Toward Next Generation Web Service Technology Overly Network: A Quantitative Approach

Fardin Abdali Mohammadi[1], Naser Nematbakhsh[2] and Mohammad Ali Nematbakhsh[3]

Department of Computer Engineering, Faculty of Engineering,
University of Isfahan, Iran
[1]ebdalimo@ce.sharif.edu, [2]nemat@eng.ui.ac.ir, [3]nematbakhsh@eng.ui.ac.ir

Abstract

Service is the core concept of future Internet that named Internet of Services. This concept refers to software components as well as processing capacity and any other resources that can be offered through the Internet. There is a need for a third party that make registering and searching services easy and feasible. Web Service Technology (WST) is one of those current solutions to the Internet of services that has attracted much attention. Registering and searching of the services are feasible by use of Service Directory. The current structure of WST that is mainly based on UDDI suffers from some structural shortcomings. From a structural point of view, service directory is implemented as a centralized node that is a performance bottleneck and single point of failure. In this paper, shortcomings of the current UDDI structure are analyzed through some quantitative experiments. Then, in order to address this shortcoming, a structure is presented by interlinking services using some sort of semantic relation that can be established between WST entities. Different aspects of structure and its performance enhancement are shown through experiments.

Keywords: *Peer-to-Peer Networks; Semantic Relation; Service Directory; Web Service*

1. Introduction

Web Service Technology (WST) is one of the current technologies that make the Service Oriented Architecture feasible through the Internet. In this technology three main entities named service provider, Service Directory, and service consumer, play roles. Service providers provide some services and create an interface for each of them to make it accessible and usable through the web. They also provide a service description for each of the services they created that describe the specification of it. These service descriptions are registered in a service directory. Service consumers search in the service directory to find appropriate services that fulfills their requirements.

WST suffers from serious shortcomings that make it inapplicable for future Internet. As the number of users, data, and service providers grow, the unevolved overlying structures of WST can not handle the huge amounts of requests generated by parties. Instead, it requires facilities that will enable it to work reliably and transparently. From a structural point of view, current service directories such as UDDI are centralized. So, it is a single point of failure and may become overloaded. Also, service directories do not monitor the status of the registered services.

Therefore, while the user may find a suitable service using service directory facilities, the service may not be usable. This happens because of service or provider failure or even network problems that are not reflected in the service directory. From the knowledge processing point of view, the service directory acts as a simple stateless search machine and does not use the previous matching knowledge to enhance future searching. For example, to response to a user query, some matching will take place, revealing some sort of semantic relationship between processed services. By managing these types of knowledge, service directories can be empowered [1] to handle failures more quickly.

The current structure of UDDI communication links is shown in Figure 1. In this figure, the service directory is shown as a big circle in the center of the structure. Service providers (*SP* nodes) publish services descriptions, and service consumers (*SC* nodes) query the directory to find their required services. Let us look closely at entities in this figure. Service consumers connect to this network, do not share any resource, and use the shared resources of others. On the other hand, service providers connect to the network, share some resources, and may use the resource offered by other providers. Lastly, the service directory handles the search queries.

At the first glance, this structure can be viewed as a superpeer based peer-to-peer network. This motivate some researchers to distribute service directories or generally service oriented architectures based on these networks' techniques [2, 3]. In peer-to-peer networks, each peer joins the network, may share some resources, and searches for another resource in the network to use. For example, in a file-sharing peer-to-peer network, each client joins the network, may share some files, and search for some other files in the network to download. Downloading the files requires consuming the bandwidth and processing power and time of the host peer. In web service networks, resources are web services and clients use the computation power of the providers to run their desired services. In this structure, service directory acts as a single superpeer.

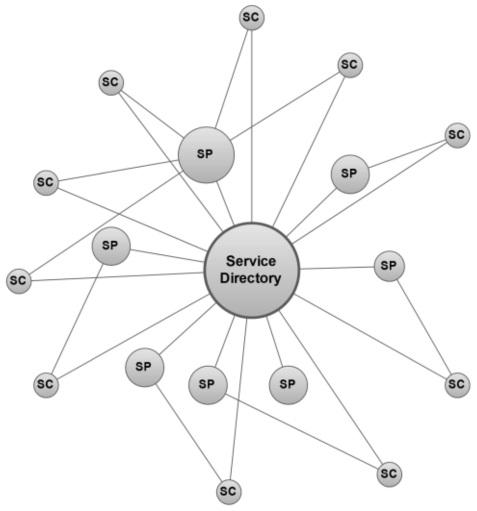


Figure 1. Current UDDI Web Service Network Structure

For designing a new web service network to overcome mentioned shortcomings, a new structure of semantic interlinked web services entities are presented. In the knowledge processing point of view of this structure, in order to enable knowledge in web service directories and use this knowledge to outperform, knowledge discovery and method of its representation based on the semantic relation between web services is presented [1]. In this method, web service entities are interlinked semantically and web services are published based on the semantic relations that exist between them. In a structural point of view of this structure, differences between web service networks and traditional peer-to-peer networks are investigated, and, with regard to the unique feature of web service networks, a structure is presented.

The main contribution of our work is that:

- We investigated the properties of web service networks and compare the characteristics of web service networks against traditional peer-to-peer networks.
- We proposed a structure for web service networks based on WST properties and a semantic interlinking technique.

