed to represent context information of static infrastructure and dynamic environment [10]. In order to differentiate relevant traffic objects on the road from other objects like trees and parked cars, algorithms for information aggregation of dynamic traffic objects and a-priori map information were introduced. Their approach allows fast information access, that can be applied for autonomous driving.

An ontology-based context awareness ADAS was presented to enable vehicles to understand the interactions between perceived entities and contextual data [11]. The ontology contains context concepts such as Mobile Entity, Static Entity, and context parameters, which enables the vehicle to understand the context information when it approaches road intersections. Experiments showed that the vehicle was able to process human-like reasoning on global road contexts with 14 rules written in *Semantic Web Rule Language* (SWRL). Their proposed ontology could be used in real-time to retrieve the key entities that a driver should consider in different traffic situations.

Ontologies have been applied in the intelligent transportation field to describe semantic knowledge of the traffic scenarios and to improve safety at intersections. Other than a sophisticated digital map that can represent the details of road networks, an autonomous vehicle also needs a trajectory and concepts of different control information to understand driving motions. Therefore, we construct ontologies for autonomous vehicles by considering these essential factors to improve safety in autonomous driving, while the above research works only considered each particular aspect. In contrast to above research, we focus on roads in urban areas of Japan. The advantages of our research are as follows:

- By accessing to the ontology-based knowledge base, we can retrieve semantic knowledge about driving environment, such as, speed limit, driving direction, lane information, and connected road segment information at real time.
- Right-Of-Way rules written in SWRL are used for inferring rules to make safe driving decisions when the vehicle receives collision warning signals. The traffic rules are general rules rather than specifically designed for special cases.
- The proposed approach can be easily extended to deal with complicated traffic situations by decomposing them into a combination of simple traffic situations.

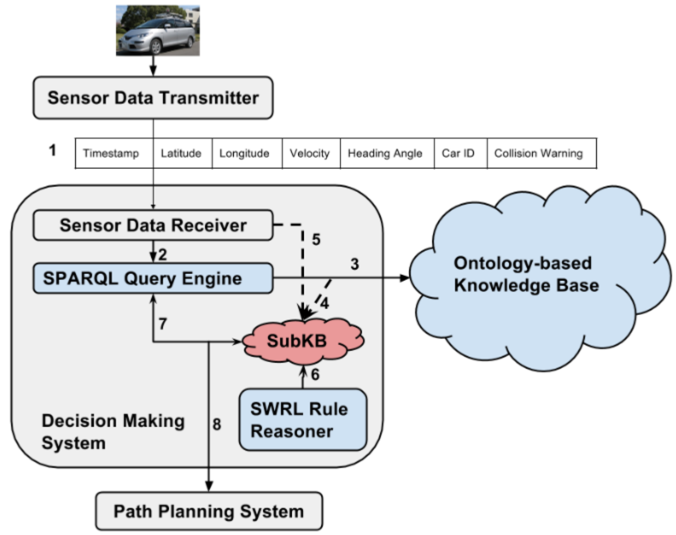


Fig. 2: Diagram of decision making system.

III. A Fast Decision Making System

In order to assist vehicles to aware traffic situations at intersections and to make a prompt decision such as “Stop”, “ToLeft”, or ”Give Way”, we developed a fast decision making system, which accesses to the ontology-based knowledge base. The ontology-based Knowledge base is constructed to enable autonomous vehicles to understand the knowledge of driving environment.

Figure 2 shows the diagram of the decision making system. The cloud shape represents data (the entire ontology-based knowledge base and a temporal Sub-Knowledge Base (SubKB)). The other rectangle shapes represent functions and the arrow lines represent processing steps. The dotted arrows are processing steps in special cases such as when the vehicle changes from one road part to another or when it receives collision warning signals. The main processing steps of the system are as follows:

