A MIDDLEWARE BASED SERVICE ORIENTED APPROACH FOR WIRELESS SENSOR NETWORK

[1]YASSER MESMOUDI, [2]YASSER EL KHAMLICHI, [3]ABDERRAHIM TAHIRI, [4]ABDELLAH TOUHAFI, [5]AN BRAEKEN

[1]National school of applied sciences of Tetouan, *Abdelmalek Essaâdi University, Morocco*
[2]Department of Electronics and Informatics, Faculty of Engineering Sciences, Universiteit Brussel (VUB), Belgium
Email: {mesmoudi.yasser, yasser.khamlichi, abderahim.tahiri, abraeken}@gmail.com, Abdellah.touhafi@vub.ac.be

Abstract— Currently, the use of sensor networks in different field, such as environmental surveillance, health care, precision agriculture, traffic monitoring, etc.... is in high increase. Each wireless sensor network application developed gives birth to new challenges such as the transfer of a large amount of data over networks, the supported data formats, the measurement procedures and the heterogeneity between the different devices used. Most of Internet of Things solutions are developed in a vertical way containing the entire process, from data collection and transmission to processing and analyzing. However, the monopoly of these solutions by some companies prevents developers from reusing these solutions and creating others while combining them, due to the incompatibility of the data sources and the heterogeneity between the different devices. Hence, the need of developing a middleware as an intermediate software layer between the sensor hardware and the sensor network applications. In this paper, we study the existing middleware approaches for wireless sensor networks and we concluded with future research directions in the domain area to achieve the requirements of emerging applications of wireless sensor networks.

Index Terms— Data format, Heterogeneity, Middleware, Smart Gateway, WSN application.

I. INTRODUCTION

The presence of sensors in several mobile devices as well as the evolution and deployment of wireless sensor networks (WSN) in several fields mean that each person or organization now has many sensors, often heterogeneous. Built on different configurations and modes of operation, these sensors are not meant to communicate with each other and generate large amounts of data with different formats. The Wireless World Research Forum predicts that by 2017, there will be 7 trillion wireless devices serving 7 billion people [1] (i.e., 1000 devices/person). This ultra large number of connected things or devices will form the Internet of Things [2].

The ability to manage and process this data is a real need faced with many limitations and problems. This encourages researchers and manufacturers to offer hardware, software or architectural solutions, in order to efficiently exploit the capabilities of the current communication technologies and provide more flexible, reconfigurable and efficient sensor networks. Besides the data management, the proposed solution should provide data compatibility, bandwidth management, ensure connectivity and solve security problems [1], [3]. Therefore, there is a need for an intermediate software layer between the WSN hardware and its applications called "middleware". The middleware is required to fully meet the design and execution challenges of the WSN [4]-[6].

This paper presents the main challenges for designing WSN middleware, describes a review of existing middleware approaches, and their main features, and then provides a comparison among them. The present paper provides another perspective to the architecture of the middleware that will be our future work. It is structured as follows. In section 2 we define WSN middleware, their classifications and different challenges in this area. Then the related work is presented in section 3. Finally and before concluding a service oriented middleware architecture proposal and our future work is outlined.

II. WSN MIDDLEWARE DESIGN

A. *WSN Middleware Problem*

In heterogeneous WSN, sensor nodes have different characteristics, like different processing power, amount of memory and available energy [3], [6]. In order to meet the needs of their customers, and to develop their infrastructure to benefit from the technological advances in the field of WSN, most of independent administrations and companies deploy their own monitoring infrastructure and software architecture. The heterogeneous sensor nodes in WSNs can impact the entire network's capability. In the case of a mismatch in data formats and structure exchange between nodes, the system should provide a mechanism for heterogeneous nodes to handle mismatch data, since all nodes communicate only with nodes of a similar data structure and exchange data formats model. The mismatching of communication types exists due to the implantation of different formats of data [7].

Figure 1 illustrates some WSN applications designed for healthcare, road traffic monitoring and environmental monitoring and as demonstrated, the sensed information coming from heterogeneous sensors is only reachable through specific application services via the company solution. Gathering such information from heterogeneous devices and sharing its can be very strategic offering advanced services, but processing them can be very complex.

