Enhanced OWL-S Semantic Web Services for Traffic Management System Utilizing Internet of Things

Abdullah H. Bin Sawad
Department of Information Systems
Faculty of Computing and Information Technology
King Abdulaziz University, Saudi Arabia

Bassam A. Zafar
Department of Information Systems
Faculty of Computing and Information Technology
King Abdulaziz University, Saudi Arabia

Sofiane Ouni
CRITAL Laboratory, ENSI, University of Manouba, Tunisia
Departement of CIT, Jeddah Community College
King Abdulaziz University, Saudi Arabia

ABSTRACT
IoT is a distributed physical network could support in various services, such as communications integration, control and information-processing via numerous transportation systems. It is an application extends to all aspects of transportation systems such as the vehicle, the infrastructure, and the driver or user.

In this paper, we approach integration of Intelligent Traffic Management Systems and IoT from semantic service viewpoint. We create the architecture system of web services based on OWL-S in IoT environment and describe the composition web service based on IoT. We have extended this web services to real-time extensions that can describe the application related to real-time management of the vehicle traffics. To validate our system, we proposed case study for Ambulance Vehicle. This case of study shows the use of the timing and extended real-time features for web service description and the related process ontology generated for OWL-S technology.

Keywords
Traffic Management System; Semantic Web Service; IoT; OWL-S;

1. INTRODUCTION
Internet of Things (IoT) is the set of technology trends that interconnect objects to the Internet. These objects can be devices, vehicles, light traffics, and other things using software, hardware, sensors, and wireless network connectivity to ensure the exchange and collection of data coming from these devices [5]. IoT can assist in the integration of communications, and information processing. It can be applied in many domains. Application of the IoT has touched every aspects of transportation systems such as vehicles, the infrastructure, and the driver. The practical cooperation between these components of a transport system allows vehicular communication, road assistance, smart traffic control, logistics management, vehicle control and safety [7].

For traffic management systems based on IoT, smart devices and sensors installed at roadsides and in vehicles provide data about real-time traffic that help to create systems. These systems able to precisely administrate, monitor, control vehicles moving, recognize operation for current traffic and flow conditions that may predict the future for traffic flow. The system knows crowded roads not only for drivers but also for authorities. Moreover, the system may issue real-time traffic information to give the driver to choose the best routes. Such System for traffic based IoT must contain every elements such as vehicles, roads, traffic signals, bridges, tunnels, and even drivers. These elements will be connected to the internet for acceptable association and management through tools like sensor devices, infrared sensors and global positioning systems. In intelligent and automatic way, IoT traffic gives traffic information collection and integration, helping the processing and analysis of the categories of traffic information on the roads in a large area [3].

In a traffic management system, collected data are significant, and thus hard work is needed for it to be exploited. Therefore, we need to move toward an information system based on knowledge with the guidance of ontologies. It will be able to access to precise information and understand the described knowledge. In the context of this work, we propose using the Ontology Web Language for Service (OWL-S) as a semantic description of the services related to the traffic management system. The OWL-S ontology can be used to describe the IoT devices as a service [1].

To create a system that facilitates traffic management systems based on IoT that provides traffic stream conditions by identifying the current or future traffic streams. Some traffic information is provided by the traffic IoT. This information allows for integrating, developing, processing and analyzing all kinds of traffic information on lanes in a big zone automatically and logically. Therefore, an intelligent transport system based on IoT is evolved in modern traffic management [3]. In this work, we need an advanced semantic description of the traffic information system that helps to collect data and save it as knowledge to be used then as a service. The effort involved in exploiting data depends on how large the collected data is in the traffic management system. Therefore, having an information system based on knowledge is necessary to guide ontologies. One of the major objectives of having this system is proving access to precise information and making sense of the described knowledge. This work suggests that the OWL-S ontology can be used as a semantic description of the services related to the traffic management system. The IoT devices are described as a service by using semantic web service as OWL-S [26]. The OWL-S ontology is a language to describe semantic web services. This language describes services using three top-level concepts: service profile, service-grounding and service model [2], [4]. The devices and services that use IoT are mapped on the previous. Another major objective is collecting and organizing data then saving it as knowledge to be used then as a service, also providing a complete architecture that can hold the traffic management system.

This paper proposes a semantic data model for ontology support, how to introduce timing and real-time issues related to web service and a methodology for extracting knowledge from ontology. Also, how to make a composition web service for expanding services to the traffic management system based on IoT. Moreover, how could be the architecture of web service based on OWL-S in IoT environment.

The rest of this paper is organized as follows. Section 2, is an overview of the concepts used in this paper and. Section 3, gives a comparison of the literature review referenced. Section 4, explains the proposed solution and contribution of the paper's research problem. Section 5, is the case of study to apply and evaluate our approach. Finally, a conclusion that summarizes the paper is mentioned in section 6.

2. BACKGROUND

In this section, we present and introduce the most important concepts that we are using in this paper. We give an overview about the semantic Internet of things description, then web of services, and semantic web services to create application based on the semantic description.

2.1 Semantic Internet of Things description

The community of semantics describes concepts and relationship between various entities in different domains. IoT domain has same methods to use semantics, but most problems of IoT related to semantic descriptions are not widely adopted as expected. The developers and users have their concern about increasing the complexity and processing time in semantics. Hence, they are inappropriate for responsive and dynamic environments like IoT. Constraints and dynamicity should be considered in IoT models. Moreover, it needs to model the concepts and relationships that provide and represent interoperability between IoT entities [25].