The rest of the paper is organized as follows: the related works have been studied comprehensively in Section 2. In Section 3, the properties of web service networks are investigate and compare the characteristics of web service networks against traditional peer-to-peer networks. In Section 4, the proposed overly network structure based on previous findings in [1] and studies of section 3 are presented. In Section 5, the system is experimented.

2. Related Works

2.1. Service Directories

Registering and matching web services are the core activities of service directories, and it dates back as far back as the provision of WST. These activities are done in service directories using prepared APIs. The service directory is a place to store a service's description and specification. Web service developers publish their services in the service directory and the users find their services by searching it using operations offered by service directories. The architecture of service directories are a great influences on core operations. UDDI [4] and ebXML [5] are some examples of two famous service directories.

2.1. Network of Web Services

Some methods are presented based on the creation of network of services with an extension of UDDI [6-10]. In [6], a semantic network consisting of service providers and services is presented. In this method, the relationship between service providers is defined, hereby showing how service providers cooperate with each other. This method is proposed in order to replace a service with its counterparts in case of failure. In [7], a centralized method is presented for service publication by using the concept of similarity of services in order to facilitate composition operations. Here, at first the similarity between input arguments of one service is measured with the output arguments of another service. Based on the degree of similarity between inputs of a service with outputs of a service a graph called semantic network is constructed. Also based on user request for service composition, this graph is scanned and required services are extracted. Finally, the extracted plan is introduced to user. The presented model applies some changes in UDDI and performs service publishing and discovery operation centrally. In [11], a centralized service directory model based on semantic relations between web services is presented. The proposed model is replaced with UDDI and also plays the role of a crawler. In the presented model, some relations are defined, and, based on those relations, a semantic network of services is constituted. Additionally, a method called Service Mining is proposed for extraction of similarity relation.

Abdali, et. al., [1] extracted WST entities and defined some sort of semantic relations between them. By using these relations, an interlinked network of WST entities named *SIG* was constructed. Using this graph, some enhanced and new operations are offered by service directories.

2.3. Peer-to-peer Networks

In the recent decade, there has been an increasing amount of published materials on peer-to-peer networks. Aberer in his recent book [12] presents good, state of the art peer-to-peer data management systems. Staab and Stuckenschmidt [13] collect some contributions in integrating peer-to-peer networks and the semantic web. Sacha [14] provides in-depth study of peer-to-peer networks and different structural maintenance algorithms. He also presents a method named *Gradient Topology* for constructing peer-to-peer networks in [14]. And, based on this concept, he presents a method for decentralizing a service oriented architectures [2].

Some works present a service discovery architectures based on peer-to-peer networks [3, 15-17]. Most of them are suitable for an exact match or ranged queries. Matching operation in real web service network is a complex task; and, by exact matching the probability of finding a service will decrease. Rodriguez, et. al., [18], by investigating some published service, shows that about 82 percent of services suffers from name ambulation and about 68 percent of them are not well documented. These problems lead to deficiency in service discovery.

3. Properties of Web Service Networks

Web service networks have unique features in comparison with similar networks such as peer-to-peer networks. Designing a structure for web service networks without considering these characteristics into account results in poor structure, or, in some cases very unnecessary and complicated structures. This section investigates these unique characteristics of web service networks with a quantitative approach.

3.1. Nature of Matching

The matching operation gives more complexity to the structure of web service networks. In peer-to-peer networks, resources are described by simple text formats such as a file name or a vector of keywords. But resources in web service networks are services and are described by a complex description languages that make the service discovery and matching operations time-consuming. The description file of each web service is written by its provider. Each provider may use a specific description language. If providers provides semantic web services, they may also use their developed ontology. So, although two services may be similar in action, they may have a different description. This phenomena makes matching more difficult.

Property 3.1 (Matching Complexity): The matching complexity of WST is more complicated and time-consuming in contrast to P2P networks.

3.2. Number of Peers

In web service networks there exist three kind of nodes as shown in Figure 1. The number of *SC* peers is more than the number of *SP* peers, and the number of *SP* peers is more than the number of *SD* peers. The following property expresses this feature:

Property 3.2 (Number of Entities): In a typical WST network, number of *SC* nodes are more than number of *SP* nodes, which is more that number of *SD* nodes itself.

$$\left|\{SC\}\right| \gg \left|\{SP\}\right| \gg \left|\{SD\}\right| \qquad (1)$$

Furthermore, each provider published a limited number of services.

3.3. Machine's Powers

Service directories are usually implemented in a high performance machines in a high capacity networks. Service providers also utilized high performance machines to handle user requests and running web services. However, consumers usually run their programs in ordinary networks that have adapted to their company usage; and, compared to service directories and providers machines are weaker in processing power. This fact can be described as follows:

Property 3.3 (Powers of entities**)**: In a typical WST network, the power of *SD* machines are greater than the power of *SP* machines, which is grater that the power of *SC* machines.

$$\|\{SD\}\| \gg \|\{SP\}\| \gg \|\{SC\}\| \qquad (2)$$

3.4. Average Record Size of the Directory

Suppose that service directories stores service descriptions in order to be able to match or index them. This section investigates the size of records (service descriptions) in the service directory. For this reason some web services datasets such as OWLS-TC 4.0 [19] and WS-DREAM [20] are used. These two test cases contain real word web services descriptions selected from real service directories and contain a total of 4821 web service descriptions. The result of this investigation is shown in Figure 2. Some services are described by WSDL language and others are described by OWLS language for evaluation of semantic matching algorithms. As shown in Figure 2, the average size of each service description corresponding to the average record size of the service directory is 10995 byte.

