Proactive Approach for Service Discovery Using Web Service for Devices On Pervasive Computing

Mohammed Fethi Khalfi
Computer Science Department
University of Djilali Liabes
Sidi bel abess, Algeria
fethi.khalfi@univ-sba.dz

S.M.Benslimane
Computer Science Department
University of Djilali Liabes
Sidi bel abess, Algeria
Benslimane@univ-sba.dz

ABSTRACT
Pervasive environments are characterized by rich and dynamic context, where users need to be continuously informed about services relevant to their current context. Proactive service Discovery is one of the most important fields of research in pervasive computing. In This paper we will present an enhancement of ubiquitous computing discovery mechanisms adding context handling capabilities to Web Services for Devices in Pervasive Computing using UPnP as an infrastructure to address these implicit requests. User preferences, network and location are described by a formal context model ontology. As compared with previous research, our method uses location aware, UPnP infrastructure, web service for devices and the notion of proactivity in pervasive computing to continuously present the most relevant services to the user or device in response to changes of context, services or user preferences.

Keywords
Web Services, Web Services For Devices, Proactive Service, SOA, UPnP, Context Awareness, Pervasive computing, Ontology, location-Based Services.

1. INTRODUCTION
Ubiquitous computing, as described by Mark Weiser [1], relies on computers present everywhere, at any times and in any things. Indeed with recent years advance in mobile communication technologies and the miniaturization of computer hardware, processing units are becoming invisible and a part of the environment.

In pervasive environments, however, user context and user preferences are rich and become essential aspects when deciding, which of the available services are of most interest to the user in a given situation. The system should discover and select services in response to changes in the user context, even if the user did not issue an explicit service discovery request to the system. We will show that the capability of capturing these implicit requests can improve user experience significantly.

eb Services for Devices [22] specially aim at permitting interoperability, since they do not rely on any specific programming language or hardware architecture. Service for Devices also suffers constraints related to devices' resource dependencies: frequent disconnections, memory limitation, narrow network bandwidth, limited power, processing capacities, etc. Therefore, the description of Service for Device must include these limitations to inform of specific constraints associated to the provided service. To go further, more specific works on the description of ontological devices and services such as are thus necessary to give a complete Service for Device description.

Devices being most often connected to the real environment of applications, associated services need to offer mechanisms which take into account applications' proactivity to environmental variations. Web services for devices thus define communication protocols using events (subscription, notification) in an asynchronous execution context and include concepts of services and event frameworks, as well as decentralized and dynamic discovery.

Besides, the location concept is largely used to determine the availability of services for devices. In software usage, location is implicitly linked to user's proximity inside his environment. Centralized service directories are difficult to keep up to date, because applications undergo frequent disconnections of devices, which leads to communication overheads while keeping coherence between pieces of information stored in a directory. Adopted solutions in WSFD hold on local and distributed discovery mechanisms between service producers and service consumers. It is the case for UPnP [23] and DPWS [24], with respectively SSDP protocol and WS-Discovery protocol.

In order to explore such a complementary views, we propose a platform that implements the feature of proactive service discovery using Web Service for Devices in Pervasive Computing and utilize UPnP as an infrastructure. The consumer does not need to specify a desired kind of service to be discovered, but it is the service that is offer to users or devices in the background.

The rest of this paper is organized as follows. Section 2, evolved standard SOA to SOA for device (SOAD). Section 3 introduces the notion of proactivity in pervasive environment. Section 4 reviews existing research on proactive services, ubiquitous computing, and context-aware computing. Section 5, introduce our vision: Proactive Approach for Service Discovery Using Web Service for Devices on Pervasive Computing. Section 6 presents a case study of our Proto-

