Volume 3, Issue 12, December 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Semantic Based Efficient Service Discovery and Recovery in Bluetooth

M.Sughasiny,
Research Scholar,
Research & Development Centre,
Bharathiar University, Coimbatore 641046,
Tamil Nadu, India

Dr.R.Dhanapal,
Professor & Director,
Research & Development,
Procademia, Quadruple Education Network
Chennai 600 024, Tamil Nadu, India

Abstract— Service discovery is one of the most important processes that have to be performed for efficient functioning of a system. Efficient and appropriate service discovery is the current necessity. While legacy mechanisms work by providing the appropriate UUID to the system for obtaining the corresponding process. The current system is moving towards semantic based processing; hence a semantic based process is the current need in service discovery. The current proposal presents a Semantic Based Service Discovery (SBSD) and a recovery mechanism that provides a flexible system for the user to present their queries in plain lexical format. The nodes present in the current network are initially clustered, and the cluster head is selected. All the services hosted by the nodes are then registered. When the user requests for a service, the requirements and specifications are lexically matched to provide the appropriate services. On event of failure, the nodes and the cluster heads are intimated of the failure, and alternate services are provided to the nodes.

Keywords— Service discovery, Bluetooth, Semantic based matchmaking, Clustering.

I. INTRODUCTION

Bluetooth [1] is a short-range wireless technology that allows devices to exchange data and voice in real-time. The Bluetooth Special Interest Group (SIG)[2] is responsible for developing specifications of this technology. Many mechanisms were proposed for efficient service discovery and device discovery. Key challenges in wireless mobile ad hoc networks are computational resource constraints, power limitations, and efficient service discovery techniques[3]. The short range radio network technology Bluetooth suffers from long service discovery delays and high power consumption due to necessary connection establishment between discovering and discovered entity.

Bluetooth Service Discovery Protocol (SDP)[2] enables a client application on a device to discover information about services on other Bluetooth devices. Every service is represented by a profile, identified by a 128-bit Universally Unique Identifier (UUID). A match occurs on a peer device if and only if at least one UUID specified by the client is contained in one or more of its service records. Efficient service discovery helps in efficient and longer usage of the network.

The continuous and rapid evolution of service-oriented and mobile technologies drives the development of new methods and techniques to improve serving of nomadic user requests. Architectural and functional requirements for designing multi-channel adaptive information systems, where services can be accessed through mobile terminals, in an ubiquitous and itinerant way, from different kinds of devices (e.g., laptops, palmtops, cellphones, smart cards) through different channels (internet, wireless networks, etc.) have been widely investigated in [18]. In particular, service discovery is considered one of the major crucial issues. Specifically, totally or partially automated techniques and tools are required to effectively locate services that fulfill a given user request.

Ability of understanding service requests and advertisements is strictly necessary to adapt and enhance service provisioning. To this purpose, the use of semantic descriptions of services has been widely motivated and recommended for automated service discovery under highly dynamic and context-dependent requirements in distributed environments. Semantic-enriched frameworks are considered a key issue to enforce timely discovery and dynamic composition of services. The ontology description languages OWL, OWL-S and, more recently, WSML have been proposed and several approaches based on these languages are being developed.

Automated techniques and tools are required to effectively locate services that fulfill a given user request in a mobility context. To this purpose, the use of semantic descriptions of services has been widely motivated and recommended for automated service discovery under highly dynamic and context-dependent requirements.

Locating a network service or a device on demand is a challenging task for enabling mobile and pervasive computing. There exists a variety of network protocols and architectures that enable an application to discover a network service with little manual configuration. Most existing work supports an attribute-based discovery as well as a simple name lookup to locate a service. Usually there are only a set of primitive attribute types, such as string and integer, to characterize a service. Thus, the service discovery process is primarily done by type matching, string comparison, or integer comparison. When there are complex attribute types, such as in Jini (Edwards, 2000), the service discovery is done by exactly matching attribute values in a query with those in a registration. For representing real world objects such as network services, it is necessary to have more complex data structures to capture richer semantics. Moreover, a user may not be able to specify the exact values of interested attributes. Thus, approximate matching are desirable. Semantic