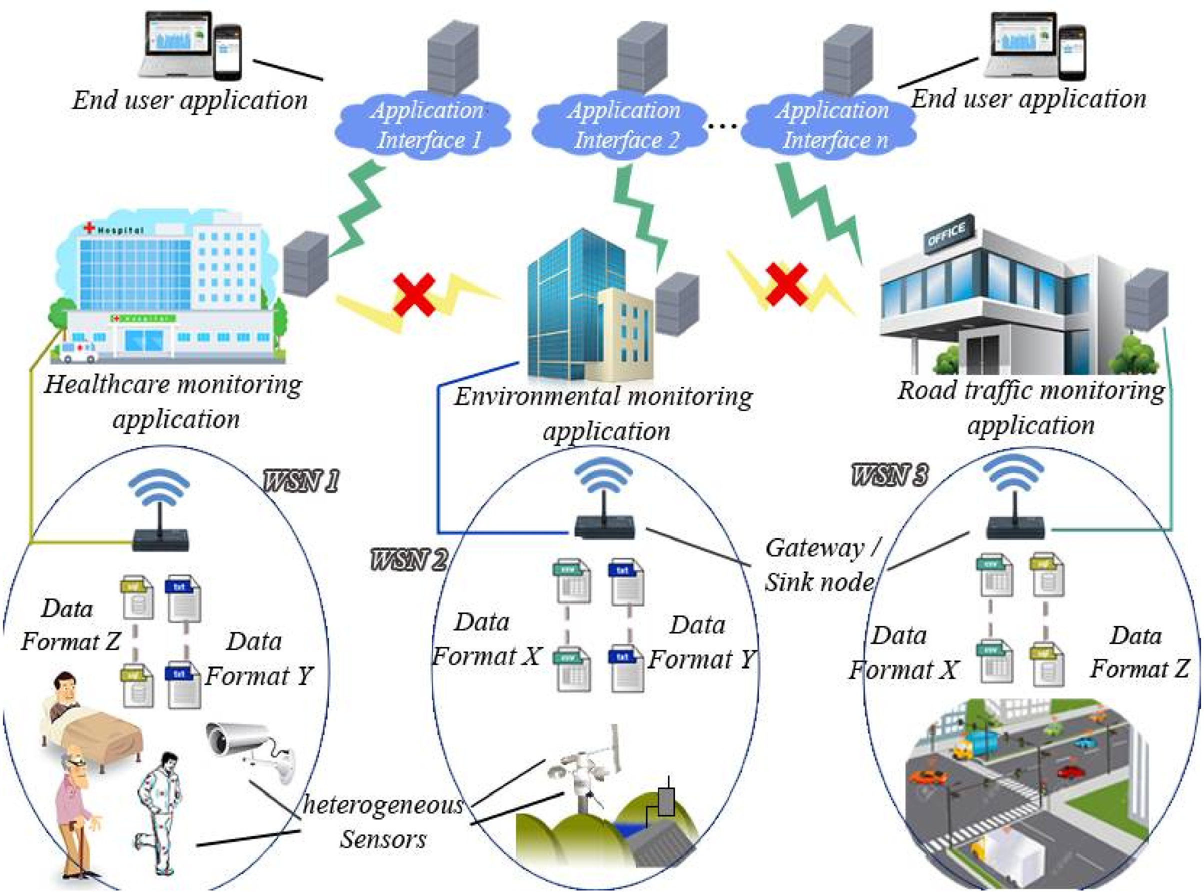

Figure1. Architecture design of WSN applications.

Therefore, the features provided by computing and communication hardware require to be matched by an appropriate software layer in the wireless sensor network system called middleware. This should enable programmers to easily and efficiently exploit the capabilities of the underlying hardware and other opportunities provided by the current communication technologies [4].

Middleware refers usually to software that sits on top of the operating systems and network protocols and below the application level. It solves problems of incompatibility and heterogeneity of hardware and networks connectivity. Its main functions at the WSN level are to ensure and facilitate the development, maintenance, execution and deployment of sensor-based applications [4], [8]. Based on [3], the proposed middleware should provide : (i) appropriate system abstractions, so that the application developer can focus on the application logic without having to deal with the lower level implementation details, (ii) standard and reusable services for several applications, so that developers can deploy and execute the application without worrying about complex, error-prone and tedious functions, (iii) runtime environment able to manage the execution of multiple applications, (iv) mechanisms for network infrastructure management and adaptation to allow the efficient use of WSN resources. It should also support interoperability with external networks, as the Internet, or enterprise systems.

B. Middleware designing challenges

Generally, the middleware performs the role of a translator that fills up the gap between the high level requirements of different applications running on wireless sensor networks and the complexity of different operations in the underlying sensor node hardware [4]. Figure 2 represents a logical architecture of a middleware within a WSN, on the one hand, on the sensor side, the middleware needs to deal with many challenges related to WSN characteristics and on the other hand, on the end user side, it should deal with the applications characteristics. Designing WSN middleware needs to meet several challenges [6]-[8]:

- Managing limited battery power and resources: With smaller and more compact sensors, the available battery power is always limited, middleware designed for this kind of nodes should provide a suitable mechanism for the use of limited memory and ensure efficient power consumption.
- Scalability, mobility and dynamic network topology: Middleware designed should be capable to maintain the required performance while the network environment is growing up. Scalability is challenged when any change occurs on large-scale networks.

