Towards a Semantic and Dynamic Cluster based Web Service Discovery System for Ubiquitous Environments

Rim Helali[1], Nadia Ben Azzouna[2] and Khaled Ghedira[1]
[1]*Higher Institute of Management, Research Laboratory SOIE, Tunis University, Tunis, Tunisia*
[2]*National School of Computer Science, Research Laboratory SOIE, Manouba University, Manouba, Tunisia*

Keywords: Ubiquitous Environments, Web Service Discovery, Clustered Architecture, Semantic Service Publication and Discovery.

Abstract: Ubiquitous computing aims to exchange and share services anywhere, anytime. However, discovering the appropriate service in ubiquitous environments poses very specific challenges. In this paper, we propose a new Semantic and Dynamic Cluster based Web Service Discovery System (SDC- WSDS) which deals with dynamicity and scalability challenges. Our discovery system is based on a clustered architecture and supports semantic publication and discovery queries. We, hence, present a new clustering algorithm for ubiquitous environments inspired by the Weighted Clustering Algorithm (WCA) and a semantic service publication and discovery model. This model is based on attribute value pairs and a semantic distance function.

1 INTRODUCTION

"Ubiquitous computing is a technology in which invisible computers are embedded and connected with all things so that anyone can communicate, exchange and share information anywhere, anytime" (Kang et al., 2007). Currently, we are surrounded by computers, mobile phones and personal digital assistants (PDA). It becomes difficult to manage these devices heterogeneity, configure different types of applications and dynamically discover appropriate services. Web service technologies seem to be adequate to address the heterogeneity problem and to help implementing pervasive applications. It is due to the fact that a web service is like an abstract middleware hiding the details about the application platform and programming language (Xu et al., 2008). In addition to devices heterogeneity, many other challenges related to ubiquitous environments were addressed in several articles (Estrin et al., 2002) (Satyanarayanan, 2001). In fact, while discovering web services we are faced to very specific ones. We focus in this paper mainly on those that we consider the most critical for web services discovery such as:

- **Dynamicity:** refers to the system's ability to manage the mobility of devices in the environment and their unexpected appearance or disappearance. Due to providers and directory nodes dynamicity (nodes that maintain descriptions of web services), the system must be able to replace them at any time (Abdennadher, 2008). Users mobility must be also taken into account in order to ensure web services availability wherever users move.
- **Scalability:** refers to the system's ability to maintain good performance with the increased number of service publication and discovery queries (Gao and Steenkiste, 2004). In fact, the majority of ubiquitous environments are characterized by the abundance of users and their interactions, and by the big number of connected devices and services. To ensure scalability, centralized solutions are to be avoided in order to prevent the problems of bottleneck and Single Point Of Failure (SPOF). Priority should also be given to localized interactions; we refer here to localized scalability meaning that remote queries should be eliminated. In fact, the density of interactions should decrease as soon as the user moves away; otherwise the network will be overwhelmed by costly interactions that are of little interest (Satyanarayanan, 2001).

In this paper, we present our Semantic and Dynamic Cluster based Web Service Discovery System (SDC-WSDS) which is based on a clustered architecture and supports semantic publication and discovery queries. Because of the abundant number of users characterizing the majority of ubiquitous environments, a clustered architecture is adopted to reduce the cost

Helali R., Ben Azzouna N. and Ghedira K..
Towards a Semantic and Dynamic Cluster based Web Service Discovery System for Ubiquitous Environments.
DOI: 10.5220/0003972902950300
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 295-300
ISBN: 978-989-8565-11-2

of searching services. Services are mapped using a semantic distance function in order to improve the service searching efficiency. Dynamicity of nodes, and especially directory nodes, is managed by applying a dynamic clustering algorithm inspired from the Weighted Clustering Algorithm (WCA) originally proposed for ad hoc mobile networks (Chatterjee et al., 2001). We adapted the WCA algorithm to cope with the specificities of ubiquitous environments. Hence, we aim in this work to define a system that manages the high dynamicity of the environment while guaranteeing the scalability requirement.

The rest of the paper is organized as follows. Section 2 addresses related work and identifies some limitations. Section 3 describes the SDC-WSD system. Finally, Section 4 concludes the paper and discusses some future directions.