ICCASA 2014, October 15-16, Dubai, United Arab Emirates

DOI 10.4108/icst.iccasa.2014.257492

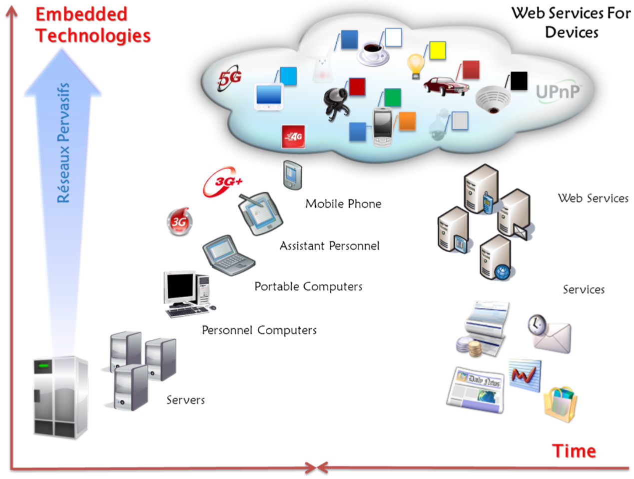


Figure 1: : From Services to Web-Services for Device.

type. Section 7 Conclusion and future work are presented in the last section.

2. FROM SERVICES TO WEB-SERVICES FOR DEVICE

2.1 Service

A service is a software entity which can be invoked, which has a specific functionality, and a well-defined interface [21]. The encapsulation principle gives some freedom in implementations: a given service can actually be provided by different implementations as far as the interface is respected; this gives some freedom in the selection of service provider. Every service is described in a description language fixed by a service framework. For better reutilisability, extensibility and dynamicity, services are not linked together to create applications. Interactions between services are specified independently of the functional code and handled by the service infrastructure. This is what is called loose coupling.

2.1.1 *Service for Hardware*

Services are limited to the computing virtual world. To diversify their functionalities, the service architecture can also be applied on devices. This induces some difficulties. Events are needed to increase proctivity. Discovery protocols also need to be adapted to find a potentially mobile device in the wideness of the world Figure.1.

Device We call input/output devices, or simply devices, equipment's interacting with their physical environment. Devices are basically similar to services: they provide functionality, a way to access it, and are loosely coupled between each other's. A device can appear or vanish at any time. This can be due to real mobile devices or to specific constraints inherited from the real world such as an energy saving policy for example. To increase proactivity, device can use asynchronous communication to notify applications of changes in their state.

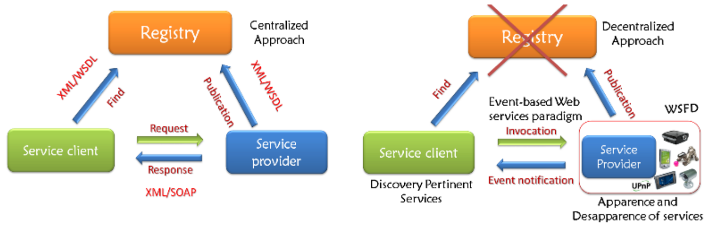


Figure 2: SOA to SOA for device.

2.1.2 *Web Service-Oriented Architecture for Device (WSOAD)*

For many years, service oriented architectures (SOA) have been used in home automation, mobile and ubiquitous computing to represent as services the sets of functionalities offered by devices. They offer lots of features such as encapsulation, dynamicity, discoverability and interoperability. They evolved from standard SOA to SOA for device (SOAD) by adding two main features: decentralized, proactive discovery and asynchronous communications, Figure.2.

2.1.3 *Discovery and Publication*

Devices and services must be statically known for any application to be able to run. Publication is done by service producer; they inform others of their available services. They can either register to a central server (yellow pages approach) or advertise themselves by broadcasting periodically on the network. On the other hand, discovery is done by service consumer which proactively searches for available services. Contrarily to SOA for which service discovery was most of the time assumed by a central service directory, SOAD prefers decentralized discovery mechanisms not to be dependent on any infrastructure.