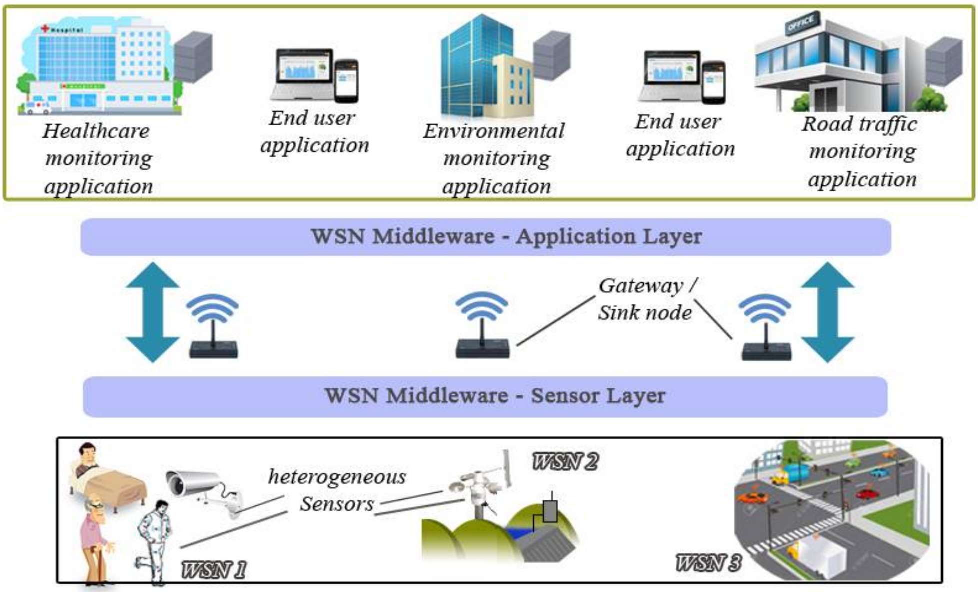

Figure 2. Logical architecture of a middleware

- Heterogeneity: The heterogeneity among the hardware, communication devices and configuration operations, have to be granted for the middleware.
- Real world integration: Middleware designed should provide real time services to be adapted to the eventual changes and data up to date.
- Data Aggregation: The aggregation of data within the network ensures that redundant data is not generated in the memory, saving costs through memory usage and energy through processing time, permitting to reduce the volume of data to transmit over the network.
- Application knowledge: The middleware must include mechanisms for injecting application knowledge of WSN infrastructure. This allows developers to map application communication requirements to network parameters which enable them to fine tune the network monitoring process.
- Quality of Service (QoS): The quality of service is important on the application level as well as on the network level. It is important for the WSN to support QoS since it defines the accuracy of data, coverage and tolerance.
- Security: The integration of security parameters in the system's design is necessary to achieve protection, since the collected information transmitted over the networks can be easily hacked by malicious intrusions and internet attacks.
- Fault tolerance: The integration of recovery methods in the system is necessary for enabling failure-resistant networks.

III. RELATED WORK

C. Classical WSN Middleware approaches

Based on [6], WSN middleware approaches can be classified into two main groups, based on how they work: those that run only inside the wireless sensor network (classical approaches) and those that integrate and process sensor data but work only outside the WSN. There are many proposals of middleware inside WSNs, with different architecture approaches such as database abstractions, mobile agents, virtual machines and application-driven and message-oriented middleware illustrated in Figure 3. Based on [1], [6] and [7] WSN middleware approaches on the inside are classified as follows:

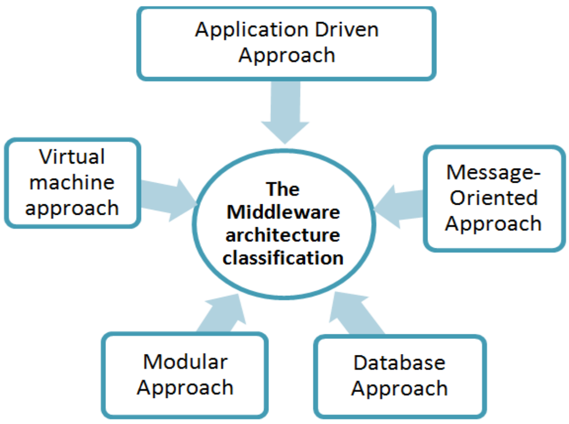

Figure 3. Classical WSN Middleware architectures

1) Database approaches (TinyDB [9], Cougar [10], SINA [11] and DsWare [12]) treat the sensor network as a virtual database, queried through an SQL-like language. They view the whole network as a virtual database system and provide an easy-to- use interface that lets the user issue queries to the sensor network to extract the data of interest but they do not support heterogeneity.
For instance, TinyDB only supports TinyOS and a single TinyOS network. This approach is good for regular queries, but does not support rendering data in real-time, it only provides approximate results [6], [8].

2) Modular Approach / mobile agent: The key to this approach is that applications are modular, and each module can be distributed through the network (mobile code). Transmitting small modules and providing simple and light updates, consumes considerably less energy than a whole application. It uses either the mobile agents or the codes that are injected directly into the WSN to collect the data. The mobile agent concept for sensor networks has been explored extensively in various approaches such as: Agila [13], Impala [14], SensorWare [15], COMiS [16], ScatterWeb [17], RUNES [18], MiSense [19], EMMON [20] and TeenyLIME [21]. The TeenyLIME approach also called Tuple space approach is a new middleware for sensor networks based on the tuple space model. It allows applications to add and view data in a common space of tuples using the 'in' and 'out' operators. The agent-based approach provides efficient mechanisms for network updates, in order to support dynamic applications. However, the nature of its code does not allow hardware heterogeneity [6].

3) The Virtual Machine Approach is flexible and contains virtual machines (VMs), and similar to the mobile agent concept, where arbitrary code can be run, virtual machine middleware is more general because it does not associate code updates with specific structure. But, since virtual machine execution involves code interpretation, there is a significant runtime overhead cost compared to native binary code.