discovery mechanisms have been invented to overcome UDDI's syntactic discovery solution by providing more precise results. Automated discovery of services with desired functionality is an active research area because it is crucial for realizing the envisioned advantages for functional reuse and automated composition. A service discovery request typically consists of a functional requirement, a set of additional requirements, input and output (IO) requirements, Quality of Service (QoS) and other non-functional requirements. For a given web service discovery request, matchmaking is the process of discovering a set of web services from one or more repositories such that all the matched services provide the required functionality and satisfy the constraints imposed by the request. Earlier approaches to address the matchmaking problem adopted a keyword-based search mechanism [1].
The remainder of this paper is structured as follows; section 2 provides the related works under the current study, section 3 provides overall system architecture, section 4 provides the complete working mechanism of the semantic based service discovery and recovery mechanism and the section 5 presents the conclusions for the current study.

II. RELATED WORKS

It proposes a novel distributed service discovery method for the pervasive computing environment. The method is based on the concept of small world, policy-based advertisement and semantic-based intelligent forwarding of service request. It utilize the policy-based proactive advertisement method to establish the service community of every node, which fully consider the node capability of computation and communication. For service beyond service community, each node maintains a few distant nodes called contacts to create a small world network for increasing the semantic coverage view.
Several OWL-S based matchmaking approaches have been proposed for service discovery [4, 5, 6, 7, 8, 9, and 10]. The approaches fall under two major categories: Input-Output (IO)-based and description-based. IO-based approaches [4,5,6]select matches by determining the relationship between the IO profile of a request and the IO profile of a service. While IO-based matching allows eliminating some services from consideration, in general, a large number of inappropriate services may remain because services with identical IO signatures yet different functionality cannot be differentiated.
Description-based approaches [7, 8, 10, and 9] adopt a more rigorous logical inference process to eliminate false positives by inferring subsumption relationships between a request and a service specification. However, services with functionality contradicting the request may be returned, and a matching service may be ranked poorly if its description does not completely satisfy a request [9, 11, 12, and 13].
In [19], a set-theoretic matchmaking approach is used to assess suitability of a service with respect to a request. In this scheme, a service profile (or a goal) is modelled as "intentions", that is, a universally (or existentially) quantified expression over a set of *relevant objects* (service capabilities), where the quantification determines whether a profile matches a goal. The main drawback of this approach is that explicit intention modelling is required; the absence of such information reduces the problem to subsumption reasoning and may result in counter-intuitive matches (see Section 3). There is no explicit consideration of gathering intention and its representation in [16]. [2_43] presents a two-stage process, where in the first stage, a set of conceptual service profiles that seem to match the request are shortlisted, and subsequently, in the second stage, the concrete information about these services is obtained by querying the services directly to determine whether the services do meet the requirements or not.
This process (shown in Figure 1) utilises both the general service descriptions and detailed information obtained from services. The conceptual description of a service holds information that represents terminological or general knowledge about it.