Figure 1 shows an example of the IoT-Lite based web services which is the instantiation of the lightweight of Semantic Sensor Network (SSN). From this description, we extract the basic web services to read and write from the sensor device. In this case, it will be the vehicle distance to read from the sensor. This sensor is used for the traffic light management to know the proximity of the vehicle and react relative this distance.

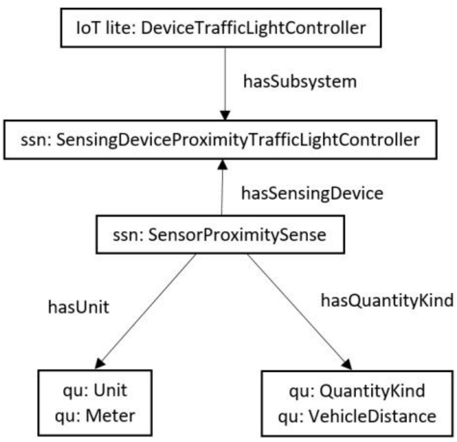

Fig 1: Device Traffic Light Controller

2.2 Web Services

With the popularity of the software systems in the present times, a variety of services can be proposed. Moreover, it has also been felt that with time, the software system has also become complicated and thus there is a need for increasing the standard for bridging the gap between the various operations [6]. Web Services are also an independent platform that allows convenient encapsulation of the existing codes and the services of the web application helps to access to various web functions with the help of using accessibility standards. This is required for making interaction independent from the implementation procedures such as the use of programming language the platform used for the operating system [12]. Platform for an operating system.

The web services are based on some protocols:

- Hypertext Transfer Protocol (HTTP): it has been seen that the general transport protocols can be used as the HTTP component such as the Simple Mail Transfer Protocol (SMTP) or the File Transfer Protocol (FTP). HTTP is considered to be one of the most important protocols used in the web services because of its protection against firewalls. Any server or internet browser can also support it. HTTP is thus ubiquitous and helps to address issues in error handling [17]. It covers the requirements of the descriptive format of the Simple Object Access Protocol (SOAP).

- Simple Object Access Protocol (SOAP): it is used for sending as well as receiving messages by the exchange of the XML messages that are coded, and it also helps to identify the link to the HTTP for enabling the application and communication protocol for the application running in heterogeneous operating systems that have different technologies and programming language. SOAP is also platform independent and can be supported by the latest technologies such as CORBA or Java. It does not require any specific operating system and it is not need specific programming language. With the SOAP, a complex extension model is provided for achieving flexibility and a higher degree of acceptance without the necessity of adhering to particular data application.

- Web Service Description Language (WSDL): it enables web service in the well-structured manner to deal with the process of managing the technology. Interface Definition Language (IDL) is very powerful, and it provides the capacity to the machinery to provide clear and suitable mechanical service to the process of information technology management. In the process of managing the technology, it is very effective to make proper integration between the service competencies [18]. In the sorts segment, by utilizing XML diagram, all information sorts information and yield that will be utilized as a part of the messages of the administration. The operations are declared through XML mapping not the same as the other XML sentence structure portrayal dialects where giving framework sort which can be an exceptionally complex direct utilized for determining for compound information sorts likewise, indicating fundamental information sorts like string, dates, and numbers.

- Universal Description Discovery and Integration (UDDI): in the process of finding web service UDDI is the significant part of XML. The process also used in the technological era with the process of making stranded deviation process. In the era of publishing, describing and web service the use of XML is very effective. The information of the ear is included three elements such as white, yellow, and green pages. Every part has their different role and responsibility in the process of managing the mechanism. In the white page, essential information is included such as name, telephone number, address and other information. Yellow pages contain more details and description about the service. In green pages, including technical information on a web service such as various interfaces and URL locations.

- Representational State Transfer (RESTful): it is REST architecture based Web Services [23]. In RESTful, everything is a resource and ease of adaption (lightweight infrastructure) no need to buy a web service just require a browser to get started. Moreover, RESTful are very usually used to create APIs for applications web-based. There are four principles can define the scalability and success of the HTTP implementing them:

1) Resource Identification through URI
2) Uniform Interface for all resources:
   - GET (Query the state, idempotent, can be cached)
   - POST (Modify, transfer the state)
   - PUT (Create a resource)
   - DELETE (Delete a resource)
3) "Self-Descriptive" Messages over Meta-Data.
4) Hyperlinks to determine the state of an application.

2.3 Semantic Web Services Approach

This section, representing OWL-S, WSMO, and METEOR-S. Then the composition and the comparison of each type [24]. After that, mention the reason for the choice OWL-S in this work.

2.3.1 OWL-S, WSMO, and METEOR-S

- OWL-S: enable users and software agents with high automation. Moreover, composing, discovering, invoking and monitoring particular web service resources and its properties.
- Web Service Modeling Ontology (WSMO): it is standard for describing Semantic Web Services by European initiative. It is operated by the SDK cluster based on the work of Fensel and Bussler [20].
- Managing End-To-End OpeRations (METEOR): in Georgia University, a project at the LSDIS Lab focuses on a procedure to track management for transactional workflows [21]. The tracking project management from semantic web view is called "METEOR-S".

