PAPER *Special Section on Frontiers in Agent-based Technology*

Semantic Integration of Sensor Data with SSN Ontology in a Multi-Agent Architecture for Intelligent Transportation Systems

Susel FERNANDEZ[†,††a)], ***Nonmember and*** **Takayuki ITO**[†], ***Member***

SUMMARY Intelligent transportation systems (ITS) are a set of technological solutions used to improve the performance and safety of road transportation. Since one of the most important information sources on ITS are sensors, the integration and sharing the sensor data become a big challenging problem in the application of sensor networks to these systems. In order to make full use of the sensor data, is crucial to convert the sensor data into semantic data, which can be understood by computers. In this work, we propose to use the SSN ontology to manage the sensor information in an intelligent transportation architecture. The system was tested in a traffic light settings application, allowing to predict and avoid traffic accidents, and also for the routing optimization.

key words: *intelligent transportation systems, sensors, semantics, ontology, agents*

1. Introduction

Intelligent Transportation Systems (ITS) are the set of applications and technological systems created with the aim of improving safety and efficiency in road transport. These systems allow to control, manage and monitoring the different elements of roads.

The continuing evolution of ITS has ushered in a new era of interconnected intelligent systems, which certainly has been a quantitative leap in safety of road transportation. These systems enable the exchange of information between different applications, and the subsequent analysis allows improving the safety of drivers and eases travel and comfort in road travel.

One of the most important information sources in ITS are sensors. Sensors can be located into vehicles or as part of infrastructure elements, such as bridges, roads or traffic signs. Sensor can provide information related to the weather conditions and the traffic situation, which is useful to improve the driving process.

A sensor network is a network of tiny computers (nodes) equipped with sensors, which collaborate on a common task [1]. These nodes have certain sensory and wireless communications capabilities that enable ad hoc networking without any preset physical infrastructure or central administration. Sensor networks have been used in data acquisition and processing for multiple applications in different fields such as industry, medicine, home automation, military environments, environmental detection, etc.

Since one of the most important information sources in ITS are sensors, the integration and sharing of the sensor data become a big challenging problem in the application of sensor networks to these systems.

Due to its high degree of expressiveness, the use of ontologies is crucial to ensure greater interoperability among software agents and different applications involved in Intelligent Transportation Systems. Ontologies provide a common vocabulary in a given domain and allow defining, with different levels of formality, the meaning of terms and the relationships between them [2]. Ontologies also facilitate the design of exhaustive and rigorous conceptual schemes to allow communication and information exchange between different systems.

Although there are several studies that apply ontologies to intelligent transportation systems, most of them do not use information from sensors. Ontologies can give a semantic interpretation to the information collected by the sensors, facilitating safe driving and improving traffic performance.

In order to make full use of the sensor data, is crucial to convert that sensor data into semantic data, which can be understood by computers.

In this work, we propose to use the SSN ontology to manage the sensor information in an intelligent transportation architecture. Our proposal is a semantic integrator module to generate an instance of SSN ontology from the raw data collected by the sensors. The whole system performs the automatic traffic light settings allowing to predict and avoid traffic accidents and routing optimization.

The remaining of the paper is organized as follows. Section 2 presents the related work in the use of ontologies on ITS. The description of the SSN ontology is presented in Sect. 3. Section 4 describes the process of integration of the sensor data to semantic knowledge in SSN ontology. In Sect. 5 the use case of the traffic light setting system is shown. Some experiments are presented in Sect. 6. Finally, the conclusions and lines of future work are summarized in Sect. 7.

2. Related Work

In the last years, there has been an increasing interest on ontology for road transportation systems. In [3], an ontology to represent traffic in highways has been developed. Its aim was the construction of a reliable Traffic Information

Manuscript received January 26, 2017.
Manuscript revised June 8, 2017.
Manuscript publicized September 15, 2017.

[†] The authors are with the Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.
[††] The author is with the University of Alcala, Spain.
a) E-mail: susel.fernandez@uah.es

DOI: 10.1587/transinf.2016AGP0005

System providing information about roads, traffic, and scenarios related to vehicles in the roads. It also facilitates the Traffic Information System to analyze how critical a specific situation is.

In [4] the authors proposed a high-level representation of an automated vehicle, which can assists drivers in taking "unorthodox" but practical relaxation decisions. The principal weakness of this approach is the lack of rules representing prior traffic regulations. They just defined a set of traffic regulations violations, which allow classifying the given motion as "legal" or "illegal".

The work proposed in [5] is an approach to create a generic situation description for advanced driver assistance systems using logic reasoning on a traffic situation knowledge base. As a restriction, in this work, the passing destination road over the intersection has to be known for each vehicle, so modeling different possibilities according to the real situation of the intersection is not possible.