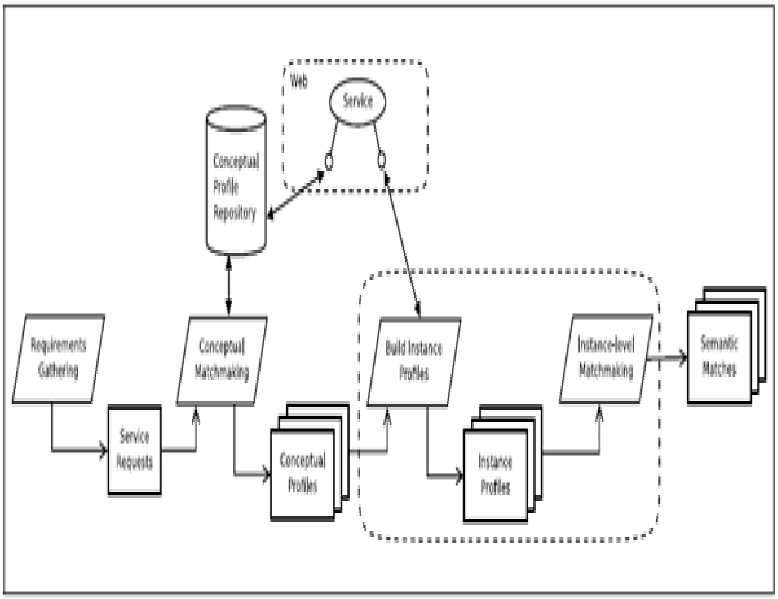

Figure 1: Consistency-based Matchmaking process

Given a set of service descriptions and a discovery request, this matchmaking approach identifies the set of matching services profiles in two stages: first, it determines the set of conceptual descriptions that are consistent with the request (*potential matches*), and second, it restricts the selected candidates to those that indeed offer concrete descriptions that are consistent with the request (*concrete matches*).A. et al. [3] proposed a new technique for the discovery of web services that account for the need of composing several services to satisfy a client query. The proposed algorithm makes use of OWL-S monologies, and returns the sequence of atomic process invocations that the client must perform in order to achieve the desired result.

Wei, D.P. et al. [4] proposed a new method to enhance the semantic description of service by using the semantic constraints of service I/O concepts in specific context. Ma, Q. et al. [5] presented to define QoS data into service descriptions and adopted the ontology reasoning to change previous syntactic matchmaking into a semantic way. Ma, J.G. et al. [6] used a novel approach to partition a large set of search results into a set of smaller groups by employing a clustering approach. Also, there are many other researches for semantic service discovery [7], [8], [9], [10], [11], [12]. [7_15] presents a heuristic algorithm based on iteration for semantic telecommunications services discovery, which can improve the efficiency of services discovery effectively. It is also found that the implied relevance during the matching process, is not concerned by others, while the current proposal considers this process for additional efficiency and accuracy. [3_78] presents a distributed registry and forecast algorithm for web services based on swarm system to answer a user request about finding web services.

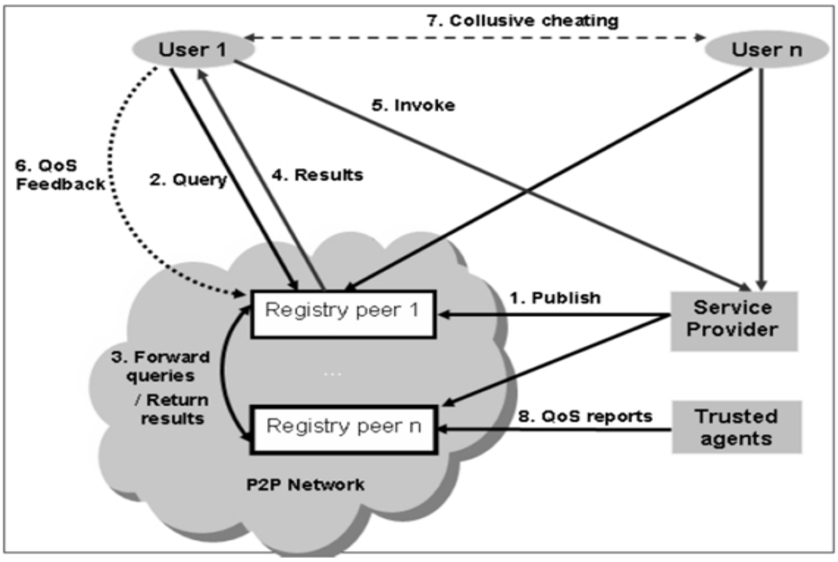

Figure 2: Framework model

The evaluation of QoS reports by the registry peers has to account for malicious reporting and collusive cheating of users to get a correct view of the QoS properties of a service. Additionally, it also allows trusted agents in the model to provide QoS monitoring for certain services in the system.

Different matchmaking approaches have been developed aiming at improving keyword-based techniques provided by the UDDI registry. In general, service matchmaking strategies that are based on purely logic deductive facilities [7, 14] present high precision and recall and are characterized by a good trade-off between expressiveness and computational complexity, but are often characterized by low flexibility. In [14] a service matchmaking strategy based on the OWL-S service profile and on a DL reasoned is proposed. The overall DAML+OIL expression representing a service profile is consistently mapped into a single DL expression and DL-based deductive facilities are applied to check if the description of request is equivalent, subsumed or consistent with the descriptions of service advertisements. In [7] the requested service profile and the provided one are expressed by means of DL expressions.