   Virtual machine approaches such as Maté [22], SwissQM [23] and Squawk [24] provide a flexible programming paradigm. They allow the development of distributed algorithms and hide the heterogeneity of the runtime environments and the hardware resources. However, the virtual machine approaches add a considerable code size and performance overhead, and applications need to be programmed in detail.

4) Message-oriented / Event based approach: When designing real-time systems, the time-triggered approach is expensive in the case where the expected rate of primitive event occurrence is low. An alternative is to use an event-triggered approach, where the execution is driven by the events. Messages that circulate through the middleware are asynchronous and the communication is based on the paradigm publish / subscribe.

   Event-driven communication is an asynchronous paradigm that decouples senders and receivers. This paradigm is based on publisher events (sensor node) and subscriber events (the sink node) [6]. To explore this concept, many approaches have been developed such as Mires [25], ATaG [26], TinySOA [27] and USEME [28].

5) Application driven approach: This middleware allows the application to identify their QoS requirements then can modify the network according to application needs. Thus, the application manages the network while giving the quality of the service an extreme priority. The Middleware Linking Application and Network (MiLAN) is one of the examples of the application driven [29]. This approach does not support the heterogeneity of sensors hardware [6], [7].

Comparative study of different wsn middleware approaches was presented in [3], [4] and [6]-[8] that describes the advantages and limitations of each approach. In our case we focus on the heterogeneity constraint. Many of those approaches were developed for specific platforms. They do not offer heterogeneity support and interoperability between heterogeneous parts of the network. Consequently, developing and deploying end-to-end applications for sensor networks remains highly complex [6].

D. WSN Middleware Approaches outside the network

1) Internet-Based Integration of Sensor Data: Traditionally, sensor networks are not IP based. Their integration into IP-based WAN infrastructures requires the deployment of proxies at the edge of both networking domains that transform between non-IP communication in the sensor network and IP communication in the Internet. For this problem a generic approach to connect all the devices has to be developed. Several research projects were developed such as: GSN [30], Borealis [31], IrisNet [32], Hourglass [33], HiFi [34], SStreaMWare [35], EdgeServers [36], ESP framework [37]. These middleware were developed to integrate sensor data into the Internet. Consequently, they are only focused on wrapping data coming from sensor sources for sharing and processing over the Internet. Those works provide heterogeneity out of WSN [6].

2) IP-Based Homogeneous Middleware: There have been several efforts to implement the Internet protocol stack on small constrained devices. The 6LoWPAN [38] ports the IPv6 protocol to small devices. This enables running services on the application layer directly on sensor nodes, which enables the integration of a Service-Oriented Architecture (SOA) into the design of WSN middleware [6].

E. The Service oriented middleware

The Service oriented middleware (SOM) architecture is the best platform to develop WSN applications to address hardware challenges such as QoS, security, and heterogeneity [7]. It’s one of the most recent approaches that provide high flexibility for adding new and advanced functions to WSN middleware. SOM logically views the WSN as a service provider for consumer applications. The adoption of service oriented approach provides WSN users with a unified protocol to access and communicate with the WSN components and developers with a flexible programming model to build efficient and scalable WSN systems [3].

Table 1 shows some WSN middleware based SOA approach with their different advantages and limitations. Some approaches do not provide any mechanisms that are independent of the middleware. instead, they depend on particular operating systems.

SOM Architecture	The Features and Advantages	Disadvantages	Operating System Independence	Data/ service Aggregat -ion support	Heteroge -neity support
OASIS [39]	1. Development of environment based on separation of concerns 2. Supports the node management 3.Supports QoS 4. Dynamic service discovery 5. Failure detection	1. Not provides a secure communication/execution 2. Cannot integrates with other systems 3. Not supports self-organization mechanisms 4. Not supports interoperability with various systems and devices	Independent	√	√
MiSense [40]	1. Content based publish/subscribe service 2. Provide programming API 3. Supports data management	1. Not support configurable services 2. Not supports self-organization 3. Not provides a secure communication/execution 4. Not support QoS 5. Increase power consumption and processing time	Built on top of TinyOS operating system	√	X
SOMDM [41]	1. Decreased the data processing load by using multi-component architecture 2. Notification and data filtering techniques 3. Handle a large of data and high communication loads efficiently	1. Not support configurable services 2. Not supports self-organization 3. Not provides a secure communication/execution 4. Not support QoS	Independent	X	X
Mob-WS [42]	Increases the scalability	1. Not provides a secure communication/execution 2. Not support QoS	Independent	X	X
SensorsMW [43]	1. The QoS configuration is provided by service level 2. Providing mechanism for the application to manage WSNs	1. Not supports self-organization 2. Not provides a secure communication/execution 3. Not support nodes with low capacity	Independent	√	X
SOMM [44]	1. Supports multimedia transmission 2. Ability to reduce the cost of development applications 3. Supports scalability	1. Overhead 2. Not support massive data and high communication loads efficiently. 3. Not very easy to use due to its implementation that used a comprises byte code	Independent	X	√
SAWM [45]	Provides secure architecture and modifiable	Not provides a secure communication	Independent	X	√
TinySOA [46]	1. It provides web service for internet Apps to access WSN 2. Supports multiple programming language	1. Not support configurable services 2. Not supports self-organization 3. Not provides a secure communication/execution 4. Not support QoS	Independent	X	X
ESOA [47]	1. Allows users to develop new applications through mix-and-match services. 2. Execute various applications on multi-platforms 3. It can integrate with other systems 4. Provides a secure communication/execution through QoS composition	1. Do not provides mechanism to handle a data collection of user to the services 2. Not applied in real time	WSN Built on top of LiteOS operating system	√	√