Regarding the use of ontologies with sensors information, we can mention the work in [6]. In this work, the authors propose an ontology for traffic management that combines both traffic concepts and general sensor ontology A3ME [7]. The added concepts are specializations of position, distance and acceleration sensor classes, and the different actions that take place in the car motions. The ontology is developed in OWL, using the JESS reasoner with SWLR [8] rules.

Despite the great effort in applying ontologies to ITS, integration and sharing the sensor data is still a big challenging problem in the application of sensor networks. In order to make full use of the sensor data, is crucial to convert the sensor data into semantic data, which can be understood by computers.

There have been some studies focused on transform the sensor data into semantic data for applications in different fields. Most of them have emerged in the field of linked sensor data, such as Esk Hydrological sensor network [9] and SENSEI [10], which focused on the transformation of the sensor data to linked sensor. The study of the semantic sensor network has been also applied in other research fields, for example, the semantic medical monitoring system model [11] based. These studies proposed different approaches to transform sensor data to RDF [12].

Most of the works mentioned above focused on mapping the sensor data to an existing ontology using a mapping language such as D2RQ [13], R2RML [14], [15] among others. In [17], they use D2RQ language and D2R server. RML [17] is an extension of R2RML and was designed as a general mapping language for heterogeneous data. Although these kinds of mapping languages are general for different data sources and different ontologies, the principal limitation is that they are not especially designed for the sensor data.

In the work in [18], they proposed a method to map sensor data to SSN ontology, translating the sensor data in databases into the RDF which conforming to SSN ontology. They first extract the sensor data from data sources, and manually annotate the sensor data using a mapping schema. Then a mapping file based on the mapping schema is generated, and finally the data in database is translated to RDF. In the final step, the system takes the mapping file as input, and use the predefined correspondences between elements of the mapping schema and SSN ontology to transform sensor data to RDF. The principal limitation of this work is that the process is not completely automatic. A human intervention is needed in the first step for the semantic annotation of the sensor data in data bases according to SSN ontology using a predefined mapping schema.

In [19] our preliminary idea for the semantic integration of the sensor data was presented. In this work, the sensor data is automatically translated to RDF, and then the semantic content is added according to SSN ontology through the annotation process manually.

3. The SSN Ontology

The Semantic Sensor Network (SSN) ontology was developed by the Semantic Sensor Networks Incubator Group [20] belonging to the World Wide Web Consortium (W3C). The ontology can model sensor devices, systems, processes, observations and knowledge of the environment. Figure 1 shows the principal concepts in the skeleton of the SSN ontology.

In this ontology the concept *stimulus* represents detectable changes in the environment. In fact, a stimulus is produced by a sensor.

Sensors transform an incoming stimulus into another. The *observable properties* of the sensors include their survival range or accuracy of measurement under defined external conditions. Many sensors need to keep track of time and location to produce meaningful results and, hence, are combined with further sensors to sensor systems such as weather stations [18]. In the case of ITS, the sensors can be located into the vehicles and into different infrastructure elements, for example as part of bridges, roads, signs, etc.

Observations act as the connection between incoming stimulus, the sensor, and the *sensor output*. *Properties* are qualities that can be observed by a certain type of sensors. *Features of Interest* are entities in the real world that are the target of sensing. The features have to be fixed by the sensing procedure. *Sensing* is a description of how a sensor works, i.e., how a certain type of stimuli is transformed into a digital representation, perhaps a description of the scientific method behind the sensor.

Sensing methods can also be used to describe how observations where made: e.g., how a sensor was positioned and used. *Sensor Output* represents the result of the observation. The results can act as stimuli or input for other sensors.

4. Semantic Integration

Integration and sharing the sensor data become a big challenging problem in the application of sensor networks. In