It considers different matchmaking models: (1) a deductive model to determine the kind of match; (2) a similarity-based approach, exploiting retrieval metrics to measure the degree of match between services and (3) a hybrid model combining the previous ones to mix deductive precision with similarity-based flexibility. In this model, first a description logic-based classification is performed to precisely establish the kind of match between the request and each advertised service, then services with partial match are ranked on the basis of their similarity degree.[14] The use of different matchmaking models aims at improving searching results and can be used in conjunction with optimization and ranking strategies. The application of the different models produces different results depending on the level of flexibility expected from the requester.

III. SYSTEM ARCHITECTURE

The current process deals with providing the user with a fast and efficient service discovery. When the network is initialized, all nodes are separate entities. These are then grouped together to form clusters. This grouping occurs on the basis of distance between the nodes. After this grouping process, a cluster head is selected as described in [7]. After this process, all the nodes that come under the same cluster head registers their services to their corresponding cluster head.

Registering the services correspond to specifying their UUID along with the semantic relations that correspond to the service When a process requests for a service, it describes the service semantically. This description is passed to the stemming engine for removing the unnecessary prefixes and suffixes. After this, the matching engine checks for the services that correspond to the provided description. These services are initially checked in the local cache, and then the global cache. If it is not available in both the caches, the global repository is queried and the services are obtained. The services that best corresponds to the current specification is provided to the user.

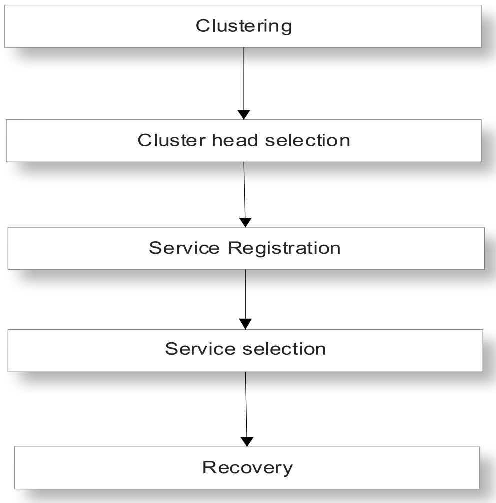

Figure3: System Architecture

When failure occurs in a node, the system using the service detects the failure and intimates it to the corresponding cluster head for performing the required actions and to initiate the recovery mechanism.

IV. SEMANTIC BASED EFFICIENT SERVICE DISCOVERY AND RECOVERY IN BLUETOOTH

The semantic based service discovery and recovery mechanism can be performed in four phases. The initial phase deals with clustering the available nodes and allocating the corresponding cluster heads, the next phase provides mechanisms for service registrations from each node. After this phase, service request processing can be carried out by the nodes. When a node fails, the recovery mechanisms that are to be carried out are discussed in the final phase. Domain ontology, representing the computer science domain is taken as the base database for processing. In general, a domain based ontology provides the users with a domain based knowledge, i.e. knowledge relating to a specific subject alone. Since we consider services related to computer science, our current ontology relates to the computer field. A generic framework has been proposed in the current study, hence any type of ontology can be used as a base for processing.

A. Clustering

Clustering is the process of grouping related entities, such that they can be treated like a single entity. The grouping is performed based on certain constraints provided by the user. Each node is considered as a single entity. In our process, clustering is performed by grouping the entities with respect to the node distances. Hence all nodes that have distances under a certain threshold are grouped together. After the grouping, each group selects a cluster head. The cluster head becomes the representing entity for that group. The formation of the cluster heads is carried out in four phases. They are

- **Floodmax -** Each node locally broadcasts its WINNER value to all of its 1-hop neighbors. After this process is carried out, the node chooses the largest value among its own WINNER value and the values received in the round as its new WINNER. This process continues for d rounds.
- **Floodmin -** This follows Floodmax and also lasts d rounds. It is the same as Floodmax except a node chooses the smallest rather than the largest value as its new WINNER.
- **Overtake -** As flooding occurs in the network, WINNER values are propagated to neighbouring nodes. At the end of each flooding round a node decides to maintain its current WINNER value or change to a value that was received in the previous flood round. Overtaking is the act of a new value, different from the node's own id, being selected based on the outcome of the information exchange.
- **Node Pairs -** A node pair is any node id that occurs at least once as a WINNER in both the 1st (Floodmax) and 2nd (Floodmin) d rounds of flooding for an individual node.

After the creation of clusters and the formation of cluster heads, the network becomes ready for transmission.