Illustrating the number of occurrences of each record size in the registry offers better intuition. The investigation, as shown in Figure 3, indicates that more than 75 percent of records are less than 10KB in size and most of them are between 2KB to 5 KB. As a result, the record size of web service networks is in order of KB, which is bigger than the record size of file-sharing peer-to-peer networks.

Property 3.4 (Average Record Size): In Contrast to P2P networks, the service directories in WST networks have an average recode size that are significantly greater.

3.5. Nature of Network Traffic

This section investigates the messages and packet size that are exchanged through the web service networks. Another distinguished behavior of web service networks are the nature of traffic in them. Unlike peer-to-peer networks where traffic is mainly file transfer (thus making

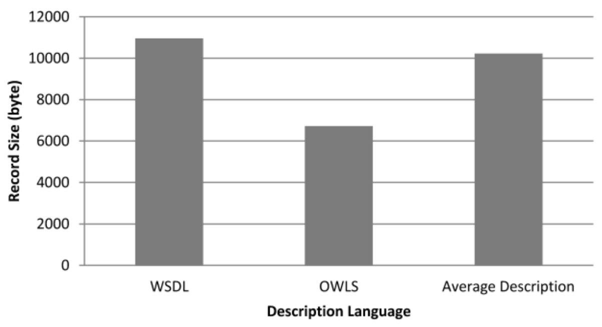


Figure 2. Average Size of Service Description in WSDL and OWLS Described Services

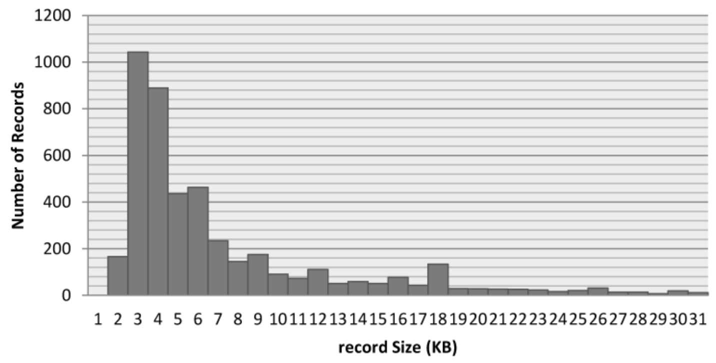


Figure 3. Number of Occurrences of Each Record Size

bandwidths very important), in web service networks, packets are mainly service descriptions that are sent and registered by service providers, user queries that search for a services, and finally execute requests that users send to service providers. The average size of these different packages is shown in Figure 4. As the figure shows, the maximum averaged packet size is 17.5 KB which occurs rarely and is composed of text characters. Even with compression, this can be reduced to 2.2 KB. But as the histogram of reply messages shows in Figure 5, most of the messages are below 4 KB in size.

Property 3.5 (Network Traffic Volume): In Contrast to P2P networks, service directories in WST networks have a significantly lighter traffic average.

3.6. Network Delay and Operation Cost

The response time of the service directory has been calculated previously in [1]. For this experiment, the work of Zheng [20] is used in which real word services invocations are monitored. They recorded invocation of more than 1.5 million services and registered the response time of each service call. As shown in [1], we have the following:

$$Rt = t_{Request} + t_{Reply} + t_{Processing} \tag{3}$$

where half of the *Rt* can be considered a good upper time of the average network delay which is 1100 ms.

