I J C T A, 9(9), 2016, pp. 4163-4173

SSASD: Semantic and Syntax Aware Service Discovery for Heterogeneous Application in Pervasive Computing

Subashchandar A.[1] and Jayakumar C.[2]

ABSTRACT

Pervasive computing is an emerging technology growing speedily and integrating IoT with Cloud. Home automation, Medical industry, research institutes like ISRO, ADA, HAL, ISAC and NAL are connected into cloud through pervasive environment in order to provide and fetch various kinds of service to perform their necessary actions to the industry. Normally, service discovery in pervasive computing is a challenging task, but in heterogeneous network discovering and providing a service is too difficult said in various existing research works which are discussed in literature survey. This problem is taken in to account and this paper motivated to provide a better solution for service discovery and service provision. In order to provide an efficient service discovery a Semantic and Syntax Aware Service Discovery (SSASD) method is proposed. The service request format, meaning and other log information is compared with the service information to fetch the accurate service. SSASD is simulated in DOTNET framework to verify the results and the performance is evaluated by comparing the obtained results with the the results of the existing approaches.

Keyword: Pervasive Computing, Service Discovery, Heterogeneous Networks, Cloud Computing, Internet of Things.

1. BACKGROUND STUDY

Service discovery is an important job in pervasive computing environments. An effective service discovery models can make it easy, fast and enables ubiquitous and heterogeneous computing devices where the services can be fetched and used effectively. All the service installed in the pervasive environments is treated as building blocks can accomplish complex services. Here it is described that hos services are integrated, analyzed and categorized to improve the efficiency of service discovery. In pervasive computing different kinds of devices surround people. All these devices can communicate each other and communicate with people also [2]. Nowadays Laptops, personal computers, mobile phones and personal data assistance are the most important devices are called as personal devices connected in pervasive environment. Microscope, Macroscopic, ECG, EEG, EMG monitoring, X-Ray, CT, MRI, Ultrasound and other medical instruments are interconnected in pervasive computing. Satellite, temperature monitoring sensor devices, antennas are agricultural devices interconnected to pervasive computing. Also additional networked computers from tiny sensors to large dynamic potent devices can provide various kinds of services interconnected in pervasive computing. Due to this large and different kinds of devices are interconnected in pervasive environment it becomes overwhelming to maintain, configure, and interconnect and dynamically finding these devices are too difficult in pervasive computing. Devices in the pervasive environment are enabled by the service discovery protocols to use by the users. Also these protocols facilitate the devices to be used by the users whenever they require under zero administration overhead.

[1] Research Scholar, Faculty of Computer Science and Engineering, Sathyabama University, Chennai, Tamil Nadu, India, *Email: subashchandar@gmail.com*

[2] Professor, Department of Computer Science and Engineering, Sri Venkateshwara College of Engineering, Chennai, Tamil Nadu, India, *Email: cjayakumar2007@gmail.com*

In case of large number of devices and large number of requests service discovery is more critical to the success of pervasive computing [3].

In this paper it is considered a basic problem of service discovery. The famous Alice and Bob are taken as people interconnected into pervasive computing. Bob visits Sathyabama University (SU) and turn on his Laptop and it is treated as a Laptop client. Bob browse the web and find out an online SU map. The map picture is small while view in the laptop. So that Bob look for a printer in a nearby computer lab in order to print SU map. The laptop locates the printer installed room. More number of printers installed in the computer lab provides printing services. All the printers cannot be connected since they are privileged. Laptop allows only few public printers can be provides and Bob access the printer as a guest. Bob accessed only the closest print to him. But the corresponding driver is not available in the laptop, so that the printer should be connected to the laptop without Bob's involvement. Finally the SU map is printed successfully. From the above discussion, it is clear that, Bob explicitly utilize the printing service and implicitly utilize the wireless connection service available on the campus and the location information service is helps him to find a printer. All the service discovery protocols are designed to reduce the administrative overhead and increase the usability. Layer based interaction can also increases the protocol efficiency. From the above discussion it is clear that any user in pervasive environment can connect to the services explicitly or implicitly. But the problem is:

- Make sure that the service availability
- Service Discovery
- Service Provision

The required service should be available in the pervasive environment, the available service should be discovered to the requests and the appropriate services should be provided to the requests. The contribution of this paper is:

- Service Declaration and Definition
- Service Selection
- Service Interaction
- Security

Till now various research works proposed various service discovery languages like Web Ontology Language for Services (OWL-S) [4], Web Ontology Language (OWL) [5], Business Process Execution Language (BPEL) [6], Syntactic discovery is based on interface matching techniques like UDDI, ebXML, WSDL, IDL, RMI interfaces [7, 8, 9],and so on. Services are described using various description schemas like Universal Plug and Play (UPnP) [10, 11], Bonjour [12-14], Quality of Service [15] and DEAP-space [16-18]. Some of the research works considered contextual information to do service discovery. The Jini service discovery concentrates on analyzing the attributes of the services, so that the users can fetch the services using attribute names. Some other earlier works proposed location-aware services for finding nearby hotels, petrol bunks and hospitals etc. But this research work concentrates service discovery where heterogeneous devices are interconnected pervasive environment. Medical, Agriculture and education based devices are considered as the heterogeneous devices in pervasive environment. Even though various research works are discussed here one of the latest earlier researches is taken to compare the proposed results to evaluate the performance.