B. Service Registration

Each node in the network has the capacity to perform services, when requested. Service registration is the process of communicating the details about the services hosted by each node. The node initially collects all the services that it can perform and registers it in the local cache. These details include its UUID, and a detailed description about the service.
This description helps during the semantic matching process. These details are then transferred to the cluster head. The cluster head maintains the details presented to it, (UUID and the service description), along with the details about the

node hosting the service. This makes it easier for the cluster head to transfer the control to the corresponding node when a service request arises. The cluster head maintains all the service details contained in the current cluster and details about services hoisted in other clusters. This external cluster details might not be complete.

After a cluster head has received all the internal service information, it is considered to be fully configured. After the complete configuration, the nodes wait for service requests

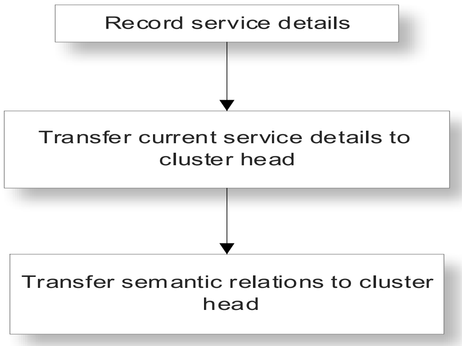

Figure 4: Service Registration

C. **Service request processing**

The current proposal accepts a service request in the form of keywords that best describes the service. The services that contain the closest correspondence to the keyword are listed to the user. The keywords provided by the user have the highest probability of containing ambiguities and unnecessary prefixes or suffixes. These should be removed from the input for performing the search efficiently. This is carried out by the stemming engine. The matching engine selects the corresponding synonyms depending on the current context and adds it to the search query. The query is initially executed in the current cache to determine the availability of the service. It is then executed in the cluster head, for performing the search in the global cache for determining a similar service launched by other nodes. If no service is found in the local or the global cache, then, an information broadcast is performed to obtain services from the repository. All services that correspond to the current keyword search are returned. These service details are registered in the global cache of the cluster head. The cluster head then ranks these services based on their usage and are presented to the node. The corresponding node selects the appropriate service required. This process increments the rank of the particular service with respect to the queried keyword. This helps the other users in obtaining better service.

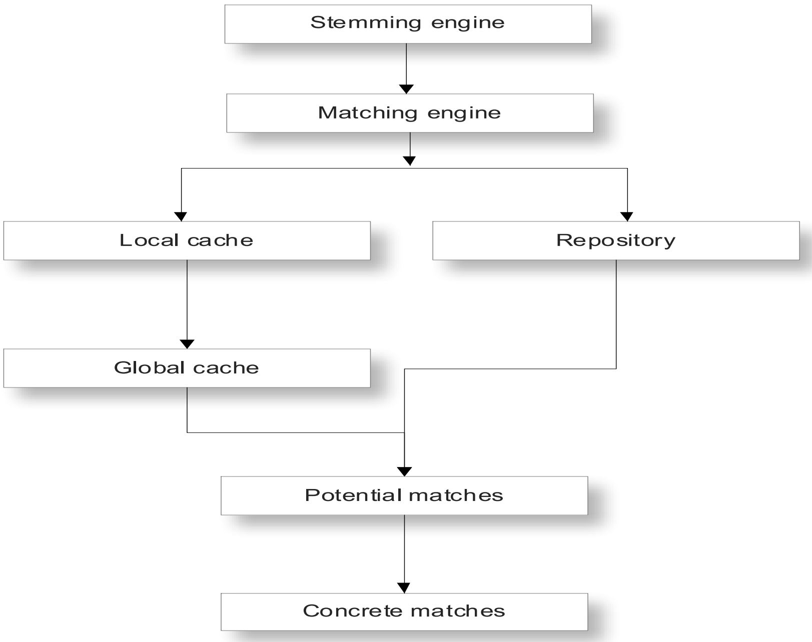

Figure 5: Service request processing

D. **Recovery**

A service failure occurs once a node hosting the particular service has failed. This mechanism is triggered once a service failure is detected by a node. The node initially detecting the failure intimates the cluster head of the failure, triggering it to remove the service details that correspond to the failed node. The cluster head then sends a broadcast message intimating all the nodes under its cluster about the failure. Hence every node using the service hosted by the failed node terminates it and retransmits its query to the cluster head for the next available service. The cluster head obtains these services from the global cache and passes it to the nodes. This information is definitely available in the global cache, since the searching process has already been performed by the cluster head.