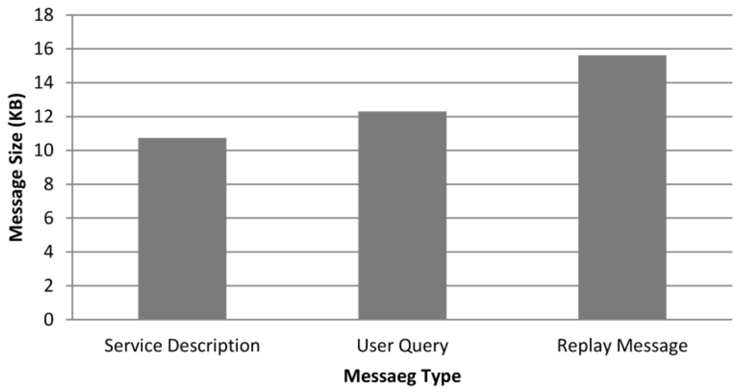


Figure 4. Average Size of Studied Message Typed in Web Service Networks

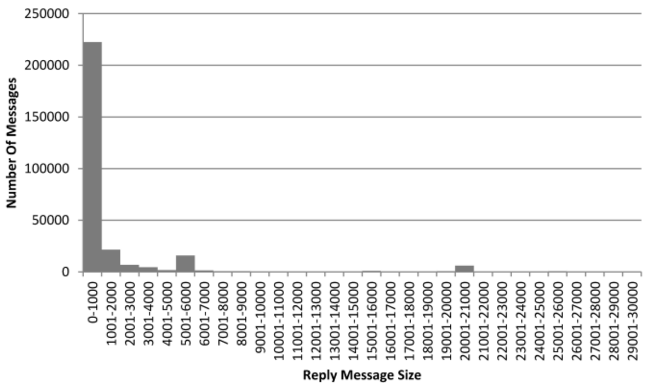


Figure 5. The Histogram of Reply Messages Size

By using the jUDDI implementation of UDDI download-able at *http://juddi.apache.org/* and equipping it with 2000 service descriptions, average service directory delay is calculated. The service descriptions are selected from WS-DREAM dataset. By averaging the response time of performing some query on this service directory plus the required time of performing exact matching, we reveal the average response time of the service directory (without considering the network delay because service directory was implemented locally) to be 66453 ms [1].

Property 3.6 (Source of Delay): Service discovery and matching is a time-consuming task that takes more of the processing power of the service directory.

3.7. Machines Availability

The churn rate of *SC* peers is very high compared with *SP* peers, and *SD* peers leave the system rarely (usually in case of failure or possible system upgrade). For calculating the availability of providers, *Seekda* crawler is used. *Seekda* is one the web service crawlers that searches the web continuously, registering each service that it finds. Furthermore, it stores some details about the services and their providers. *Seekda* also logs the availability of the registered services by checking their status' daily. At the time of writing this article, *Seekda* claimed that it monitors 28606 web services from 7739 providers.

Seekda calculates availability of the web services, not of the providers. However, this information can be used to calculate a lower bound of provider availability. Suppose that a provider sp_i has k services, $\{s_{i1}, \ldots, s_{ik}\}$, and $A(s_{ij})$ is the availability of *jth* service of the sp_i. Then a lower bound of the availability of sp_i based on availability of its services would be:

$$A(sp_i) = \arg\max_{1 \le j \le k}(A(s_{ij})) \tag{4}$$

This is because a provider may be alive while its services temporarily are out of reach.

A random sample of providers with more than one service was recruited from different countries. Then, the availability of service providers was calculated by equation 4. The result of this experiment showed that above 70 percent of the providers have an availability of more than 80%. So, we have the following property:

Property 3.7 (Machines Availability): The availability of providers and service directories are higher than peer-to-peer networks peers.

3.7. Motivation

Returning to the mentioned feature of web service networks in this section, it is now possible to state that the general structure of specific distributed frameworks such as peer-to-peer networks can not be directly applied to web service networks. In designing a structure for web service network, these features must be taken into account. The following section show how semantic links introduced in [1] can be used to interconnect services and machines. Following this, a structure for web service networks is proposed.

4. Semantic Interlinked Network of Web Service Entities and Service Descriptions

In [1] a graph of web service entities named *SIG* is used to empower service directories in order to offer more powerful operations. In fact, some semantic relations are defined between WST entities and a semantic graph of interlinked entities is constructed. An ontology named OWL-SD is presented to make it possible to formally define semantic relations and other necessities. Furthermore, RDF are used to interlink WST entities. By using the RDF technology, it is possible to interlink WST entities all over the world. Also, SPARQL are used to query *SIG* and use the knowledge stored in it. Abdali, et. al., [1] has shown how this semantic interlinked graph (*SIG*) can enhance WST operations.

In this paper we utilize only *isSimilarTo* and *isReacherThan* semantic relation among others to interlink services. Here, services are web service operations and machi nes. Other semantic relations are used to empower service directory operations and are not in the scope of this paper.

In this paper, the network nodes implemented with the findings of [1] are described in the following section.

4.1. Network Nodes

First, we categorized the types of nodes that are in the network as follows:

- Service directory nodes implemented by Service Directory Machines or *SDM*
- Service provider nodes implemented by Service Provider Machines or *SPM*
- Service consumer nodes implemented by Service Consumer Machines or *SCM*

Each *SDM* contains a web service description of the registered services in the network. Moreover, in order for creation, preservation and use of the *SIG*, some process is created in the service directory. Figure 6 shows the general model for a *SDM* as described in [1]. The component of *SDM* would be Ontologies, Monitoring Process, Inference Engine, SPARQL Endpoint and service descriptions DB which are explained in detail in [1].

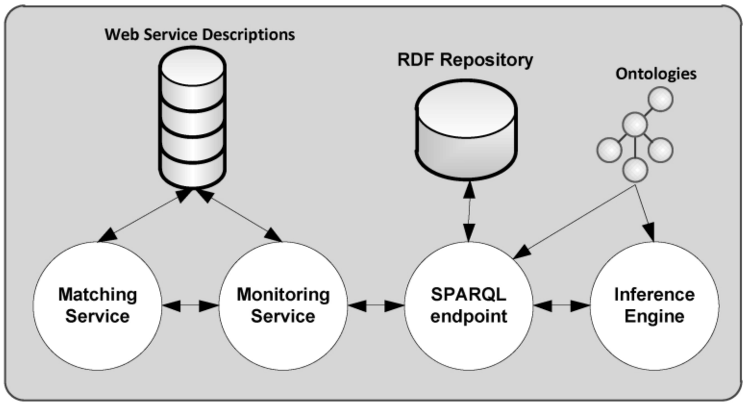


Figure 6. Model of a Service Directory Machine (*SDM*) as in [1]

SPM machines store the services and execute user queries. The model of *SPM* machines are depicted in Figure 7. Here, RDF *Repository* store the semantic relation between web services and between providers on the net if such relations exist. *Monitoring Service* monitors the availability of web services and other providers in the network where there exists a semantic relation between their web services to other machines. The *Ontology* is used to make RDF triples understandable by machine.

