A Semantic MSOAH-IoT Design for Improving Efficiency and Solving Heterogeneity within IoT Applications

Mohammed Lamnaour[1], Mohammed Achkari Begdouri[2], Yasser Mesmoudi[1], Yasser EL Khamlichi[1], and Abderrahim Tahiri[1]
[1] Department of computer science, ENSA Tetuan, Abdelmalek Essaâdi University, Morocco
[2] Department of computer science, FP Larache, Abdelmalek Essaâdi University, Morocco
Email: mohammed.lamnaour@gmail.com; achkari.med@gmail.com; ymesmoudi@uae.ac.ma; yelkhamlichi@uae.ac.ma; t.abderrahim@uae.ac.ma

Abstract —The integration of Internet of Things (IoT) technology in order to develop and manage different real life applications such as smart cities, environmental surveillance, health care, precision agriculture and traffic monitoring, has proven to be a game changer. Different IoT hardware and software solutions are merging to meet different application requirements. However, with each new IoT application, there are new challenges to solve, because of the variety of communication protocols, data formats and the heterogeneity of smart objects on the network. Many companies develop middleware platforms to overcome the heterogeneity problems. However, most of these solutions are proprietary, monopolized and developed as vertical proprietary application stacks, so it prevents developers from reusing modules. This paper demonstrates some recent middleware approaches, especially the Service Oriented Architecture for Heterogeneity Issues within the Internet of Things (MSOAH-IoT), designed to solve hardware and software heterogeneity issues within IoT applications. Then proposes an enhancement of the MSOAH-IoT middleware, by implementing new features such as semantic web and context awareness, in order to establish a smart gateway, able to make decisions for automated management, while ensuring connection between heterogeneous smart objects.

Index Terms—SOM middleware, MSOAH-IoT, smart services, smart objects, heterogeneity

I. Introduction

Recently, thanks to rapid advances in sensor technology and wireless communications, devices with embedded sensors and connection capabilities are expanding widely in many fields, forming a network known nowadays as the Internet of Things (IoT) [1], [2]. IoT solutions face multiple critical issues such as heterogeneity, data management, data compatibility, bandwidth management, connectivity, and security problems [3], [4]. Most of manufacturers provide complete IoT solutions that consist on hardware, software and architectural solutions. However, most of them belong to specific applications, thus it restrains the reuse of these solutions, especially on the software layer level called "middleware". One of the most challenging issues regarding the use of such sensors regards the possibility to seamlessly make them interoperate to reach a specific goal. This objective could be difficult to achieve, due to the lack of a universally accepted standard for sensor communications [5].

A middleware is a software layer located between the sensor hardware layer and end user application layer. It manages data collected by sensors in the lower level layer, and translates it to a higher-level layer [6]. As defined in an early work related to middleware architecture [4], a middleware should provide reusable services for several applications, and a runtime environment able to manage the execution of multiple applications, but above all, it must solve problems of incompatibility and heterogeneity of hardware and network connectivity. Designing a middleware should take into consideration multiple mechanisms to handle the following important issues as specified in [7], [8]: Heterogeneity - Data Aggregation - Managing limited battery power and resources - Real world integration - Scalability, mobility and dynamic network topology - Quality of Service (QoS) – Security - Fault tolerance and Application knowledge.

In our previous work [1], we have developed a service oriented architecture based middleware to deal with heterogeneity issues within the IoT, which we labelled MSOAH-IoT. We used Java/JavaEE to make it portable and interoperable with various platforms. The main goal was to handle heterogeneous smart objects (IP and not IP based) connected through different wireless networks (WIFI, Bluetooth, ZigBee), as well as generating a unified data format. The MSOAH-IoT model is based on the REST architecture and JSON for lightweight communications within limited resources IoT network. We made connected IoT devices available as web services, and then we offered a desktop web application to the end user for displaying and managing the entire network.

The final objective of our work, is to develop a smart gateway easy to install, compatible with various platforms, able to: deal with sensor and wireless networks heterogeneity; analyse all collect information; create patterns of habitual actions, and make smart decisions based on thresholds, rules and patterns analysis. For

Manuscript received December 12, 2021; revised May 15, 2022.
Corresponding author email: yelkhamlichi@uae.ac.ma.
doi:10.12720/jcm.17.6.443-451

example if the gateway is implemented on a smart home and there is a gas leak when no one is home, the gateway should open windows and alerts authorities.

This paper presents the developed middleware (MSOAH-IoT) and its main features, then propose an enhancement, by integrating semantic web and context awareness technologies to provide smart services to the end user. Related work is presented in section 2. Section 3 describes the proposed model's architecture, and results. Then section 4 discusses the MSOAH-IoT enhancement and future work, followed by the conclusions of this work.