2.1.4 *Evening Mechanism*

Synchronous communication mechanism is well fitted for services-based applications. Devices do not only provide functions, they may also need to advertise from a modification of their state. Consequently, communications need to occur in both ways, and may happen when the device needs it, without an external request. This is a crucial point for the reactivity of the system. The asynchronous messaging mechanism used by services for devices is event notification. Consumers interested in receiving events from a device have to subscribe to them. Services for devices thus have to handle subscriptions along with event sending. Events generally notify changes from the physical constraints of a device (for example a battery level, a sensor information change).

2.2 Interoperability of Devices in a Service Oriented Architecture

Web Service suffers from the same problem than SOA when trying to interact with devices. Web Services provide mechanism essentially for synchronous communications that is why Web Service for Devices (WSD) was introduced. The WSD model has two major implementations: Universal Plug and Play8 (UPnP) and Device Profile for Web Services (DPWS) that we will describe in more details in following sections.

2.2.1 *Universal Plug and Play (UPnP)*

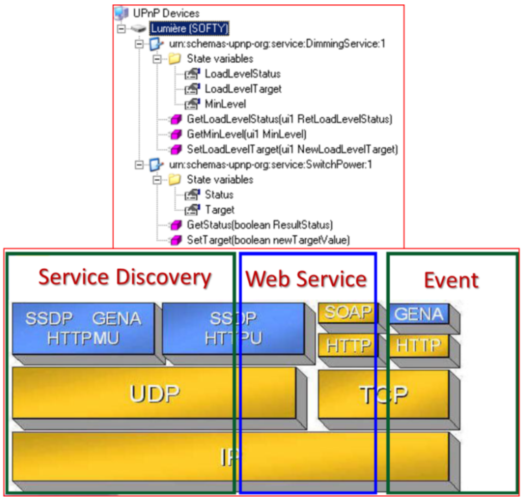


Figure 3: UPnP Architecture.

UPnP was firstly created to extend the Plug and Play concept to peripherals on a network Figure.3. It defines device profiles, in a specific language (though XML), for example for printers, or gateways, which are fixed by the UPnP Consortium, and implemented by UPnP servers in such lightweight devices [23]. The Simple Service Discovery Protocol10 (SSDP) used by UPnP permits to discover and search devices on a local IP network, using multicast UDP messages. It can moreover specify a type filter for searches, and then find only devices or services matching this type. UPnP uses Simple Object Access Protocol SOAP for classical Web Services requests. The eventing layer is managed by the General Event Notification Architecture11 (GENA). Subscription to events is required, and leased. However, this protocol has some drawbacks: events are related to state variables, and must be defined before use. When a service contains several evented variables, all their values are sent at once even if only one value has changed.

2.2.2 Device Profile for Web Service (DPWS)

Due to the limitations previously exposed for UPnP, and to the non-standard descriptions and protocols used, the DPWS technology has emerged with an aim of replacing UPnP. DPWS was created more as a Web Service than as a plug and play protocol [24]. It is based on Web Services standards, such as SOAP for data transfers and WSDL for descriptions. This Web Services kernel is extended by some WS-* specifications: WS-Discovery12 and SOAP-over-UDP for peer-to-peer service discovery at a local network scale, WS-Eventing13 for managing subscriptions for event channels, WSSecurity14 for security, which was cruelly missing with UPnP, WS-Addressing15 for advanced endpoint and message addressing, WS-Policy16 for policy rules and exchanges, WS-Transfer17 and WS-Metadataexchange18 for device and Service description.

UPnP is an open architecture for pervasive network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. UPnP standard is chosen for its flexibility, standards-based connectivity, web services-like architecture and available support from the industry. There are many UPnP enabled devices available in the market today and UPnP SDKs already available for different programming environments and platforms, to ease the development process.

3. PROACTIVITY

Proactivity is a key concern of ubiquitous computing, both for adaptation triggering and for adaptation time. In a pervasive environment, the user invokes services. The system combines these services and deliveries. Reactive systems expect an explicit order of the user to initiate an action. A proactive system provides action based on contextual information (location user preferenceĚetc.), Figure 4.