2.3.2 Composition

- OWL-S have three types of processes Atomic, Simple and Composite. An atomic process has no sub-processes and process in single operation. A simple process is a process not invocable, but it is used as abstraction element. A composite process used for expanding association and planning purpose.
- WSMO Choreographies and Orchestrations to describe WSMO based on the Abstract State Machines. The principles for Choreography model are:
  - It is state-based.
  - The state represented by algebra.
  - By define algebra changing the relations and function values that change the state through guarded transition rules.
- METEOR-S in composition stage includes creating representation web processes. The METEOR-S framework has chosen Business Process Execution Language for Web Services to create process abstract and constructing model workflow patterns. Design processes abstract include the following.
  - Creating process flow by using the control flow constructs.
  - By specifying service templates, each service in the process represent the requirements that allow to the process designer link a known web service or define web semantic service description.
  - Specifying the constraints to optimization.

2.3.3 Comparison

- OWL-S is to define an ontology for describing web services that enable users and software agents with high automation. Moreover, composing, discovering, invoking and monitoring.
- WSMO is to define an ontology for describing semantic web services contain composition, discovery, and invocation, it is focused on solving the interoperability problem. OWL-S and WSMO propose a new framework to achieve goals by describing Web Services.
- METEOR-S propose a framework to introduce semantic information to the process web lifecycle by constructs for adding semantics to current industry standards.

2.3.4 Why OWL-S

The next following mention the reason for choice OWL-S in this work.

- Recommended by W3C.
- Flexible, no restriction of the way to implement the web of services.
- Rich description of the service composition process.
- The implementation is orchestration mechanism.
- Service composition process description is near the language programming so that it will be accessible to implementation orchestration program.

3. RELATED WORKS

The cooperative traffic is one of the main challenges in the Intelligent Traffic Management System (ITMS). An idea, which is used to collaborate with ITMS, was a cause to begin in automated highways by the concept of combined where the vehicles can get a signal from the road. The research laboratory of the General Motors presented the documentation of the first ideas on the automated highway [19]. In situations, ITMS environment requires the intelligent action to improve, as soon as they are collected the data and automatic decision making it cooperates with the traffic system. The objective of the cooperative driving looks after on early detection and prevents the risks. Author has stated that the limited channels' bandwidth has to be distributed to the entire participating node in the ITS in high amount. However, the author does not clarify full information in this study. Hence, it is improved to the efficiency of the diverse alternatives. It helps to get better the ITS sufficient capacity and helps to develop the transport system [9].

In [22] the paper introduced intelligent internet-of vehicles management system (IIOVMS). It is an intelligent technique to monitor and to improve urban road traffic. This proposed system contains three essential components that are cloud computing, internet of things (IoT) and wireless sensor network (WSN). [15] Cloud computing in private and public for managing and operating a platform to integrate with the all system information, storage and data processing working in the cloud. [22] Private cloud will use subsystem for traffic management, data processing, virtual storage, and maintenance. This platform only permits access and supervise by the staffs who have authorization. Public cloud to display road condition, registration and log in for end users, information queries and intelligent routes navigation. Wireless sensor networks consist of many sensor nodes installed in cars which have functional modules for processing, sensing, and communication. Also, delivering and transferring real-time data to users.

In [29], the authors suggest control system to develop traffic based on IoT and PROTEUS technology which is simulation software on the platform of field-programmable gate array (FPGA). FPGA is a chip that does not need to design personnel to carry the risk and reducing the cost of the chip [31].

According to [13] system for emergency vehicle needs more improvement in interoperability and performance. So, authors proposed a prototype of Bevor. It is a smart semantic middleware emergency vehicle in ITS based on "National Transportation Communications for ITS Protocol (NTCIP)". Bevor designed in a framework with semantic web 3.0 and Extensible Markup Language (XML) to exchange and allowing simple access to semantic information such us events, policy matching, and composition. The essential architecture Bevor system consists of traffic management center, information centers, environmental sensors and environmental devices. Traffic management center contains sensor context and ontologies include NTCIP definitions, policy roles [14] and identifiers. Information center could be police, emergency center, and hospital to exchange data on request from client or Traffic management center. Environmental devices include message boards, signals, and ramp meters. Manager agent handles all the operations.

In [28] the authors introduced on semantic web technologies in interoperability with IoT as a comprehensive solution. In [8], the authors designed an application for the Emergency Planning Department of the Essex County Council (ECC). In emergency situations, the traditional way to handle the cause was through a personal phone and fax by Geographical Information Systems (GIS). However, this includes definite steps, is slow and there is the chance of errors occurring. In this research, the authors aim to create an application for emergency management using a decision support system (DSS) based on a semantic web service (SWS) with GIS. SWS will help emergency services officers in their tasks (displaying, processing and retrieving) so they can handle emergency cases more quickly and accurately. The language that will be used for creating ontologies is OCML. The limitation of the study is that in some situations an agency needs authority access for different agencies and not all agencies will accept cooperation.

In [9], the authors proposed an UbiRoad middleware system solution to the following interoperability: interoperability between the in-car and the roadsides devices that provided and programmed by different vendors. UbiRoad uses semantic language and the technologies of semantics for the declarative specification of services and devices' behavior. It also provides data level interoperability and financial level coordination. In [11], the authors developed a model composed of an ontology, representing the vehicles, the infrastructure and implementation for the regulation of traffic were in OWL with Protégé. The inference rules were implemented in semantic web rule language. The authors used vehicles that were equipped with actuators and sensors which can communicate with others and the infrastructure. In each vehicle, they focus on an internal part and its decisional part.

According to [10], authors proposed a traveler information system as a multi-agent platform to allow travelers to find information for traffic road by using SWS profiles. They implemented a hybrid matching algorithm of different weights and measures to profile the web services parameters based on their relevance, type, and nature. In addition to this, they used the 'sibling' relationship to improve precision, recall and reduce run time.