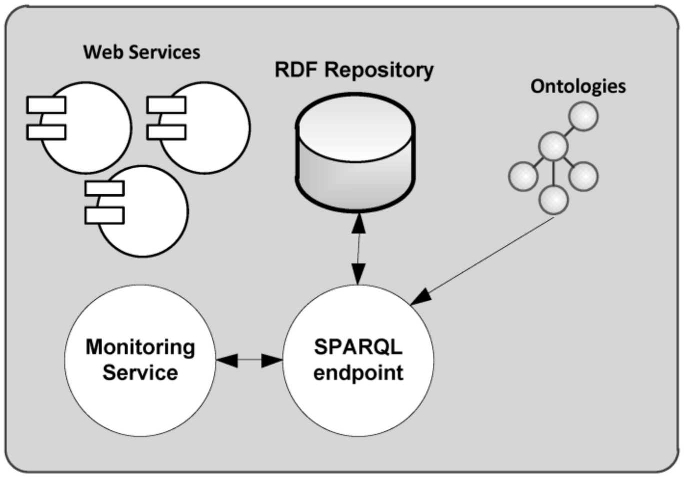


Figure 7. Model of a Service Provider Machine (*SPM*)

Finally, *SCM* machines are user machines. They may have any internal structure and use any tools necessary to handle their needs. They usually join the network, perform their computations, and leave the networks after that.

4.2. Proposed Topology

The presented architecture of Semantic Overly Network (*SON*) is a three layer structure. The core layer is a network of *SDM* machines calls Service Directories Layer. The middle layer is a network of *SPM* machines or Service Providers Layer, and the outer layer is the network on *SCM* machines or Service Consumers Layer.

SDM nodes are usually high performance machines that have been setup by some companies. Some works distribute directory service databases over the providers and remove the service directory machine [2]. Although the record size of service directories are not big (Property 3.4) and almost any provider can save all of the directory records, agreeing to a directory can reduce the performance of a provider dramatically without any special benefit. This is because matching is a time-consuming operation that may affect the quality of the provider and disturb its normal servicing (Property 3.6). Therefore, it is more beneficial for a provider to subscribe in a cloud of service directories and pay a charge for each binding of users that directories have. Hence, we tack into account powerful *SDM* machines (Property 3.3) to construct the core layer of the *SON*. *SDM* machines are interconnected and construct a *Gradiant Topology* network [2].

SPM machines construct the middle layer. Each *SPM* machine is connected to some other *SPM* machine to maintain this layer. The neighbors of a *SPM* machines are machines in which there exist at least one semantic relation between their services or service operations (section 4). So, *SPM* machines are interlinked based on semantic relations introduces in section 4.

SCM machines have high churn rates and are not contain a structured layer. High churn rate nodes impose high traffic and management cost. Consequently, this layer is unstructured and get its desired service from two other layers.

The general architecture of *SON* is shown in Figure 8. As shown in this figure, *SDM* machines constructs the Service Directories Layer on top of the network and *SPM* machines constructs the Service Providers Layer according to semantic relations. Here, only limited number of relations [1] are used and all the links in *SON* are indirection.

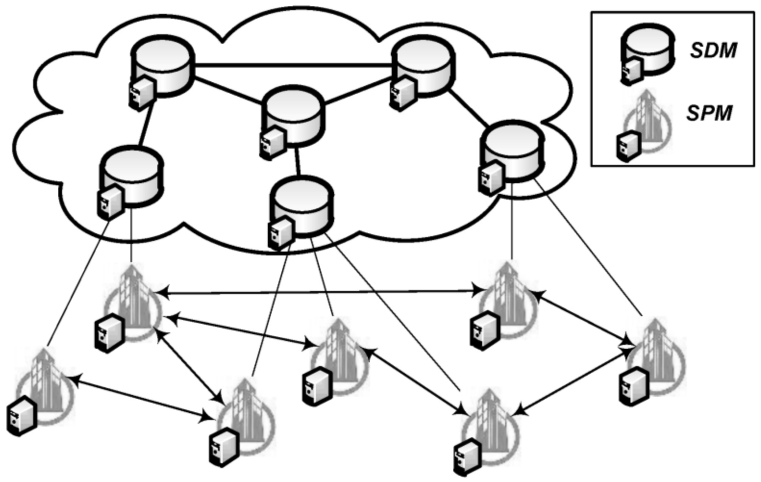


Figure 8. Architecture of Semantic Overly Network (*SON*)

As the result of property 3.4, all *SDM* machines are equipped with a complete copy of web service descriptions and RDF *Repository* as illustrated in Figure 6. There is no difference between *SDM* machines except their performance criteria; that is, the remaining number of requests it can handle. So, Unlike *Gradient Topology* that some nodes role as superpeers, here gradient topology is used only to give structure to the Service Directories Layer. Hence, there is no need to have a superpeer threshold or superpeer election algorithms. *SPM* machines see the whole Service Directories Layer as a peer. They recognize *SDM* machines by using a DNS bootstrap node and sending their requests to those nodes.