II. Related Work

Middleware approaches are classified in [7], based on their position on the network, some run inside the network (classical approaches), while others integrate and process sensor data but work outside the network. Authors in [1], [3] and [7]-[10] describes most of these approaches, each one is developed based on different architecture approach. The classical approaches are as follows:

- Database approach: Middlewares based on this approach do not support heterogeneity neither rendering data in real-time.
- Application driven approach: It allows the application to identify their QoS requirements, and then modify the network according to application needs. Most of middlewares within this approach do not support the heterogeneity of sensors.
- Modular approaches (mobile agent), are developed in modular way to minimize the energy consumption; however, the nature of its code does not allow hardware heterogeneity.
- Virtual Machine approaches provide a flexible programming paradigm, to hide the heterogeneity of the runtime environments and the hardware resources, but add a considerable code size and performance overhead.
- Message-oriented (Event based) approaches are real-time systems where the processing trigger is event. This paradigm is based on publisher events and subscriber events. They do not offer heterogeneity support or interoperability between heterogeneous parts of the network.

Other middlewares were developed to integrate not IP based networks, into IP-based network such as: GSN [11], Borealis [12], IrisNet [13], Hourglass [14], HiFi [15], SStreaMWare [16], EdgeServers [17], ESP framework [18]. However, they only focus on wrapping data coming from sensor nodes for sharing and processing over the Internet.

Several efforts were conducted to implement the IPv6 protocol for constrained IoT devices. The result is 6LoWPAN [19]. This protocol enables running services on the application layer directly on smart objects and the integration of a Service-Oriented Architecture (SOA) into the design of the middleware [8].

The Service oriented middleware (SOM) architecture is the best platform to develop IoT applications [10]. It logically views the network as a service provider for the end user application. Authors in [20] have studied a large number of proposed SOA based middleware for IoT solution, belonging to the same research domain, each distinguished by technical specifications depending on the context of use. Table I, presented in [1] demonstrates the features of most recent and popular SOM middle-wares. The majority were developed using JAVA programming language, based on the REST architecture and provide a mobile application for users.

TABLE I: COMPARATIVE STUDY OF EXISTING MIDDLEWARE

SOM Middlewares	Supported Communication Protocols
Sitewhere [21]	REST; Web-socket; MQTT; AMQP
DeviceHive [22]	REST; MQTT
Konker [23]	REST; MQT
Kaa [24]	REST
Nimbits [25]	REST
Nitrogen [26]	MQTT; HTTP
OpenIoT [27]	REST; GSN
Webinos [28]	REST
Linksmart [29]	REST

III. The MSOAH-IoT Middleware

As described in [1], the development of a middleware must meet several criteria and solve several issues. In our case, we focused on the first place on solving heterogeneity of sensor hardware and network interfaces. The middleware is implemented in a gateway, within unlimited resources and connected via Ethernet, to a local server for storage purposes and can benefit from cloud services.

The gateway can be a Smartphone, a Laptop or a Raspberry PI working on specific operating system like windows or Linux, and equipped with different communication technologies such as Bluetooth and WiFi. In [1], we proposed the design and the implementation of new middleware labeled MSOAH-IoT to address heterogeneity issues within the IoT.

A. The MSOAH-IOT Architecture

MSOAH-IoT [1] architecture is carried out through three phases, as described in Fig. 1. In the first one, we programmed the gateway and the objects to initialize all services related to each communication interface, so links could be created between detected smarts objects and the gateway. We deployed a web service to discover and initiate WiFi connections by using a web server.

On the other hand, we developed a script to discover Bluetooth connections and create Java WebSocket in order to exchange data. In phase 2, we developed the middleware core based on three main functions as follows:

- *ThingDiscovery ():* Allows the collection of detected smart objects details such as the communication interface, universally unique identifier (UUID), name, operating system, and the proposed services etc. all collected data is saved into a database.

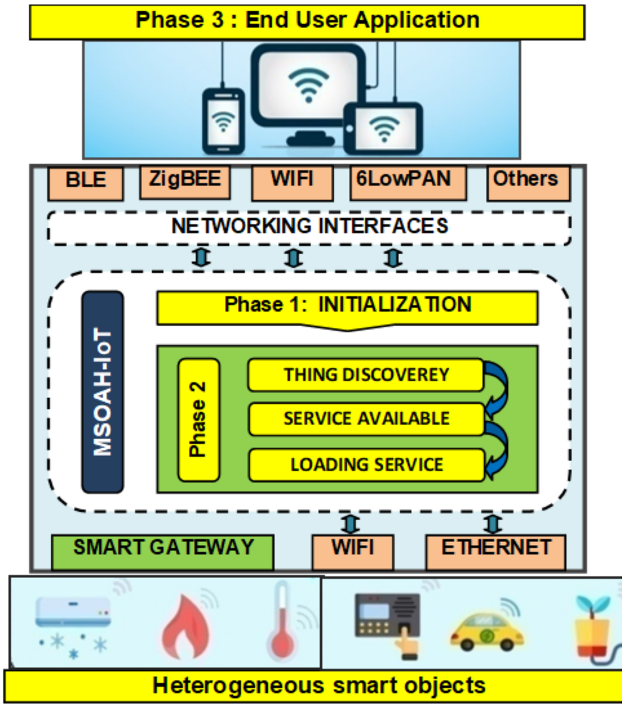

Fig. 1. The MSOAH-IoT implementation architecture

- *ServiceAvailable():* We create this method to allow authorized users to consult connected objects and all available services registered in the gateway database.
- *LoadingService():* Provides specific implementation (I.e. getter and setter methods) for service applicant using specific environment (specific end user browser). It contains get and set methods offering data collection and controlling services.