1) The sensor data receiver gets sensor data via the sensor data transmitter, which also sends detected vehicle’s information while there is a potential collision. The sensor data are converted into RDF stream data format.
2) The sensor data receiver sends the data to the SPARQL query engine.
3) The SPARQL query engine accesses to the knowledge base to retrieve information of our vehicle’s current lane, next lane, and driving direction, etc.
4) If the vehicle is at the point where it changes from current path segment to the next path segment, we recreate the temporal Sub-Knowledge Base (SubKB), which contains only nearby road segments and the SWRL rules. By narrowing down the searching space of knowledge base, we can reduce rule reasoning time for decision making.
5) When the sensor data receiver gets collision warning signal, we add the driving situation information such as collision warning and the other vehicle’s position,

velocity, and driving direction into the SubKB. For example, if we detected that our vehicle (carX) has a collision warning with carY, we add a triple <carX, control:collisionWarningWith[1], carY> to the SubKB.
6) The SWRL rule reasoner performs reasoning on the updated SubKB and newly inferred information is added to the SubKB. For example, decisions such as "Stop", "ToLeft", or "Give Way" with the other vehicle's ID.
7) The SPARQL query engine accesses to the inferred SubKB to retrieve the decisions and the vehicles that our vehicle should give way to.
8) The decision signals are sent to the path planning system via the sensor data transmitter to update driving path or driving behavior. Newly added inferred knowledge is removed from the temporal SubKB.

The decision making system mainly consists of a sensor data receiver, a SPARQL query engine, a temporal ontology-based SubKB, and a SWRL rule reasoner. In the following, we describe each component in detail.

A. SPARQL Query Engine

The SPARQL query engine contains many predefined SPARQL queries that are used to retrieve knowledge from the knowledge base. SPARQL is a powerful RDF query language that enables Semantic Web users to access to the ontology-based knowledge base. The following SPARQL query in Example 1 is used to retrieve the next path segment with two variables - current path segment (tempaku:currPS) and current pathSegmentID ("currentID"^^xsd:int). The first pathSegmentID is 0 and increments by 1. By assigning the pathSegmentID, we can easily identify the next path segment even the current path segment is revisited.

```
Example 1:
SELECT          DISTINCT ?next
WHERE {
tempaku:currPS  control:nextPathSegment   ?next.
tempaku:currPS  control:pathSegmentID     "currentID"^^xsd:int.
?next           control:pathSegmentID     ?nextID.
Filter(         ?nextID = (currentID + 1)     ) }
```

The following SPARQL query in Example 2 is used to retrieve all the cars that our experimental vehicle (car:ToyotaEstima) should give way to, when it receives a collision warning signal. Here, car:ToyotaEstima is a variable that changes according to different experimental vehicles. The triple including the object property control:giveWay is added to the SubKB when the decision making system infers according to the Right-of-Way rules in SWRL.

```
Example 2:
SELECT DISTINCT ?cars
WHERE { car:ToyotaEstima  control:giveWay  ?cars. }
```

We also constructed other SPARQL queries to retrieve the types of a path segment, triples of an instance, and relations between instances, etc. Jena API[2] is used for executing SPARQL queries on the knowledge base.

TABLE I: Statistics of the dataset.

Name:	Dataset for Safe Autonomous Driving
Homepage:	http://www.toyota-ti.ac.jp/Lab/Denshi/COIN/Ontology/
Data Dump:	TTICore-0.1/
VoID Description:	void.ttl
RDF Triples:	37566
Entities:	1424
Classes	149
Data Properties	40
Object Properties	35

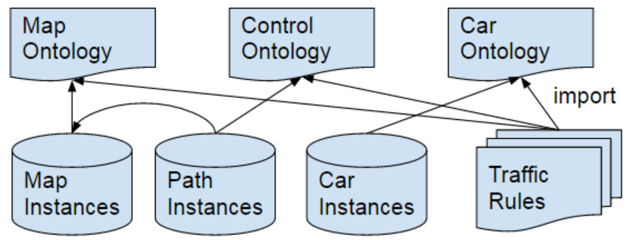


Fig. 3: Structure of the Semantic Knowledge Base.

B. Temporal Sub-Knowledge Base

We constructed machine-understandable ontologies and ontology-based semantic knowledge base for autonomous vehicles [7]. The statistics of the ontology-based knowledge base is described with VoID vocabularies as shown in Table I. VoID is an RDF Schema vocabulary for expressing metadata about RDF datasets [12]. There are 37,566 RDF triples, 1424 entities, 149 classes, and 75 properties (40 data properties, and 35 object properties) in the data dump. The repository of the dataset is available on the website "`http://www.toyota-ti.ac.jp/Lab/Denshi/COIN/Ontology/TTICore-0.1/`". The ontologies can represent knowledge of sophisticated maps, paths and driving control concepts that are necessary for autonomous driving.