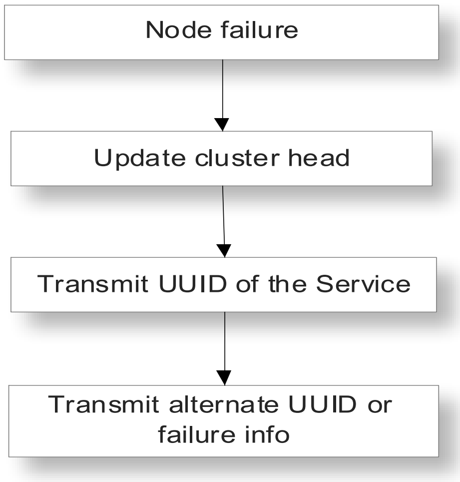

Figure6: Recovery Mechanism

V. RESULTS AND DISCUSSION

The SBSD method is compared with the COMPAT method and their average response time is calculated. The figure below shows that the response times of SBSD and the COMPAT are almost the same, while SBSD shows comparatively slight variations over COMPAT.

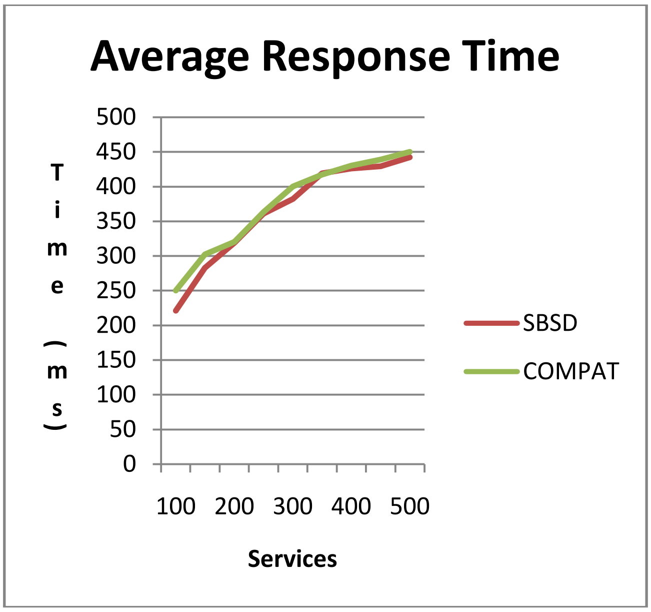

Figure 7: Average Response Time

The figure shows the comparison between potential matches generated vs the concrete matches that are shortlisted. Concrete matches are said to be the matches with highest potential for selection. From the graph we can see that around 90% of the potential matches are shortlisted as concrete matches.

TABLE 1: SBSD and COMPAT

Services	SBSD	COMPAT
100	221	250
150	283	302
200	319	320
250	361	363
300	382	400
350	419	417
400	426	430
450	429	439
500	442	450

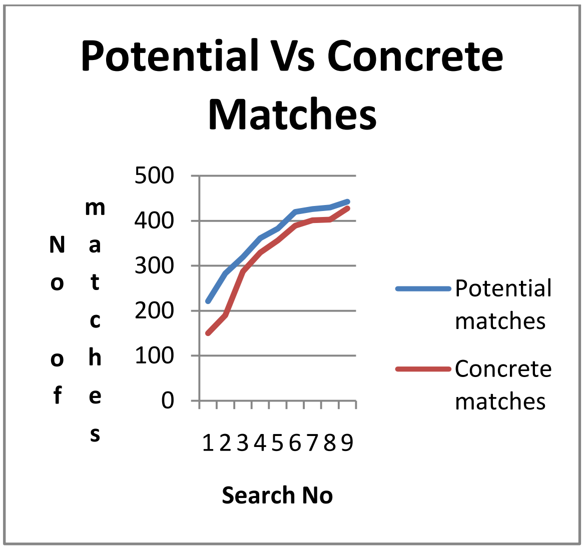

Figure 8: Comparison between potential and concrete matches

TABLE 2: Potential matches and Concrete matches

Potential matches	Concrete matches
221	150
283	190
319	287
361	329
382	356
419	389
426	401
429	402
442	427

The current process is evaluated with various sets of data containing different number of data items and the obtained values are recorded in a confusion matrix.