2 RELATED WORK

Web service discovery systems proposed in the literature used a diversity of architectures ranging from centralized systems to fully distributed systems. Despite the fact that availability and rapidity of access are guaranteed with systems using centralized directories (Chen et al., 2006), (Hwang et al., 2007), these systems suffer from problems like SPOF and bottleneck as the number of registration and discovery queries increases. To address the scalability issues related to centralized architectures, solutions have been proposed dealing mainly with decentralization of directory nodes (Kang et al., 2007), (Xu et al., 2008), (Gao and Steenkiste, 2004), (Kim et al., 2005), (Liu et al., 2002). Kim et al. proposed a fully distributed architecture by electing for every service requestor or provider, K volunteer nodes (directory nodes) to which publication and discovery queries are sent (Kim et al., 2005). Problems of SPOF and bottleneck are thus avoided but this decentralization has its cost because voluntary nodes are announced by flooding the query and even in case of updating the volunteers list, flooding is used. To avoid the flooding problem, Xu et al. and Kang et al. proposed web service discovery systems based on a clustered architecture that reduces the search space and thus avoids overloading the network (Kang et al., 2007), (Xu et al., 2008). To add semantic search capability, Kang et al. proposed a clustering method based on semantic similarity where each cluster contains descriptions of the most semantically similar services. Services are in fact, defined by attribute-value pairs (AV- Pairs) and semantics of each attribute is described using an ontology (Kang et al., 2007). In order to manage nodes dynamicity characterizing ubiquitous environments, many solutions have been proposed ranging from choosing nodes that have the highest resources capacity and the lowest mobility to be directory server nodes (Kim et al., 2005), to proposing a cluster formation algorithm inspired from the cluster head selection method (Liu et al., 2002).
In our web service discovery system SDC-WSDS, we propose 1) a clustering algorithm inspired from the Weighted Clustering Algorithm (WCA) originally proposed for ad hoc mobile networks and which we adapted to meet with ubiquitous environments challenges and 2) a semantic service publication and discovery model based on AV-Pairs and a semantic distance function.

3 SDC-WSDS DESCRIPTION

In this section, we present our web service discovery system SDC-WSDS which is designed to enhance scalability, and dynamicity while discovering web services in ubiquitous environments. To provide scalable web service discovery, we adopt a clustered architecture in order to reduce the number of publication and discovery queries (the access is limited to the directory nodes). Each cluster is organized in a two level hierarchical architecture containing a Directory Server (DS) and a number of web service servers (WSS) (Xu et al., 2008). Thus, the ubiquitous environment is partitioned into dynamic clusters using a clustering algorithm as detailed in section 3.1. In our proposed architecture, as described in figure 1, each node can be mobile and can take one of four roles:

- A Directory Server (DS): is responsible of a certain region in the ubiquitous environment. It maintains descriptions of all existing web services distributed within its region (Xu et al., 2008). Different from those in USDM-PerCom (Xu et al., 2008), each DS contains, in addition to the descriptions of web services within its region, descriptions of the most semantically similar web services according to the AV-Pairs assigned to each DS. In fact, in order to be able to calculate the semantic distance, AV-Pairs (provided from the first service description that a local WSS sends) are assigned to DSs. After registering its first service, each DS sends the AV-Pairs, that it will be responsible of, to all other DSs. The DS having not yet been assigned any values, when receiving a new service description, checks the values supported by the other DSs by calculating a semantic distance between them and the received values. If the calculated distance is lower than a

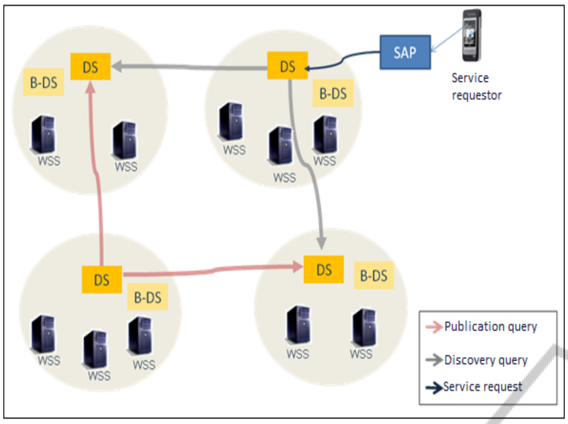

Figure 1: SDC-WSDS architecture.

predefined threshold, the DS registers the service description without setting it as its supported values and waits for another service description.

- A backup Directory Server (backup DS): maintains the same service descriptions as the local DS (periodic messages are sent from the DS to the backup DS containing a list of the new registered services). It is used to replace the DS in case of failure in order to guarantee service availability and to enhance stability of the discovery system.
- A Web Service Server (WSS): contains a number of web services and their descriptions. A WSS is assigned to a DS in the ubiquitous environment. This DS becomes its gate to the entire ubiquitous environment (Xu et al., 2008).
- A service requester: accesses to the environment through a portal and specifies the characteristics of the desired service (Xu et al., 2008). The portal is responsible of articulating the request into AV-Pairs and sends the query to the nearest DS .