Table 1. Comparative of WSN middleware based SOA approach

The ESOA framework is built on LiteOS while MiSense is built over TinyOS. The support for heterogeneous multi-service composition highlights the enhancement of service interworking and provisioning to end-users, enabling service orchestration, and discovery at the middleware level. However, these mechanisms are only provided in OASIS, and ESOA approaches. On the other hand, the security mechanisms have been taken into account through different SOM architectures approaches like SOMM, ESOA, and SAWM. Data or service aggregation is supported in approaches like OASiS, MiSense, SensorsMW, and ESOA. However, most of these approaches do not provide specific implementation and mechanism details [7]. The SOM architectures for WSNs should provide different functionalities that support the system. However, most of the studies on SOM architectures approaches do not provide all functionalities, including: Interoperability heterogeneity, scalability, discovery, security, massive data support and QoS support.

The most disseminated technology for service implementation is Web Services technology. Web service technology is often used to connect and access sensors and actuators through the Internet. Two standards in this area have emerged; the Representational State Transfer (REST)-based approach and Simple Object Access Protocol (SOAP) based web services. Rest utilizes the common Hypertext Transfer Protocol (HTTP) methods (get, post) to transfer data, and SOAP uses messages to communicate between services. The structure of the messages and the way to handle those are predefined in the web service specification. SOAP is less flexible than REST.

Recently, a REST-based web transfer protocol was developed called Constrained Application Protocol (CoAP) [48]. CoAP includes the HTTP functionalities which have been re-designed taking into account the low processing power and energy consumption constraints of small embedded devices. Since the Web services technology uses XML as the encoding system, data is easily exchanged between computing systems with incompatible architectures and incompatible data formats. Another lightweight protocol that shows efficiency instead of XML format is called JavaScript Object Notation (JSON) [7].

Web Service Definition Language (WSDL) completely describes the Web service interface, while SOAP completely describes parameters, data types and exceptions included in a message being exchanged between Web services [3]. These works assume that sensor nodes are powerful enough to run a μ IP protocol. Depending on the hardware platform, nodes may not have enough resources to support the overhead introduced by the IP protocol stack. IPv6, REST and CoAP are important advances in WSN. They provide an infrastructure, but without further software, operations and interactions between nodes must still be hand-programmed [6].

IV. MIDDLEWARE ARCHITECTURE PROPOSAL

As presented above, the presence of several heterogeneous sensor nodes in disparate networks generates a huge amount of data with different formats. Heterogeneity makes providing a common data management system a challenging task as it should be flexible and configurable. Besides, these solutions are mostly deployed for specific application; it requires a specific data format, which avoids programmers to reuse them. Moreover, collecting and processing data into a WSN should meet this heterogeneity and must be externally to the network. So, it takes advantage of other resource capabilities.

With the emergence of new programming trends, techniques such as web services, semantic web, linked data and cloud computing have been adopted to improve software programming. In our future work illustrated on Figure 4, we focus on the development of Distributed Context-Aware Sensor Service Bus.

The sensor platform should be future proof by adding reconfigurability, modularity, extensibility, and integration ability, allowing a whole set of parameters to be measured and allowing collaboration with other data sources on the Web. It will be IP based and sensors will play the role of native web servers via HTTP or CoAP or they will be linked to a gateway that exposes their data on the Web. Common data formats such as XML and JSON will be applied in order to facilitate the interoperability in the global network and to avoid vendor lock-in.

The sensor/actuator node will contain a local-context processing engine, such that local sensor-data can be processed and interpreted while taking local-context information into account. This context awareness will help to detect tampering attempts and causes of failure and contribute to the robustness by improving the failure and tampering resilience of the network. It can also assist in starting an appropriate reconfiguration mechanism.

Using metadata and semantic annotations to describe sensor data, data coming from other sources like the (social) web and in general physical world resources on a scalable heterogeneous platform, will help to understand the connection between the descriptions in the model and the data stream operations such as discovery, indexing and querying from applications, services or systems using these data. The device discovery requires the ontology to represent sensor types, models, methods of operation and metrological definitions such as accuracy, precision measurement range like for allowing sensor capabilities to be defined relative to prevailing conditions.

Finally, appropriate security mechanisms will allow secure remote reconfiguration of the node, to transmit and store encrypted and authenticated data on the node. Also local access control mechanisms will be identified on the node itself in order to get immediate secure access to it.

CONCLUSION

Nowadays sensor networks have found its application in nearly every field, such as, home automation, industrial automation, medical aids, mobile healthcare, elderly assistance, intelligent energy management and smart grids, automotive, traffic management, and many others. In order to integrate heterogeneous computing and communication devices, and support interoperability within the diverse applications and services, it's obligatory to develop middleware that can manage and process this great amount of data.