TABLE 3: Confusion Matrix

		Predicted	
		Positive	**Negative**
Actual	Positive	TP	FP
	Negative	TN	FN

Where,
TP - True positive, FP- False positive, TN – True Negative and FN – False Negative.
The two performance measures, sensitivity and specificity are used for evaluating the results.
Sensitivity is the accuracy on the positive instances.

$$Sensitivity = TP/TP + FN$$

where TP is True Positive Rate and FN is False Negative Rate.
Specificity is the accuracy of the negative instances

$$Specificity = TN/TN + FP$$

where TN is True Negative Rate and FP is False Positive Rate.

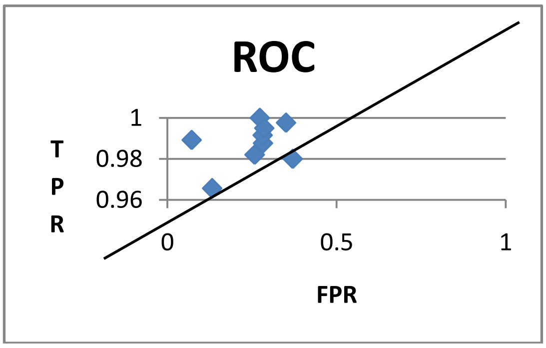

Figure 9: Comparison of TPR and FPR

From the ROC plot, we can see that the points are mostly concentrated towards the north-west corner of the graph. Hence, this implies that the accuracy rate of the current system is high.

TABLE 4: FPR and TPR

FPR	TPR
0.131579	0.965517
0.071429	0.989189
0.272727	1
0.282051	0.987578
0.28125	0.991429
0.258065	0.981959
0.37037	0.97995
0.285714	0.994924
0.35	0.99763

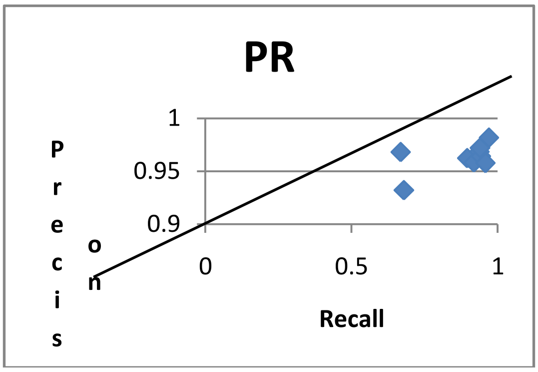

Figure 10: Precision vs. Recall

Precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance. Hence we can use this measure to find the relevance of the readings.

TABLE 5: Values of Precision, Recall, Accuracy

Precision	Recall	Accuracy
0.933333	0.679612	0.932127
0.963158	0.667883	0.968198
0.958188	0.895765	0.962382
0.966565	0.919075	0.958449
0.974719	0.937838	0.968586
0.979434	0.943069	0.9642
0.975062	0.958333	0.957746
0.975124	0.940048	0.972028
0.983645	0.970046	0.9819

Average accuracy of the current system is found to be 0.962846.

VI. CONCLUSION

The current system is found to perform automated service discovery in an efficient manner. The current approach is a semi-supervised approach, which requests the user to select the appropriate service from the list of available concrete services. The current proposal is designed as a learning system that improves over time. Every time a customer selects a service, the rank entry for the current service increases with respect to the searched keyword, hence the accuracy rate of the current system is directly proportional to the time of deployment.

REFERENCES

[1] R. Mettala. *Bluetooth Protocol Architecture*. Http://www.bluetooth.com/developer/whitepaper/, (1999).
[2] Sasikanth Avancha, Anupam Joshi, Timothy Finin, *"Enhancing the Bluetooth Service Discovery Protocol".*

[3] Igor Sedov,Stephan Preu, Clemens Cap,Marc Haase, Dirk Timmermann,*"Time and Energy Efficient Service Discovery in Bluetooth".*

[4] Sasikanth Avancha, Anupam Joshi, and Timothy Finin*,"Enhanced Service Discovery in Bluetooth", IEEE, Published in* : Computer (Volume:35 , Issue: 6) , Jun 2002, 96 – 99, 10.1109/MC.2002.1009177.