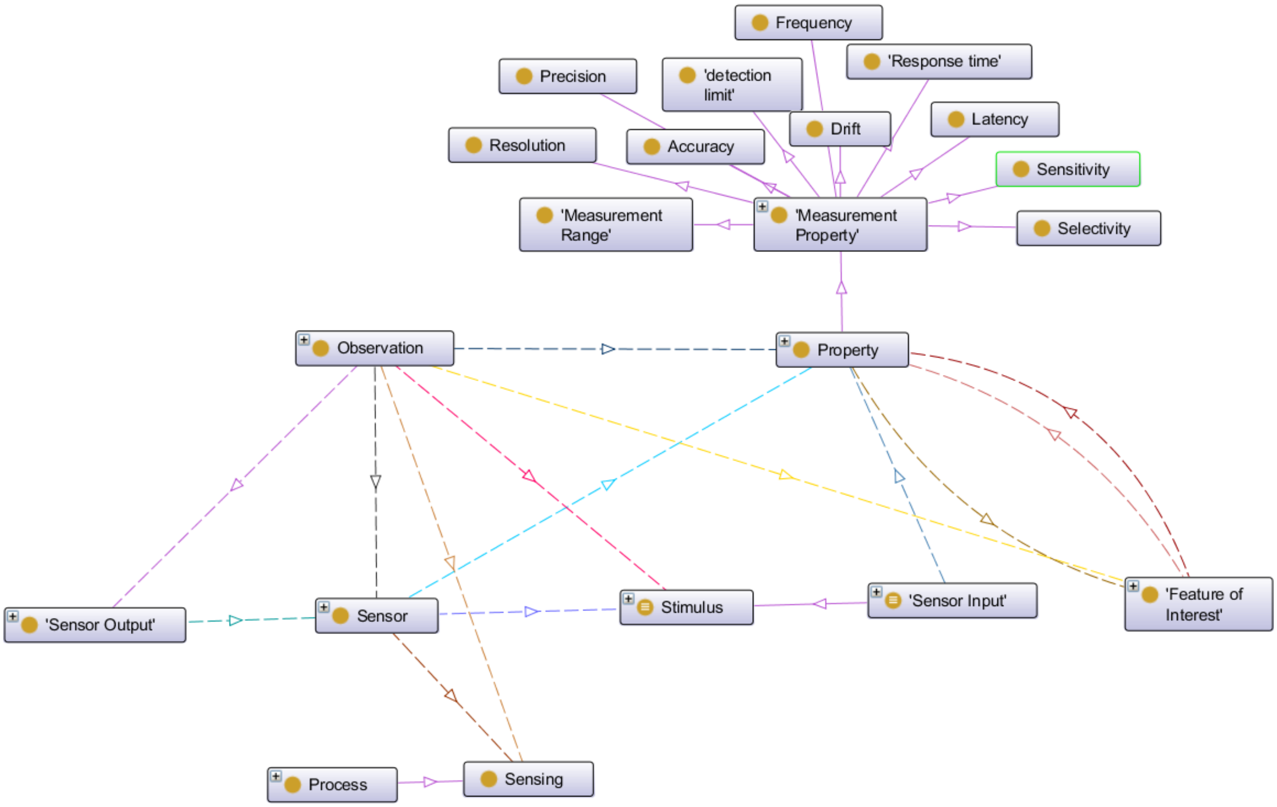

Fig. 1 Skeleton of SSN ontology

order to make full use of the sensor data, is crucial to provide sensor data with semantics.

In this work we proposed a method to map sensor data to SSN ontology in automatic way. Our proposal is a semantic integrator module to generate an instance of SSN ontology. This ontology will be use as knowledge representation form in a multiagent architecture for Intelligent Transportation Systems.

Figure 2 shows the structure of the semantic integrator. As we can see in the figure, the main steps for the semantic integration are:

1. *RDF translation*. In this step, the main task is to translate the information of the sensor data from the database, into a Resource Description Format (*RDF*) file. This step allows the further semantic treatment of the sensor information and the reusability.
2. *Mapping*. The principal objective of this phase is to automatically find correspondences between the RDF file of the sensor database with the elements of the SSN ontology.
3. *Alignment file*. In the final step, the system creates the alignment file, according to the correspondences found in the mapping phase. Using this file, the instance of the SSN ontology can be generated.

After the last step, the algorithm output represents an instance of the SSN ontology according to the sensor data. The instance of the SSN ontology provides semantic knowledge about the sensors to be used by the applications involved on ITS. The different phases of the semantic integration process are explained in more detail in next subsections.

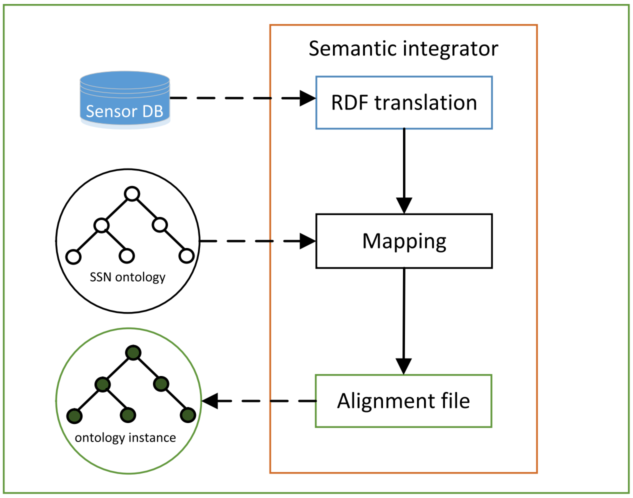

Fig. 2 Structure of the semantic integrator module

4.1 RDF Translation

The Resource Description Framework (RDF) [12] is a standard model for data interchange on the Web. It has features that facilitate data merging even if the underlying schemas

Table 1 Correspondence rules to translate the sensor information in data bases into RDF triples.

Database element	RDF subject	RDF Predicate	RDF Object
Table	Row_IRI	rdf:type property	Table_IRI
Literal	Row_IRI	Column_IRI	rdf:literal
Reference	Row_IRI	Column_IRI	Ref_Row_IRI

differ, and it supports the evolution of schemas without requiring all the data consumers to be changed. Therefore, the first step in our semantic integrator is to translate the information of the sensor data from the relational database into RDF.