MSOAH-IoT is developed using the Java Platform Enterprise Edition (Java EE) architecture that permits to build distributed web and enterprise applications, which helps to focus on application issues, and allows interoperability between systems.

The model uses JAX- RS (Java API for RESTful Web Services), that provides portable APIs for developing, exposing and accessing Web applications, designed and implemented in compliance with principles of REST architectural style [30]. Then, we used the JSF framework 2.2, which simplifies development and integration of web-based user interfaces. Developers can build web applications by assembling reusable UI components in a page, connecting these components to an application data source and wiring client-generated events to the server-side event handlers [31].

Using JSON - JavaScript Object Notation API (an implementation for data stores and data structures) permits to optimize HTTP requests, both in terms of the number of requests and the size of data packages exchanged between clients and servers [32]. Finally, the middleware uses SQLite for storage, especially that the smart gateway has limited resource components, such as memory capacity and CPU. SQLite is a relational database management system embedded into the end program, and does not have a separate server process and reads/ writes directly to ordinary disk files [33].

B. Results

We conducted many scenarios to prove the efficiency of the proposed middleware. The MSOAH-IoT was able to discover all smart objects within its sensing range, through Wifi, Bluetooth, and Zigbee depending on the smart gateway interfaces.

In [1] we developed two scenarios to test the ability of the MSOAH-IoT to handle heterogeneity of sensing hardware, where different sensors providing various services were used. The middleware was able to detect all sensors in the scenario, even if they were working under different Operating System, collect data from the sensor nodes and then sending commands. In the second scenario, we tested the capability of the MSOAH-IoT to support heterogeneous networking interfaces such as WIFI, Bluetooth and Zigbee within the same IoT application. In order to evaluate the proposed middleware MSOAH-IoT, we compare its functionalities with other similar approaches proposed by other well-known research works. We observe on one hand, that our model relays on the most used communication protocols architecture in IoT solutions: REST and Web-socket.

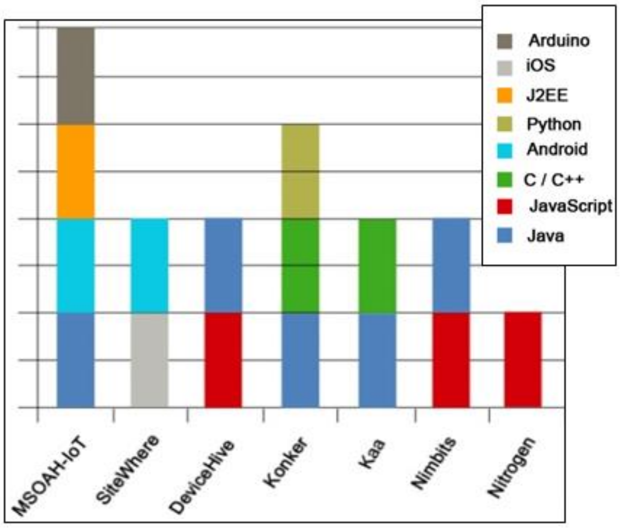

Fig. 2. Programming languages used in existing IoT Middleware

On the other hand and as described in Fig. 2, the proposed approach MSOAH-IoT combines different programming languages: Java, J2EE, Arduino and Android, which makes it more flexible than others, and adaptable to various situations and technologies depending on the application requirements.

IV. The MSOAH-IoT Enhancement

After solving heterogeneity issue by designing the MSOAH-IoT middleware, our objective is to add more features to the proposed solution so it can make smart decisions after analysing the collected information coming from all the surrounding smart objects. We will adapt the developed middleware, so it can provide more smart services to the end user, by implementing technologies such as semantic web and context awareness, which allow building a context-aware system. This system uses context to provide relevant information and services to the user, where relevancy depends on the user's task.

A. Semantic Web

The goal of the Semantic Web is to give a new form of content that is meaningful for humans, computers, and things [34]. The use of semantic web technologies [35] in the internet of things increased exponentially since 2010.The most cited and relevant use cases for semantic web technologies are interoperability, data storage, data integration, data abstraction and access, semantic reasoning and interpretation, resource/service search, discovery and scalability. The two main areas of an IoT system that benefit the most, from the integration of semantic technologies are as cited in [34], firstly the representation and description of the Thing, its capabilities and environment, and secondly Semantic annotation of the data that the Thing produces.

Ontology is defined in [36] as: "formal, explicit specification of a shared conceptualization. A conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon". Integrating ontology in middleware ensures the expressiveness, the flexibility and extensibility.

There are four main components that compose ontology: Classes, relations, attributes and individuals. Classes are the main concepts to describe. Each class can have one or several children, known as subclasses, used to define more concepts. Classes and subclasses have attributes that represent their properties and characteristics. Individuals are instances of classes or their properties. Finally, relations are the edges that connect all the presented components [37].