To enhance the effective search capability, the most relevant service according to the query has to be discovered without visiting all the DSs in the environment. Thus, publication or discovery queries are only sent to the most semantically similar DSs. A semantic distance function is used to calculate the similarity between registered services and publication and discovery queries. The following subsections describe in details the proposed solutions to realize our SDC-WSDS.

3.1 Clustering Algorithm

To form the clusters of the environment, we defined a clustering algorithm inspired from the WCA algorithm. The choice of WCA is based on the fact that this algorithm takes into consideration several parameters like transmission power, mobility and battery power of mobile nodes (Chatterjee et al., 2001). These parameters are very significant for the election of the cluster-heads (DSs in our case), especially in ubiquitous environments where nodes are characterized by a high dynamicity.

3.1.1 DSs Election and Clusters Formation

Initially, there are no DSs in the environment. The DSs list is generated for the first time by invoking the DSs election procedure at the time of the system activation. Each node in the environment broadcasts to its immediate neighbors (i.e. one-hop neighbors) its ID and chooses only the nodes with a transmission delay lower than a predefined threshold to form its neighbors set. This restriction on the neighbors set is mainly motivated by reducing the distance between a node and its neighbors. The procedure, as described in Algorithm 1, consists of electing the DSs on the base of a calculated value Wv. In the original WCA algorithm, the value Wv calculated for every node is a combination of components with certain weighting factors chosen according to the system needs (Chatterjee et al., 2001). Due to the dynamicity of the ubiquitous environment and in order to guarantee the system stability, we judge, in our proposed algorithm, that the nodes mobility and the degree of nodes (i.e. the number of neighboring nodes dv) are more important than battery power in the DSs election procedure. Therefore, the value Wv is calculated as :

$$Wv = \frac{(1 + b\Delta V + cMv)^2}{Pv} \tag{1}$$

where b and c are calibration factors. The first component Δv computes the degree-difference for each node v by calculating the difference between the ideal number of nodes a DS can support δ and the number of neighbors dv. It helps, thus, choosing the node with the nearest degree to the ideal degree δ as a DS. This is to ensure that the DSs are not over-loaded and the efficiency of the system is maintained at the expected level. The component Mv represents the mobility of nodes. A node with less mobility is always a better choice to be a DS. To calculate nodes mobility, we chose, rather than using the mobility metric proposed in the original algorithm that is based on a localization system, to use the mobility metric proposed in (Li et al., 2007). The choice of this specific metric is justified by the fact that it does not depend on any location system (e.g. GPS) and it can fully capture the relative motions between a node and its neighborhood, in real-time, using simple triangulation (Li et al., 2007). The last component Pv, represents the remaining bat-

Algorithm 1: Clustering algorithm proposed for the SDC-WSDS.

Input : listID, n, TD, Pv, δ
Initialization : dv=0
for $i = 1$ to n **do**
  for $j = 1$ to $n-1$ **do**
    if Transmission Delay $\langle$ TD **then**
      $dv = dv + 1$
      $add(listID(j), N)$
    end if
  end for
  Δv = $|dv$-$\delta|$
  $Mv(t-\Delta t) = \frac{1}{|U|}\sum_{i,j\in U}\frac{M_{ij}(t-\Delta t)}{\Delta t}$
  $Wv = \frac{(1+b\Delta V+cMv)^2}{Pv}$
end for

listId = list of the identifiers of one-hop neighbors, TD = a predefined transmission delay threshold that a node cannot exceed to be considered as a neighbor, n = the total number of one-hop neighbors, N = the set of one-hop neighbors of the node v with transmission delay $\langle$ TD, Pv = Battery power at the time t.

tery power of a node. Electing the node which has the highest battery power is more relevant in the DSs election than choosing the one which has consumed the lowest amount of battery power(used in the original algorithm). It is due to the fact that a node may initially have limited battery power. So knowing the consumed battery power of a node does not reflect the time during which a node can play the role of a DS after being elected. After calculating its Wv, each node sends its value to its neighbors (nodes forming the set N). This exchange of messages permits the nodes to be aware of the node with the lowest Wv in their neighbors set which is elected as a DS. Non elected nodes (i.e. WSSs) send a membership query to their local DS. Thus, each new elected DS is aware of its cluster members. If a node is elected as a DS (i.e. receives at least one membership query) despite the fact that it has not the lowest Wv in its neighbors set (it has the lowest Wv in another node neighbors set), the elected node has to disjoin the cluster to which it has been assigned as a WSS by sending a disjunction query to the local DS. Each elected DS chooses from the cluster the node that has the second lowest Wv as a backup DS. If a node has an empty neighbors set due to transmission delay restriction, it is then elected as a DS in its region.