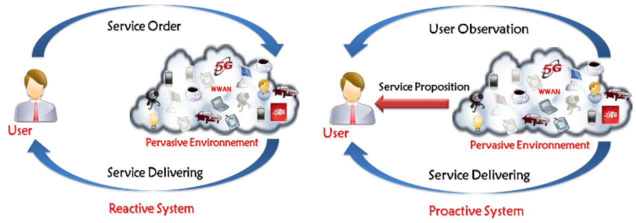


Figure 4: Proactivity Mechanism.

4. RELATED WORK

Several prototypical systems for ubiquitous computing environments have been intensively investigated in recent years [2, 3]. Early discovery approaches were based on the functional description of services, for instance service category, semantic description and key words. For enhancing service discovery, many research efforts involved context in different phases. However, as far as we know, few works has acknowledged context as a first-class criterion and motivating factor in service discovery. In other words, explicit, request-driven service discovery approaches are predominant in pervasive environments. GAIA [4] is middleware support for active space environments such as smart rooms and living environments. GAIA uses a component repository and centralized approaches to events, and services discovery. The context is described using data collection called predicates. Some example context predicates are: Context (location, Amine, entering, room 3231). First order logic formula may be applied on these predicates in order to deduce the action to be taken function on the context. CORTEX [5] proposes a novel sentient object model to address the emergence of a new class of application that operates independently of human control. Infrastructure based and ad-hoc based wireless environments are considered to address mobility. The middleware reacts on events by changing the behavior of objects. In order to select the relevant context, CORTEX uses filters that provide a basic mechanism to allow objects to express interest, or lack thereof, in events of a certain type and zones that introduce a means of scoping or limiting the propagation of event notifications in the system. Aura [6] is a context-aware middleware which can be used to create mobile applications. It represents the user by its aura, like a Personal Area Network (PAN), and brings the appropriate resources from the services of the environment to support the user's task. Context changes are notified by events, and tasks can change while context is evolving. A key context element is considered to be the user location information and various methods may be used in order to locate the user.

Another key aspect of the context is considered to be the time (ex. scheduled tasks). Other context aspects are related to the user profile, personal data (that should be also secured).

The following two examples address service discovery with QoS and context information, but they don't consider the impact of continuous context changes on service discovery. Mokhtar et al. [7] proposed the EASY (Efficient semantic Service discoverY) framework which takes QoS and context into account. EASY relies on semantic modeling using OWL-S to describe context, QoS and functionalities of services. User preference is not considered in EASY.
Park et al. [8] presented the concept of Virtual Personal Space (VPS) to extend the scope of service discovery. VPS conceptually extended the concept of space beyond the location domain by including QoS, user rating and service load in service discovery. Hyun-Yong Noh et al. [17] proposes a new indoor localization technique specific to context awareness.

The Context attributes proposed by Lee and Helal [9] are an early effort of service discovery in smart home environments. Context attributes extend the Jini [10] service discovery protocol to include domain specific context information in the service description. Chung-Pyo Hong et al. [18] purpose of this research is to propose an efficient management scheme for Virtual Personal. World (VPW) which is a model focused on service continuity of personal world.
Broens et al. [11] enhanced the idea of context attributes with an ontology-based context model. The model is able to describe the relationships between different context attributes, and in turn support the semantic matching of service and context description. All of these works focus more on an extension of existing service discovery protocols. Kimberly Garcia et al. [16] proposed a semantic approach consisting of a customizable model expressed as a set of ontologies that suits many different type of organizations, from small ones to large and complex organizational structures.

