Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Study of Semantic Middleware for Internet of Things

Viraj Avachat*, Poonam Gupta
Dept. of Computer Engg, G. H. Raisoni College of Engg & Mgmt,
Pune, Maharashtra, India

Abstract— The Internet of Things is a revolutionary paradigm that makes day-to-day physical objects smart, which communicate with each other with self-autonomy and achieve common goals. Large number of applications can be thought of with IoT Paradigm in various domains such as Transportation, Logistics, Healthcare, Home and Offices. Middleware for IoT is a software layer which acts as a bridge between the IT infrastructure and IoT Application. It helps in hiding the technological implementation details for the programmers. The Ideal Middleware consists of various Ideal Characteristics such as Scalability, Interoperability etc.

Keywords— Internet of Things, Ubiquitous Computing, Middleware, Semantic Web, Soft Computing

I. INTRODUCTION

Internet of Things is a novel and revolutionary paradigm in the ubiquitous computing era. In this paradigm objects or things can be uniquely identified and with the help of Internet they can connect and communicate with each other without any human intervention. The "Internet of Things" term was first introduced by Kevin Ashton as a title of his presentation in 1999 [1].

The concept of the Internet of Things paradigm is that everyday objects with the help of agents i.e. sensors or actuators can be uniquely identified with unique addressing schemes and are capable of interacting and co-operating with each other to reach common goals [2].

Some of the authors have defined Internet of Things. These definitions are as follows:

Definition by [3]: "Things have identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environment, and user contexts."

Definition by [4]: "The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service."

The Internet of Things paradigm can be viewed as one vision, many paradigms [2]. The paradigms can be classified as Internet-oriented, Things-oriented and Semantic-oriented. The focus of Internet-oriented paradigm is to uniquely identify the things and hence it focuses on IP for Smart Objects and Web of Things. The Things-oriented paradigm mainly focuses on the objects such as sensors, actuators etc. While the Semantic-oriented paradigm mainly focuses on the Smart Internet of Things which are capable of taking independent decisions by reasoning over data and machine learning.

With the Internet of Things paradigm there can be potentially huge number of applications in multiple domains such as Transportation and Logistics, Health Care, Smart Home and Office etc.

The Middleware of the Internet of Things is a software layer or set of sub layers which bridge the IT infrastructure of IoT and IoT Application [2]. The main idea of using a middleware for IoT is hiding the technological implementation details of the IT infrastructure, so that the programmers can easily develop number of different IoT Applications. The main aim of the middleware is to provide abstraction as well as provide multiple services.

There are several reasons why Middleware for IoT is necessary [5]. They are as follows:

- Middleware helps to define and enforce a common standard among various heterogeneous devices.
- It acts as glue to bond different types of objects together.
- It provides interface for communication between devices and their services, hiding all the details of diversity.

With the help of Semantic web technologies, Middleware can provide the interoperability between the objects and the information. The IoT applications can become context aware, which helps each searching for the required services. IoT applications become autonomous and taking intelligent decisions without human intervention. The semantic data is better understood by the "things" as well as humans [6].

The next section defines various characteristics that are essential in building a middleware for IoT. In Section 3, we discuss some existing Semantic Middleware solutions. In Section 4, we compare these existing Middleware solutions with the required characteristics and see whether they fulfil them or not. In Section 5, we discuss the future scope in the research of Middleware for IoT.

II. CHARACTERISTICS OF IDEAL MIDDLEWARE

The following is the list of desired characteristics that the Middleware should possess [5], [7], [8].

A. Scalability

Scalability is one of the desired characteristics that a Middleware should possess. The IoT paradigm supposes to have thousands of devices collaborating and communicating with each other at a given time to accomplish some goals. Hence the Middleware is supposed to make this work smoothly. It should effectively function and manage the objects so that it can smoothly operate in small-scale as well as large-scale environments.