The RDF data model [12] is based on making statements about resources (in particular web resources) expressions, known as triples, following a *subject–predicate–object* structure. The subject denotes the resource, and the predicate denotes traits or aspects of the resource, and expresses a relationship between the subject and the object.

In this work the translation to RDF is made following the principle that each database tuple (table row) produces a set of RDF triples (subject node, predicate or property arc and object node). In the process of translating the first step is create IRIs (*Internationalized Resource Identifier*) for identifying the tables, the columns in a table, and each row in a table. Then, the algorithm also uses a set of correspondence rules that are shown in Table 1, to create the RDF triples from the database elements.

As we can see in the table, there are three types of RDF triples that can be formed from database elements. These are:

- *Table triples*: Those RDF triples that are generated from each row in tables. In this case the predicate will be the rdf:type property; and the object, the IRI of the table.
- *Literal triples*: The RDF triples generated from each column with a non-null value, including the primary key. In this case the predicate will be the IRI of the column; and the object, the rdf:literal.
- *Reference triples*: The RDF triples which represents columns that constitute a foreign key and columns with non-null values in the row. In this case the predicate will be the IRI of the column; and the object, the IRI of the reference.

For example, if we have a sensor table, in which the primary key is *sensor_id*, and the rest of the attributes are: *sensor_location*, *observation_value*, and *observation_time*. One of the RDF triples that we can obtain from this table is shown in Fig. 3. This triple represents a specific observation value of a sensor. The subject is the primary key value (*sensor_id*), the predicate is the attribute *observation_value* and the object is the measured value, in this case 375.

4.2 Mapping

The second step in the semantic integration is to map the RDF sensor database to the SSN ontology. The ontology mapping can be defined in one sentence as follows: given two ontologies that describe a set of discrete entities, ontology mapping is the process of finding correspondences between them [21]. This process can be addressed by finding relationships between entities belonging to different sources ontologies and taking advantage of these relations. One of the key factors to consider for this process is to find similarity measures that allow to obtain precise correspondences between entities of different ontologies.

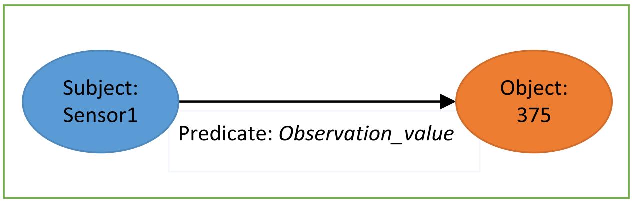

Fig. 3 Example of RDF triple

Table 2 Correspondences between the elements in the RDF sensor database and the classes and properties in SSN

RDF database	SSN element	Description
sensor	ssn:class:Sensing device	Sensor Identifier
measure	ssn:class:Property	type of measure of the sensor
observation_value	ssn:class:SensorOutput	output of the sensor
observation_time	ssn:class:Observation	time of the observation data
measure_unit	ssn:class:UnitOfMeasure	unit of the observation data
sensor_location	ssn:class:Deployment	Location of the sensor
has_measure	ssn:property:Observes	Type of the measure of the sensor
has_output	ssn:property:isProducedBy	The output of the sensor
has_observation_value	ssn:property:hasValue	The sensor output value
has_observation_time	ssn:property:ObservedProperty	The observation time
has_location	ssn:property:hasDeployment	The sensor location

Taking into account the imprecise nature of the similarity measures, we decided to use *FuzzyAlign* [22] for the matching process. It is a multi-layer fuzzy rule-based system which uses three different layers to perform the ontology alignment process.

On each layer, fuzzy logic techniques are used to combine different similarity measures to obtain more accurate alignments. *FuzzyAlign* combines basically two similarity measures for establishing correspondences between entities from different ontologies. The first measure is the terminological similarity, which takes into account the linguistic and semantic information in the context of the concept names, and the second one is the structural similarity, which uses both the taxonomic hierarchy of concepts and their internal structure, represented by their properties, types and cardinality restrictions.

Table 2 shows some of the correspondence found as result of the mapping process between the elements of the RDF sensor database and the classes of the SSN ontology.

4.3 Alignment File

The last step of the semantic integration is to build the alignment file. The alignment rules are constructed using the Alignment Format API [23]. The API was designed as a for-

```
<ruleml:imp>
   <ruleml:_body>
     <swrl:datavaluedPropertyAtom
          swrlx:property="http://..RDF_DB#has_location"/>
        <ruleml:var>x</ruleml:var>
        <ruleml:var>y</ruleml:var>
     </swrl:datavaluedPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
      <swrl:datavaluedPropertyAtom
           swrlx:property="http://..SSN#hasDeployment"/>
       <ruleml:var>x</ruleml:var>
       <ruleml:var>y</ruleml:var>
     </swrl:datavaluedPropertyAtom>
  </ruleml:_head>
</ruleml:imp>

<ruleml:imp>
   <ruleml:_body>
      <swrl:classAtom>
         <owllx:Class
            owllx:name="http://..RDF_DB #Sensor"/>
          <ruleml:var>x</ruleml:var>
      </swrl:classAtom>
   </ruleml:_body>
   <ruleml:_head>
      <swrlx:classAtom>
          <owllx:Class
            owllx:name="http://..SSN#Sensing_device"/>
          <ruleml:var>x</ruleml:var>
      </swrl:classAtom>
  </ruleml:_head>
</ruleml:imp>
```