2. EXISTING APPROACH

Evan Magill et al. (2016) addressed a rule conflict within sensor networks. It is well known that the behavior of the sensor nodes is changed when they are in situ. This is too important in certain applications like healthcare, for example in Ambulance Assessment where the entire network is tuned to be individuals. This can be obtained by applying rules, where each rule is processed from the distributed network. These rules dictate the behavior of the sensor nodes according to the sensor data values and control the entire network behavior. Since pervasive environment has different number and different kinds of heterogeneous devices it is difficult to control all the devices by a single rule. Even though adding separate rules for all the devices it brings conflict where more devices and more service are available under same category. So that, this paper concentrates on service discovery for heterogeneous device based pervasive computing in terms of scalability.

2.1. Proposed Approach: SSASD

More efficient way of discovering a service is accurate service, fast, nearby and cost less. In this paper the proposed SSASD challenging the issues occur while service discovery, resolve all the problems and provide the services. In order to provide an efficient service discovery a Semantic and Syntax Aware Service Discovery (SSASD) method is proposed. The service request format, meaning and other log information is compared with the service information to fetch the accurate service. In this paper it is assumed that there are two log files are maintained at the pervasive environment. One is for incoming request and the other is for persisted services in the pervasive environment. SSASD follows the declaration and description, service selection and service assignment.

3. SERVICE DECLARATION AND DESCRIPTION

All the services enter into PE are organized easily by declare it. A variable is assigned to declare the services. Then the semantics of the services are described for accessing the services. Finally the syntax of the services is assigned to access the services accurately. Declaring and describing a service is the foundation of service discovery. Declaration has only the identity variable (service name) and the description of the service has the building blocks of information about a service. The entire information about a service is available in the description. Also it includes organization, semantics and content of the description, query form and query processing. The semantic of the service description presents the information in a description. Whenever a query comes the query language and the service matching engine describe the service from the user’s point of view. Using this description the service engine assesses the matched services to the request. The description also has two portions as describing the references and the expressions with operations (functions).

3.1. Service Selection

After successful declaration and description of the services it has been assembled, processed and distributed if there are multiple service choices are exists. The selection component is responsible for deciding which service can be used by the client. The main idea used to select the service is using a simple decision making. Decision making is applied strictly finding the comparison results of the semantic, syntax and log information of the requests and the services in the PE.

3.2. Service Interaction

After successful service selection, a requested user is assigned by the service and the user can interact with the services. Interaction of the service dictates how it can be accomplished. The method of interaction decides how much information is available about a service, familiarity with the service and interaction requirements. The client-service interaction adds another dimension of information for the description and selection components. Assigning and accessing the service discovery needs flexibility in service interaction and it allows the clients can employ various techniques during service access. In this paper, it is considered that heterogeneous devices are connected. So that it is considered that:

Let PE is the pervasive environment having N number of Services as:

$$PE = \{\{NETs\}, \{Ss\}, \{Ds\}, \{Sers\}, \{DCs\}, \dots .\}$$

NET - Denotes the set of networks

Ss - Denotes the set of Services

Ds - Denotes the set of Devices

Sers - Denotes the set of Servers

DCs - Denotes the set of Data Centers

$$S = \{S_1, S_2, \dots, S_i, \dots ., S_N\}, \ \forall\, i = 1 \text{ to } N$$

If a service Si is newly enter in to PE then the value of N becomes N = N + 1 and Si is added into S, means the service Si is deployed in the PE. Each service Si is declared for indexing. The set S contains different types of services like devices, software and middleware. According to the service category it is easy identify the services. In this paper the service discovery is divided into different types as Service declaration and description, locating the services, service selection and service interaction. The declaration and description of services are depicted in Figure-1.

A new request enters into PE the request analyzer analyzes the request and obtains all the information regarding the query like service ID, service name and other meta information and compared with the service ID table. If all the information matches between the request and the services then that particular service is enabled to the request.