The Middleware should deal with four major scalability issues:

- a. Large Network Size
- Massive Number of Events
- Mobility Rate
- Heterogeneous Devices

B. Spontaneous Interaction

As many objects interact with each other in IoT, a single object will have to interact with one or more than one objects simultaneously. The Middleware should have the capability to let a single object interact with multiple objects simultaneously, without affecting the outcome of one object due to some other object that it is interacting simultaneously.

Also the Middleware should mange new interactions coming from the new discovered objects along with its current on-going interactions.

C. Zero Infrastructure

The objects that participate in an IoT application should be capable of discovering other objects as well as should understand what services these objects are providing. Also it should be capable to announce its presence and publish its services, so that the other objects can discover and utilise its services.

This is in contrast with the traditional distributed environment, where the resource and services publication, discovery and communication is managed by a dedicated server.

D. Multiplicity

The objects that are participating in an IoT application often have to rely on the services offered by different objects. When selecting an object from multiple objects which offer same service can be a big challenge. Also the object has to deal with large number of results returned by different services at a given time. This can be further challenging if the result from different services for a given request may contradict with each other.

Another challenge face by the object is to connect and communicate with multiple objects simultaneously.

E. Interoperability

Interoperability is a huge challenge that middleware for IoT applications face as numerous heterogeneous objects collaborate together which communicate and exchange information. The middleware is expected to accommodate large number of diverse devices to which it has to collaborate. It is also expected that the middleware accommodates newer unknown devices that will be discovered in the near future.

Interoperation can be classified under three categories i.e. network, syntactic and semantics. The interaction and communication related protocols come under network interoperation. The structures of information to be exchanged among the objects come.

F. Context Detection

The situation of an object can be understood with the help of Context, where object can be person, place, or things that are relevant in the IoT application. The middleware is expected to be context aware for working into smart environments.

Context Detection and Context Processing are the steps to achieve Context Awareness. The data that is collected is processed, the data is identified which has the significant impact for the response. While the context data is processed to identify the meaning and accordingly the Context processing performs or takes decision based on that.

G. Device Discovery and Management

The IoT middleware is expected to detect all the neighbouring objects as well as it have to make its presence felt to each of its neighbouring objects in the network.

H. Security and Privacy

Security and Privacy is very important challenge that middleware for IoT should fulfil. The communication between the objects takes place automatically without any human intervention can possess a real huge challenge in terms of trust, privacy and security. Confidentiality, Authenticity and Non-Repudiation are the important factors that are necessary to be considered while implementing security and privacy.

I. Abstraction

The middleware should provide abstraction which will be helpful for the developers to develop IoT applications without getting delved into deeper IT infrastructure implementation of IoT.

J. Managing Large Volume of Data

IoT middleware should be designed in such a way that it can manage large volume of data. The objects that take part in IoT are increasing day by day and it is believed that trillions of objects will be part of this huge network. As more and

more objects get interconnected, more and more data will flow and objects will be required to handle huge amounts of data. The middleware will have deal with challenges like context detection and processing for enormous flow of data, as well as querying, indexing, modelling and transactional handling of data.

III. EXISTING WORK

A triple space-based distributed middleware for the IoT [9] is based on Tuple space-based computing. In the tuple space-based computing instead of message based communication between the objects, the communication is done using shared space. The objects read and write data from the shared space. The proposed middleware use Semantic Web vision, hence it expresses the semantic data in RDF (Resource Description Framework) triples, i.e. "the subject, the predicate and the object". This middleware offers reference autonomy (the objects can communicate with each other without knowing anything), time autonomy (asynchronous communication between objects) and space autonomy (objects can be executed in different computational environments). The objects that need to offer its services will register its services in a shared space. Such objects are referred as service providers. While the objects which needs the services would create an invocation and advertise it. Such objects are called the consumers. The service provider will recognize the needs of the consumer; it will get the input data from the consumer and perform the desired service. This middleware proposes API for the devices to run this middleware. The API is divided into two sets: basic primitives and advanced primitives. The mobile and embedded devices are allowed to run this middleware. But, some devices cannot use the advanced primitives as they do not have the capacity to run them. The advance primitives include service management and complex queries. The Triple space-based framework offers following characteristics Interoperability, Abstraction, Spontaneous Interaction, Zero Infrastructure, Multiplicity, Security and Privacy, Device Discovery and Management and Context Detection but it lacks Scalability which is its major drawback.