In this paper, we have discussed the main challenges of designing a WSN middleware. We surveyed the existing approaches and some examples of each approach and a comparative study is presented. Finally, we presented a new direction of research which is the development of

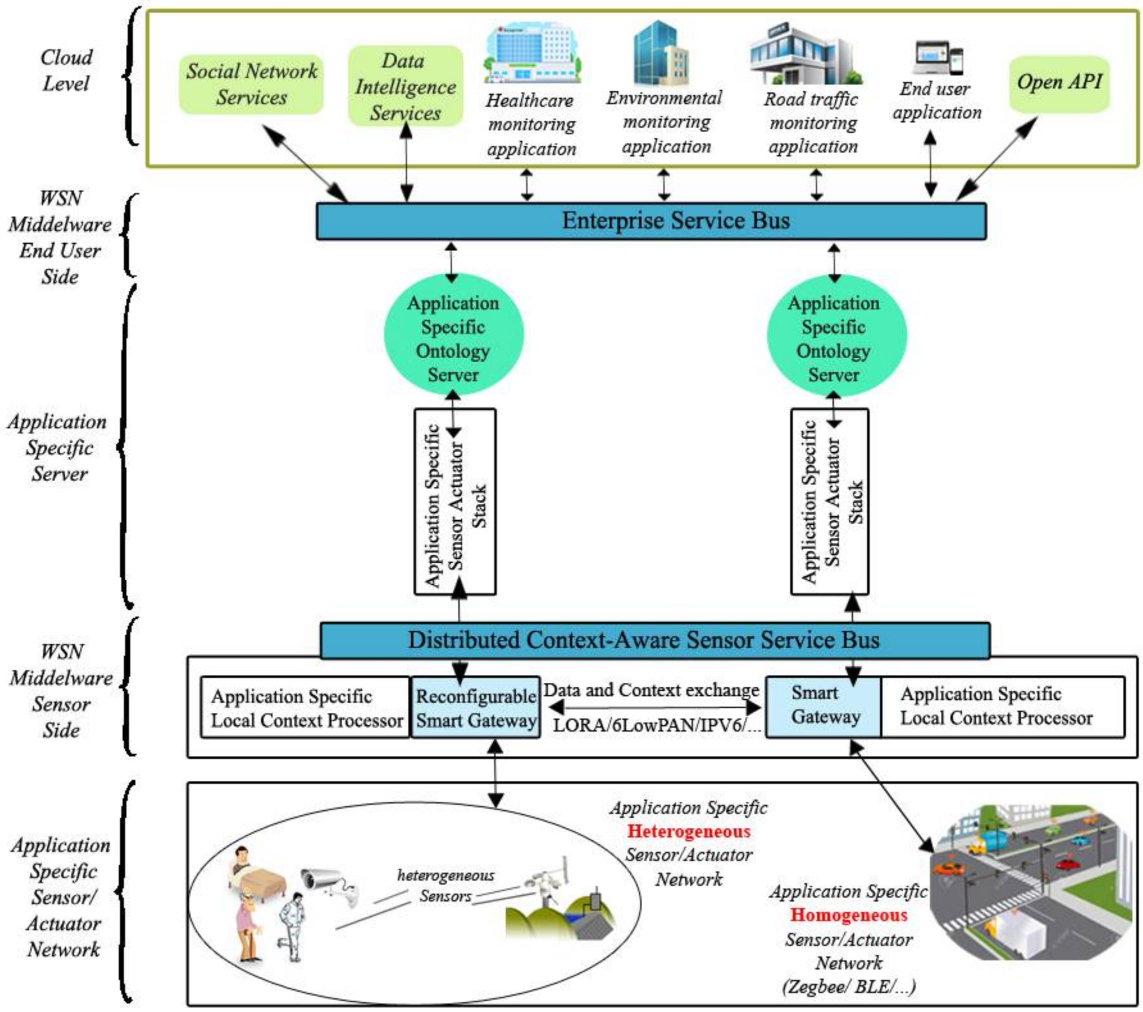

Figure 4. The SOM architecture proposa

distributed Context-Aware Sensor Service Bus and Application Specific Ontologies to meet the requirements of emerging applications. We will focus on the basic building blocks required to implement the proposed model, this will be realized by developing two elements. The first one which we are working on, is a set of reconfigurable smart gateway interconnected with a local context processor. The second is a set of application specific ontology servers. The result of this work will be published as an article.

REFERENCES

[1] 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, 2016. Doi: 10.1109/JIOT.2015.2498900J.

[2] K. Paridel, E. Bainomugisha, Y. Vanrompay, Y.Berbers and W. De Meuter. "Middleware for the internet of things, design goals and challenges", Electronic Communications of the EASST, 28, June 2010.

[3] F.C. Delicato, P.F. Pires, A.Y. Zomaya. "Middleware Platforms: State of the Art, New Issues, and Future Trends". In: H. Ammari (eds). The Art of Wireless Sensor Networks. Signals and Communication Technology, pp 645-674, 2014. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-40009-4_19.

[4] 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. DOI: 10.5923/j.jwnc.20140401.02.

[5] A. Raj Prajapati, P. Arvindkumar Prajapati. "Middleware: Middleware Approaches and Evaluation Framework for Wireless Sensor Networks". International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 5, pp. 428-430, 2014.