In the IoT-based smart city in suggestion scenarios [6], Adaption composition framework was proposed for environmental services and user tasks. The framework depends on context-Aware web Service description Language (WEASEL). OWLS – TC4 used to evaluate the framework through matchmaking mechanisms based testbed by merging composite and simple services. Including performance and quality, the result of evaluation shows more accurate composition for the end user than other approaches. In experiment result [30], shows a framework to process the transaction in IoT environment using an OWL-S profile with service status ontology to describe information such as current status and waiting time of the entities in the dynamic model.

Most of the related works on semantic services for IoT traffics are made to specific applications and specific needs. The created services are associated with the context of these applications. This limits the extensibility of such services. The present study fills a gap in the literature. We aim for designing a semantic web service that can be adapted to different user needs even if it is not considered before. This can be made by the creation of an efficient and an evolutionary web services for Traffic Management System. That is the scope of what we want to design in this paper. So, to reach the purpose of this work, four steps are needed. The purpose can be explained in detail with the help of the following steps:

- By developing a complete architecture for holding the system of the traffic management and explore the similar works for having the complete vision regarding the infrastructure of the hardware and the software programming applications [31].
- It is necessary to understand the interfaces of the different applications of the software programming and to understand the framework for implementing the OWL-S for selecting one suitable for them.
- Designing the semantic services of the OWL-S for ensuring the response on the real-time basis for vehicle traffic and the use simulation to check evaluation the effectiveness of the performances.

4. OUR CONTRIBUATION

This section includes our contributions introducing timing and real-time issues related to web service and a methodology for extracting knowledge from the OWL files that describes the services running. Also, by the use of OWL-S, we describe the web service composition for IoT. Moreover, we create the architecture of web service based on OWL-S in IoT environment to be applied for Traffic Management System.

4.1 System Architecture

We create a new architecture for web service based on OWL-S in IoT environment. In Figure 2, the user chooses one of the established queries and then, compose the services or processes. For example, we can have composite services of the queries as simple services (1, 2, 3 and 4). After that, passing to the web service. On the other hand, we have a designer do a semantic description of the services and composition the process. Composition links with timing description to descript the services. For instance, creating service 1 and service 2 and how much time to implement them. The designer cannot describe the service directly by the classical web service (as Restful). But, the designer does semantic description by OWL-S for the profile, model, and grounding. Moreover, we have added the description of timing. After that, mapping to Restful can be run directly. The next step will be the transferring to execution. Finally, the monitoring is made to evaluate the running of web service if it is made with success related to the timing issues. When the evaluation result did by the user and not acceptable then, we have remotely existed that make an update to the composition to have better composition process.

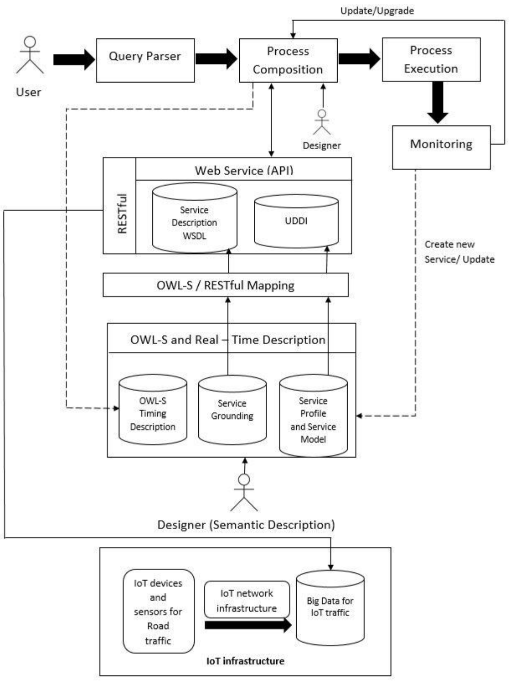

Fig 2: Architecture based on OWL-S in IoT environment

4.2 Real-Time Extensions for OWL-S

A hard real-time system is essential for the responses to occur in identified deadline [27]. For soft systems, in case the deadlines are missed, the system will keep performing in the right way. Some systems could have both hard and soft real-time sub systems with hard and soft deadlines. In our context, it is possible for a web services to have a deadline (soft or hard) and timing constraints related to their responses time.

In fact, the real time system can be found in the IoT devices or computers interacting with the environment with sensors, such as temperature and pressure transducer and an actuator for Vehicle traffic control applications. There are different usages of real-time systems for instance communication, process control, manufacturing, multimedia and common everywhere [16].

There are some requirements like the language and run-time support in order to allow the program to perform functions as follows:

- Specifying times for actions to be either performed or completed.
- Supporting repeating work (periodic or aperiodic).

The previous facilities called real-time control features that allow the program to synchronize at the same time. Inspired by the programming languages for real-time systems, we have defined the description of the compositions of web services according to timing features.

In Figure 3, we present the process ontology with Real-Time Construct extensions. Adding RealTimeConstruct to control construct very important to serve the goal of the traffic management system. The Real-Time Construct depends on the time at which the result is produced and the logical result of the computation. It contains Timing Construct and Waiting Construct.

- Timing Construct contains Delay and Timeout as shows in the description.
  - Delay: Pause or sleep for a certain time (relative time). For example, the delay in the traffic light is used to transition from one light state to another.
  - Timeout: The system waits an amount of time for a specific event to occur.
- Waiting: The waiting construct consider the waiting time and waiting condition.
  - Waiting Time: waiting for a specific time to resume execution (the used time is absolute time)
  - Waiting Condition (watchdogs): the available amount of time to performing some activity and possibility of offering an alternative behavior if this deadline not met.

