

A Novel Deep Learning Approach for Semantic Interoperability Between Heterogeneous Multi-Agent Systems

Noureddine El Abid Amrani

SSDIA Laboratory, ENSET Mohammedia,
Hassan II University, Casablanca, Morocco
and IGA, Casablanca, Morocco
noureddine.elabid@iga.ac.ma

Oum El Kheir Abra

SSDIA Laboratory, ENSET Mohammedia,
Hassan II University
Casablanca, Morocco
abra.oek@gmail.com

Mohamed Youssfi

SSDIA Laboratory, ENSET Mohammedia
Hassan II University,
Casablanca, Morocco
med@yousffi.net

Omar Bouattane

SSDIA Laboratory, ENSET Mohammedia
Hassan II University,
Casablanca, Morocco
omar.bouattane@gmail.com

Abstract—This article focuses on the issue of semantic interoperability in heterogeneous distributed multi-agent systems. Existing middleware technologies offer programming models that strongly combine agents' learning models and communication models, which can lead to performance weaknesses when the number of agents is very important. Moreover, existing methods in the field of semantic interoperability solve the problem of understanding messages exchanged between distributed agents with heterogeneous ontologies, using several techniques to combine these ontologies. The first category of these methods relies on the fusion principle, others use alignment, and finally, there are those founded on Semantic Web technique. All these methods are limited to abstract concepts and do not deal with concrete concepts such as those represented by images. We propose in this paper a new approach that addresses the problem of semantic interoperability between heterogeneous distributed agents based on two principles: At first, the communication aspect of the agent from the learning aspect is separated. Then, we propose extending semantic interoperability to concrete concepts by combining two techniques: Semantic Web technology, which allows terms representing abstract concepts to be interpreted and deep learning technology, which is introduced as a new method to ensure semantic interoperability in the case of concrete concepts such as images. A detailed description of the proposed approach is provided, showing that it is very useful in solving the disadvantages of existing multi-agent platforms.

Keywords—multi-agent system; middleware; semantic interoperability; ontology; web semantic; deep learning

I. INTRODUCTION

Multi-agent systems (MASs) are an interesting solution for the modeling of a complex and massively distributed system.

The distributed elements of such a system need to communicate and interact with each other despite their technological heterogeneity. Interoperability between distributed information systems has been the subject of several studies. It is first concerned with technological interoperability (transport of messages from one system to another and necessary communication protocols for exchanging messages). Two solutions can be used to satisfy this need: (a) Web services [1], agents can publish their abilities as invocable Web services by other agents and this independently of the conceptual (architecture, interaction model, etc.) and technical characteristics of these agents (platform, programming language, etc.). (b) Middleware technology. Indeed, the only mode of communication between heterogeneous multi-agent systems lies in sending messages asynchronously. So a Message Oriented Middleware (MOM) can be used as a broker to transport messages between two heterogeneous platforms. Based on the comparative in [2], we have opted for middleware using the AMQP protocol such as RabbitMQ and excluded other solutions such as JMS which is monolingual and therefore does not ensure any interoperability, and ActiveMQ which may be an open source solution, but authors in [2, 3] confirmed the presence of unidentifiable problems at the queue level.

Interoperability between heterogeneous MASs also concerns semantic interoperability, which means interpretation of exchanged messages between different agents at the meaning level. This problem has been raised by many organizations such as the Foundation for Intelligent Physical Agents (FIPA) [4]. FIPA provides standards and requirements needed to ensure interoperability between multilevel MASs: FIPA-ACL [5] at the communication level, DAML + OIL [6]

Corresponding author: Noureddine El Abid Amrani

at the semantic level, AAA (Abstract Agent Architecture) at the architectural level, etc. FIPA has also encouraged the development of different platforms within the specifications it has proposed such as JADE [7] and ZEUS [7]. These middleware offer programming models that strongly couple agent learning models and their communication models. In the majority of cases, when the number of agents is very important, the communication models used have drawbacks in term of performance. Furthermore, various studies have proposed ontologies as a mean to solve the semantic interoperability problem between heterogeneous agents. Most of these approaches have presented methods for interpreting the conceptualizations of a given ontology based on mapping, merging [8-10] or aligning ontologies [11-13].