[6] J. Cecílio, P. Furtado. "Existing Middleware Solutions for Wireless Sensor Networks". In: Wireless Sensors in Heterogeneous Networked Systems. Computer Communications and Networks. pp 39-59, 2014, Springer, Cham. Doi: 10.1007/978-3-319-09280-5.

[7] R. Alshinina, K. Elleithy. "Performance and Challenges of Service-Oriented Architecture for Wireless Sensor Networks". Sensors, vol. 17, no 3, pp. 536-575, 2017. Doi:10.3390/s17030536

[8] M.E.K. Ajana, H. Harroud, M. Boulmalf, M. Elkoutbi. "Middleware Architecture in WSN". In: D. Benhaddou, A. Al-Fuqaha (eds) "Wireless Sensor and Mobile Ad-Hoc Networks", pp 69-94, 2014, Springer, New York. DOI 10.1007/978-1-4939-2468-4_4.

[9] SR. Madden, MJ. Franklin, JM. Hellerstein, W. Hong. "TinyDB: an acquisitional query processing system for sensor networks". ACM Trans Database Syst, vol. 30, no. 1, pp.122–173, 2005.

[10] Y. Yao, J. Gehrke. "The cougar approach to in-network query processing in sensor networks". ACM SIGMOD Rec. vol. 31, no. 3, 9,2002.

[11] CC. Shen, C. Srisathapornphat, C. Jaikaeo. "Sensor information networking architecture and applications". IEEE Pers Commun, vol. 8, no.4, pp. 52–59, 2001.

[12] X. Yu, K. Niyogi, S. Mehrotra. "Adaptive middleware for distributed sensor environments". In: IEEE distributed systems online, vol 4, no. 5, 2003.

[13] CL. Fok, GC. Roman, C. Lu. "Rapid development and flexible deployment of adaptive wireless sensor network applications. In: Proceedings of the 25th IEEE international conference on distributed computing systems ICDCS05. IEEE, Los Alamitos, pp 653–662, 2005.

[14] T. Liu, M. Martonosi. "Impala: a middleware system for managing autonomic, parallel sensor systems". System, vol. 38, no.10, pp. 107–118, 2003.

[15] A. Boulis, CC. Han, R. Shea, MB. Srivastava. "SensorWare: programming sensor networks beyond code update and querying. Pervasive Mobile Comput, vol. 3, no. 4, pp. 386–412, 2007.

[16] D. Janakiram, R. Venkateswarlu, S. Nitin. "COMiS: component oriented middleware for sensor networks". In: Proceedings of the 14th IEEE workshop on local area and metropolitan networks (LANMAN). IEEE, Piscataway, 2005.

[17] J. Schiller, A. Liers, H. Ritter, R. Winter, T. Voigt. "ScatterWeb – low power sensor nodes and energy aware routing". In: Proceedings of the 38th annual Hawaii international conference on system sciences 2005 HICSS 05, vol 00, no. C. IEEE, Los Alamitos, pp 1–9, 2005.

[18] F. Oldewurtel, J. Riihijarvi, K. Rerkrai, P. Mahonen. "The RUNES architecture for reconfigurable embedded and sensor networks". In: Proceedings of the 2009 third international conference on sensor technologies and applications. IEEE, Piscataway, pp 109–116, 2009.

[19] KK. Khedo, RK. Subramanian. "A service-oriented component-based middleware architecture for wireless sensor networks". J Comput Sci, vol. 9, vol. 3, pp. 174–182, 2009.

[20] S. Tennina, M. Bouroche, P. Braga, R. Gomes, M. Alves, F. Mirza, V. Ciriello, G. Carrozza, P. Oliveira, V. Cahill. "EMMON: a WSN system architecture for large scale and dense real-time embedded monitoring". In: Proceedings of the 2011 IFIP 9th international conference on embedded and ubiquitous computing. IEEE, Piscataway, pp 150–157, 2011.

[21] P. Costa, L. Mottola, AL. Murphy, G. Pietro Picco. "TeenyLIME: transiently shared tuple space middleware for wireless sensor networks". In: Proceedings of the international workshop on middleware for sensor networks. ACM, New York, pp 43–48, 2006.

[22] P. Levis, D. Culler. "Maté: a tiny virtual machine for sensor networks". In: ASPLOSX proceedings of the 10th international conference on architectural support for programming languages and operating systems, vol 36, no. 5. ACM, New York, pp 85–95, 2002.

[23] R. Müller, G. Alonso, D. Kossmann. "SwissQM: next generation data processing in sensor networks". In: Proceedings of the 3rd biennial conference on innovative data systems research CIDR07, Asilomar pp 1–9, 2007.

[24] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, D. White. "Java on the bare metal of wireless sensor devices: the squawk Java virtual machine". In: Proceedings of the 2nd international conference on Virtual execution environments, pp 78–88, 2006.

[25] E. Souto, G. Guimarães, G. Vasconcelos, M. Vieira, N. Rosa, C. Ferraz, J. Kelner. "Mires: a publish/subscribe middleware for sensor networks". Pers Ubiquit Comput vol. 10, no. 1, pp. 37–44, 2005.

[26] A. Bakshi, VK. Prasanna, J. Reich, D. Larner. "The abstract task graph: a methodology for architecture-independent programming of networked sensor systems". In: Proceedings of the 2005 workshop on end-to-end sense-and-respond systems applications and services (EESR 05), Berkeley, pp 19–24, 2005.