Fig. 4 Fragment of the alignment file in SWRL

mat for expressing alignments in a uniform way. The goal of this format is to be able to share on the web the available alignments and to facilitate the comparison of the results of different alignment methods through the generation of outputs in different formats such as OWL and SWRL.

Figure 4 shows a fragment of the alignment file generated in one of the experiments performed in the testing phase of the system. In this example we can see two alignment rules between the RDF sensor's data base and the SSN ontology.

In this case we have used the SWRL format to express the output alignments. The first rule represents the alignment of the properties *has_location* and *hasDeployment* in the database and the SSN ontology respectively; and the second rule represents the alignment of the class *Sensor* in the database, and the class *Sensing_device* in SSN ontology.

5. The Automatic Traffic Light Setting Application

This scenario is related to the traffic light adjustment, taking into account the traffic flow and the weather conditions. Figure 5 shows the use case diagram of this task, in which three agents are involve: the traffic flow agent, the road agent and the weather agent.

To accomplish the traffic flow detection task, first the agent queries to the traffic ontology to obtain the information about the traffic situation. The description of the whole traffic scenario (e.g. roads, lanes, traffic signs and infrastructure elements) is provided by the traffic ontology, while the information concerning the sensors is provided by the SSN ontology. The different agents obtain the knowledge from the ontologies through SPARQL queries.

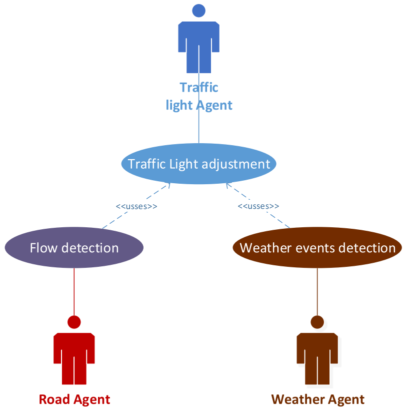

Fig. 5 Use case diagram of the traffic light setting task

As we explained before, the raw data collected by the sensors is translated to semantic knowledge and stored as an instance of SSN ontology to be processed by the agents. In this case, the road agent makes queries to the SSN ontology to retrieve the information about the number of vehicles detected by the current road sensor.

Taking into account the length of the lane and the number of vehicles detected by the road sensor, the traffic light agent can compute the real density of the lane. If the current density is greater or equal than the maximum density, it is assumed that lane is congested. If the traffic light is red, and the road segment is congested, then the duration time of the light is decreased; however, if the traffic light is red and the road segment is not congested, then the duration time of the light is maintained in the default value. If the traffic light is green and the road segment is congested, then the duration time of the light is increased, while if it is not congested, the duration time is maintained.

Figure 6 shows some of the most important elements (concepts, individuals and relations) present in the instance of SSN ontology, generated in this scenario. The right corner of the figure shows the types of the relations (arcs) identified with different colors. The principal relations identified are: *is_produced_by* (to describe a sensor observation); *observes*, *observed_by* (to connect the concepts *sensor*, *observation* and *property*); *has_value* (to connect the *sensor_output* and *observation_value*); and *observation_result* (the relation between *observation* and *sensor_output*). For example, in this view, at the time *Time1*, the sensing devices observe the *Car_flow* property, producing sensor outputs, with different values (*value1* to *value8*).