There are few works that have addressed proactive discovery as a complement to explicit request-driven service discovery; however our work distinguishes itself substantially. Bellavista et al. [12] proposed a user-centric service view for explicit discovery. Discovery scope and service view are the keys to indicate the changes of context information and its impact on available services. Hesselman et al. [13] presented the idea of a persistent discovery request but did not provide further details. Chii Chang et al. [14] propose a proactive mobile Web service-based approach for mobile social network in proximity (MSNP). Evangelos Niforatos et al. [15] introduce a platform for the rapid prototyping of proactive location based service discovery. Sebastian VanSyckel et al. [19] presented a comprehensive approach to configuration management for proactive adaptation in pervasive environments. Joonhee Kwon [20] proposes a method for effective proactive service in pervasive computing environment. this method uses social network and context in pervasive computing.

Conclusions about the existent solutions: However our work is based on a comprehensive formal model of context (Meta Model) and an efficient infrastructure able to reflect the changes in context, user preference, services in real time and integrates intelligent devices that may appear and disappear dynamically (UPnP). Our prototype is implemented with a strong focus on maintaining context information, detecting changes and more frequent proactive discovery can be performed to improve user experience.

5. PROPOSED ARCHITECTURE:

Figure.5 how the overall architecture of our Proactive Service Discovery Using Web Service for Devices system that supports pervasive applications from different domains and propose to the user or device select efficiently the most relevant services that may be found around the user at a given moment. It consists of the following components.
After analyzing our problem, we have decided that the following issues need to be solved: o Our architecture adopts decentralized approach based on the UPnP devices/services discovery mechanisms. o Propose a context model to add the necessary meta-data entities (devices, services, users, etc.). o Integrate the proactive mechanism the user is not obliged to given his request, but this is the system that offers the most relevant services.

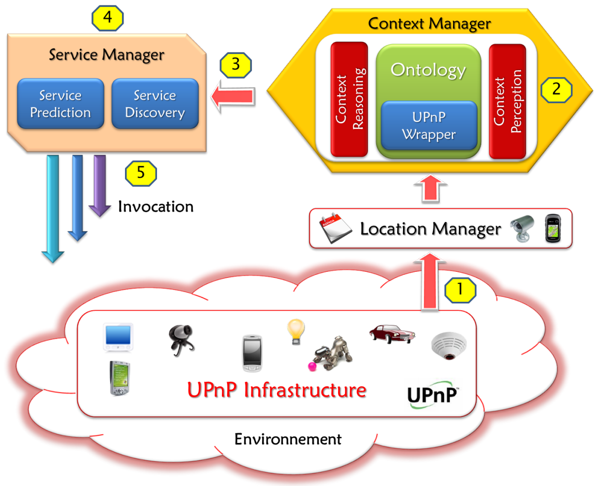


Figure 5: Our conceptual framework for context aware proactive service for WSFD.

The proposed architecture includes the following parts, Figure.6:

5.1 Location Manager:

A range of context provider gives atomic information about the user's or devices location. Context providers cover from low level data obtained from sensors, GPS, RFID, WiFi, etc. The system provides two levels of inferences. At the first level, atomic context information infers new information while in the second the inferred context information is used to infer higher-level information.

5.2 Context Manager:

the context manager collects information from different entities that affects context (e.g., sensors, users, devices, etc.).

Then, it uses context ontology to provide contextual information formatted in this ontology, it includes the Context Interpreter, Context Reasoner, and Model Context Ontology. Each component are described in details in the following subsections:

5.2.1 *Context Interpreter:*

Context information from different data sources has diverse chronological features and data formats. The system needs a type conversion to map the value of one input type to another value of another input type to allow new and innovative information sources to be used. The context interpreter is responsible for converting raw data to meaningful context information that can be put into ontology for use by other components of the system. This can be done by using Web Ontology Language (OWL) which has capability to specify characteristics and functionalities of all the information sources.

5.2.2 *Ontologies:*

OWL ontologies define the context. They represent both data directly sensed from context providers and information inferred from this data using inference rules. The context ontology describes generic concepts and consists of atomic information from the information sources, inferred information from the atomic information using the inference rules and the inference rules. Figure 6.