Nevertheless, the limitations of these approaches lie in the fact that they rely on interpretation or translation techniques such as WordNet, allowing to search for synonyms that represent a concept or an ontological relationship. The case of concepts represented by concrete objects such as images, to the best of our knowledge, has not yet been processed. It would be interesting to be able to deal with concrete objects because of the spectacular evolutions of information technologies: Internet of Things (IoT), mobile applications and social networks. In this context, a comprehensive approach is proposed, which addresses the problem of interoperability between heterogeneous agents by using semantic web technology to interpret terms representing abstract concepts and Deep Learning technology to recognize and interpret concrete concepts such as images. The recognition and interpretation of image-type objects would be useful in the automated management of fields like road traffic, for the interpretation of road signs by driverless vehicles, medical imaging diagnostics for anomaly detection, etc.

II. RELATED WORK

Semantic interoperability between MASs solves the problem of understanding messages exchanged between agents with heterogeneous ontologies. In order to improve semantic interoperability in heterogeneous distributed systems, many techniques for combining these ontologies have been introduced and can be classified into three main categories: First, we find those based on the principle of fusion, others use alignment, and finally those based on the Semantic Web. In this section, we present a brief overview of these techniques:

A. Ontology Fusion

This technique is typically used when source ontologies cover similar or related domains. The principle is to merge source ontologies to produce a new coherent ontology that represents the union of these ontologies. In [8], each agent has his own ontology created by its owners. When agents meet other agents with ontologies in the same domain, they exchange their ontologies and decide to merge them. This merge is performed as follows: identical concepts, syntactic or semantic concepts already known to the current agent are identified. The syntactical comparison is performed using a syntactic similarity measure using the Levenshtein distance modification method. The semantic comparison is performed using the WordNet dictionary. Once all similar concepts are

identified, various relationships between these concepts and already known concepts are created (by merging with their corresponding classes). Then, for all the unknown concepts of the agent at the moment, they are incorporated into the knowledge base and learn new relationships between them and already understood concepts. Finally, to link the classes added to the ontology with the classes that already exist, WordNet is still used to identify the types of relationships that should be used such as the relationships of hypernymia (super-class), hyponymy (sub-class), holonymy (is-part-of), meronymy (has-for), etc. Authors in [9] presented a method of fusion of heterogeneous ontologies to build a more complex ontology, based on the WordNet tool. The principle of this method is to merge heterogeneous domain ontologies according to a methodology that includes four main steps: WordNet mapping, selection of concepts, similarity calculation and rebuilding hierarchy. Using the values obtained in the previous steps, the two domain ontologies are merged in a newly rebuilt hierarchy.

B. Ontology Alignment

The ontology alignment takes two input ontologies and outputs the result of the alignment between the incoming ontology entities. This technique is typically utilized when source ontologies cover complementary application domains. It is about establishing semantic links between these ontologies so that the one uses the knowledge of the other. There are several ontology alignment algorithms. One of the most used techniques is the approach in [10] where the authors propose an algorithm working on distributed agent's knowledge bases to establish ontology correspondences. This algorithm is applied at the reception level of each message exchanged between two agents. The receiving agent will bring into play the different mechanisms to understand the message it receives and updates his ontology by adding the new knowledge. It begins by checking, in his ontology, the existence of the terms necessary for the interpretation of a message. In case of failure, the receiving agent transmits all unknown terms to the issuing agent for clarification. The latter searches for synonyms in its own ontology and sends clarifications on the terms found. If unknown terms persist, the receiving agent may request an external text service such as WordNet to obtain other synonyms and other words to correctly interpret the message. Finally, the receiving agent can introduce new terms into his ontology in order to update it. Authors in [13] present a dynamic multi-strategy ontology alignment framework, named RiMOM. The key insight in this framework is that similarity characteristics between ontologies may vary widely. They propose a systematic approach to quantitatively estimate similarity characteristics for each alignment task and propose a strategy selection method to automatically combine matching strategies based on two estimated factors while they take into account the textual and structural characteristics of ontologies.

C. Ontology Semantic Web Techniques

Ontology semantic web methods use the semantic Web RDF and OWL languages to implement source ontologies and propose techniques to align or merge. Thereby, in [11] the authors use the OWL semantic Web language to implement two source ontologies and propose an algorithm that receives two input (.owl) files and applies a 4-step strategy: lexical

alignment, semantic alignment, verification of similarity, and heuristic functions. To eliminate unnecessary comparisons, the authors propose browsing the first file from top to bottom and the second from bottom to top. First, the output file is initialized by the first ontology O1. Then, a lexical analysis between two classes C2 of O2 and C1 of O1 is launched. In case of failure of the lexical analysis, the semantic analysis is executed instead. When C2 of O2 is compared to all C1 classes of O1, a heuristic function is called to write the output file. To solve the data translation problem, authors in [14] propose the "Processing ontology alignments with SPARQL" approach, where the SPARQL query language is used for extracting data from ontology, and through its CONSTRUCT statement, for generating new data. To deal with the limitations of this language, the authors propose to combine two recent extensions of SPARQL in order to be able to transform data according to complex alignments: SPARQL++ provides aggregates, value generating built-ins and (possibly recursive) processing of mappings expressed in SPARQL and PPARQL providing queries on path expressions (made from regular expression patterns) which are sufficient for expressing those of the expressive language.