The structure of semantic knowledge base is shown in Fig. 3, which is based on three main ontologies: map ontology, control ontology, and car ontology. The map ontology is used to create map instances and to define traffic rules. The control ontology is mainly used to describe path instances and also used to create traffic rules. The car ontology is used to create car instances and traffic rules with other ontologies. The path instances are constructed based on the map instances and control ontology. The knowledge base contains these three ontologies, instances about maps, paths, cars, and traffic rules written in *Semantic Web Rule Language* (SWRL).

The map will be extended and more traffic rules will be added to the knowledge base, which will increase the reasoning time to make a decision. In order to reduce decision making time, the system should access to a small portion of

[1]PREFIX control:<http://www.toyota-ti.ac.jp/Lab/Denshi/COIN/Control#>

[2]http://jena.apache.org/documentation/ontology/

```
Input  : currRS # Current Road Segment
Output: SubKB # Sub-Knowledge Base
dirList ← getConnectedRS(currRS);
rsList ← dirList;
SubKB ← ∅
foreach rs ∈ dirList do
 | rsList.add( getConnectedRS(rs) );
end
foreach rs ∈ rsList do
 | SubKB.add( getAllInfo(rs) )
 | if <rs, map:hasLane, lane> then
 |  | laneList.add( lane )
 | else
 |
 | end
end
foreach lane ∈ laneList do
 | SubKB.add( getAllInfo(lane) )
end
SubKB.add( SWRLRules )
return SubKB
```

Algorithm 1: Sub-Knowledge Base construction.

the knowledge base. Therefore, we create a temporal Sub-Knowledge Base (SubKB), which only contains nearby map information and traffic regulations.

Algorithm 1 describes the procedure of constructing a temporal *SubKB* based on current position. The current road segment (*currRS*) can be a lane or an intersection. At first, we get a list of road segments (*dirList*) directly connected to *currRS* using the following SPARQL query in function *getConnectedRS(currRS)*. Then, we get the connected road segments for each road segment (*rs*) in the *dirList* and put them all in the *rsList*.

```
SELECT DISTINCT ?connectedRS
WHERE {
{ currRS map:isLaneOf ?rs .
  ?rs map:isConnectedTo ?connectedRS. }
UNION
{ currRS map:isConnectedTo ?connectedRS. } }
```

For each road segment (*rs*) in the *rsList*, which are connected to *currRS* within 2-depth, we get all the triples of them and add them in the temporal *SubKB* using *getAllInfo(rs)*. If the road segment has lanes, we put all the lanes into the *laneList*. Then, for each lane in the *laneList*, we get all the triples of the lane instance and put them into the *SubKB*. The function *getAllInfo(rs)* retrieves all the triples of the instance *rs* from the knowledge base. We also add all the SWRL rules into the SubKB and return it as a temporal knowledge base for current position. In Fig. 4, if our vehicle is running on the lane B, the SubKB contains all the information of 2-depth connected lanes (B, C, D, E and lanes below F), road segments (B+C, D+E, and the road segment below F), an intersection A, a crosswalk F, and a lane adapter G.

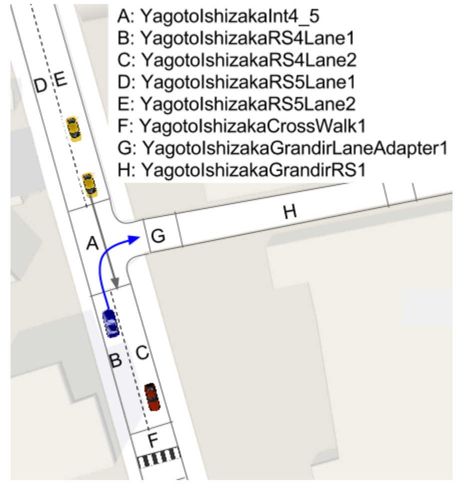


Fig. 4: Experimental area.