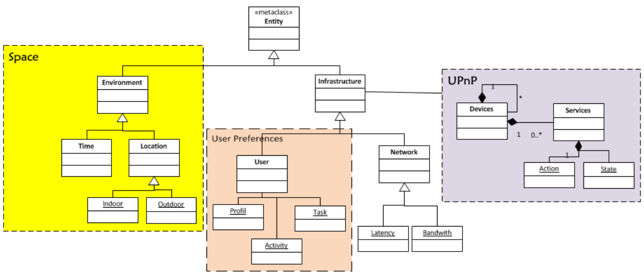


Figure 6: Partial Ontology for Context Manager.

5.2.3 *Context Reasoner:*

This module is responsible for inferring new higher level context from given atomic context information. The context reasoning is based on inference rules. This module may trigger the execution of rules based on the current context information which in turn infers new contexts. It enables the context-aware system to be tailored for specific application scenarios. The context reasoner can also infer context properties using ontology reasoning by specifying inter-ontology relationships.

5.3 Service Manager:

includes Service Discovery and Service Prediction Figure.7.

5.3.1 *Service Discovery:*

Our framework provides lookup functionality to locate appropriate services and devices that are relevant to client needs and in accordance with user context. Using UPnP service discovery mechanism, Services are discovered.

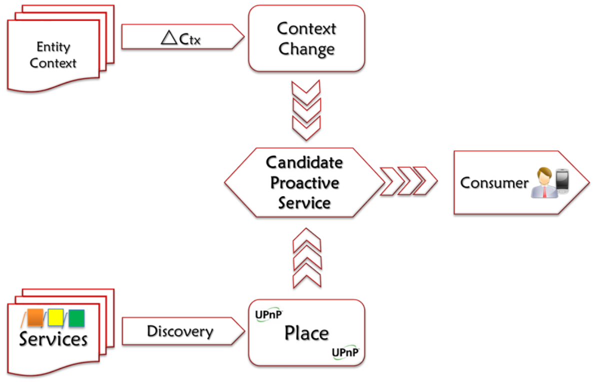


Figure 7: : Illustration of Service prediction.

5.3.2 *Service prediction:*

This approach recommends proactively a service that can satisfy user's future needs. It is based on the assumption that common situations can be detected, even in a dynamic infrastructure based on UPnP.

We propose also in this approach recommends proactively a service that can fulfill user's or devices invoqued. We define the notion of contextEntity (Ce), satisfied by a specific service (SPS) resulting from a previous discovery process based an infrastructure UPnP <Ce, Sd>.

The proactive service candidate algorithm that we propose performs a semantic matching process in order to offer the most appropriate service. The goal is to rank the available service discovered an infrastructure UPnp based on their contextual information ans select the most one for consumer. This algorithm compares service context and content entity that contain location, user preference, (Ontology). The figure present the proposed algorithm for all the services available at the UPNP Space, we calculate the matching score of best service. First we detect change of context than we calculate the matching score between context service and context entity, Figure.8

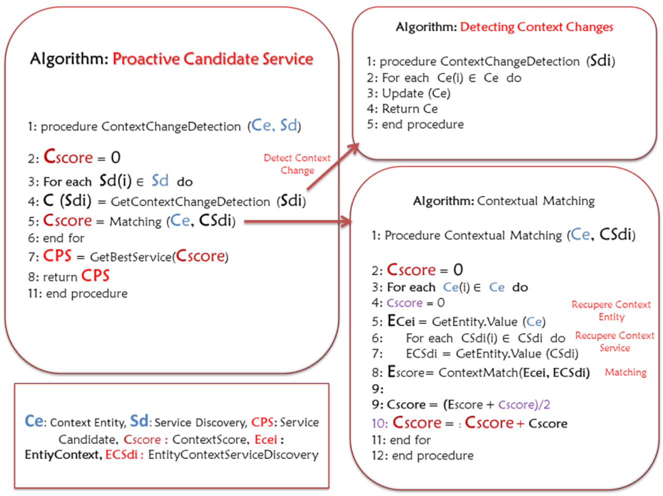


Figure 8: Proactive service discovery algorithm.