D. Conclusion

In this section, we presented a brief review of the literature of some existing methods for interpreting ontology conceptualizations such as fusion, alignment, and semantic web. Other approaches rely on interpretation or translation techniques to search for synonyms that represent a concept or an ontology relationship. The main goal of these approaches is the semantic interpretation of concepts, while most of them are based on a concept-to-concept negotiation technique. The scope of these methods is restricted to term type concepts. This bibliographic study shows that the existing techniques of interpreting concepts of a given ontology do not deal with those represented by concrete objects such as images. With the continual progress of the IoT, mobile applications and social networks, we believe that the use of image-like concepts, such as in the medical field and road safety, is booming. In this context, we consider that the proposed approach for semantic interoperability would be useful for heterogeneous distributed applications exchanging image-type information. The proposed approach therefore addresses the problem of interoperability between heterogeneous agents by using semantic WEB methods to interpret terms representing abstract concepts and Deep Learning methods to interpret objects representing concrete concepts.

III. MAIN CONCEPTS AND TOOLS OF OUR APPROACH

In the following, we give some definitions of the main concepts and tools utilized in the design of our approach, and then we focus on the description of the two technologies we adopted to ensure semantic interoperability: Semantic Web and Deep Learning.

A. Main Concepts

1) Agent

An agent can be a physical or virtual entity that can act, perceive its environment (in a partial way) and communicate

with others, is autonomous and has skills to achieve its goals [15].

2) Multi-Agent Systems

A MAS is defined as: A multi-agent system is a network of problem-solving entities (agents) that work together to find answers to problems that exceed the individual capabilities or knowledge of each entity (agent). A MAS agent must be able to interpret messages sent by other agents, based on his knowledge, which is considered as a certain knowledge and knowledge of these agents. In this way, this technique allows the agent to acquire the skills necessary to interpret new knowledge and understand messages sent by other agents [16].

3) Agent Communication Language

In an MAS, agents are heterogeneous because each agent must meet very specific needs of an application. Therefore, the agent implementation is dependent on its field of application. To ensure reliable cooperation between heterogeneous agents, they have to communicate their needs to each other through a Agent Communication Language (ACL). Such a language must follow a well-defined syntax so that agents can interpret the messages in the same way. Two most commonly used languages in MASs are KQML (Knowledge Query and Manipulation Language) [17] and FIPA-ACL. FIPA-ACL provides a set of messages with associated semantics, i.e. conditions that the sender must respect and expected effects on the recipient.

```
(Inform
:Sender Sign agent
:Receiver Conductor Agent
:Language XML
:Ontology SignTrafficOnto
:Content (<TrafficSign name="A14">
  <GeomForm> triangle </GeomForm>
  <BgColor> wihte </BgColor>
  <BorderColor> red </BorderColor>
  <UrlImg> http://.../A14.png </UrlImg>
</TrafficSign>
)
)
```

Fig. 1. An example of a FIPA-ACL code

4) Ontology

Ontology is a formal specification of a shared conceptualization [18]. In MAS, ontology is a description of the concepts and relationships that can exist for an agent or a community of agents. In our approach, ontology represents a specific vocabulary to describe a conceptualization at the sending level. This vocabulary will be used to interpret a message at the recipient's level.

5) RabbitMQ Broker

RabbitMQ [2] is an open source message broker based on the Advanced Message Queuing Protocol (AMQP). A message broker is an architectural model for the validation, transformation and routing of messages. It gives applications a common platform to send and receive messages and a safe place to store messages until they are received. AMQP is a message protocol that manages traffic between producers and consumers. Producers provide the messages and consumers

retrieve them and process them. RabbitMQ is responsible for ensuring that messages reach the right consumers. To do this, RabbitMQ uses two key components: exchanges and queues. The main function of AMQP is the routing of messages using its different exchange modes as required, independently of the TCP/IP, HTTP, and SMTP transport protocols.