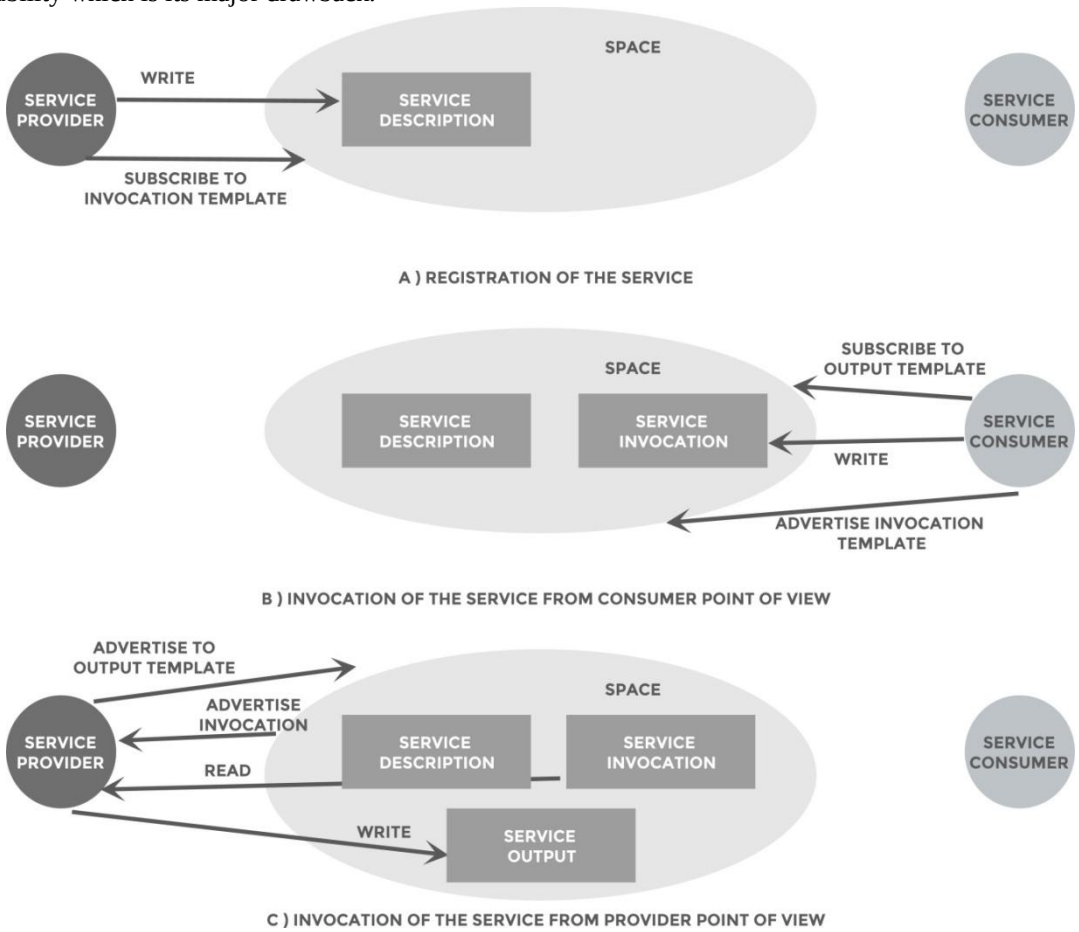

Fig. 1 Triple Space-based distributed middleware - Services

The UBIWARE research project proposed in [10] has the vision to create self-managed, autonomous and proactive components that can do the tasks of automatic integration, orchestration and composition of complex systems on the Internet of Things. This middleware uses the "agent technology". With the help of the agent technology the objects can easily and automatically discover each other and hence complex functionalities can be configured on those objects. In this middleware, each object has an autonomous software agent which represents that object. This software agent has various responsibilities such as monitoring the state of the object, discovering service requests, making decisions and requesting external help if it is needed. The object is connected with its software agent through the adapter or interface. The adapter or the interface includes sensors, actuators, data structures and semantic adapter components. Hence the object and it software agent along with its adapter wholly become a "smart resource" which is proactive and self-managing. With the implementation of agent technology many possibilities like mobility of services between

heterogeneous platforms, service discovery in decentralized manner, negotiation-based service integrations etc. have become true. The following characteristics are offered by the middleware: (i) interoperability which is possible using metadata and ontologies, (ii) Spontaneous interaction, (iii) Scalability, (iv) Zero Infrastructure, (v) Multiplicity, (vi) Context Detection, (vii) Device Discovery and Management and (viii) Abstraction. The major drawback this middleware faces is security and privacy.

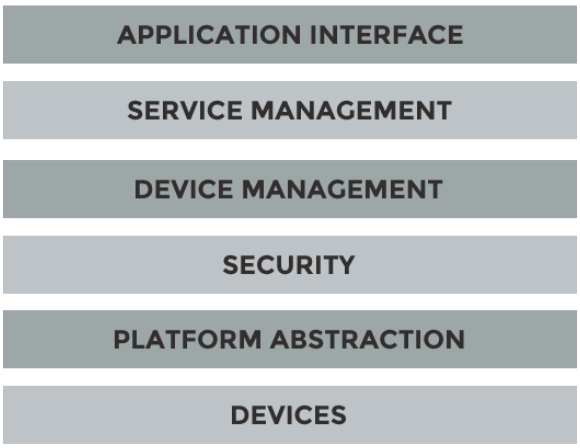

Fig. 2 Layers in SOA based Middleware