It is assumed that the services are heterogeneous; in order to improve the efficiency all the heterogeneous devices are categorized according to the area like medical, agriculture and education by adding a mnemonics as prefix to the service ID. The mnemonics are “MED”, “AGR” and “EDU” used to represent medical, agriculture and education respectively. It is assumed that in this paper the request query should indicate the area of the service requested. A request is parsed and analyzed, where analyzing the request the prefix of the parsed service ID is examined first. For example, if it is “MED” then the service analyzer filter the service ID having only “MED”. It makes easy and fast comparison between the request and the services. Next the service ID and the service name is looked in to the service declaration, if it available then it looks

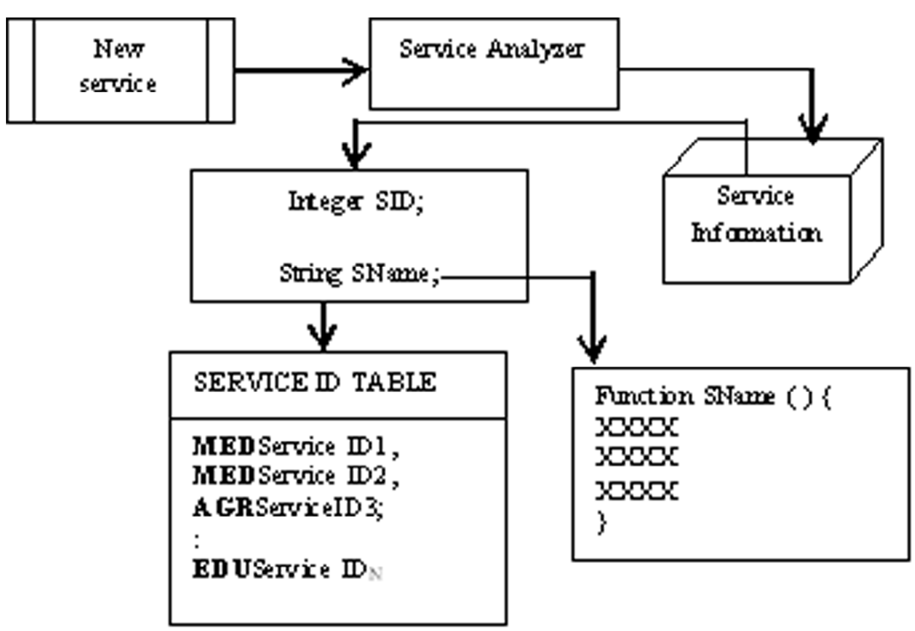

Figure 1: New Service Entering to Pervasive Environment

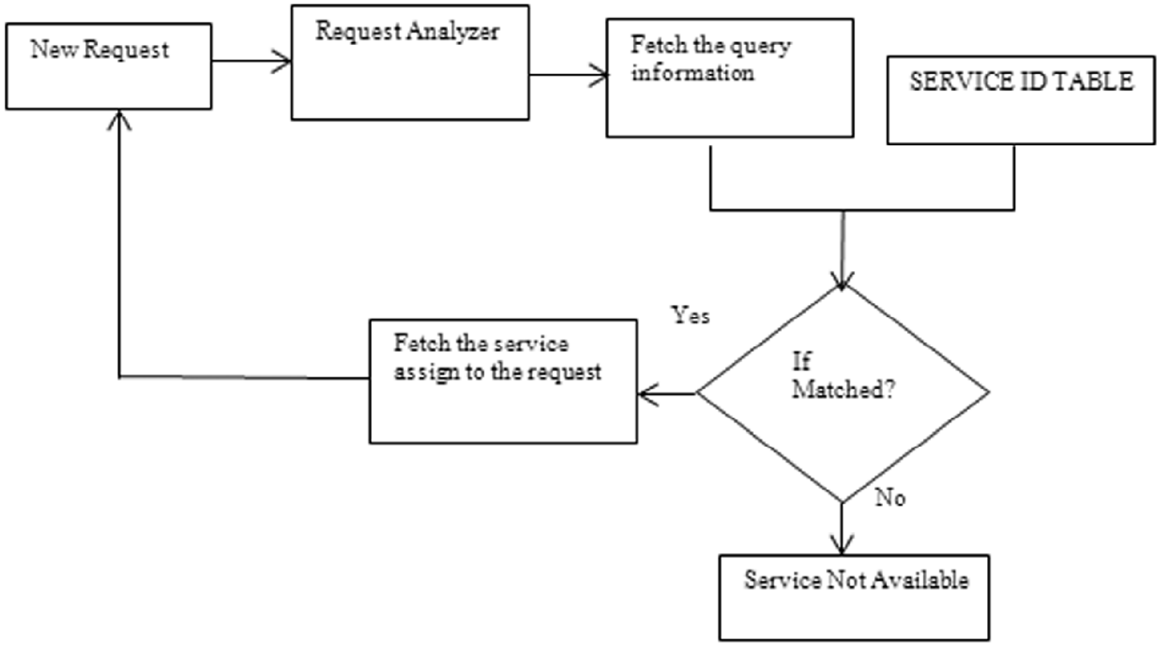

Figure 2: New Request Entering to Pervasive Environment

for the corresponding description of the services. It eliminates the service comparison in terms of searching with prefix, and declaration verification. If the required service is available in the declaration then it is very easy and fast to fetch the description of the service, it calls the implementation and assign to the requests effectively.