B. Semantic Web

Authors in [19], define Semantic Web as follows: “The Semantic Web will bring structure to the meaning full content of the web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users”. The WWW was developed for humans. Documents on the web are machine readable but not machine understandable. The main aim of Semantic Web is to enrich documents with semantic information about the content and to develop powerful mechanisms capable of interpreting this information. These goals are achieved through implementation of models, standards as well as annotation of resources at layers [20]. The Semantic Web uses tools to describe document content, such as the Resource Description Framework (RDF), which is an entity relationship model used to represent resource information on the World Wide Web. The basic principle is that everything on the Web can be unique, identified with URIs (Uniform Resource Identifiers) and then described in terms of triples representing resources, their properties and values. For Semantic Web purposes, RDF properties can be organized into classes and subclasses, with attributes and values. Languages such as RDFS, DAML+OIL, or recently OWL, allow a complete description of the complex ontological relationships between RDF properties, in an RDF/XML format.

C. Deep Learning

The problem of learning visual information is generally categorized in image classification, location and detection of objects, semantic segmentation and instance segmentation, etc. (machine learning before Deep Learning). Deep Learning [21] is a learning technique that allows a machine or a program to give a semantic and exact interpretation at the meaning level of an image, a video or understand the spoken language. It can be supervised as the learning techniques used in the AI domain, but the structure is different, it is a complex system composed of thousands of units, the neurons, which perform each of the elementary calculations. The specificity is that the results of a layer of neurons will serve as input to the calculation of the next layer. As it can be unsupervised, the technique used relies on data that is not labeled. The interpretation of visual information has been approached in several ways [22] but the goal remains the same: examine images for the purpose of identifying objects and judging their meaning.

IV. THE PROPOSED APPROACH

This work presents a contribution to the development of a new asynchronous communication middleware model adapted to massively distributed architectures based on micro-agents. The proposed middleware should provide an API, with

different multi language implementations, which allows exchanging messages between micro-agents implemented with different languages while respecting MAS standards recommended by FIPA. In this context, our goal is to ensure interoperability between agents deployed in heterogeneous multi-agent platforms based on existing protocols such as AMQP. To ensure technological interoperability, we chose to use a Message Oriented Middleware (MOM) that relies on the AMQP protocol, such as RabbitMQ broker. With AMQP protocol, RabbitMQ ensure the transport of end-to-end messages independently of platforms, languages and communication protocols such as HTTP, SMNTP, etc.

Below, focus is given on the communication aspect of an agent and its learning aspect to achieve our main goal, which is semantic interoperability. The model we propose is built on the principle of separating communication aspect and learning aspect of an agent (an extract of class diagram in Figure 2). This allows agents to adapt to expected changes and to integrate future learning solutions.

With this middleware, it is possible to create and deploy heterogeneous agents evolving in distributed containers. To communicate with the other agents of the platform, each agent can use polymorphic communication models adapted to its context. To achieve this, we use the pattern “Strategy” to define a communication adapter family. Each adapter, relating to a given communication protocol, is encapsulated. These adapters are dynamically interchangeable and offer an open model for expansion. The developer using this middleware can thus create his own adapter according to the communication model that he wishes to implement. Currently, we think it is appropriate to create implementations of messaging adapters for exchanging ACL messages and messaging protocols like AMQP, STOMP and MQTT using the RabbitMQ broker.

Using the same principle adopted for the communication aspect, we use a polymorphic and scalable learning model based on the “Strategy” pattern. An agent can use at any time any learning model among the “AbstractLearningStrategy” implementations such as Deep learning, Semantic web, etc. depending on its context. We therefore, ensure an open extension middleware by offering scalability for the community of developers using it. To achieve this objective, we must implement an ontology adapted to this context. Ontology is a vocabulary in a given field presented as a set of related concepts which can be either concrete or abstract. A concrete concept is represented by a concrete object such as a term, an image, etc. An abstract concept can only be represented by a general term such as “vehicle”. Hence, ontology can be considered as a graph of nodes and arcs (relations). A node can be seen as a set of words that designate the same concept, such as a car that is represented by the set {car, automobile ...}. To interpret a message sent by an agent, we need a system with a tool to implement such ontology and another tool to query the implemented file to answer a question about an incomprehensible word. To do this, we investigate two technologies:

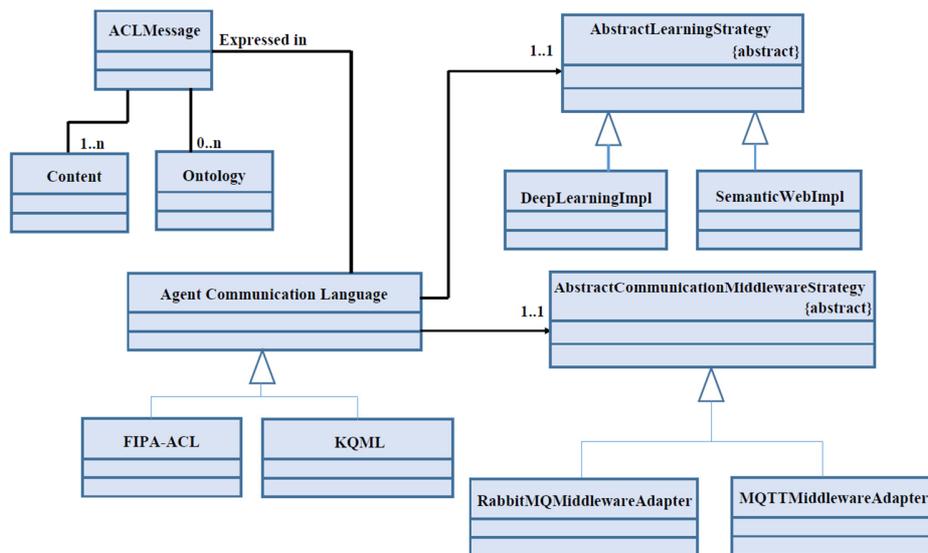


Fig. 2. An extract of class diagram of our approach

- Semantic Web which is a technology that has appropriate languages to meet our needs. RDF, RDFS and OWL languages are used to implement the agent ontologies and the SPARQL language is used to query the implemented files.
- Deep Learning is a technique based on a trained neural network to recognize an object (image) that represents a concrete concept. The outputs obtained are produced in the form of terms. Among the obtained results, it is possible to find terms that the agent cannot understand. In this case, the Semantic Web will be used for the correct interpretation of these terms.

So, our strategy (Figure 3) is: the semantic web technology languages can be used to solve the problem of semantic interoperability in case of abstract concepts. In fact, RDF, RDFS or OWL languages can be used to implement source ontologies at sending level and SPARQL queries can query these ontologies to determine the list of synonyms for the purpose of interpreting these concepts. And in the case of a concrete concept represented by an image for example, a trained neural network is implemented to recognize this concept.

A. Architecture of the Proposed Approach

Our proposed approach scheme of interoperability is depicted in Figure 4 according to two levels:

- Mediation level is the intermediate layer between multi-agent systems achieving technological interoperability that addresses technical problems of link between these systems, interface definition and data format. It also provides protocols and standards that allow messages to be transported and exchanged reliably. In our approach, we opted for RabbitMQ Broker, which allows exchanging messages based on AMQP protocol and hence provides interoperability between different systems. The message can be deposited in Exchange and from that point, the

message can borrow a path and is then placed at the end of queue of receive agent before processing and consumption.

- Semantic level layer ensures semantic interoperability between agents. The main objective of this layer is to guarantee that exchanges between agents keep their meaning using Semantic Web and Deep Learning.