C. SWRL Rule Reasoner

When the autonomous vehicle receives a collision warning signal from a collision detection system, the SWRL rule reasoner is executed according to different types of traffic situations. The SWRL rule reasoner performs reasoning on the temporally created knowledge base SubKB rather than the whole knowledge base. According to Fig. 4, the following situations may occur:

- Before an intersection: Give way or move forward in comply with Right-of-Way rules.
- At an intersection: Stop and give way to the other cars when upcoming collisions are detected.
- On a two-way lane: Move to the left side and give way to the other cars coming from the opposite side of the two-way lane.

When the rule inferencing is performed by the SWRL rule reasoner, we use a SPARQL query to check whether we need to give way to the other cars or not. SWRL rules in the knowledge base are inferred with Pellet reasoner, which provides standard and cutting-edge reasoning services for OWL ontologies [13]. Pellet API[3] and OWL API[4] are applied for reasoning. According to surveys, Pellet is an open source, which supports SWRL rules and OWL API [14].

IV. Experiment

In this section, we discuss the experimental results of the improved fast decision making system. First, we will introduce the experimental area and the sensor data. We conducted experiments with real-world sensor data by driving an Estima car on the predefined driving path.

A. Experimental Area

The experiment started from Toyota Technological Institute (TTI) and ended at TTI resident apartment near Yagoto station in Nagoya city of Japan. The following experimental

[3]http://clarkparsia.com/pellet/
[4]http://owlapi.sourceforge.net/

TABLE II: An example of transmitted sensor data

Timestamp	Latitude	Longitude	Velocity (m/s)	Heading Angle	Car ID	Collision Warning
1712884	35.13467	136.9641	1.406401	-348.869	0	1
1712884	35.13444	136.9641	3.894356	194.5781	1	1
1712985	35.13467	136.9641	1.478781	-349.403	0	1
1712985	35.1345	136.9639	3.73306	194.0251	1	1
1713076	35.13467	136.9641	1.629769	-350.565	0	1
1713076	35.13473	136.9638	3.611545	194.9413	1	1
1713156	35.13467	136.9641	1.784153	-351.792	0	0
1713237	35.13467	136.9641	1.931167	-353.119	0	0
1713328	35.13467	136.9641	2.074262	-354.594	0	0
1713419	35.13467	136.9641	2.144296	-355.393	0	0
1713510	35.13468	136.9641	2.277657	-357.105	0	1
1713510	35.13482	136.9641	4.012344	194.3415	1	1
1713601	35.13468	136.9641	2.406614	-358.897	0	0
1713783	35.13468	136.9641	2.53862	-0.84693	0	0
1713874	35.13469	136.9641	2.669124	-2.92268	0	0
1713954	35.13469	136.9641	2.726286	-3.95872	0	0
1714045	35.13469	136.9641	2.837927	-6.08027	0	0
1714136	35.13469	136.9641	2.948412	-8.28217	0	0
1714227	35.13469	136.9641	3.055672	-10.6708	0	0

data is from one part of the Yagoto area, which contains both an uncontrolled intersection and a narrow two-way lane as shown in Fig. 4. A lane adapter (G: YagotoIshizaka-GrandirLaneAdapter1) connects an uncontrolled intersection (A: YagotoIshizakaInt4_5) and a two-way narrow road (H: YagotoIshizakaGrandirRS1).

B. Data

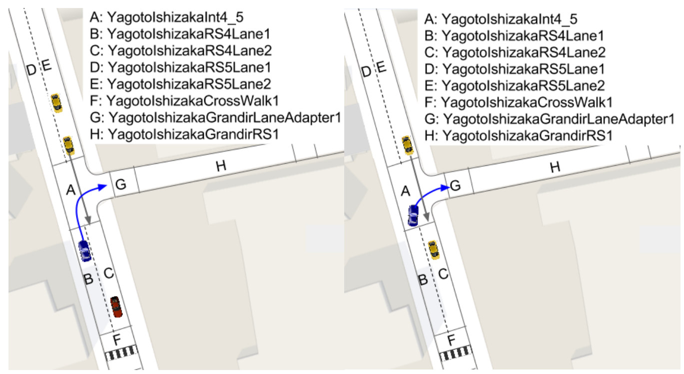
Table II shows the format of sensor data transmitted at real-time to the sensor receiver of decision making system. At each timestamp, the sensor data transmitter sends data in the order of timestamp, latitude, longitude, velocity, heading angle, carID, and collision warning signal. Here, if the collision warning signal is 0, it means that no upcoming collision is detected. Otherwise, it means that an upcoming collision is detected and sends the detected vehicle's information along with our experimental vehicle's information. The default ID for our experimental vehicle is 0 and the other detected vehicle's IDs are non-zero integers.