There are already several projects that index ontologies and vocabularies on the Web. One of the most used schemas on the Web is provided by Schema.org and is defined in [34] as: "a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on Web pages, in email messages, and beyond". Linked Open Vocabularies (LOV) is an online catalogue provided by Open Knowledge Foundation, LOV facilitate of any kind of vocabularies used for data description on the Web [34]. The use of semantic web in designing middleware has increased to overcome IoT issues [38]-[41]. For example, UbiROAD[38] is designed for smart traffic environments. It uses semantic web technologies to enable autonomous co-ordination capabilities amongst the devices installed in as smart traffic infrastructure e.g. discovery, adaptability, and negotiation. OpenIOT [39] uses W3C Semantic Sensor Networks (SSN). It handles many issues such as data management, service discovery, and interoperability. Ubiware [40] is designed to deal with distributed resources and heterogeneous services. Authors in [41] demonstrated Hydra middleware designed for network-embedded systems that allow building ambient intelligence applications based on heterogeneous devices.

Several ontologies are developed for internet of things and can be used to build new ontologies and design semantic middleware. [39] Defines the device that constitutes the two basic components of an IoT, the system, composed of both physical and virtual elements; Sensors are devices that collect data for the system and Actuators are devices that enable the system to act on the physical world. In the paradigm of pervasive computing, many distributed things perform computations. Most of these things being physical devices have limited resources. A complete modelling of the system should include a description of their energy consumption since energy management is a crucial topic in IoT systems.

B. Context Awareness

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object considered relevant to the interaction between a user and an application, including the user and applications themselves. A system is context-aware if it uses context, to provide relevant information and/or services to the user, where relevancy depends on the user's task [42]. The context life cycle consists of four steps as shown in Fig. 3.

- Context Acquisition: is the first step in a context awareness system, it consists of gathering the context from the environment and users [36].
- Context Modelling: also referred as context representation, is defined as the context representation that provides assistance in the understanding of properties, relationship, and details of context [43].

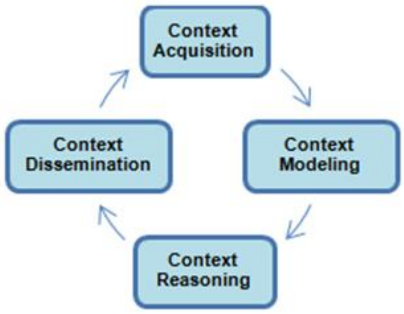

Fig. 3. Life cycle of context

- Context Reasoning: Context reasoning can be described as the extraction of new knowledge from the available context and extraction of context sets from high-level context for better understanding [43].
- Context Dissemination: Context distribution delivers context to consumers [43].

In the following, some popular middlewares based on context-awareness as described in [44], [45]. MoCa is a service based distributed middleware, it employs ontologies to model and mange context. COSMOS is a middleware that enables the processing of context information in ubiquitous environments. It consists of three layers: context collector (collects information from the sensors), context processing (obtain high-level information from raw sensor data), and context adaptation (provides access to the processed context for the applications). SOCAM is an ontology-based middleware.

It uses ontology-based method for both context modelling and context reasoning. Hydra comprises a Context Aware Framework (CAF).CAF consists of two main components: Data Acquisition Component responsible for connecting and retrieving data from sensors and the Context Manager is responsible for context management, context awareness, and context interpretation.

C. Semantic MSOAH-IoT Architecture

Researchers within the IoT community have proposed many middlewares in recent years, the use of semantic web in the design of middleware helps to overcome some issues. In order to design the architecture of our proposed model, we start by defining the functional requirements, needed by the system.

Our model is based on the functional requirements, used to design CA4IOT [46]. Protect users' data - Manage problems of overload - Ensure the continuity of service - Connect sensors to the IOT middleware- Understand and maintain context information- Understand user requirement / request / problem- Understand to fill the gap between high-level user requirement and low-level sensor capabilities- Understand high-level context information using low-level raw sensor data - Manage users.

The proposed Semantic MSOAH-IoT architecture consists of four layers: Communication Layer, Controllers Layer, Storage and Decisional Layer. However, there are two external layers interacting with the whole system: The End User Layer and the Sensing Layer as described in Fig. 4. As defined in [1] the MSOAH-IoT contains already all layers cited except for the controller and the decision layer.

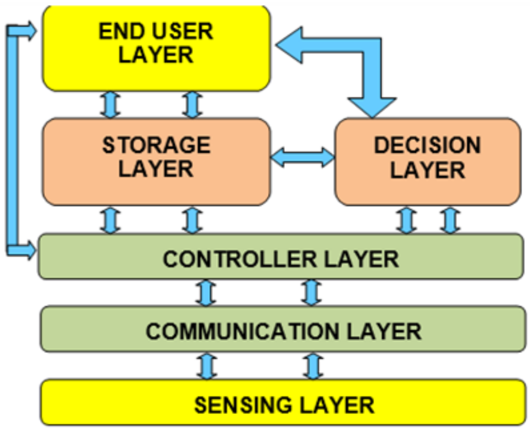

Fig. 4. Layers of the proposed semantic MSOAH-IoT

V. FUTUR WORK

The following describes roles, functionalities and components of each layer as well as interactions, starting from low-level sensor layer to high-level application layer as demonstrated in Fig. 5.