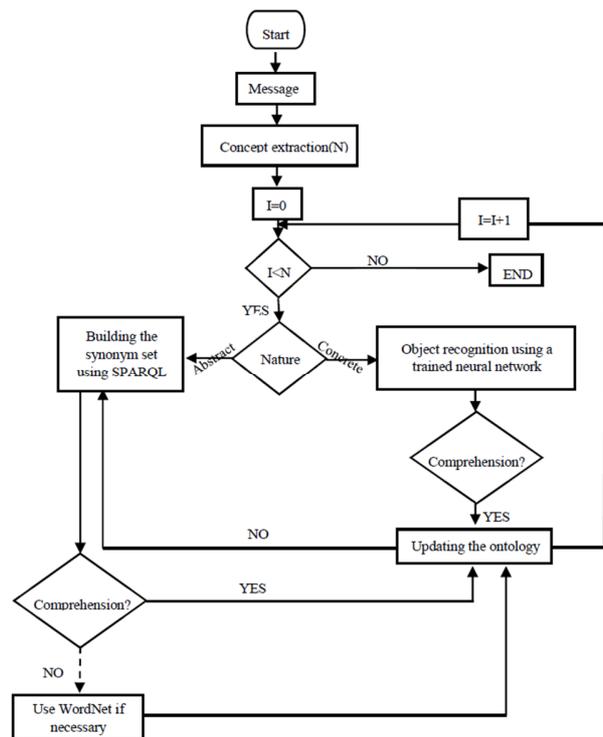


Fig. 3. Simplified diagram of our approach

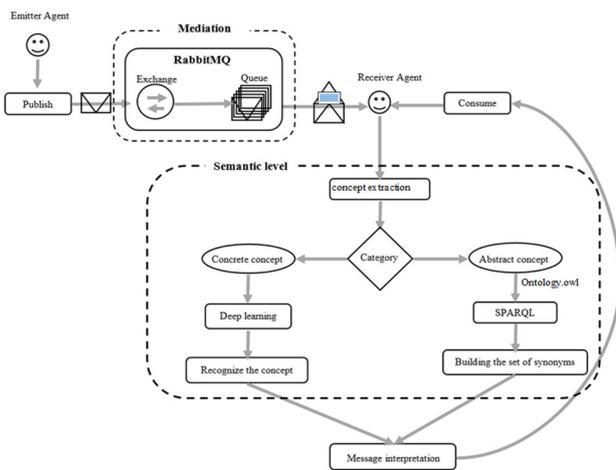


Fig. 4. Architecture of the proposed approach

B. Illustrative Examples

1) Abstract Concept Case

RDF, RDFS or OWL files that implement the ontology of a sending agent designate an abstract concept by a set of synonyms that indicate its semantics and vocabulary to facilitate understanding meaning of that concept. SPARQL language queries are used to question these files to elucidate messages not understood by the receiving agent. SPARQL queries produce a list of synonyms provided by the sending agent on this concept. If the list includes one or more words that are incomprehensible by the receiving agent, the use of a translation tool such as WordNet is proposed. To illustrate our approach, let us consider the field of automatic detection and recognition of road signs by driver software agents of autonomous vehicles. For this purpose, we need to build a traffic codes ontology.

a) Traffic code

Road signs are classified into three categories, shown in Figure 5. Road signs get generally the form of a square, a triangle or a circle. Detection of these shapes in an image provides an interpretation of traffic signs. Each category groups a number of subcategories (Figures 6-8).

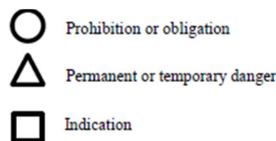


Fig. 5. Categories of road sign forms

b) Proposed ontology

In this context, we propose the ontology shown in Figure 9.

c) Implementation of the Proposed Ontology

To implement our ontology, we select a free and open source tool “Protege” which is an ontology and knowledge editor. In the following, we present an OWL code snippet of our application context (Figure 10).

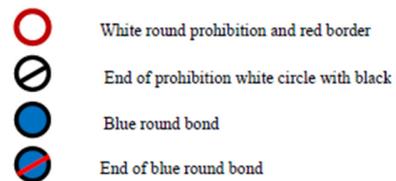


Fig. 6. Prohibition/obligation signs

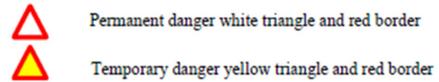


Fig. 7. Danger signs

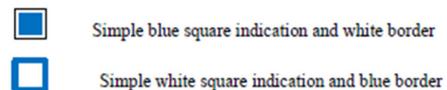


Fig. 8. Indication signs

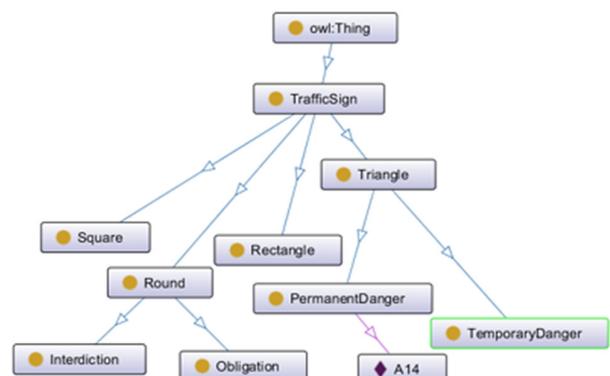


Fig. 9. Proposed ontology

```
<?xml version="1.0"?>
<rdf:RDF ...>
...
<owl:NamedIndividual rdf:about=".../TrafficSign#A14">
  <rdf:type rdf:resource=".../TrafficSign#PermanentDanger"/>
  <hasBgColor rdf:datatype="http://...#string">white</hasBgColor>
  <hasBorderColor
    rdf:datatype="...#string">red</hasBorderColor>
  <hasDescription rdf:datatype="...#string">other
  dangers</hasDescription>
  <hasImage rdf:datatype="...#string">exclamation
  mark</hasImage>
</owl:NamedIndividual>
...
</rdf:RDF>
```