The instances in the semantic knowledge base for experiments are based on the map ontology and control ontology. The experimental vehicle is assigned a path file for experiments. The instances of other vehicles that have collision warnings with our experimental vehicle are added to the temporal SubKB at real-time and then deleted from the SubKB after driving decisions are inferred.

C. Real-World Data Experiment

To evaluate whether the decision making system can make correct decisions at real-time, we tested with the intelligent vehicle (Toyota Estima) using predefined driving path. The intelligent vehicle is equipped with many sensors such as Velodyne Lidar, GPS-IMU, and cameras.

The sensor data transmitter sends sensor data in the format as shown in Table II while the intelligent vehicle runs on the path as shown in Fig. 5. The path segments for the experiments are B $->$ A $->$ G $->$ H, and so on. Table III shows the decisions made in different timestamps according to the data in Table II. The SWRL reasoner of the decision making system is executed only when the vehicle receives a collision warning. From timestamp 1712884 until 1714045, our vehicle waits and gives way to the other vehicles until the warning is cleared for a specific time period. Here, we assume that the detected vehicles run straight from E (YagotoIshizakaRS5Lane2) to A (YagotoIshizakaInt4_5).



(a) Before YagotoIshizakaInt4_5. (b) On YagotoIshizakaInt4_5.

Fig. 5: Encountered situations during real-world experiment.

TABLE III: Experimental results with real-world data.

Timestamp	Estima Position	Detected Vehicle	Decision
1712884	YagotoIshizakaRS4Lane1	YagotoIshizakaRS5Lane2	Wait, Give Way
1712985	YagotoIshizakaRS4Lane1	YagotoIshizakaRS5Lane2	Wait, Give Way
1713076	YagotoIshizakaRS4Lane1	YagotoIshizakaRS5Lane2	Wait, Give Way
1713156	YagotoIshizakaRS4Lane1	N/A	Receive
1713237	YagotoIshizakaRS4Lane1	N/A	Receive
1713328	YagotoIshizakaInt4_5	N/A	Receive
1713419	YagotoIshizakaInt4_5	N/A	Receive
1713510	YagotoIshizakaInt4_5	YagotoIshizakaRS5Lane2	Wait, Give Way
1713601	YagotoIshizakaInt4_5	N/A	Receive
1713783	YagotoIshizakaInt4_5	N/A	Receive
1713874	YagotoIshizakaInt4_5	N/A	Receive
1713954	YagotoIshizakaInt4_5	N/A	Receive
1714045	YagotoIshizakaInt4_5	N/A	Receive
1714136	YagotoIshizakaInt4_5	N/A	Go
1714227	YagotoIshizakaInt4_5	N/A	Receive

In the following, we describe the decisions and situations in different timestamps.

- **Timestamp:** 1712884 $\sim$ 1713076

Situation: As shown in Fig. 5a, the intelligent vehicle is running on B (YagotoIshizakaRS4Lane1) and is going to run on the uncontrolled intersection A (YagotoIshizakaInt4_5). We detected a potential collision with another vehicle, which is running straight from E (YagotoIshizakaRS5Lane2) to A (YagotoIshizakaInt4_5).
Decision: Wait and give way to the other vehicle.

- **Timestamp:** 1713156 ∼ 1713237
  Situation: The vehicle is running slowly on the lane B (YagotoIshizakaRS4Lane1) and no collision warning is detected. But keeps waiting for the "GO" decision.
  Decision: Receiving sensor data.
- **Timestamp:** 1713328 ∼ 1713419
  Situation: The vehicle is running on the intersection A (YagotoIshizakaInt4_5) and no collision warning is detected.
  Decision: Receiving sensor data.
- **Timestamp:** 1713510
  Situation: As shown in Fig. 5b, the intelligent vehicle is running on the intersection A (YagotoIshizakaInt4_5), and the other vehicle is running straight on E (YagotoIshizakaRS5Lane2).
  Decision: Wait and give way to the other vehicle. (Our vehicle can move if the other vehicle stopped for a specific period,i.e. 500ms, or until receiving five non-collision-warning signals continuously)
- **Timestamp:** 1713601 ∼ 1714045
  Situation: The vehicle is running on the intersection A (YagotoIshizakaInt4_5) and no collision warning is detected.
  Decision: Receiving sensor data.
- **Timestamp:** 1714136
  Situation: We send Go decision if we don't receive collision warning in the following five continuous sensor data.
  Decision: Go.