Algorithm SSASD (String Request, P1, P2, …, Pn)

{

// P1, P2,…Pn are the input parameters passed

// **initialize:** total cost as tot-Cost, cost per unit of time as initial-cput

1. input service ID table, Declaration table, Description table and implementation location
2. tic
3. Request parsed and Var ={sid, sname, sloc, sser, sdis, sem1, sem2};
4. pre = leftstring(3,“sid”);
5. if (pre == “MED”) then
6. for i=1 to m
7. retrieve all MEDServiceID(i)
8. end i
9. else if (pre == “AGR”) then
10. for i=1 to m
11. retrieve all AGRServiceID(i)
12. end i
13. else if (pre == “EDU”) then
14. for i=1 to m

15. retrieve all EDUServiceID(i)
16. end i
17. If (Var(request) == Var(service)) then
18. request← service
19. end if
20. toc // display the time to compute the efficiency
21. tot-Cost = initial-cput * toc

}

The above algorithm simply describes the SSASD functionality and it can be programmed and the performance is evaluated. The following section discuss about the experimental results and discussion.

EXPERIMENT AND RESULTS

In this paper SSASD algorithm is implemented and experimented by simulating in Network Simulator – 2.34, software. The area of the PE networks is 1500 × 1500. Systems are treated as nodes in the network where 200 nodes are deployed in the network. Some of the nodes are treated as servers where all the normal nodes are connected to servers. Also 25 nodes are treated as devices and some software services are interconnected in the PE where each node can communicate with one another. Here 20 nodes are privileged and 5 are can be accessed using guest role. Part from this all the systems are assumed as clients and can send request to the server for the services (devices). The proposed schemes have been experimented in the simulation environment in ns2. The simulation parameters are shown in Table 1.

The main contribution of the paper is to declare and describe the services. All the permanent services are declared and described but some of the temporary services will be connected in a particular of each day or in specific days within a week. It is not necessary to declare and describe the temporary services in the PE since they are going to be used by certain known people or known company for their own purpose. Since number of services declared and described against the number of services deployed in the PE are

Table 1
Simulation Parameters

Parameter	*Level*
Area	1000m × 1000m
Speed	1 to 15 m/s
Radio Propagation Model	Two-ray ground reflection
Radio Range	250 m
Number of Nodes	20 to 1000
MAC	802.11
Application	CBR, 100 to 500
Packet size	50
Simulation Time	100 s
Placement	Random
Malicious Population	Upto 5%
Sybil Ids per malicious node	2
Pause Time	2ms

calculated. Here the services are heterogeneous and it is calculated how many medical, agriculture and education services are available. Also it is calculated which kind of services are frequently used in order to increase the number of services to satisfy the customer requirement.

All the services are accessed only by comparing the syntax and semantics of the services. If the service is not matched by the service name or by service ID, it is compared by the semantic parameters given in **Var** set in Algorithm_SSASD. These parameters sem1 and sem2 represents semantic 1 and semantic 2 of the corresponding services. The number of services accessed from the number of services declared is shown in Figure-3. The services may be declared and described but cannot be accessed due to many reasons like not matching semantically or syntactically. The number of services declared and described out of the total