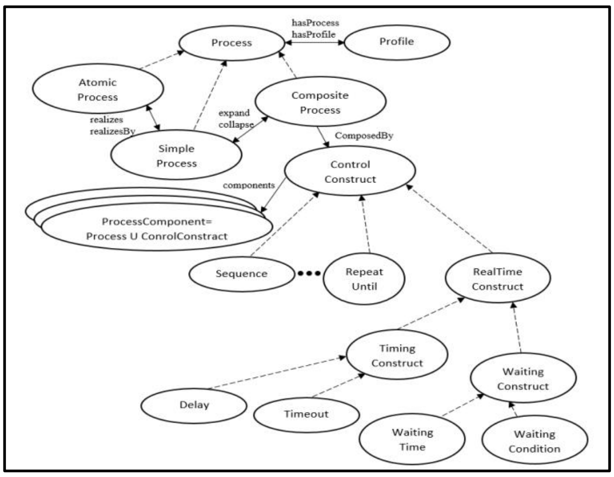

Fig 3: Timing extensions in The Process ontology

4.3 Web services orchestration

Orchestration describes the aspect of creating business processes from composite web services [32]. Orchestration interacts with internal and external web services that refer to an executable process where the interaction happens at the message level. They cover task execution, business logic and they can cross applications and organizations to define a long-lived, transactional, multistep model the process. Orchestration represents control from one party's viewpoint.

We give the answer to the question: How to run composite processes of web services? For that, we will use centralized orchestration process that interprets the OWL processes description file.

Below in Figure 4, the algorithm that runs the list of web service as centralized orchestration. We assume that the list of web services the starting from the first web service in the list, then go to the next through different structures such as While, for and if-else condition with specific waiting and time. After each call for web service, it is deleted from the list and then move to the next until finishing the list.

```
List_WebServices= get_list(OWL_Process_description)
While (List_WebServices != empty) do
    WS= getTheFirst(List_WebServices)

    if WS Typeof atomicProcess then
        CallWebService(WS)
        delete first(List_WebServices)
    end if
end while

    if WS Typeof ifCondition then
        (Cond, Num_WS_then, Num_WS_else) = CallWebService(WS)
        delete first(List_WebServices)

        if Cond== true then
            for i=  to Num_WS_then do
                W= getTheFirst(List_WebServices)
                CallWebService(W)
                delete first(List_WebServices)
            end for

        else
            for i= to Num_WS_else then do
                delete first(List_WebServices)
        end if
    end if

    if WS Typeof delay then
        Time= CallWebService(WS)
        Wait (Time)
        delete first(List_WebServices)
    end if
```

Fig 4: Centralized Orchestration Process

5. CASE STUDY AND EVALUATION

We suppose a problem for vehicle traffic management to show that we can reach many kinds of applications to manage them. These case study show the use of the timing and extended real-time features for web service description and the related process ontology generated for OWL-S technology. Thus, we are trying to perform some functions related to real-time as waiting time and delay.

5.1 Traffic Light Adaptation for Emergency Vehicles

In emergency cases, the ambulance car is required to reach a particular location as soon as possible. Therefore, the ambulance car turns on its siren to notify the drivers of other vehicles, so they move away from the ambulance car. Even if the traffic lights are red, the ambulance car and other vehicles (if any) in front of the ambulance car need to pass the red traffic lights. This situation is dangerous because these cars can be involved in an accident.

Our system is proposed to help the ambulance car arriving at the destination safely and fast. The ambulance car sends a request to the transceivers to pass through an intersection. Then, the traffic signal states change from red signal to green. After that, the ambulance car passes more safely and rapidly while the signal controllers change states of traffic signals more appropriately. This action is made through the use of sew service for IoT devices for Vehicle Traffic management.

Activity diagram in Figure 5, shows the system depends on three main components: Emergency Car, Traffic Management System IoT Devices, and Traffic Controller. The driver in an emergency car, press enable emergency button. Then, the system trying to detection of the signal controllers for the traffic light in the distance of 500m. Next, the traffic controller changes the signal controllers for traffic light from red to green. After that, the system wait for detection of near signal controller of traffic light the distance of 5m for 20 seconds. Then, the traffic management system checks if the emergency vehicle was receiving signal in 5 seconds. If the signal received, that means the emergency vehicle does not cross the traffic light yet and the system recursive 20 seconds till the emergency vehicle does not receive signal. At the end, the system makes the signal for traffic light red and return to the default setting.