[27] A. Rezgui, M. Eltoweissy. "Service-oriented sensor–actuator networks: promises challenges, and the road ahead". Comput Commun vol. 30, no. 13, pp. 2627–2648, 2007.

[28] E. Cañete, J. Chen, M. Díaz, L. Llopis, B. Rubio. "A service-oriented middleware for wireless sensor and actor networks". In: Proceedings of the 2009 sixth international conference on information technology new generations, vol 25, no. 6. IEEE, Piscataway, pp 575–580, 2009.

[29] A. Murphy, W. Heinzelman. "Milan: middleware linking applications and networks". Technical report, University of Rochester, Rochester, NY, USA, pp 1–16, 2002.

[30] K. Aberer, M. Hauswirth, 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. [Online]. Available: http://lsirpeople.epfl.ch/hauswirth/papers/LSIR-REPORT-2006-006.pdf.

[31] DJ. Abadi, Y. Ahmad, M. Balazinska, J. Hwang, W. Lindner, AS. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, S. Zdonik. "The design of the borealis stream processing engine". Time, pp 277–289, 2005.

[32] PB. Gibbons, B. Karp, S. Nath, S. Seshan. "IrisNet: an architecture for a worldwide sensor web". IEEE Pervasive Comput vol. 2, no. 4, pp. 22–33, 2003.

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

[34] MJ. Franklin, SR. Jeffery, S. Krishnamurthy, F. Reiss, S. Rizvi, E. Wu, O. Cooper, A. Edakkunni, W. Hong. "Design considerations for high fan-in systems: the HiFi approach". In: CIDR 2005 proceedings of second biennial conference on innovative data systems research, Asilomar, pp 290–304, 2005.

[35] L. Gurgen, C. Roncancio, C. Labbé, A. Bottaro, V. Olive. "SStreaMWare: a service oriented middleware for heterogeneous sensor data management". In: Proceedings of the 5th international conference on Pervasive services, pp 121–130, 2008.

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

[37] T. Kobialka, R. Buyya, C. Leckie, R. Kotagiri. "A sensor web middleware with stateful services for heterogeneous sensor networks". In: Proceedings of the 2007 3rd international conference on intelligent sensors sensor networks and information. IEEE, Piscataway, pp 491–496, 2007.

[38] G. Mulligan. "The 6LoWPAN architecture". In: Proceedings of the 4th workshop on embedded networked sensors. ACM, New York, pp 78–82, 2007.

[39] X. Koutsoukos, M. Kushwaha, I. Amundson, S. Neema, J. Sztipanovits. "Oasis: A service-oriented architecture for ambient-aware sensor networks". In Lecture Notes in Computer Science (LNCS); Kordon, F., Sokolsky, O., Eds.; Springer: Berlin/Heidelberg, Germany, pp. 125–149, 2008.

[40] A. Malatras, A. sgari, T. BaugÉ. "Web enabled wireless sensor networks for facilities management". IEEE Syst. J., vol. 2, pp. 500–512, 2008. DOI: 10.1109/JSYST.2008.2007815

[41] K.P.A.S.V. Patel. "A novel design of service oriented and message driven middleware for ambient aware wireless sensor network". Int. J. Recent Trends Eng. Vol. 1, pp. 313–317, 2009.

[42] F. Aijaz, S.M. Adeli, B. Walke. "A service-oriented approach for in-network computations in wireless networks". In Proceedings of the International Conference on Wireless and Optical Communications Networks (WOCN), Cairo, Egypt, vol. 28, no. 30, pp. 1–6, 2009.

[43] G. Anastasi, E. Bini, A. Romano, G. Lipari. "A service-oriented architecture for QoS configuration and management of wireless sensor networks". In Proceedings of the IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010), Bilbao, Spain, pp. 1–8, 2010.

[44] M.M. Faghih, M.E. Moghaddam. "Somm: A new service oriented middleware for generic wireless multimedia sensor networks based on code mobility". Sensor, vol. 11, pp. 10343–10371, 2011. doi:10.3390/s111110343.

[45] F. Sheikhha, M.E. Moghaddam. "Service-oriented wireless multimedia sensor network middleware using infra-red cameras". In Proceedings of the Third International Conference on Mobile, Ubiquitous, and Intelligent

Computing (MUSIC), Vancouver, BC, Canada, pp. 230–235, 2012.

[46] E. Avilés-López, J.A. García-Macías. "Tinysoa: A service-oriented architecture for wireless sensor networks". Serv. Oriented Comput. Appl. Vol. 3, pp. 99–108, 2009. DOI 10.1007/s11761-009-0043-x.

[47] V. Vanitha, V. Palanisamy, N. Johnson, G. Aravindhbabu. "Liteos based extended service oriented architecture for wireless sensor networks". Int. J. Comput. Electr. Eng. Vol. 2, pp. 432–436, 2010. DOI: 10.7763/IJCEE.2010.V2.173.

[48] Z. Shelby, C. Bormann, B. Frank. "Constrained application protocol (CoAP), 2011.. In: An online version is available at: https://datatracker.ietf.org/doc/rfc7252/?include_text=1.

★★★