In the next part, the main operations of the structure are explained.

4.3. *SON* Operations

4.3.1. Bootstrap Mechanism: The DNS mechanism is used as bootstrap utility. The IP address of leaves in Service Directories Layer are stored in the DNS system. The leaves are machines which have no incoming connections from other *SDM* machines. Such nodes store their address in the DNS. Each *SDM* or *SCM* that joins the network or has a request, finds appropriate entry points in the Service Directories Layer by asking DNS servers.

4.3.2. *SDM* leave/join: A *SDM* can join the network by making a connection to a *SDM* machine that can be found using a DNS bootstrap. Based on the capabilities of this new *SDM* machine, it will find its correct place in the Service Directories Layer and begin its activities. If it becomes a leaf, it registers its address as a bootstrap node. This new *SDM* receives a replica of service descriptions and RDF repository from a stronger *SDM* machine (a machine that is currently attached to it). Because the churn rates of *SDM* machines are low, this process occurs rarely.

When a *SDM* machine leaves the network, the network updates its structure. If the leaving node is a leaf, the attaching machine removes it from the DNS. Otherwise, the disconnected nodes repair their connections by finding an entry point using DNS.

4.3.3. *SPM* leave/join: The *SPM* machine join the network by the following steps:

1. It selects one entry *SDM* using DNS bootstrap.
2. It publishes service descriptions to the *SDM* machine.
3. The *SDM* machine registers the service descriptions, updates, and informs other *SDM* machines.
4. The *SDM* machine finds possible semantic relations that exist between newly published web services with registered web services of other providers.
5. If there exist such relations, *SDM* sends the list of this relations to *SPM*, therefore, the *SPM* machine makes a connection with the other *SPM* machines funded by the *SDM*.
6. If there was no semantic relation funded, *SDM* offers some powerful *SPM* machines and the new *SPM* machine makes some connection to those machines.

While they are in the networks, *SPM* machines regularly check connections to ensure that its neighbors and interlinked web services are alive.

SPM machines may leave their networks for any reason. If this happens, its neighbors become aware of this event and report it to some *SDM*. The *SDM* machines then check the status of the failed provider and, in the case of problem repair the network connections. By this mechanism there is no need to waste the *SDM* machine's time to pulling the providers, and the network traffic will be reduced.

4.3.4. *SCM* join or Service Discovery: *SCM* machines join networks to find and execute web services. First, *SCM* finds an entry point to the Service Directories Layer using DNS bootstrap mechanism. Then, the *SCM* sends a request specification to a related *SDM* machine. The *SDM* machine then searches in its data base, trying to find the best possible matched web service to then send the corresponding *SPM* machine to the user. Users utilize this web service by sending requests directly to the *SPM* machine.

SDM machines use the semantic relations between services to speed up the search process. In order to search the service directory to discover the relevant services, there is no need to compare the service specification with all other services. By using the properties of relations, this operation can be done with lower cost as described in [1].

4.3.5. Fault Management: A fault can be due to three main reasons as follows:

1. An error occurred during the execution of a web service. In this case, *SPM* can re-execute the service without *SCM* awareness.
2. The web service encountered a fault that can not be recovered. In this case *SPM* finds a suitable similar web services by using *SON* links.
3. *SPM* may be down or there is a network failure. This phenomena is reported to the *SDM* machines either through other linked *SPM* machines or polling made by *SDM* and is reflected in them. Through this, the *SDM* machines temporarily remove the provider and search for other, similar services.

5. Evaluation

In this section, the proposed system has been experimented and evaluated. The response time of the system for service discovery and fault management has been investigated and compared to the Empowered Service Directory (*ESD*) proposed in [1]. For this reason, we augmented both systems with *SIG*, an interlinked graph of WST entities generated from OWLS-TC 4.0 dataset. Some parameters were defined to evaluate the proposed system and are shown in Table 1.

Table 1. Parameters of System Evaluation

Parameter	Meaning
N_S	The number of registered web services in the network
N_{SDM}	The number of *SDM* machines in the network
N_{SPM}	The number of *SPM* machines in the network
N_{SCM}	The number of *SCM* machines in the network
F_{SDM}	The fail rate of a *SDM* machines in the network
F_{SPM}	The fail rate of a *SPM* machines in the network
F_S	The fail rate of a web service

The value of N_S is equal to 1083 because there exist 1083 web services in the OWLS-TC dataset. For the sake of simplicity, we suppose that each *SCM* issues only one request. So, the number of *SCM* machines corresponds to the number of concurrent queries submitted in the network. We suppose that *SDM* are highly available and set their F_{SDM} equal to 1%. Based on the findings of section 3.7, we allocate a random number of availability for each web services in the [80%-100%] interval. The availability of *SPM* machines is calculated using Equation 4. Using calculated availabilities, the F_{SPM} and F_S values are calculated. Finally, Because each *SPM* machines run its services and their processing power does not affect the overall search response time, we set the number of *SPM* fixed and set N_{SPM} equal to 10. All web services are distributed among these *SPM* machines randomly according to a uniform distribution. To simulate running of web services, based on the finding of section 3.6, a random processing time in the interval of [500ms,1100ms] is assigned to each web service.