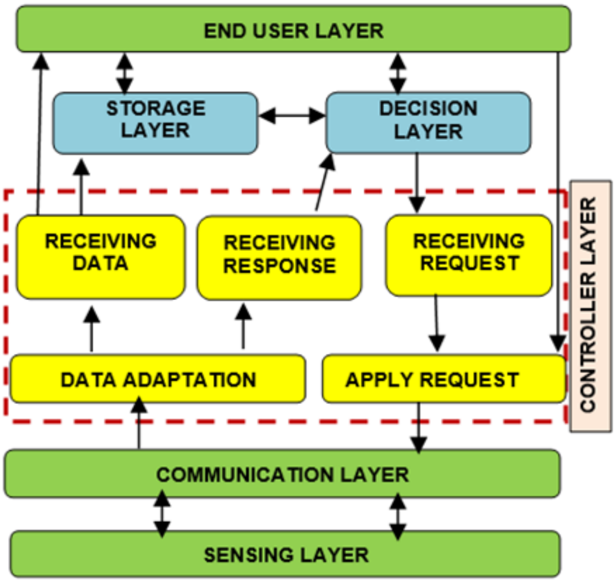

Fig. 5. Interactions between Controller layer components and other layers within the proposed Semantic MSOAH-IoT

- ***Sensing Layer***: An external layer, which represents all software and hardware sensors, embedded in the environment.
- ***End User Layer***: This layer is not in the core of our system, it represents human users, applications or services interacting with the system. A web application and an android app are developed in the first version of MSOAH-IoT.
- ***Communication Layer*** is responsible for communication with connected objects installed in the Sensing Layer; it permits collecting/sending data from/to various connected smart objects/things through heterogonous networking interfaces. It deals with the heterogeneity issues. This layer is the core of the MSOAH-IoT.
- ***Storage Layer*** permits saving collected data from things installed in the Sensing Layer and the actions triggered by users. This data will be used by the decision layer to make the perfect reactions at the perfect time.

Layers that follows are the new features that need to be developed and implemented from scratch, the first is the Decisional Layer and the second one the Controller Layer.

- ***Decision Layer:*** This component is the brain of our system. It analyses the stored data, creates patterns, uses predefined thresholds and rules then makes decisions. For example: an old man is used to open a door at 7 o'clock, if he doesn't, some actions must be triggered, like checking proximity sensors, sending alert to his phone to check his position, checking his heartbeat, etc. Fig. 6 describes an example of contexts definition in the case of Ambient Assisted Living - AAL application after analysing and processing data.

- ***Controller Layer*** should contain many components:
    - Receiving data: to save information into the database or send it to the end user application;
    - Data adaptation: permits the identification of the information from data;
    - Receiving request: allows to receive from decisional automatic actions triggered by a given event in order to make new measures;
    - Apply request: to identify things to contact when more measures are needed by the decisional layer;
    - Receiving response sends needed measures to the decision layer;

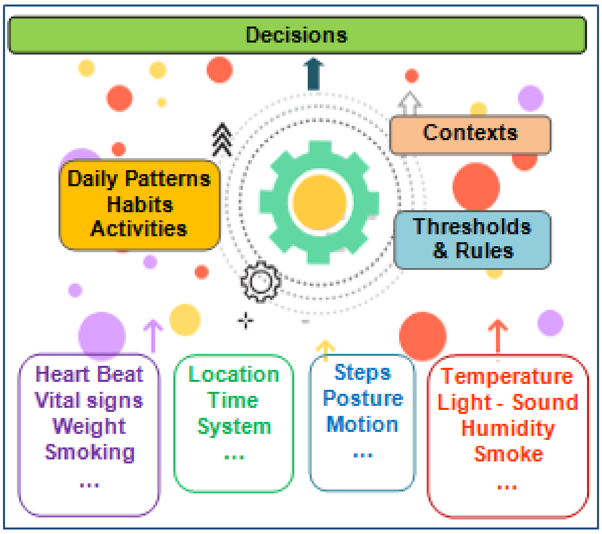

Fig. 6. An example of contexts definition in the decisional of AAL.

In consequence, users can read data in real time, consult all registered measures in data base, request for various measures directly from connected things and modify a wrong decision when it's made by the decision layer. Our objective is to create a smart gateway, able in the first time, to collect measurements and information from heterogeneous objects, within the sensing range of its various networks (WIFI, Bluetooth, ZigBEE, etc..), analyze the data and detect critical events, then make smart decisions, such as asking for more data, sending alerts, shutting doors, opening windows, etc.

We use Node-RED, which is a flow-based programming tool to design our IoT architecture. Developed by IBM's Emerging Technology Services team and now a part of the OpenJS Foundation, this tool offers an attractive Web based user interface to execute IoT service-based workflows [47]. The Node-RED allows users to replace common coding tasks with visual drag-and-drop to connect web services and gadgets. Various components are linked together to create a flow using Node-Red editor. Besides, it allows wiring together hardware devices, APIs and online services in new and interesting ways, with over 225,000 modules in Node's package repository [48], [49].