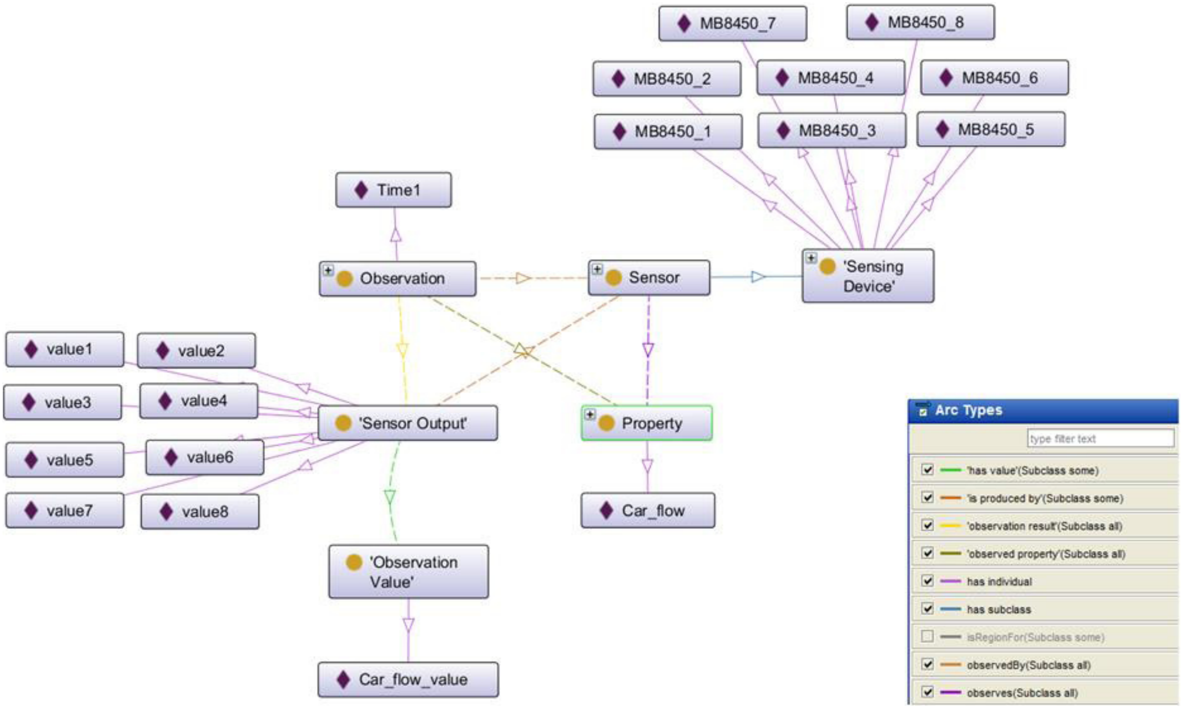

Fig. 6 Instance of SSN ontology

6. Experiments

The semantic integration system of sensor data with SSN ontology can be useful in several scenarios which involve sensors or sensor networks, such as industry, medicine, home automation, military environments, environmental detection, intelligent transportation systems (ITS), etc. In this work, we have focused in the specific field of ITS. We have performed several experiments by simulating different traffic scenarios to prove the automatic synchronization of the traffic light duration time.

The experiment phase was divided into two parts: the first one has the aim of testing the correctness and completeness of the semantic integration module and the second one is to test the full system performance. The results of both phases are presented below.

6.1 Semantic Integration Tests

There are many ways to qualitatively evaluate the returned alignment results. One possibility consists of proposing a reference alignment (R) that is the one that people must find. The result from the evaluated alignment algorithm (A) can then be compared to that reference alignment. The alignments A and R are considered to be sets of pairs.

The most commonly used and understood measures in this type of tests are coming from the information retrieval field. These are *Precision* and *Recall*, which have been adopted for ontology alignment too. They can be seen as extended versions of accuracy, a simple metric that computes the fraction of instances for which the correct result is returned.

When using *Precision* and *Recall*, the set of possible labels for a given instance is divided into two subsets, one of which is considered "relevant" for the purposes of the metric. Given a reference alignment R and the resultant alignment A, these similarity measures are defined as follows:
- *Precision* describes exactness. It is computed as the fraction of correct instances among those that the algorithm believes to belong to the relevant subset [24]. Given a reference alignment R, the precision of some alignment A is given by:

$$Precision(A, R) = \frac{|R \cap A|}{|A|}$$

- *Recall* describes completeness. It is computed as the fraction of correct instances among all instances that actually belong to the relevant subset [24]. Given a reference alignment R, the recall of some alignment A is given by:

$$Recall(A, R) = \frac{|R \cap A|}{|R|}$$

- *F-Measure* is used in order to aggregate the result of *Precision* and *Recall* [24]. Given a reference alignment R and a number α between 0 and 1, the *F-Measure* of some alignment A is given by:

$$F_\alpha(A, R) = \frac{Precision(A, R) \cdot Recall(A, R)}{(1 - \alpha) \cdot Precision(A, R) + \alpha \cdot Recall(A, R)}$$

The semantic integration system has been tested using different ontology matching algorithms to compare the results

Table 3 Results of the semantic integration of RDF sensor data with SSN ontology in terms of *Precision*, *Recall* and *F-Measure*.

Source1	Source2	System	Precision	Recall	F-Measure
RDF_DB	SSN	*FuzzyAlign*	0.85	0.92	0.88
		CODI	0.88	0.50	0.64
		SOBOM	0.66	0.39	0.49

Table 4 Overall results of the experiments in simulation.

Number of experiments	200
Number of vehicles	350
% gain time	80%
Average gained time (seconds)	137
% not gain time	22.8%

and choose the one with better performance in terms of *Precision*, *Recall* and *F-Measure*. The chosen alignment methods to be compared were CODI [25] and SOBOM [26].