6. APPLICATION USE CASES: INTERACTIVE MUSEUMS

We present in this section the scenario of our context-aware proactive service discovery using web Service for Devices. Then, we discuss our experimental results. We consider, for example, an application for visiting a museum.

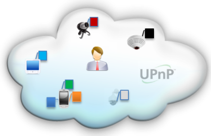
Service Name	ContextService	ContextEntity	Score
S1	-Location.Museums=Zoological -Profile.Age = 21 -Location.City = SBA -Location.Country = AIGERIA -Resource.Screen = 16 -Device = Intel Core 2 duo -Resource.Memory = 2048 -Resource.Network = Ethernet	 -Location.Museums=Archaeology -Profile.Age = 25 -Profile.Langage = Arabic -Profile.Role = Tourist -Location.City = SBA -Location.Country = AIGERIA -Resource.Network = 3G -Device = Iphone 5 -Resource.Memory = 16 -Resource.Screen = 5	2
S2	-Location.Museums=Archaeology -Profile.Age = 24 -Profile. Langage = Arabic -Profile.Role = Citizen -Location.City = TLEMCEN -Location.Country = ALGERIA, -Resource.Network = 3G -Device = Ipad 2 -Resource.Memory = 32 -Resource.Screen = 5		4
S3	-Location.Museums=Archaeology -Profile.Age = 25 -Profile. Langage = Arabic -Profile.Role = tourist -Location.City = ORAN -Location.Country = ALGERIA -Resource.Network = Wifi -Device = Iphone 5 -Resource.Memory = 16 -Resource.Screen = 5		8

Figure 9: analyses after executing algorithm of our approach Proactive Candidate Service.

We present 3 service in an infrastructure UPnP, the analyses after executing algorithm of our approach Proactive Candidate Service illustrated in figure.9 demonstrate that our proposition present a more interesting result; we believe that our proactive service discovery mechanism allow selecting appropriate service provided by web service for devices which is more transparent for users and devices.

7. CONCLUSION

Pervasive computing refers to the seamless integration of devices into the user everyday life. One field in the wide range of pervasive computing is the context-aware system. In recent context aware computing, researchers have proposed vision, such as proactive computing. This paper presents a novel proactive approach service discovery for pervasive environments using Web Service for Devices via UPnP as an infrastructure. The goal is to proactively and continuously discover and offer services that fit the ever-changing context and preferences of users.

Future works will be concentrated on the improvement of our experiment results, we expect to evaluate our proactive service discovery in more interesting real world scenarios, and these results will be tested on a more important of services.

8. REFERENCES

[1] Weiser, M.: The computer for the twenty-first century. Scientific American 265(3) (Sep 1991) 94-104.

[2] Ververidis, C., Polyzos, G.: Service discovery for mobile ad hoc networks: a survey of issues and techniques. IEEE Communications Surveys Tutorials 10(3), 30-45 (2008)

[3] Zhu, F., Mutka, M., Ni, L.: Service discovery in pervasive computing environments. IEEE Pervasive Computing 4(4), 81-90 (2005)

[4] Manuel Román, Christopher Hess, Renato Cerqueira, Roy H. Campbell, Klara Nahrstedt, "Gaia: A Middleware Infrastructure to Enable Active Spaces", IEEE Pervasive Computing, 2002.

[5] Verissimo Paulo, Cahill Vinny, Casimiro António, Cheverst Keith, Friday, Adrian and Kaiser Jörg. "CORTEX: Towards supporting Autonomous and Cooperating Sentient entities." In: Proceedings of European Wireless 2002 (EW2002), 26-28 February 2002, Florence, Italy, 2002.