As the experimental results shown above, the decision making system makes correct decisions at the uncontrolled intersection cases. By performing reasoning on SubKB, the calculation time is reduced comparing with the system which uses the whole knowledge base [8]. The whole knowledge base we currently use is about 407kb, while the size of a temporal SubKB is about 19kb ∼ 40kb. Therefore, the size of knowledge base for reasoning is reduced to about 1/20 ∼ 1/10.

Table IV shows the calculation time for making a decision using previous system introduced in [8] and our current decision making system with SubKB. As the comparison result shows, the decision making time depends on the size of the knowledge base. By only considering nearby road segments for decision making, we can reduce the calculation time by providing the same decisions. The average time for making a decision is about 53ms, which is close to the sensor data transmission duration.

TABLE IV: Comparison of decision making time.

	Whole Knowledge Base	Sub-Knowledge Base
Maximum	965ms	236ms
Minimum	305ms	37ms
Average	470ms	53ms

V. Conclusion and Future Work

We introduced a fast decision making system that accesses to the ontology-based knowledge base. The ontology-based knowledge base is constructed based on map, control, and car ontologies, and also includes traffic regulations written in SWRL. The decision making system can promptly make a driving decision by using only the temporally constructed Sub-Knowledge Base (SubKB), which avoids collisions by following Right-Of-Way traffic rules.

In future work, we will work on controlled intersections with various traffic situations. Furtheremore, we will extend the knowledge base by adding additional information and apply probablistic reasoning to solve traffic uncertainty problems to improve our decision making system.

References

[1] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic web revisited," *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 96–101, May 2006.
[2] D. Allemang and J. A. Hendler, *Semantic Web for the Working Ontologist - Effective Modeling in RDFS and OWL*, 2nd ed. Morgan Kaufmann, 2011.
[3] L. Zhao and R. Ichise, "Ontology integration for linked data," *Journal on Data Semantics*, vol. 3, no. 4, pp. 237–254, 2014.
[4] C. Gutierrez, C. A. Hurtado, and A. Vaisman, "Introducing Time into RDF," *IEEE Transactions On Knowledge and Data Engineering*, vol. 19, no. 2, pp. 207–218, 2007.
[5] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.
[6] *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, http://www.w3.org/Submission/SWRL/.
[7] L. Zhao, R. Ichise, S. Mita, and Y. Sasaki, "Core ontologies for safe autonomous driving," in *14th International Semantic Web Conference*, 2015, posters and Demos.
[8] L. Zhao, R. Ichise, T. Yoshikawa, T. Naito, T. Kakinami, and Y. Sasaki, "Ontology-based Decision Making on Uncontrolled Intersections and Narrow Roads," in *Intelligent Vehicles Symposium*. IEEE, 2015, pp. 83–88.
[9] R. Regele, "Using Ontology-Based Traffic Models for More Efficient Decision Making of Autonomous Vehicles," in *4th International Conference on Autonomic and Autonomous Systems*. IEEE Computer Society, 2008, pp. 94–99.
[10] S. Ulbrich, T. Nothdurft, M. Maurer, and P. Hecker, "Graph-based context representation, environment modeling and information aggregation for automated driving," in *Intelligent Vehicles Symposium*. IEEE, 2014, pp. 541–547.
[11] A. Armand, D. Filliat, and J. Ibañez-Guzman, "Ontology-Based Context Awareness for Driving Assistance Systems," in *Intelligent Vehicles Symposium*. IEEE, 2014, pp. 227–233.
[12] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao, *Describing Linked Datasets with the VoID Vocabulary*, W3C Recommendation, 2011, http://www.w3.org/TR/void/.
[13] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A Practical OWL-DL Reasoner," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.
[14] S. Abburu, "A Survey on Ontology Reasoners and Comparison," *International Journal of Computer Applications*, vol. 57, no. 17, pp. 33–39, November 2012.