Table 3 shows the result of the three systems in the integration between the *RDF* sensor database and the SSN ontology. In the case of *F-Measure* we have used $\alpha = 0.5$ (This is the *Harmonic Mean* of *Precision* and *Recall*). The reference alignments were defined manually by us at the beginning of the tests and were the same for the three used alignment algorithms to guarantee the same conditions in the evaluation.

As we can see in the table, the values of *Precision*, *Recall* and *F-Measure* obtained by the system with *FuzzyAlign* were very good. Although the best results were obtained by CODI in terms of *Precision*, which means that CODI was better in exactness, *FuzzyAlign* was the best in *Recall* (completeness) and *F-measure* (overall). That means that the system with this alignment algorithm can detect most of the existing alignments with high rate of accuracy and the best overall performance.

6.2 Full System Tests

The second testing phase is to verify the performance of the full system in simulation. The simulation module, taking into account the location and speed of the vehicle, and also the length of each lane segment and the traffic light duration time, can estimate the arrival time of each vehicle to destination, according to the route.

The description of the whole traffic scenario (e.g. roads, lanes, traffic signs and infrastructure elements) is provided by a traffic ontology [27], while the information concerning the sensors is provided by the SSN ontology. The different agents obtain the knowledge from the ontologies through SPARQL queries.

In the simulation phase 200 experiments were conducted, with 350 vehicles for which different traffic scenarios were defined. These scenarios were composed of 55 routes within a radius of 10 km, with 10 intersections regulated by traffic lights (40 traffic lights). For each experiment, the chosen initial configuration for traffic lights, and the position and route of the vehicles was random.

Table 4 shows the results obtained in the experiments. The table also shows the % of vehicles that gained time using the automatic synchronization of traffic lights; the average of time gained by vehicle; and the % of vehicles that did not gain time. In the experiment, the total time from the origin to destination for each car was considerably less in the case of intelligent traffic light adjustment.

As we can see in the table, 80% of vehicles experienced a gain of time, and the average value of gained time was approximately 137 sec. No vehicle experienced a loss of time in the experiments, while vehicles that did not gain time (22.8%) were those who moved along uncongested routes.

7. Conclusions and Future Work

In this work, we propose the use of the SSN ontology to manage the sensor information in an intelligent transportation architecture. The work focused on the translation of the raw data collected by the traffic flow sensors into semantic knowledge to be stored and processed as instance of SSN ontology. Our proposal is a semantic integrator module to generate an instance of SSN ontology from the raw data collected by the sensors. This semantic integration is performed in three principal steps: the first one is the automatic translation of the sensor information in databases into RDF, the second one is the automatically mapping of the RDF sensor data with the elements of the SSN ontology; and the then create the alignment file, according to the correspondences found in the mapping phase. The semantic integration system was applied as part of an automatic traffic light settings application, with the objective of predict and avoid traffic accidents and perform routing optimization on ITS.

The tests of the semantic integration module showed good results in terms of *Precision*, *Recall* and *F-Measure*, being able to align most of the elements of the databases with the SSN ontology. The success of the semantic integration allows the semantic treatment of the sensor data to be used for the different applications, enhancing their functionalities and capabilities to make decisions.

In the case of the full system performance, the experiments demonstrated that the intelligent traffic light settings using ontologies and taking the traffic flow data into consideration can improve the traffic circulation. In case of congestion, vehicles have to wait for shorter periods of time at red lights, while they can move for longer periods of time with green lights. This adjustment gives higher priority to congested traffic segments over the rest of the traffic, allowing to avoid traffic jams.

The principal advantage of this approach is that allows to maximize the intelligence of ITS through the semantic annotation of the data collected by the sensors to be understood by the applications involved on ITS.

However, the process of semantic annotation of the sensor data is too difficult and still needs some human participation. Thus, as future work, we are planning the design of a rule based system combined with machine learning techniques to improve the semantic annotation process, reducing the human intervention in the semantic annotation phase.

Taking into account the versatility of the system, another line of future work is to make some experiments in other scenarios such as Smart Homes and some other IoT applications.

Acknowledgments

This work was partially supported by Research and Development on Utilization and Fundamental Technologies for Social Big Data by NICT (National Institute of Information and Communications Technology), and the Fund for Strengthening and Facilitating the National University Reformations by Ministry of Education, Culture, Sports, Science, and Technology, Japan.

References

[1] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, Wireless sensor and actuator networks: technologies, analysis and design. Academic Press, 2010.

[2] R. Studer, V.R. Benjamins, and D. Fensel, "Knowledge engineering: principles and methods," Data & knowledge engineering, vol.25, no.1, pp.161–197, 1998.

[3] G. Sérgio and S. Ícaro, "An Ontology for a fault Tolerant traffic Information System," Proc. 22nd International Congress of Mechanical Engineering (COBEM 2013), Ribeirão Preto, SP, Brazil, Nov. 3–7, 2013.