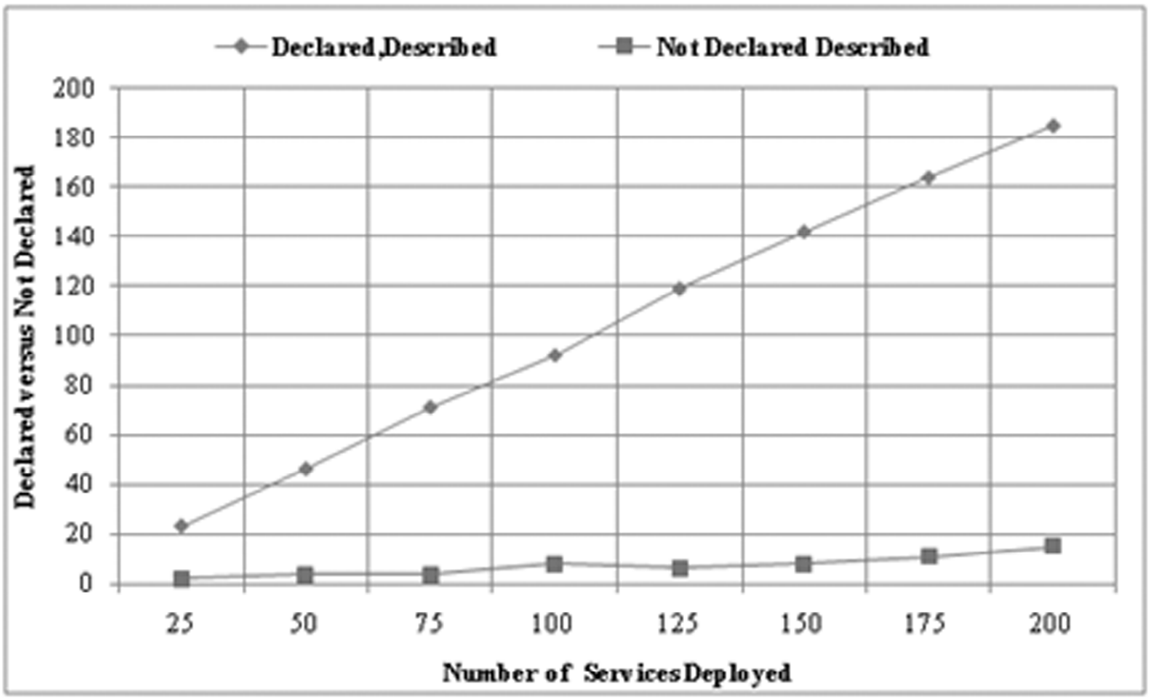

Figure 3: Number of Services Permanent versus Not Permanent

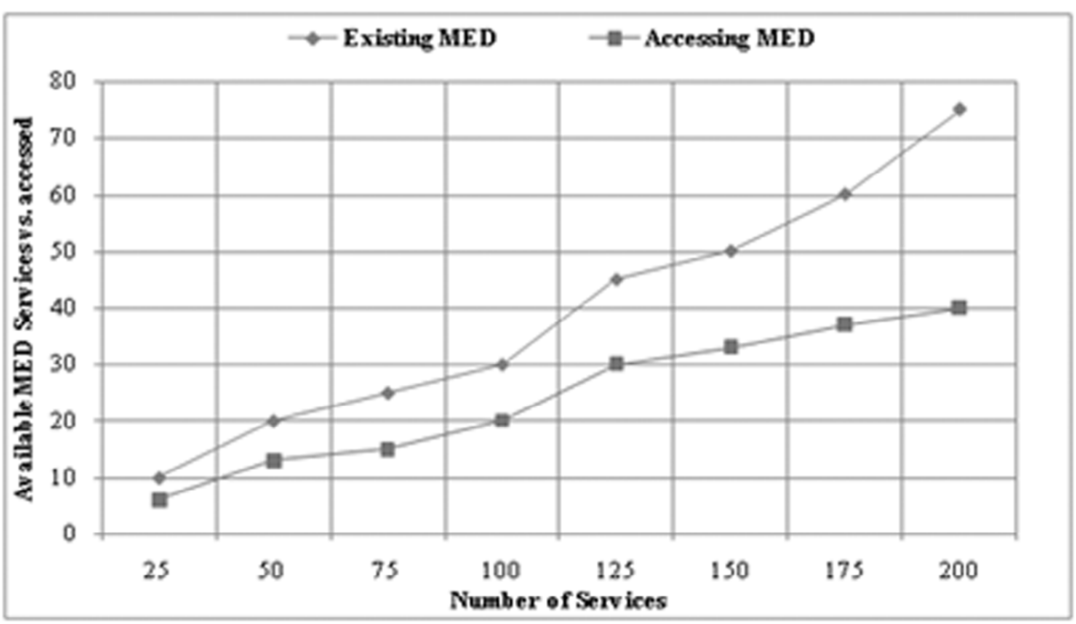

Figure 4: Available MED Service vs. Accessing MED Services

number of services in the PE is gradually increased from 25 to 200 increased by 25. The number of declared and described services are lesser than the total number of services because some of the services are temporary and they are used in specific time. The number of declared versus described services is proportionally increased according to the number of services deployed in increased manner.

The services are mainly considered as hardware devices belonging to various areas such as medical, agriculture and education. The number of services deployed is depending on the number of requests coming to the PE every day. According to the market survey, the requirement for medical services is more than the agriculture and education whereas the number of agriculture is more than the number of education services. In order to analyze the type of services requested by the market, here it is calculated the number of MED, AGR and EDU services deployed and requested by the user. The MED service requests, AGR requests and