Till this date, we are still in the phase of development of these new features and defining ontologies. We use a low level ontology that defines nodes for the controller layer, and a high level ontology for the higher level layer in order to classify the user patterns and habits. We use Protégé for designing ontologies. It is one of the top editing tool utilized for ontology in research. Protégé provides a graphical class hierarchy and helps to understand the dependencies [50]. Otherwise same tools used in [1], are being used in the development of the proposed middleware, such as: JAX- RS (Java API for RESTful Web Services) the Java programming language API for creating RESTful web services, and JSON (JavaScript Object Notation API) which is a specification that exposes an implementation for data stores and data structures and permits to optimize HTTP requests. For the storage layer we use SQLite as an adequate database management system for IoT solutions. Finally the middleware is being tested on the Raspberry Pi.

VI. Conclusion

Heterogeneity issues are still the most important challenges to solve while designing IoT middleware, whether it related to networking interfaces, operating systems or data formats. In this paper, besides presenting the main challenges of designing an IoT middleware and its several approaches, we presented our model labeled MSOAH-IoT, its architecture, and its efficiency. Then we proposed to implement new features to the MSOAH-IoT, to provide a smart system capable of sensing then making smart decisions.

We present layers and components of the new model, based on semantic web and context awareness technologies, to provide services automatically to the end users. In future work, we intent to implement our new model labeled semantic MSOAH-IoT, in an Ambient Assisted Living AAL application, aiming to improve the quality of life for older people. This interest is due to the number of elderly people, growing more rapidly compared to other age groups. A supervisor can rely on it to make smart decisions, it can check periodically the patient heartbeat, and his position, or remind him of his medicines, and many other smart services.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Mohammed Lamnaour and Yasser Mesmoudi carried out the software programming. Mohammed Achkari Begdouri and Yasser EL Khamlichi helped in the implementation and the testing of the code components. Abderrahim Tahiri conceived the original idea and supervised the project. All authors contributed to the design of the research and to the writing of the manuscript. All authors had approved the final version.

ACKNOWLEDGMENT

An earlier version of this manuscript was presented at the 4th International Conference on Cloud Computing Technologies and Applications, Cloudtech'2018, (26-28 Nov. 2018, Brussels, Belgium), as conference in "Design and implementation of a smart gateway for IoT applications using heterogeneous smart objects".

REFERENCES

[1] Y. Mesmoudi, M. Lamnaour, Y. E. Khamlichi, *et al.*, "A middleware based on service oriented architecture for heterogeneity issues within the internet of things (MSOAH-IoT)," Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 10, p. 1219, December 2020.

[2] S. Taylor, "The next generation of the internet," CISCO IBSG, p. 9, April 2013.

[3] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70-95, 2015.

[4] F. C. Delicato, P. F. Pires, and A. Y. Zomaya, "Middleware platforms: State of the art, new issues, and future trends," in *Art of Wireless Sensor Networks. Signals and Communication Technology*, H. Ammari, Ed., Springer, Berlin, Heidelberg, 2014, pp. 645-674.

[5] B. D. Martino and A. Esposito, "Semantic techniques to support IoT interoperability," in *Semantic IoT: Theory and Applications. Studies in Computational Intelligence*, R. Pandey, M. Paprzycki, N. Srivastava, S. Bhalla, and K. Wasielewska-Michniewska, Eds., vol. 941, Springer, Cham, 2021.

[6] B. Bhuyan, H. K. D. Sarma, and N. Sarma, "A survey on middleware for wireless sensor networks," *Journal of Wireless Networking and Communications*, vol. 4, no. 1, pp. 7-17, 2014.

[7] Y. Mesmoudi, M. Lamnaour, Y. E. Khamlichi, A. Tahiri, A. Touhafi, and A. Braeken, "Design and implementation of a smart gateway for IoT applications using heterogeneous smart objects," in *Proc. 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, Brussels, Belgium, 2018, pp. 1-7.

[8] J. Cecílio and P. Furtado, "Existing middleware solutions for wireless sensor networks," in *Wireless Sensors in Heterogeneous Networked Systems. Computer Communications and Networks*, Springer, Cham, 2014, pp. 39-59.

[9] M. E. K. Ajana, H. Harroud, M. Boulmalf, and M. Elkoutbi, "Middleware architecture in WSN," in *Wireless Sensor and Mobile Ad-Hoc Networks*, D. Benhaddou, A. Al-Fuqaha, Eds., Springer, New York, 2015, pp. 69-94.

[10] R. Alshinina and K. Elleithy, "Performance and challenges of service-oriented architecture for wireless sensor networks," *Sensors*, vol. 17, no. 3, pp. 536-575, 2017.

[11] K. Aberer, M. Hauswirth, and A. Salehi, "The global sensor networks middleware for efficient and flexible deployment and interconnection of sensor networks," Technical Report LSIRREPORT- 2006-006, Ecole Polytechnique Fedéralé de Lausanne (EPFL), 2006.