Fig. 10. An OWL code extract from our ontology

d) Ontology query using SPARQL

The “Apache Jena Fuseki” tool is used to interrogate our ontology. Apache Jena Fuseki is a SPARQL query editor for ontology querying. Let’s take the example of a driver who requests all the information concerning the A14 road sign. The question is translated into a SPARQL query (Figure 11). The answer is given in a table form (Table I), which displays all the information about the road sign: type, border color, background color, image and description. From the obtained information, the driver can interpret the traffic sign and therefore comply with the traffic code.

```

prefix owl: <http://.../owl#>
prefix TrafficSign: <.../TrafficSign#>

SELECT ?predicate ?object
WHERE {
    TrafficSign:A14 ?predicate ?object
}

```

Fig. 11. An example of a SPARQL request

TABLE I. RESPONSE OF A SPARQL REQUEST IN FIGURE 11

Predicate	Object
TrafficSign:hasImage	"Exclamation mark"
http://.../#type	TrafficSign:PermanentDanger
TrafficSign:hasBgColor	"White"
TrafficSign:hasDescription	"Other danger"

If this response is sufficient to interpret the message, the Agent Driver (AD) replies by taking into account several parameters (such as the time of day), which also influence the driver's behavior, updates his ontology, and queries source ontology for next concept. Otherwise, other translation techniques can be used to interpret the concept, such as WordNet.

2) Concrete Concept Case

To make the agent driver able to interpret the road signs, a trained neural network is implemented. To achieve this goal (Figure 12), it is necessary to compile a set of learning images to perform deep learning. This set collects thousands of images of different road signs converted into data and transferred to the neural network. Artificial neurons then assign a weight to the different elements. The last layer of neurons will then gather different information to deduce the sign of the circulation.

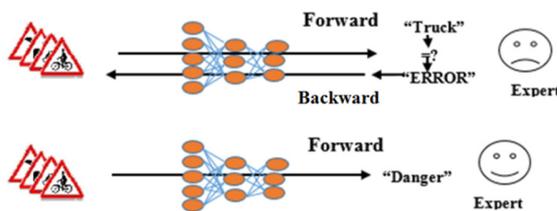


Fig. 12. Neural network training

The neural network will then compare this response to the correct answers given by the experts. If the answers match, the network keeps this success in memory and will use it later to recognize the signs. In case of failure, the network takes note of its error and adjusts the weights placed on the different neurons to correct it. The process is repeated many times until the network becomes able to recognize a road sign in various forms and circumstances. In the given example, the trained neural network returns the term "Danger" as an answer. If the receiving agent understands this response, he updates his ontology by adding the new concept. In the opposite case, the receiving agent moves to the method used in the first "abstract concept" case to search the synonyms for "Danger".

V. CONCLUSION AND FUTURE WORK

In a MAS, communication between agents is achieved by sending structured messages based on ACLs describing agent's

messages in "performative" (message content) form. Nevertheless, in an open environment, the ACLs suffer from some drawbacks in message interpretation when received by an agent, due to their ontology. This latter describes vocabulary of a given domain, useful for interpreting messages sent by agents. While many techniques for combining heterogeneous ontologies have been introduced for the purpose of improving semantic interoperability in heterogeneous distributed systems, they still are restricted to abstract concepts and don't deal with concrete concepts. To address this issue, this paper introduced a new semantic level interoperability architecture based on two principles: the first principle is the separation (decoupling) of the agent's communication model and learning pattern. The second principle is to extend semantic interoperability to concrete concepts by combining two techniques: Semantic Web and Deep Learning. The key idea of the proposed architecture is that the Semantic Web is utilized to interpret abstract concepts and Deep Learning is performed as a new method to ensure semantic interoperability in case of concrete concepts like images. The purpose of the proposed method is twofold: agent's learning model is performed independently of its communication model and the ability is given to an agent to interpret a concrete concept. The proposed approach is very suitable to ensure semantic interoperability between heterogeneous distributed applications exchanging image-type information, as in the medical domain or in road traffic. For our future work, we intend to implement and validate our approach in the field of road traffic.