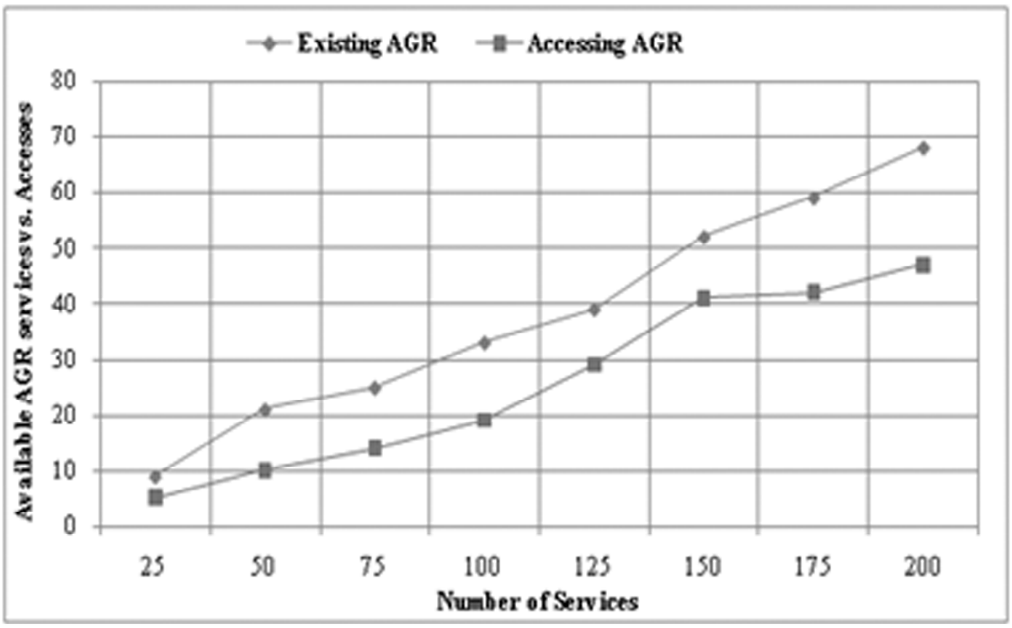

Figure 5: Available AGR Service vs. Accessing AGR Services

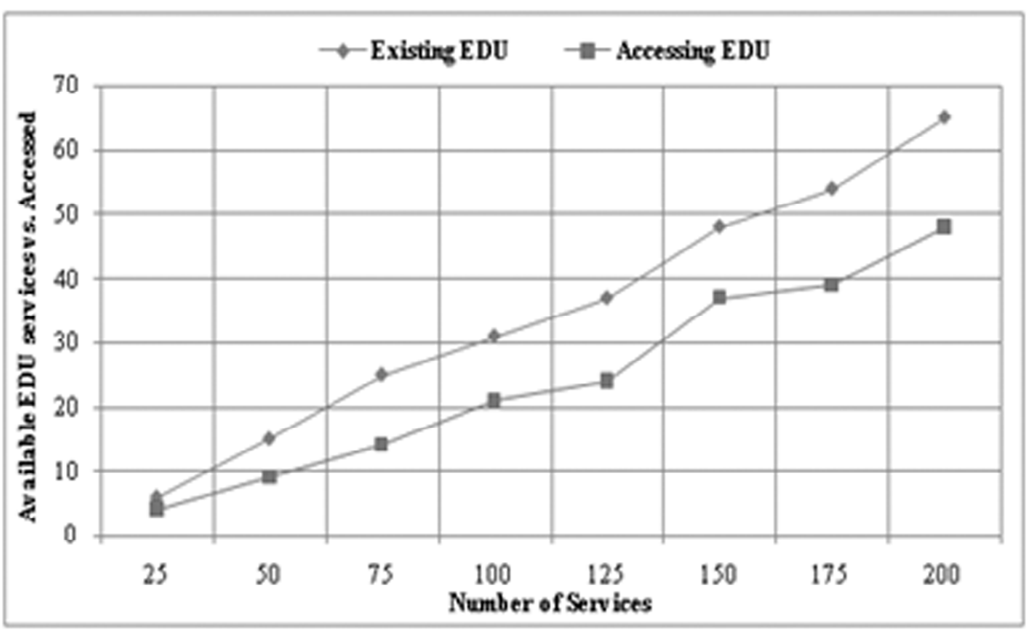

Figure 6: Available EDU Service vs. Accessing EDU Services

EDU service requests against number services are shown in Figure-4, Figure-5 and in Figure-6 respectively. According to the number of service availability the number of service accessing is gradually increased proportionally. From the three figures Figure-4, Figure-5 and Figure-6, it is clear that the MED services are used frequently and high than the agriculture and education.

The request for education and agriculture services are more or less equal in the PE. Similarly the time taken to process the service discovery and service provision is calculated and shown in Figure-7. The time taken by SSASD is very less and it is increased proportionally according to the number of requests and number of services. If more number of services exists then the time taken to discover is also increased due to the number comparison between the request and the number services. The efficiency of the proposed approach is calculated by thediscrete timer installed in-built in NS2 software. The existing approach [17] calculated time in terms of number of rules applied for service discovery. The time taken by the existing approach too high than the proposed approach and it is shown in figure-7.