3.1.2 Nodes Dynamicity Management

Due to the dynamic nature of the ubiquitous environment, the nodes as well as the DSs tend to move in different directions. So, a node may be detached from its local DS and looks for a new DS; a reaffiliation is thus done. New DSs can be added to the set if a node cannot find a DS to which it can be affiliated due to the overload of neighboring DSs. In addition to nodes mobility, unavailability of nodes (i.e. breakdown, battery power extinction) has to be managed especially for DSs and backup DSs. We discuss each of these cases in details in Table 1.

3.2 Semantic Discovery Procedure

Publication or discovery of a web service is done by sending a query containing a description of the service. We propose a semantic discovery procedure in which each query is comprised of a set of AV-pairs where service functionalities, inputs, outputs, and the characteristics of the device containing the service are described by pairs of attributes and values. The choice of this type of representation is due to the fact that it allows service providers (WSS) to precisely describe their services and requesters to easily describe their service requirements. The semantic of each attribute is represented by an ontology which provides information about the semantic relations among the concepts. The ontology trees are shared by all the nodes including DSs and WSSs. Thus, nodes are aware of which type of ontology should be used to describe the services to be published or discovered (Kang et al., 2007). A semantic distance function is used to calculate the distance between the AV-pairs of the service to discover and the AV-Pairs assigned to each DS in order to enhance the effective search capability. In fact, each DS is responsible of the most semantically close service descriptions to its supported AV-Pairs. It is made possible by calculating the semantic distance between the AV-pairs of the service to publish and the AV-Pairs assigned to the DS. Thus, the most appropriate service for the considered request is discovered without visiting all the DSs and therefore we avoid the network overload.

3.2.1 Web Service Publication

In SDC-WSDS, we register web services not only in the local DS but also in the K most similar DSs according to a calculated value D which is a combination of the semantic distance and the physical distance. The duplication of service description in K DSs is done to ensure availability of services and the choice of the K DSs on the base of semantic similarity enhances effective search capability (Liu et al., 2002). First, each WSS which has a new web service to publish sends a publication query to its local DS. The DS calculates the semantic distance between the service to publish and the AV-Pairs assigned to the DSs. After calculating the semantic distance, we choose K DSs to which the service description will be sent. The parameter K as defined by Liu et al. is

Table 1: Nodes dynamicity management in SDC-WSDS.

Condition	Considered action(s)
A node detecting a new DS closer than its local DS (a higher signal strength).	The node sends a request for affiliation. The DS accepting the request updates its list and informs the old DS.
Arrival of a new node which sends an affiliation request to all its neighboring DSs but no response is received (due to overloading of neighboring DSs).	The DS election procedure is invoked, but only neighboring DSs are involved. After reforming the clusters, each old DS sends to its local DS its registered service descriptions, and each new DS sends a DS replacement query to all other DSs in the environment.
Failure of a DS (If no messages are received by the backup DS during a specific period of time).	The backup DS initiates the DS election procedure by sending a DS election query to the WSSs members of the cluster. After electing the DS, the backup DS sends its registered service descriptions to the new DS which chooses the node with the second lowest Wv as its new backup DS and sends to the other DSs in the environment a message informing them of its new status.
Failure of a backup DS (if no acknowledgements are received during a specific period of time after sending messages containing the new registered services).	The DS selects the node that has the smallest Wv from its WSSs to be the new backup DS (Nodes have to recalculate their Wv).
Failure of both DS and backup DS (the unavailability of the DS is detected by the WSSs).	The WSSs have to elect a new DS among them by recalculating their Wv and choosing the WSS with the smallest Wv. The second WSS to have the smallest Wv is chosen as a backup DS. The newly elected DS broadcasts its information to all other DSs. Upon receiving the message, they add the new DS to their list. If it is semantically and physically close; they send their registered service descriptions to it.