[6] Garlan, D.; Siewiorek, D.P.; Smailagic, A.; Steenkiste, P.; "Project Aura: toward distraction-free pervasive computing", Pervasive Computing, IEEE Volume 1, Issue 2, April-June 2002 Page(s):22 - 31, 2002.

[7] Mokhtar, S.B., Preuveneers, D., Georgantas, N., Issarny, V., Berbers, Y.: Easy: efficient semantic service discovery in pervasive computing environments with qos and context support. J. Syst. Softw. 81(5), 785-808 (2008)

[8] Park, K., Yoon, U., Kim, S.: Personalized service discovery in ubiquitous computing environments. IEEE Pervasive Computing 8(1), 58-65 (2009)

[9] Lee, C., Helal, S.: Context attributes: an approach to enable context-awareness for service discovery. In: Proceedings of the 2003 Symposium on Applications and the Internet, pp. 22-30 (2003)

[10] Arnold, K., Scheifler, R., Waldo, J., O'Sullivan, B., Wollrath, A.: Jini Specification. Addison- Wesley Longman Publishing Co., Inc., Boston (1999)

[11] Broens, T., Pokraev, S., van Sinderen, M., Koolwaaij, J., Dockhorn Costa, P.: Context-aware, ontology-based service discovery. In: Ambient Intelligence. Lecture Notes in Computer Science, vol. 3295, pp. 72-83. Springer, Berlin/Heidelberg (2004)

[12] Bellavista, P., Corradi, A., Montanari, R., Toninelli, A.: Context-aware semantic discovery for next generation mobile systems. IEEE Commun. Mag. 44(9), 62-71 (2006)

[13] Hesselman, C., Tokmakoff, A., Pawar, P., Iacob, S., et al.: Discovery and composition of services for context-aware systems. Lect. Notes Comput. Sci. 4272, 67 (2006).

[14] , Chii Chang, Satish Narayana Srirama, Shonali Krishnaswamy, Sea Ling, Proactive Web Service Discovery for Mobile Social Network in Proximity, JNIT, Vol4,numero2, Avril 2013.

[15] Evangelos Niforatosa, Evangelos Karapanos and Spyros Siouta, PLBSD: a platform for proactive location-based service discovery, Journal of Location Based Services 2012, 1-16, iFirst.

[16] Kimberly Garcia , Manuele Kirsch-Pinheiroy, Sonia Mendoza , Dominique Decouchant An Ontological Model for Resource Sharing in Pervasive Environments, "2013 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Atlanta, GA : United States(2013).

[17] Hyun-Yong Noh, Jin-Hyung Lee, Sae-Won Oh, Keum-Sung Hwang, Sung-Bae Cho Exploiting indoor location and mobile information for context-awareness service, Information Processing and Management 48 (2012) 1-12.

[18] Chung-Pyo Hong ů Cheong-Ghil Kim ů Shin-Dug Kim

,An Effective Personalized Service Provision Scheme Based on Virtual Space for Ubiquitous Computing Environment, Wireless Pers Commun, Springer Science+Business Media New York 2013.

[19] Sebastian VanSyckel, Dominik Schafer, Gregor Schiele, Christian Becker ,Configuration Management for Proactive Adaptation in Pervasive Environments.

[20] Joonhee Kwon, A Study on Effective Proactive Service in Pervasive Computing Environment, IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.8, August 2012.

[21] Mohsen Rouached a, Shafique Chaudhry a, Anis Koubaaab ,LoWPANs Meet Service-Oriented-Architecture ,Journal of Ubiquitous Systems and Pervasive Networks Volume 1, No. 1 (2010) pp. 39-48.

[22] "Web Services for Devices Initiative," WS4D, http://www.ws4d.org.

[23] UPnP Device Architecture v.1.0.1, UPnP Forum, 2003. "The Universal Plug and Play (UPnP) Forum," http://www.upnp.org

[24] Dan Driscoll, Antoine Mensch, Devices Profile for Web Services (DPWS) Version 1.1, OASIS Standard, July, 2009.