[12] D. J. Abadi, Y. Ahmad, M. Balazinska, *et al.*, "The design of the borealis stream processing engine," *CIDR*, 2005.

[13] P. B. Gibbons, B. Karp, S. Nath, and S. Seshan, "IrisNet: An architecture for a worldwide sensor web," *IEEE Pervasive Comput.*, vol. 2, no. 4, pp. 22–33, 2003.

[14] J. Shneidman, P. Pietzuch, J. Ledlie, M. Roussopoulos, M. Seltzer, and M. Welsh. "Hourglass: An infrastructure for connecting sensor networks and applications," Harvard Technical Report TR2, vol. 1, no. TR-21–04, 2004.

[15] M. J. Franklin, S. R. Jeffery, S. Krishnamurthy, F. Reiss, S. Rizvi, E. Wu, O. Cooper, A. Edakkunni, and W. Hong. "Design considerations for high fan-in systems: The HiFi approach," in *Proc. Second Biennial Conference on Innovative Data, Systems Research CIDR 2005*, Asilomar, 2005, pp. 290–304.

[16] L. Gurgen, C. Roncancio, C. Labbé, A. Bottaro, and V. Olive, "SStreaMWare: A service oriented middleware for heterogeneous sensor data management," in *Proc. 5th International Conference on Pervasive Services*, 2008, pp 121–130.

[17] S. Rooney, D. Bauer, and P. Scotton, "Techniques for integrating sensors into the enterprise network," IEEE *Trans Netw Serv Manag*, vol. 3, no. 1, pp. 43–52, 2006.

[18] T. Kobialka, R. Buyya, C. Leckie, and R. Kotagiri, "A sensor web middleware with stateful services for heterogeneous sensor networks," in *Proc. 3rd International Conference on Intelligent Sensor Networks and Information*, IEEE, Melbourne, Qld., Australia, 2007, pp. 491–496.

[19] G. Mulligan, "The 6LoWPAN architecture," in *Proc. 4th Workshop on Embedded Networked Sensors*, ACM, New York, 2007, pp. 78–82.

[20] M. A. A. D. Cruz, J. J. P. C. Rodrigues, *et al.*, "Performance evaluation of IoT middleware," *J. Netw. Comput. Appl.*, vol. 109, pp. 53-65, 2018.

[21] DeviceHive - Open Source IoT Data Platform with the wide range of integration options. [Online]. Available: https://devicehive.com/

[22] SiteWhere Open Source Internet of Things Platform. [Online]. Available: https://sitewhere.io/fr/

[23] Konker - Construa a sua solução de IoT em dias e não em meses ». [Online]. Available: http://www.konkerlabs.com

[24] Kaa Enterprise IoT platform. [Online]. Available: https://www.kaaproject.org/

[25] Business profile for nimbits.com provided by Network Solutions. [Online]. Available: http://www.nimbits.com/

[26] Nitrogenjs - Overview, GitHub. [Online]. Available: https://github.com/nitrogenjs.

[27] OpenIoT – Open Source cloud solution for the Internet of Things. [Online]. Available: http://www.openiot.eu/

[28] Webinos | The webinos Foundation ». [Online]. Available: http://webinos.org/

[29] LinkSmart® Docs - Home - LinkSmart® Docs ». [Online]. Available: https://docs.linksmart.eu/

[30] S. Patni, "Introduction to JAX-RS," in *Pro RESTful APIs*, Springer: Apress, Berkeley, CA, 2017, pp. 49-62.