REFERENCES

- [1] A. Shajjar, N. Khalid, H. F. Ahmad, H. Sugun, "Service Interoperability between Agents and Semantic Web Services for Nomadic Environment", 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Australia, December 9-12, 2008
- [2] V. M. Ionescu, "The analysis of the performance of RabbitMQ and ActiveMQ", 14th RoEduNet International Conference - Networking in Education and Research, Net NER), Craiova, Romania, September 24-26, 2015
- [3] A. F. Klein, M. Stefanescu, A. Saied, K. Swakhoven, "An experimental comparison of ActiveMQ and OpenMQ brokers in asynchronous cloud environment", 5th International Conference on Digital Information Processing and Communications, Sierre, Switzerland, October 7-9, 2015
- [4] H. Suguri, E. Kodama, M. Miyazaki, I. Kaji, "Assuring interoperability between heterogeneous multi-agent systems with a gateway agent", 7th IEEE International Symposium on High Assurance Systems Engineering, Tokyo, Japan, October 23-25, 2002
- [5] N. E. A. Amrani, M. Youssfi, O. E. K. Abra, "Semantic interoperability between heterogeneous multi-agent systems based on Deep Learning", 6th International Conference on Multimedia Computing and Systems, Rabat, Morocco, May 10-12, 2018
- [6] E. J. Gonzalez, A. Hamilton, L. Moreno, R. L. Marichal, J. A. Mendez, V. Munoz, "Development of a multiagent system for identification and control", IEEE Conference on Emerging Technologies and Factory Automation, Catania, Italy, September 19-22, 2005
- [7] D. Camacho, R. Aler, C. Castro, J. M. Molina, "Performance evaluation of ZEUS, Jade, and SkeletonAgent frameworks", IEEE International Conference on Systems, Man and Cybernetics, Yasmine Hammamet, Tunisia, October 6-9, 2002
- [8] M. Mohsenzadeh, F. S. Aliche, M. Teshnehlab, "A New Approach for Merging Ontologies", World Academy of Science, Engineering and Technology, Vol. 4, pp.647-653, 2007
- [9] H. Kong, M. Hwang, P. Kim, "A new methodology for merging the heterogeneous domain ontologies based on the WordNet", International

- Conference on Next Generation Web Services Practices, Seoul, South Korea, August 22-26, 2005
- [10] S. C. Bailin, W. Truszkowski, "Cooperation Between Intelligent Information Agents", in: Lecture Notes in Computer Science, Vol. 2182, Springer, 2001
- [11] C. R. R. Robin, G. V. Uma, "A Novel Algorithm for Fully Automated Ontology Merging Using Hybrid Strategy", European Journal of Scientific Research, Vol. 47. No. 1, pp. 74-81, 2010
- [12] S. Bailin, W. Truszkowski, "Ontology negotiation: How agents can really get to know each other", in: Lecture Notes in Computer Science, Vol. 2564, Springer, 2003
- [13] J. Li, J. Tang, Y. Li, Q. Luo, "RiMOM: A Dynamic Multistrategy Ontology Alignment Framework", IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 8, pp. 1218-1232, 2009
- [14] J. Euzenat, A. Polleres, F. Scharffe, "Processing Ontology Alignments with SPARQL", International Conference on Complex, Intelligent and Software Intensive Systems, Barcelona, Spain, March 4-7, 2008
- [15] N. Vlassis, A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence, Morgan & Claypool, 2007
- [16] T. Agotnes, W. Van der Hoek, M. Wooldridge, "Quantified coalition logic", Synthese, Vol. 165, No. 2, pp. 269-294, 2008
- [17] S. Willmott, F. O. F. Pena, C. Merida-Campos, I. Constantinescu, J. Dale, D. Cabanillas, "Adapting agent communication languages for semantic Web service inter-communication", The 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Compiegne, France, September 19-25, 2005
- [18] X. Jin, "Ontology-based software agent for intelligent business terminal", 2nd IEEE International Conference on Information Management and Engineering, Chengdu, China, April 16-18, 2010
- [19] D. Fensel, J. A. Hendler, H. Lieberman; W. Wahlster, T. Berners-Lee, "Languages and Ontologies", in: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, The MIT Press, 2005
- [20] B. Abrahams, W. Dai, "Architecture for automated annotation and ontology based querying of semantic Web resources", The 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Compiegne, France, September 19-25, 2005
- [21] A. Gudi, H. E. Tasli, T. M. den Uyl, A. Maroulis, "Deep learning based FACS Action Unit occurrence and intensity estimation", 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, Slovenia, May 4-8, 2015
- [22] A. A. M. Al-Saffar, H. Tao, M. A. Talab, "Review of deep convolution neural network in image classification", 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications, Jakarta, Indonesia, October 23-24, 2017