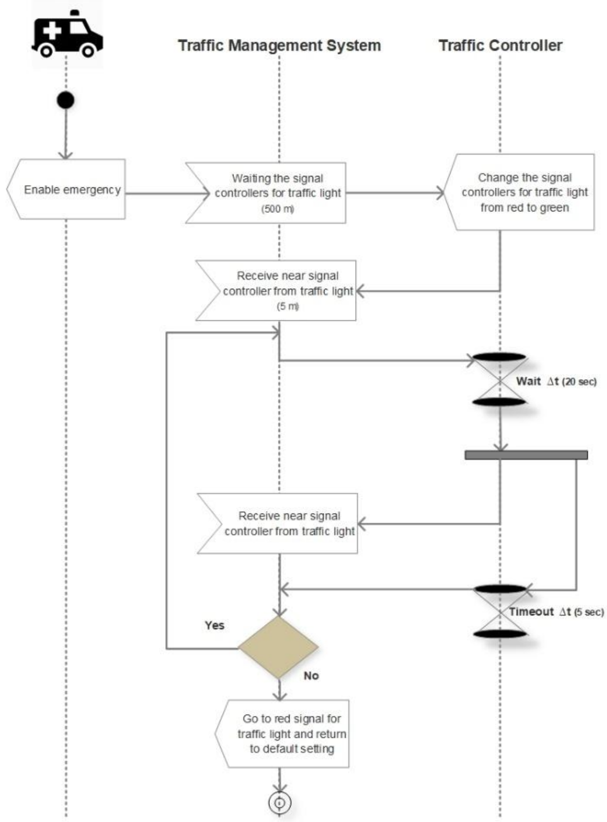

Fig 5: Activity Diagram for Ambulance Vehicle

5.2 Evaluation and Result

It is necessary to evaluate the quality of web service system and keeping the higher level of quality based on an evaluation result. In [34], the evaluation quality of service in a system can be found some metrics such as response time, reliability, efficiency or availability. In this paper, we are more interested in applying response time, and the efficiency for real-time to evaluate our system according to some parameter.

We are working on traffic management system, and we need to prove that our service is suitable for this kind of application. At first, we need to apply a real context which is hard to make it. So, we are base our work on the simulation model for vehicle traffic. To do a Simulation model, we need an approach with a fast model, low online computational and no complexity for controlling and coordinating traffic lights (Model Predictive Control) [33].

Figure 6 shows the suggestion controlling for two traffic lights. The first traffic light called TL1 and the second traffic light called TL2. In n arrive1, describing the number of vehicles arriving at the first queue (Q1) and $n_{Q1,Q2}^{arrive\ 2}$ describing the number of vehicles arriving from Q1 to queue2 (Q2). Then, the timing vehicle takes to move from Q1 to Q2 is called $Link_{1-2}$.

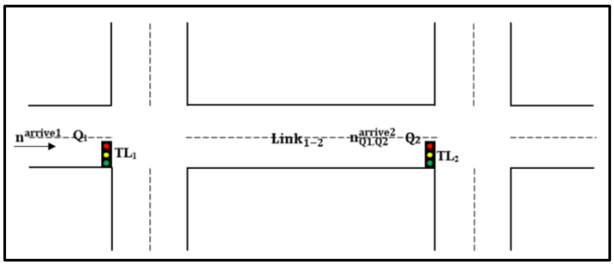

Fig 6: Simulation Model for Traffic Management System

The developed simulator is based on Simulation Model for Traffic Management System. In this simulator, we have created a simulation programmed based on NetBeans and Java.

We have three scenarios of traffic light management for Problem 1: scenario1 is a normal situation without the use of web services traffic management system, scenario2 is based in our approach of web services with real time extension and scenario3 is the use of web service without timing. The difference between our approach and others that sometimes the traffic light switches back to the red light and the emergency vehicle still stack in that traffic light. Our approach makes the traffic light still green till ensuring the emergency vehicle passing the traffic light. It is made by the timing extension that ensures the correct running. In the simulation, we use random arriving vehicles so sometimes we have more queue, and sometimes we have less queue. Two kinds of vehicles in our simulation emergency vehicle and regular vehicle.

All the results will be based on these simulation parameters in Table 1 and next figures the horizontal axis displaying time of simulation and at the same time is ID of vehicles. The vertical axis displaying variation the timing of vehicle take to move from start point to the path (endpoint).

Table 1: Simulation Parameters for Traffic Vehicle

Simulation time = 40000 s
Distance between 2 light traffics is = 1000m
average velocity= 50 km/h
The default time of traffic light change= 180s
Vehicle Arrival average rate = 0.005 V/s
emergency Vehicle Arrival rate = 0.0006 V/s

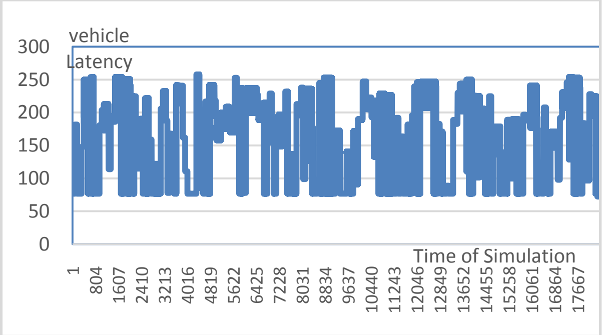

Fig 7: Regular Vehicles in Classical Case

- Scenario1: it is for the normal situation. Reviewing the timing of the vehicles (emergency vehicle and regular vehicle) without using web services traffic management system (Figure 7). We note that in many cases, we get a high latency timing value of the vehicle crossing through the path. Thus, the latency value is 254s due to the arriving vehicles in the queue before the traffic light. However, there is sometimes we have the value 77, in this case, the vehicles move rapidly without staying a lot at the traffic light.
- Scenario2: It is the evaluation of the vehicle latency (time to cross all the predefined path) using web services for light traffic management without the timing descriptions. When the queue has an emergency vehicle the light changed from red to green depends on traffic management system and light controller (Figure 8).