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

[5] M. Hülsen, J.M. Zöllner, and C. Weiss, "Traffic intersection situation description ontology for advanced driver assistance," 2011 IEEE Intelligent Vehicles Symposium (IV), pp.993–999, June 2011.

[6] A.J. Bermejo, J.E. Villadangos, J.J. Astrain, A. Córdoba, L. Azpilicueta, U. Garate, and F. Falcone, "Ontology Based Road Traffic Management," IDC, pp.103–108, Aug. 2012.

[7] A. Herzog, D. Jacobi, and A. Buchmann, "A3ME - an Agent-Based middleware approach for mixed mode environments," 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM'08, pp.191–196, Sept. 2008.

[8] SWRL, Semantic Web Rule Language, Available online: http://www.w3.org/SWRL/ (Access on June 10th, 2015)

[9] R. Dutta and A. Morshed, "Performance Evaluation of South Esk Hydrological Sensor Web: Unsupervised Machine Learning and Semantic Linked Data Approach," Sensors Journal, vol.13, no.10, pp.3806–3815, 2013.

[10] J. Jelschen, "SENSEI: Software evolution service integration," 2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), pp.469–472, 2014.

[11] G. Zhang, C. Li, Y. Zhang, C. Xing, and J. Yang, "SeanMedical: A kind of semantic medical monitoring system model based on the IoT sensors," 2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom), pp.238–243, 2012.

[12] Resource Description Framework (RDF): Concepts and Abstract Syntax. W3.org.

[13] Accessing Relational Databases as Virtual RDF Graphs. http://d2rq.org/. LastVisited, 2016.

[14] R2RML: RDB to RDF mapping language. http://www.w3.org/TR/r2rml/. W3C Member submission. LastVisited, 2014.

[15] C. Dong, X. Zhang, and W. Zhou, "An R2RML-based Mapping System from Metal Materials Database to Ontology," 2013 Ninth International Conference on Semantics, Knowledge and Grids (SKG), pp.186–189, 2013.

[16] A. Moraru, C. Fortuna, and D. Mladenic, "A System for Publishing Sensor Data on the Semantic Web," Journal of Computing and Information Technology, vol.19, no.4, pp.239–245, 2011.

[17] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle, "RML: a generic language for integrated RDF mappings of heterogeneous data," Proc. 7th Workshop on Linked Data on the Web (LDOW2014), 2014.

[18] X. Zhang, Y. Zhao, and W. Liu, "A Method for Mapping Sensor Data to SSN Ontology," International Journal of u- and e- Service, Science and Technology, vol.8, no.6, pp.303–316, 2015.

[19] S. Fernandez and T. Ito, "Using SSN Ontology for Automatic Traffic Light Settings on Intelligent Transportation Systems," 2016 IEEE International Conference on Agents (ICA), pp.106–107, Sept. 2016.

[20] W3C. Available online: http://www.w3.org/2005/Incubator/SSN/ (Access on Sept. 5th, 2015)

[21] J. Euzenat and P. Shvaiko, Ontology Matching; Springer: Berlin, Germany, 2007.

[22] S. Fernández, I. Marsa-Maestre, and J.R. Velasco, "Performing Ontology Alignment via a Fuzzy-Logic Multi-layer Architecture," Knowledge Discovery, Knowledge Engineering and Knowledge Management, vol.415, pp.194–210. Springer, 2013.

[23] J. Euzenat, An API for ontology alignment (version 2.1), Montbonnot, France, INRIA Rhône-Alpes, 2006.

[24] V.C.J. Rijsbergen, Information Retrieval, Butter-worths, Second Edition, London, 1979.

[25] J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt, "Leveraging terminological structure for object reconciliation," Semant. Web: Res. Appl., vol.6089, pp.334–348, 2010.

[26] P. Xu, Y. Wang, L. Cheng, and T. Zang, Alignment Results of SOBOM for OAEI 2010, Proc. 5th International Workshop on Ontology Matching (OM 2010), Shanghai, China, 7–11 Nov. 2010.

[27] S. Fernandez, R. Hadfi, T. Ito, I. Marsa-Maestre, and J.R. Velasco, "Ontology-Based Architecture for Intelligent Transportation Systems Using a Traffic Sensor Network," Sensors 2016, vol.16, no.8, 1287, 2016.

Susel Fernandez received her B.S. and M.S. degrees in Computer Science from The University of Oriente, Cuba, and the Ph.D. from the University of Alcala, Spain in 2013. Now she is working as assistant professor at the Nagoya Institute of technology, Japan. Her main research interest focuses in applications of Artificial intelligence, multiagent systems, and knowledge based systems, especially ontologies.

Takayuki Ito is Professor of Nagoya Institute of Technology. He received the B.E., M.E., and Doctor of Engineering from the Nagoya Institute of Technology in 1995, 1997, and 2000, respectively. His main research interests include multi-agent systems, intelligent agents, collective intelligence, social computing, crowd science and engineering, group decision support systems, agent-mediated electronic commerce, and software engineering on offshoring.