Finally the number of services discovered from the available services are calculated and depicted in Figure-8. The number of service discovery is depending on the user requests. User can request a service by means of syntax or by means of semantic. Only few number of requests request a service by semantic and syntax. The number of services discovered is up and down in Figure-8, due to the request matching. Since the number of discovered services are purely depends on the request not on the number of requests availability. Figure-8 shows the numbers of requests are fulfilled with the service discovered and provided to them. It combines all the medical, agriculture and education areas. Once the service is discovered it means that the service is provided to the request.

Finally the cost is calculated for service discovery in PE. Most of the services available in PE are accessed by making a payment in periodical basis. It is well known that recent computing technologies under cloud are used by the “Pay-n-Use” scheme. The cost of the service discovery is calculated using the formula as:

$$Total - Cost = \text{cput} * \text{time}$$

It shows that the cost depends on the time taken for the service discovery.

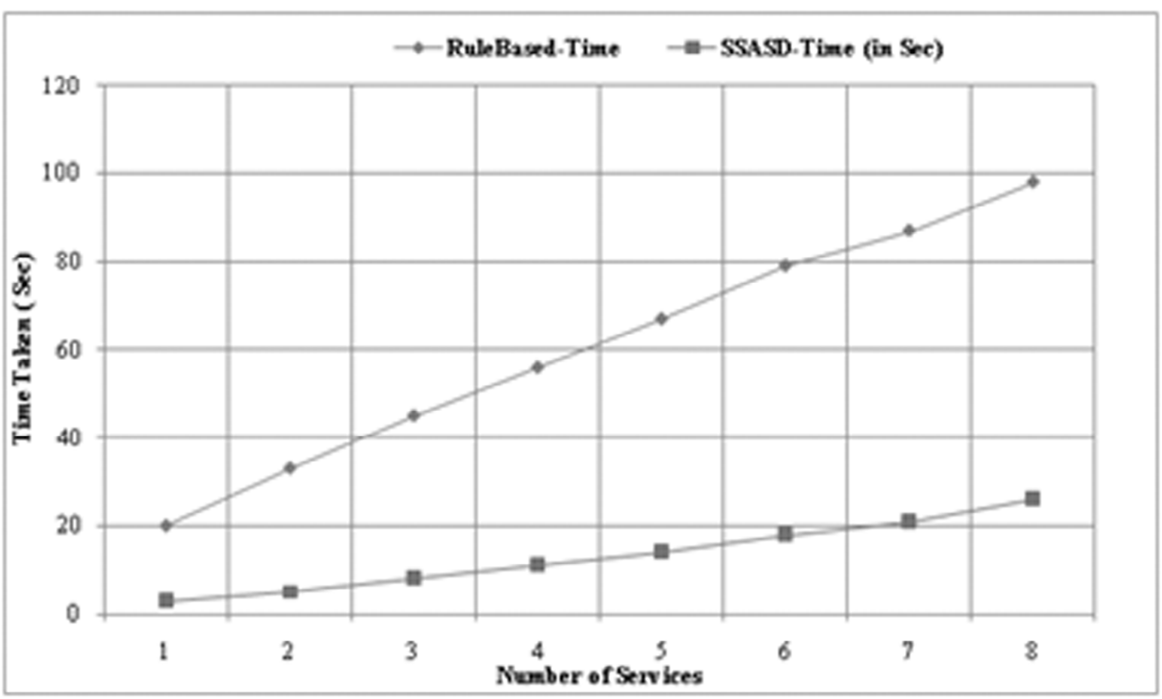

Figure 7: Time Taken for Service Discovery and Provision

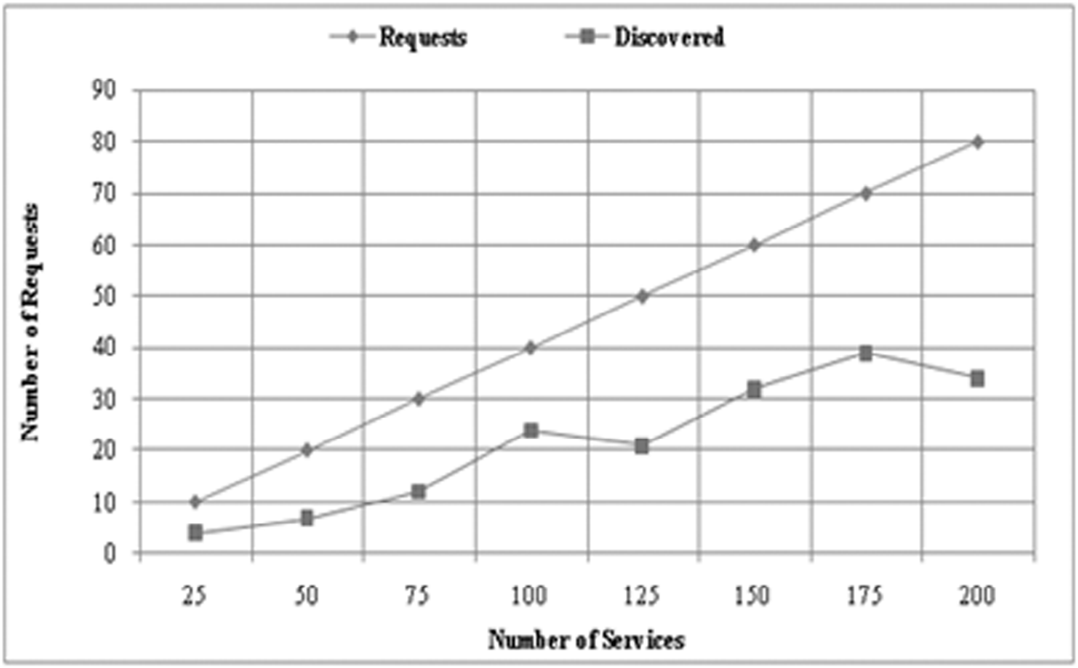

Figure 8: Number of Services Discovered

CONCLUSION

This paper aimed to discover an accurate service for the incoming request. Initially, the proposed approach SSASD verifies the declarative part, second the prefix of the service is verified, third the service availability is verified, and finally the service description is verified. These verifications involve reducing the time for service discovery and confirming the service availability in each stage. Means if the service is not in declaration no need to proceed for the same service, we can go for the other services in the PE. The main advantage of the proposed approach is reducing the time complexity and cost effective in terms of number of comparisons. Also from the experimental results it is clear and proved that the proposed SSASD is better than the existing approach.

REFERENCES

[1] E.H. Magill and J. Blum. Exploring conflicts in rule-based Sensor Networks, Pervasive and Mobile Computing, 27 (2016), 133–154.

[2] M. Weiser, The Computer for the 21st Century,Scientific American, 265/3 (1991),66-75.