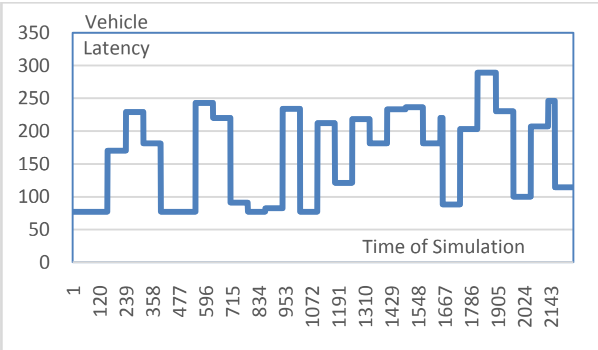

Fig 8: Other Web Services without Timing

- Scenario3: It is the evaluation of the vehicle latency with the use of web services traffic management system based on the Timing extension (our approach). In this case (Figure 9), by using timing expression, there will be more extension of green light till ensuring the emergency vehicle passing the light. This web service manages emergency vehicle at the traffic light in a better way. Most of the time for emergency vehicles will be within the lower value of latency time.

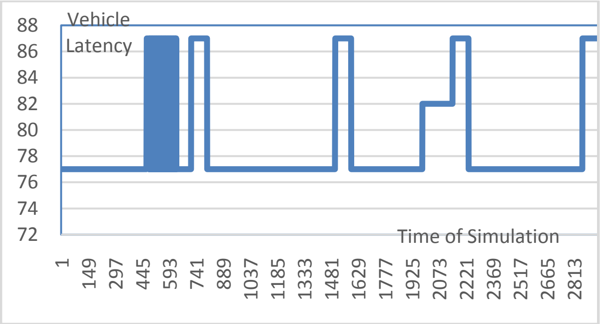

Fig 9: The Case Our Web Services Timing Extensions

In our approach, the highest traffic vehicle queue in the simulation the value reaches to 88s and in the lowest traffic vehicle queue the value is 78s. Without timing extension, the classical web service approach as in the scenario 2 has a higher traffic vehicle queue according to the simulation, the value reaches to 289s and some times in the case lowest traffic vehicle queue the latency is 79s.

Thus, our approach is better than other classical approaches without using of timing description, because it can hold better management of traffic light to extend of green light till ensuring the emergency vehicle passing the light. Thus, there is no waiting a lot of time of emergency vehicles. Indeed, the waiting time for the emergency vehicle is less than the average of the other approaches. The emergency vehicle has the lowest value of timing in our approach; it is best to use in the traffic light.

Thus, applying our real time web service system in the real environment may save many urgent cases by helping the ambulance car arriving at the destination safely and fast. It will reduce the cost of street light and accelerate the process of arriving of the emergency vehicles.

6. CONCLUSION

In this paper, we have started with the application systems on vehicles and traffics. After that, we have presented an overview of semantic web service and ontologies that to represent semantic web technology. We preferred to use semantic web service for traffic management system because of the interoperability, easy integration with other systems and the flexibility. So, we reviewed works in different Approaches and Technologies from Multiple Researchers in ITMS based on IoT.

We used OWL-S to describe semantic web services based IoT environment in traffic management system. One of the reasons made us to choice OWL-S is no restriction on the way to implement the web of services.

The main objectives of this paper are reached. We developed a complete architecture for holding the system of the traffic management and explore the similar works for having the complete vision regarding the infrastructure of the hardware and the software programming applications. Then, through the extended OWL-S proposed architecture, we focused more on the IoT web services composition by the orchestration process that depends on principles. We used centralized orchestration process that interprets the OWL processes description file to run composite processes of web services. We made an algorithm that runs the list of web service as centralized orchestration for an assumed list of web services.

Also, we introduced timing and real-time issues related to web service and a methodology for extracting knowledge from the ontology. To validate our system, we proposed case of study for Ambulance Vehicle to validate our system. This case study showed the use of the timing and extended real-time features for web service description and the related process ontology generated for OWL-S technology. To get more realistic, we created a simulation that contains a solution for an emergency vehicle at a traffic light. The simulation was build based on Simulation Model for Vehicle Traffic. This simulation performs the efficiency of our Timing extension for OWL-S to resolve the description of applications such as the vehicle traffic management systems.

7. REFERENCES

[1] De, S. ; Barnaghi, P. ; Bauer, M. ; Meissner, S., Service modelling for the Internet of Things, Conference on Computer Science and Information Systems (FedCSIS), 2011.

[2] M. A. Aslam, S. Auer, J. Shen, and M. Herrmann, “Expressing Business Process Models as OWL-S Ontologies,” vol. 1.

[3] Hasan Omar Al-Sakran, “Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2, 2015.

[4] Tesseris George, Baryannis George, “OWL-S: Semantic Markup for Web Services”, Computer Science Department University of Crete, http://www.csd.uoc.gr/~hy566/OWL-s/OWL-S.pdf.

[5] Pantos, R. and May, W., 2017. HTTP live streaming (No. RFC 8216).

[6] Lai YL, Chou YH, Chang LC. An intelligent IoT emergency vehicle warning system using RFID and WiFi technologies for emergency medical services. Technology and health care. 2017(Preprint):1-3.

[7] Ersue, M; Romascanu, D; Schoenwaelder, J; Sehgal, A (4 July 2014). "Management of Networks with Constrained Devices: Use Cases". IETF Internet Draft < draft-ietf-opsawg-coman-use-cases>. Anca Hangan, Stefan Oniga, Zoltán Gál, “eHealth Solutions in the Context of Internet of Things”, IEEE International Conference on Automation, Quality and Testing, Robotics, 2014.

[8] Vlad Tanasescu, Alessio Gugliotta, John Domingue, Rob Davies, Leticia Gutiérrez-Villarías, Mary Rowlatt, Marc Richardson, and Sandra Stin£i¢. A semantic web services gis based emergency management application. In International Semantic Web Conference, pages 959_966. Springer, 2006.