[31] J. Juneau, "The basics of JavaServer faces," in *Java EE 8 Recipes*, Springer: Apress, Berkeley, CA, 2018, pp. 103-189.
[32] B. Smith, *Beginning JSON*, Springer: Apress, Berkeley, CA, 2015.
[33] J. Feiler, "Understanding what SQLite Is," in *Introducing SQLite for Mobile Developers*, Springer: Apress, Berkeley, CA, 2015, pp. 9-14.
[34] I. Szilagyi and P. Wira, "Ontologies and Semantic Web for the Internet of Things - a survey," in *Proc. 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, 2016, pp. 6949–6954.
[35] D. Andročec, M. Novak, and D. Oreški, "Using semantic web for internet of things interoperability: A systematic review," *International Journal on Semantic Web and Information Systems*, vol. 14, no. 4, pp. 147–171, Oct. 2018.
[36] G. Fersi, "Middleware for internet of things: A study," in *Proc. International Conference on Distributed Computing in Sensor Systems*, Fortaleza, Brazil, 2015, pp. 230–235.
[37] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the internet of things," in *Proc. 8th Middleware Doctoral Symposium on - MDS '11*, Lisbon, Portugal, 2011, pp. 1–6.
[38] D. Rathod and G. Chowdhary, "Survey of Middlewares for internet of things," in *Proc. International Conference on Recent Trends in Advance Computing (ICRTAC)*, Chennai, India, 2018, pp. 129–135.
[39] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil, "Autonomy through knowledge: how IoT-O supports the management of a connected apartment," in *Semantic Web Technologies for the Internet of Things (SWIT)*, Nov. 2016.
[40] M. G. S. Ortega, L. F. Rodriguez, and J. O. Gutierrez-Garcia, "Energy-aware data collection from the Internet of Things for building emotional profiles," in *Proc. Third International Conference on Fog and Mobile Edge Computing (FMEC)*, Barcelona Spain, 2018, pp. 234–239.
[41] N. M. Htaik, N. A. M. Maung, and W. Zaw, "Enhanced IoT-based interoperable and configurable middleware using semantic web techniques," in *Proc. 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Chiang Rai, Thailand, 2018, pp. 90–93.
[42] J. Aguilar, M. Jerez, and T. Rodríguez, "CAMeOnto: Context awareness meta ontology modeling," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 202–213, Jul. 2018.
[43] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-Aware computing, learning, and big data in internet of things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.
[44] D. Huru, C. Leordeanu, E. Apostol, M. Mocanu, and V. Cristea, "BigClue analytics: A middleware component for modeling sensor data in IoT systems," in *Proc. 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Exeter, United Kingdom, 2018, pp. 891–896.
[45] P. Temdee and R. Prasad, "Context-Aware middleware and applications," in *Context-Aware Communication and Computing: Applications for Smart Environment*, Cham: Springer International Publishing, 2018, pp. 127–148.
[46] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "CA4IOT: Context awareness for internet of things," in *Proc. IEEE International Conference on Green Computing and Communications*, Besancon, France, 2012, pp. 775–782.
[47] C. Simpkin, I. Taylor, D. Harborne, G. Bent, A. Preece, and R. K. Ganti, "Efficient orchestration of Node-RED IoT workflows using a vector symbolic architecture," *Future Generation Computer Systems*, vol. 111, pp. 117-131, 2020.
[48] M. J. A. Baig, M. T. Iqbal, M. Jamil, and J. Khan, "Design and implementation of an open-Source IoT and blockchain-based peer-to-peer energy trading platform using ESP32-S2, Node-Red and, MQTT protocol," Energy Reports, vol. 7, pp. 5733-5746, 2021.
[49] Practical Node-RED Programming, Packt. March 2021.
[50] S. Nandhinidevi, K. Saraswathi, M. Thangamani, and M. Ganthimathi. (2021). Design and development of bird ontology using protégé. Materials Today: Proceedings. [Online]. Available: https://doi.org/10.1016/j.matpr.2021.01.596

Mohammed Lamnaour received his engineering degree in Computer Science in 2009 from the University of Ibn-zohr Morocco in 2008. From 2008 to 2014, he worked as an analyst engineer in CGI-IT technologies Morocco. Since 2014, he worked as an engineer in a teaching position at the Office of Professional Training and Promotion of Work OFPPT- Tetouan and pursuing his Ph.D. His current research interests are in the field of internet of things and software deployment.

Mohammed Achkari Begdouri obtained the state engineer diploma in Computer Science from the National School of Applied Sciences of Tangier, Morocco, in 2011. Then, he joined the Interior Ministry as head of the information systems department in Tangier. He received the Ph.D. degree in 2017 from Abdelmalek Essaadi University, Morocco. He is currently Professor at the Polydisciplinary Faculty of Larache (Morocco); he teaches and supervises projects on software engineering, systems administration and information security. His research area focuses on Theoretical Computer Science and Software Applications. He published various works in

international journals. He participated in national and international conferences; he was member of various program committees, organizing committees and scientific projects.

Yasser Mesmoudi received his engineering degree in Computer Science in 2009 from the University of Abdelmalek Essaädi Morocco. He obtained also the PhD degree in Computer Sciences from the same university in 2019. Currently he is a professor at the National School of Applied Sciences at the UAE Morocco. He has published numerous papers in international journals and conferences. His current research interests are in the field of wireless sensor networks, Internet of things especially in software engineering. He is at the head of the digital pole at the same university.

Yasser El Khamlichi received his engineering degree in Telecommunications and Computer Networking in 2010 from the University of Abdelmalek Essaädi-Morocco. In 2017, he obtained also the PhD degree in Computer Sciences from the same university in 2019. Currently he is a professor at the National School of Applied Sciences at the UAE morocco. He has published numerous papers in international journals and conferences. He has been a member of the organizing committee at different international conferences and journals. His current research interests are in the field of wireless sensor networks and IoT, primarily in optimization techniques.

Abderrahim Tahiri received his engineering degree and PhD in Computer Science from the University of Abdelmalek Essaâdi -Morocco. Since 2001, he has led different academic and professional projects as an IS/IT engineer at the University. Since 2010, he has been an Associate Professor in the Department of Informatics at the National School of Applied Sciences of Tetouan, a School of Engineering at Abdelmalek Essaädi University. He has been a member of the school Council and he was appointed Coordinator of the Informatics branch then chief of department. He has published numerous papers in international journals and conferences. He has been a member of the organizing committee at different international conferences and journals. His main research interests include software engineering, wireless sensor networks applications and technologies.