[3] T. Kindberg and A. Fox, System Software for Ubiquitous Computing,IEEE Pervasive Computing, (2002), 70-81.

[4] W3C,Web Ontology Language for Services (OWL-S) 1.1, http://www.daml.org/services/owl-s/1.1/, 2005.

[5] W3C, Web Ontology Language (OWL),http://www.w3.org/TR/owlfeatures/, February, 2004.

[6] T. Andrews, Business Process Execution Language Version 1.1. the BPEL4WS Specification, Available: http : // download. boulder .ibm. com / ibmdl /pub /software /dw/specs/wsbpel/ ws-bpel.pdf.

[7] S. Ben Mokhtar, Semantic Middleware for Service-Oriented Pervasive Computing. Tesis Doctoral, 2007.

[8] J. C. Corrales, D. Grigori, M. Bouzeghoub, and J.E. Burbano, Bematch: A platform for matchmaking service behavior models, In the 11th International Conference on Extending database technology: Advances in database technology (EDBT'08), (2008),695-699.

[9] W. Kokash, Leveraging web services discovery with customizable hybrid matching, presented at the In Proc. of ICSOC, (2006),522-528.

[10] UPnP Device Architecture 1.0. UPnP Forum. [Online]. Available: http://www.upnp.org /resources/documents.asp, 2003.

[11] M. Jeronimo and J. Weast, "UPnP Design by Example: A Software Developer's Guideto Universal Plug and Play", Intel Press, 2003.

[12] Apple Developer Connection, (2006), Bonjour Homepage. Apple Computer.[Online].Available: http://developer.apple.com/networking/bonjour/.

[13] S. Cheshire and M. Krockmal, Multicast DNS. IETF Internet Draf, [Online]. Available: http://files.multicastdns.org/draft-cheshire-dnsext-multicastdns, 2005.

[14] S. Cheshire and M. Krochmal, DNS-Based Service Discovery.IETF InternetDraft.[Online]. Available: http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt, 2005.

[15] J. Liu, Q. Zhang, B. Li, W. Zhu, and J. Zhang, A Unified Framework for ResourceDiscovery and QoS-Aware Provider Selection in Ad Hoc Networks, ACM Mobile Computingand Communications Review, 6/1 (2002), 13–21.

[16] M. Nidd, Service Discovery in DEAPspace, IEEE Personal Communications, 8/4 (2001), 39–45.

[17] M. Nidd, Timeliness of Service Discovery in DEAPspace, In Proc. IEEE Int'l Workshopson Parallel Processing, (2000), 73–80.

[18] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade,DEAPspace: Transient Ad-Hoc Networking of Pervasive Devices, In First AnnualWorkshop on Mobile and Ad-Hoc Networking and Computing, (2000), 133–134.