5.1. Average Query Turnaround Time

In this experiment, the turnaround time of a query execution from joining an *SCM* until leaving it is compared in *ESD* and proposed system. The main steps of a query execution are: using the bootstrap mechanism to join to a *SDM* machine, submitting the query to it, finding the appropriate service, connecting to the corresponding *SPM* machine, and finally executing discovered services and get results.

For this experiment, the number of requests(active *SCM* machines) are kept fixed during the experiment and set equal to 1000. First, the experiment start with a single *SDM* machine, and more *SDM* machines join the network at some point in time. Figure 9 shows the Average turnaround times of both systems. As shown, by increasing the number of *SDM*, the averaged turnaround time is reduced and becomes better.

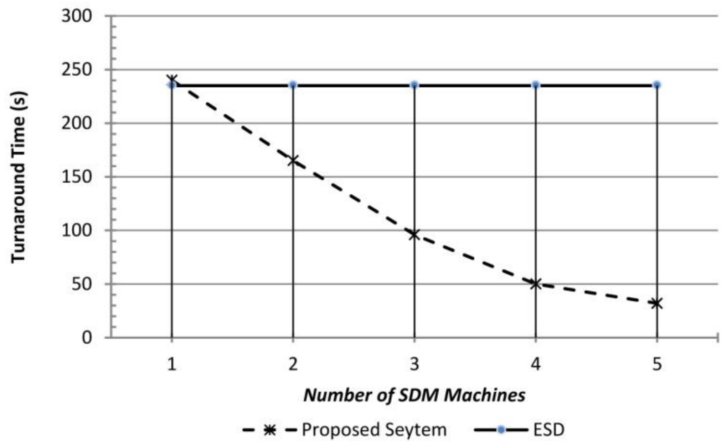


Figure 9. Comparison of Query Turnaround Time in Proposed System and *ESD*

The main reason for this phenomena is that the source of delay is service directory discovery operation. In fact, the network delay and execution time of a service is lower than the discovery process. So, the proposed system distributes the discovery operation among *SDM* machines, enhancing average turnaround time of the system.

5.2. Fault Management

In this experiment, the response time of the system in repairing faults is investigated. For this reason, the logs of failed services are extracted from the previous experiment. Then, the average time of fault recovery is calculated. All of the parameters are same as previous experiment. The result is depicted in Figure 10.

As the results show, finding a replacement in case of existence is more costly in traditional service directories. This is because in the proposed structure, the only action that is needed is to flow *isRicherThan* and *isSimilarTo* links while not performing any costly matching operation.

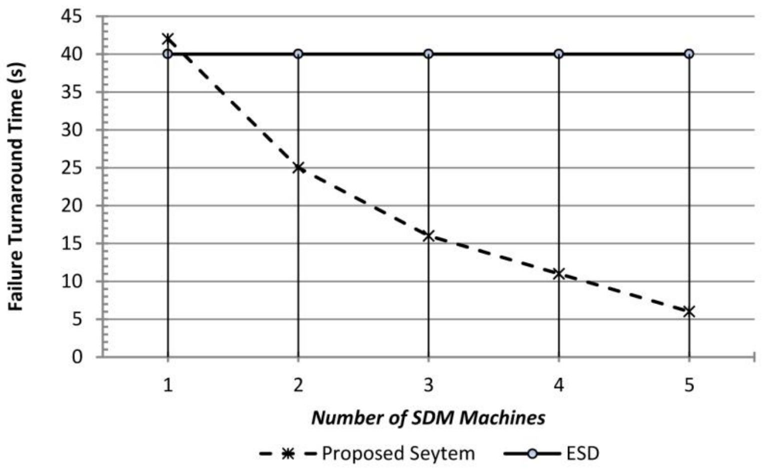


Figure 10. Comparison of the Time Required to Recover from a Failure

6. Conclusion

In this paper an overlying network for web service technology was constructed. First, important aspects of web service networks were explored by investigation of real word web services, providers, and directories. Based on the findings, a Semantic Overly Network for web service technology was presented and main operations are cleared. From a structural point of view, this overly network consisted of two main layers of network node types. In the proposed network, service directory machines are interlinked based on their performance using Gradient Topology and constructs a Service Directories Layer. Also, this interlinked service providers machines based on semantic relation between their web services. This paper used the semantic interlinked graph of entities of [1] that showed how it can be used to publish web services, use RDF to represent knowledge in it, and use SPARQL to query the knowledge. Some applications of the proposed architecture are investigated and the power of the structure to handle failures is shown. At the end, the effect of the structure in minimizing the discovery process is shown.

7. Future Works

It is recommended that further research be undertaken in the following areas:

1. A fault tolerance strategy in which \$SPM\$ machine use others processing powers.
2. Developing an accounting mechanism to make the framework usable in business based future Internet.