[9] Vagan Terziyan, Olena Kaykova, and Dmytro Zhovtobryukh. Ubiroad: Semantic middleware for cooperative traffic systems and services. International Journal on Advances in Intelligent Systems Volume 3, Number 3 & 4, 2010.

[10] J Javier Samper Zapater, Dolores M Llidó Escrivá, Francisco R Soriano García, and Juan José Martínez Durá. Semantic web service discovery system for road traffic information services. Expert Systems with Applications, 42(8):3833_3842, 2015.

[11] P. Morignot and F. Nashashibi, "An ontology-based approach to relax traffic regulation for autonomous vehicle assistance," CoRR, vol. abs/1212.0768, 2012.

[12] Carolina Tripp Barba, Miguel Angel Mateos, Pablo Reganas Soto, Ahmad Mohamad Mezher, and Mónica Aguilar Igartua. Smart city for vanets using warning messages, traffic statistics and intelligent traffic lights. In Intelligent Vehicles Symposium (IV), 2012 IEEE, pages 902_907. IEEE, 2012.

[13] Kuang-Ho Chen, Chyi-Ren Dow, Da-Jie Lin, Chen-Wei Yang, and Wei-Chun Chiang. An ntcip-based semantic its middleware for emergency vehicle preemption. In Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on, pages 363_368. IEEE, 2008.

[14] John Davies, Rudi Studer, and Paul Warren. Semantic Web technologies: trends and research in ontology-based systems. John Wiley & Sons, 2006.

[15] Marios D Dikaiakos, Dimitrios Katsaros, Pankaj Mehra, George Pallis, and Athena Vakali. Cloud computing: Distributed internet computing for it and scientific research. IEEE Internet computing, 13(5), 2009.

[16] Jie Ding, Rui Wang, and Xiao Chen. Performance modeling and evaluation of real-time traffic status query for intelligent traffic systems. In Communications (APCC), 2016 22nd Asia-Pacific Conference on, pages 238_242. IEEE, 2016.

[17] Arthur J Dock. Using gps/gis to test emergency vehicle traffic signal preemption. www. skit. cm/p0798. htm, 2007.

[18] G Ferrari, S Busanelli, N Iotti, and Y Kaplan. Cross-network information dissemination in vanets. In ITS Telecommunications (ITST), 2011 11th International Conference on, pages 351_356. IEEE, 2011.

[19] Vinita Jindal and Punam Bedi. Vehicular ad-hoc networks: introduction, standards, routing protocols and challenges. International Journal of Computer Science Issues (IJCSI), 13(2):44, 2016.

[20] KR Jothi and A Ebenezer Jeyakumar. Optimization and quality-of-service protocols in vanets: a review. In Artificial intelligence and evolutionary algorithms in engineering systems, pages 275_284. Springer, 2015.

[21] Saravanan Kannan, Arunkumar Thangavelu, and RameshBabu Kalivaradhan. An intelligent driver assistance system (i-das) for vehicle safety modelling using ontology approach. International Journal of UbiComp, 1(3):15_29, 2010.

[22] Ying Leng and Lingshu Zhao. Novel design of intelligent internet-of-vehicles management system based on cloud-computing and internet-of-things. In Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, volume 6, pages 3190_3193. IEEE, 2011.

[23] Francisco J Martinez, Chai Keong Toh, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. A survey and comparative study of simulators for vehicular ad hoc networks (vanets). Wireless Communications and Mobile Computing, 11(7):813_828, 2011.

[24] Makoto Miyawaki, Z Yamashiro, and T Yoshida. Fast emergency pre-emption systems (fast). In Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on, pages 993_997. IEEE, 1999.

[25] Brijesh Kadri Mohandas, Ramiro Liscano, and Oliver WW Yang. Vehicle traffic congestion management in vehicular ad-hoc networks. In Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on, pages 655_660. IEEE, 2009.

[26] G Hosein Mohimani, Farid Ashtiani, Adel Javanmard, and Maziyar Hamdi. Mobility modeling, spatial traffic distribution, and probability of connectivity for sparse and dense vehicular ad hoc networks. IEEE Transactions on Vehicular Technology, 58(4):1998_2007, 2009.

[27] Hassnaa Moustafa and Yan Zhang. Vehicular networks: techniques, standards, and applications. Auerbach publications, 2009.I

[28] AA Obiniyi, SK Aina, and MB Hammawa. Trends in ontology of vanet. International Journal of Computer Applications, 118(6), 2015.

[29] Haoyuan Ou, Jianming Zhang, and Yi Wang. Development of intelligent traffic control system based on internet of things and fpga technology in proteus. Traffic, 20:2, 2016.

[30] Qu, C., Liu, F., Tao, M., Deng, D. (2016) An OWL-S based specification model of dynamic entity services for Internet of Things. J Ambient Intell Humaniz Comput 7:73–82K

[31] T Thirumalai and KR Kashwan. Split ring patch antenna for fpga configurable r_d applications. Journal of Next Generation Information Technology, 6(3):47, 2015.AA Obiniyi, SK Aina, and MB Hammawa. Trends in ontology of vanet. International Journal of Computer Applications, 118(6), 2015.

[32] C. Peltz, "Web services orchestration and choreography," Computer (Long. Beach. Calif)., vol. 36, no. 10, pp. 46–52, 2003.

[33] W. Kooistra, "Simulation of coordinated traffic light control using model predictive control," no. August, 2012.

[34] IEEE Standard for a software quality metrics methodology. (n.d.). Retrieved January 17, 2018, from https://standards.ieee.org/findstds/standard/1061-1998.html