Middleware based on Service-Oriented Architecture (SOA) is proposed in [11]. In this approach each objects offers its functionality as a service. This improves the reactivity i.e. discovery of the services that are available and its invocation. The SOA-based vertical integration reduces the cost and the effort that is required to realize a business scenario because device drivers or third-party solutions are not required. The proposed architecture of the middleware can be identified as layers. Hence this middleware architecture supports open and standardized communication. The layers are: Application Interface, Service Management, Device Management, Security, Platform Abstraction and Devices layer. In the Application Interface layer, the objects in the network are connected with the help of web services. This layer allows the objects to directly participate in the business processes without knowing the underlying hardware details. The architecture features a messaging system and a service catalogue. The messaging system helps the objects get notifications of events from different objects. It uses a system cache, so that the objects will not miss any notifications of events from different objects which it received at the same point. The service catalogue helps the objects to know about the services that are available. The Service Management layer of the architecture provides the services directly or it wraps the object's functionality into a representation of service. The Device Management layer helps with dynamic discovery of the objects and their status. The Security layer tries to implement security through authentication so that only eligible objects can access the object's services. The Platform Abstraction layer provides the abstraction of the IT Infrastructure such as communication protocol, data type and definition etc. The following characteristics are satisfied by the SOA Middleware: interoperability, scalability, spontaneous interaction, zero infrastructure, multiplicity, device discovery and management, abstraction, security and privacy. It does not support the Context Detection characteristic.