equal to $log_{\,1-p}\, A$ with p is the unavailability probability of a DS and A is the availability requirement of a DS (Liu et al., 2002). We propose to calculate a distance D which value is given by :

$$D = P_1 \times d + P_2 \times hopcount \qquad (2)$$

where P_1 and P_2 are weighting factors with the sum equal to 1, d is the semantic distance calculated according to a semantic distance function such as the distance function used in (Kang et al., 2007) and hopcount is the physical distance between the local DS and each one of the other DSs. This distance D is used to choose the K DSs for the service publication. It is a combination of the semantic distance and the physical distance. The choice is justified by the fact that the most semantically close DSs to the service to publish could be the most distant physically to the DS. Therefore, a compromise has to be established between semantic and physical distance in order to reduce the number of hops when publishing a service, and according to the provider needs, a bigger importance can be accorded to one of the two distances by assigning a larger weighting factor. The K DSs with the smallest value D are chosen to publish the services.

3.2.2 Web Service Discovery

When looking for a service, a service requester accesses to a portal in which he expresses his request by specifying the functionalities of the required service. The portal takes the responsibility of articulating the request in the form of AV-Pairs and sends it to the nearest DS. The DS calculates the semantic distance between the service description and the assigned values of all the DSs in the environment and sends a multicast query to the K first DSs with the smallest semantic distance d. Each DS checks its list of registered services to see whether there are any services matching the query. A DS finding a matching service sends a response to the DS originator of the query with the service ID. After receiving all the responses, the DS proceeds to a service selection based on contextual attributes such as the WSS status (busy or not), the physical distance and so on. A query is then sent to the DS responsible of the selected service which sends a service invocation to the WSS containing the web service. The response of the WSS is then routed to the service requestor.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a semantic and dynamic cluster based web service discovery system named SDC-WSDS which is based on a clustered architecture and supports semantic publication and discovery queries. Therefore, we proposed a clustering algorithm permitting the partition of the environment into dynamic clusters and the management of nodes dynamicity. Services are semantically defined using attribute-value pairs and the semantic similarity between them is measured by a semantic distance function. Publication and discovery queries are thus routed only to the most semantically similar nodes in order to reduce the number of visited nodes and to enhance searching capability. Our future work will first include the implementation and the evaluation of SDC-WSDS by testing the stability of the system in terms of number of nodes reaffiliations and DSs set updates and by testing the scalability of SDC-WSDS in terms of number of publication and discovery queries. Second we propose the extension of the proposed system to support quality of service during the web service selection in order to meet with the invisibility challenge.

REFERENCES

Abdennadher, I. (2008). Une approche pour l'assurance des qualites de services des systemes publier/souscrire deployes sur un reseau mobile ad-hoc. Master's thesis, Ecole Nationale d'Ingnieurs de Sfax, Tunisie.

Chatterjee, M., Das, S. K., and Turgut, D. (2001). Wca: A weighted clustering algorithm for mobile ad hoc networks. *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 5:193–204.

Chen, I. Y., Yang, S. J., and Zhang, J. (2006). Ubiquitous provision of context aware web services. *IEEE International Conference on Services Computing*, pages 60–68.

Estrin, D., Culler, D., Pister, K., and Sukhatme, G. (2002). Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1:59–69.

Gao, J. and Steenkiste, P. (2004). Design and evaluation of a distributed scalable content discovery system. *IEEE Journal on Selected Areas in Communications*, 22:54–66.

Hwang, Y.-Y., Oh, I.-J., Im, H.-J., Lee, K.-C., Lee, K., and Lee, S. (2007). Uws broker for ubiquitous web services dynamic discovery. *International Conference on Intelligent Pervasive Computing*, pages 204–209.

Kang, S., Kim, D., Lee, Y., Hyun, S. J., Lee, D., and Lee, B. (2007). A semantic service discovery network for large-scale ubiquitous computing environments. *ETRI Journal*, 29:545 – 558.

Kim, M., Kumar, M., and Shirazi, B. (2005). Service discovery using volunteer nodes for pervasive environments. *International Conference on Pervasive Services*, pages 188–197.

Li, Z., Sun, L., and Ifeachor, E. C. (2007). Gps-free mobility metrics for mobile ad hoc networks. *IET Communications*, (5):1–18.

Liu, J., Zhang, Q., Li, B., Zhu, W., and Zhang, J. (2002). A unified framework for resource discovery and qos-aware provider selection in ad hoc networks. *ACM Mobile Computing and Communications Review*, 6:13–21.

Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8:10–17.

Xu, T., Ye, B., Kubo, M., Shinozaki, A., and Lu, S. (2008). A gnutella inspired ubiquitous service discovery framework for pervasive computing environment. *8th IEEE International Conference on Computer and Information Technology, CIT*, pages 712 – 717.