[5] George C. Polyzos, Christopher N. Ververidis, Elias C. Efstathiou, *"Service Discovery and Provision for Autonomic Mobile Computing"*, I. Stavrakakis and M. Smirnov (Eds.): WAC 2005, LNCS 3854, pp. 226 – 236, 2006.

[6] Rupali Favade, Vikrant S. Kaulgud, Sakib A. Mondal *"An Enhanced Service Discovery Protocol for Bluetooth Scatternets "* , 1-4244-0237-9/06/$20.00 ©2006 IEEE

[7] Reen-Cheng Wang · Yao-Chung Chang · Ruay-Shiung Chang, J Intell Manuf ,"*A semantic service discovery approach for ubiquitous computing",* (2009) 20:327–335 ,DOI 10.1007/s10845-008-0213-2

[8] Yeo Bong Yoon and Hee Yong Youn, O. Gervasi et al., *"A New Service Discovery Scheme Adapting to User Behavior for Ubiquitous Computing"*, (Eds.): ICCSA 2005, LNCS 3482, pp. 19 – 28, 2005. Springer-Verlag Berlin Heidelberg 2005.

[9] Baopeng Zhang, Yuanchun Shi, and Xin Xiao, J. Indulska et al*.,"A Peer-to-Peer Semantic-Based Service Discovery Method for Pervasive Computing Environment",* (Eds.): UIC 2007, LNCS 4611, pp. 195–204, 2007. Springer-Verlag Berlin Heidelberg 2007.

[10] Kyriakos Kritikos and Dimitris Plexousakis, E. Di Nitto and M. Ripeanu *,"A Semantic qos-Based Web Service Discovery Engine for Over-Constrained qos Demands",* (Eds.): ICSOC 2007 Workshops, LNCS 4907, pp. 151–164, 2009. Springer-Verlag Berlin Heidelberg 2009

[11] Rajesh Thiagarajan,Wolfgang Mayer, and Markus Stumptner, Q. Li et al. *,"Semantic Service Discovery by Consistency-Based Matchmaking", (*Eds.): apweb/WAIM 2009, LNCS 5446, pp. 492–505, 2009. Springer-Verlag Berlin Heidelberg 2009

[12] Dong Cao, Xiaofeng Li, Xiuquan Qiao, and Luoming Meng,W. Liu et al. *,"A Heuristic Algorithm Based on Iteration for Semantic Telecommunications Service Discovery"* (Eds.): WISM 2009, LNCS 5854, pp. 139–148, 2009, Springer-Verlag Berlin Heidelberg 2009.

[13] Le-Hung Vu, Manfred Hauswirth, and Karl Aberer, C. Bussler et al. , *"Towards P2P-Based Semantic Web Service Discovery with qos Support",* (Eds.): BPM 2005 Workshops, LNCS 3812, pp. 18–31, 2006, Springer-Verlag Berlin Heidelberg 2006

[14] Devis Bianchini · Valeria De Antonellis · Michele Melchiori , "*Flexible Semantic-Based Service Matchmaking and Discovery"* , worldwide Web (2008) 11:227–251 DOI 10.1007/s11280-007-0040-y

[15] Wang Chu, Depei Qian , "*Semantics Based Enterprise Modeling for Automated Service Discovery and Service Composition"* , *2007 IEEE Asia-Pacific Services Computing Conference*, 0-7695-3051-6/07 $25.00 © 2007 IEEE DOI 10.1109/APSCC.2007.23

[16] M. Sughasiny and Dr. R. Dhanapal, "*Performance Improvisation for Bluetooth Service Discovery Using Single Hop and Multi Hop Methods*", European Journal of Scientific Research ISSN 1450-216X Vol.94 Issue 1 January 2013,.

[17] Alan D. Amis, Ravi Prakash, Thai H.P. Vuong, Dung T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks", (2000).

[18] M. Sughasiny and Dr. R. Dhanapal, " *Performance Evaluation of an Alternative Controller for Bluetooth Service Discovery",* ICTACT Journal on Communication Technology – Volume 3,Issue 2,pp.529-534,(2012).