TABLE I THE MIDDLEWARE SOLUTIONS AND THE CHARACTERISTICS THEY POSSESS

Characteristics	**Middleware Approach**		
	Triple Space Based Distributed Middleware	**UBIWARE**	**SOA Based Middleware**
Scalability	No	Yes	Yes
Spontaneous Interaction	Yes	Yes	Yes
Zero Infrastructure	Yes	Yes	Yes
Multiplicity	Yes	Yes	Yes
Interoperation	Yes	Yes	Yes
Context Detection	Yes	Yes	No
Device Discovery and Management	Yes	Yes	Yes
Security and Privacy	Yes	No	Yes
Abstraction	Yes	Yes	Yes

IV. FUTURE SCOPE

The Various Middleware Architecture that we have discussed still has some drawbacks associated with them and is not complete. The triple space-based distributed middleware lacks scalability, while UBIWARE faces security and privacy issues whereas SOA based Middleware does not support Context Detection and Processing.

There are still some open issues in the design of Middleware for IoT that needs to be resolved. Standardization is one such issue. A single standard will not exist for the middleware of IoT as large different types of applications are possible with IoT. UBIWARE tries to provide a standardized solution for semantic web application domain while SOA based middleware ties to provide a standardized solution in Business development.

Managing large volumes of Data and Storage Capacity is one of the open issues that need to resolve. Similarly Security and Privacy is also an important issue that needs to be addressed. The IoT applications and objects should have strong security and privacy policies implemented with less overhead on the performance of Middleware.

V. CONCLUSION

This paper is the survey of the existing research work done in designing semantic web based middleware systems for the IoT. The ideal characteristics of a middleware i.e. Scalability, Spontaneous Interaction, Zero Infrastructure, Multiplicity, Interoperation, Context Detection, Device Discovery and Management, Abstraction, Managing Large volumes of Data, Security and Privacy were discussed. We discussed three middleware solutions i.e. triple space-based semantic distributed middleware, UBIWARE and SOA based Middleware, where we understood the architecture and found out which characteristics they possess. Future Scope and Open Issues were also highlighted. This existing work will definitely help us in designing an ideal middleware solution in the future, for that we need further investigate the open issues and drawbacks that each solution possess and suggest possible solutions.

REFERENCES

[1] Kevin Ashton, *That 'Internet of Things' Thing.* Available: http://www.rfidjournal.com/articles/view?4986

[2] Luigi Atzori, Antonio Iera and Giacomo Morabito, "*The Internet of Things: A survey*," in Elsevier, Computer Networks, May 2010.

[3] T. Lu and W. Neng, "*Future internet: The internet of things*," in 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 5, August 2010.

[4] P. Guillemin and P. Friess, "*Internet of things strategic research roadmap*," in The Cluster of European Research Projects, Tech. Rep., September 2009.

[5] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti and Subhajit Dutta, "*Role of Middleware for Internet of Things: A Study*", in International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.3, August 2011

[6] Zhexuan Song, Alvaro A. Cardenas, Ryusuke Masuoka, "*Semantic Middleware for the Internet of Things*", in The Internet of Things - IOT , 2010

[7] Moumena A. Chaqfeh and Nader Mohamed, "*Challenges in Middleware Solutions for the Internet of Things*", in Proc. of The 2012 International Conference on Collaboration Technologies and Systems (CTS 2012), Denver, Colorado, May 21- 25, 2012.

[8] Koosha Paridel, Engineer Bainomugisha, Yves Vanrompay, Yolande Berbers and Wolfgang De Meuter, "*Middleware for the Internet of Things, Design Goals and Challenges*", in Proceedings of the Third International DisCoTec Workshop on Context-Aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS 2010)

[9] A. Gómez-Goiri, D. López-de-Ipiña, "*A Triple Space-Based Semantic Distributed Middleware for Internet of Things*," In LNCS Vol. 6385, pp. 447-458. Springer, July 2010.

[10] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V.Y. Terziyan, "*Smart Semantic Middleware for the Internet of Things*", in Proc. ICINCO-ICSO, 2008, pp.169-178.

[11] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, V. 1736 Trifa, "*SOA-based integration of the internet of things in enterprise services*," in: Proceedings of IEEE ICWS 2009, Los Angeles, Ca, USA, July 2009.