Although by distributing service directories, the average response time of the system can be improved in case of any failure; however failure recovery can also be enhanced by using other *SPM* machines processing power. For example, any two *SPM* machines, can keep a copy of other machines' data and services to run in case of failure or traffic overload. Another property of real Internet is that many resource and services are not free. Companies can execute a service with a small cost and service directory may need to be paid for each binding of a *SCM* to a *SPM*. Therefore, the need of an accounting mechanism seems to be necessary for a distributed WST network.

Acknowledgments

We appreciate Jan Sacha for his code in [2] which is the base of our overly network. The first author would like to thank Professor Karl Aberer and Dr. Zoltan Miklos, in the Distributed Information Systems Laboratory (LSIR), School of Computer and Communication Sciences (IC), École Polytechnique Fédérale de Lausanne (EPFL) for preparing an excellent research environment to accomplish parts of this work at LSIR.

References

[1] A. M. Fardin, N. B. Naser and N. M. Ali, “Empower service directories with knowledge”, Know.-Based Syst., vol. 30, (**2012**)**,** pp. 172-184.

[2] J. Sacha, B. Biskupski, D. Dahlem, et. al., “Decentralising a service-oriented architecture”, Peer-to-Peer Networking and Applications, vol. 3, no. 4, (**2009**) October, pp. 323-350.

[3] S. Sioutas, E. Sakkopoulos, C. Makris, et. al., “Dynamic Web Service discovery architecture based on a novel peer based overlay network”, Journal of Systems and Software, vol. 82, no. 5, (**2009**), pp. 809-824.

[4] L. Clement, A. Hately, C. v. Riegen, et. al., "UDDI Version 3.0.2", http://uddi.org/pubs/uddi-v3.0.2-20041019.htm, **(2012)** July.

[5] F. Najmi and N. Stojanovic. "OASIS ebXML RegRep Service and Protocols (ebRS) Version 4.0", http://docs.oasis-open.org/regrep/regrep-core/v4.0/csd01/regrep-core-rs-v4.0-csd01.html, (**2012**) July.

[6] Y. Wang and J. Yang, "Relation Based Service Networks for Reliable Service Selection", (**2009**) Vienna, Austria, pp. 209-214.

[7] H. N. Talantikite, D. Aissani and N. Boudjlida, “Semantic annotations for web services discovery and composition”, Computer Standards & Interfaces, vol. 31, no. 6, **(2009)**, pp. 1108-1117.

[8] S. Chen, Z. Feng, H. Wang, et. al., "Building the Semantic Relations-Based Web Services Registry through Services Mining", (**2009**), pp. 736-743.

[9] Z. Maamar, L. K. Wives, Y. Badr, et. al., “LinkedWS: A novel Web services discovery model based on the Metaphor of "social networks",” Simulation Modelling Practice and Theory, vol. 19, no. 1, (**2011**), pp. 121-132.

[10] Z. Maamar, N. Faci, L. Krug Wives, et. al., "Towards a Method for Engineering Social Web Services Engineering Methods in the Service-Oriented Context", IFIP Advances in Information and Communication Technology J. Ralyté, I. Mirbel and R. Deneckère, eds., Springer Boston, (**2011**), pp. 153-167.

[11] M. Sellami, S. Tata, Z. Maamar, et. al., "A Recommender System for Web Services Discovery in a Distributed Registry Environment", (**2009**) Venice/Mestre, Italy, pp. 418-423.

[12] K. Aberer, “Peer-to-Peer Data Management”, Peer-to-Peer Data Management, (**2011**).

[13] S. Staab and H. Stuckenschmidt, “Semantic Web and Peer-to-peer: decentralized management and exchange of knowledge and information”, (**2006**).

[14] J. Sacha, "Exploiting Heterogeneity in Peer-to-Peer Systems Using Gradient Topologies", Dept. of Computer Science, Trinity College Dublin, (**2009**).

[15] J. Liu and H. Zhuge, “A semantic-link-based infrastructure for web service discovery in P2P networks”, in Special interest tracks and posters of the 14th international conference on World Wide Web, (**2005**), pp. 940-941.

[16] S. Sioutas, E. Sakkopoulos, L. Drossos, et. al., “Balanced distributed web service lookup system”, J. Netw. Comput. Appl., vol. 31, (**2008**) April , pp. 149-162.

[17] K. Verma, K. Sivashanmugam, A. Sheth, et. al., “METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services”, Inf. Technol. and Management, vol. 6, (**2005**) January, pp. 17-39.

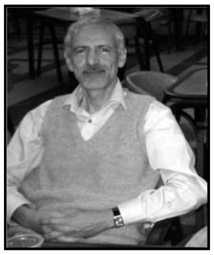
[18] J. M. Rodriguez, M. Crasso, A. Zunino, et. al., “Improving Web Service descriptions for effective service discovery,” Sci. Comput. Program., vol. 75, (**2010**) November, pp. 1001-1021.

[19] K. Matthias , F. Benedikt and S. Katia, “OWLS-MX: A hybrid SemanticWeb service matchmaker for OWL-S services”, Web Semantics: Science, Services and Agents on theWorldWideWeb, vol. 7, (**2009**), pp. 121–133.

[20] Z. Zheng and M. R. Lyu, “Collaborative Reliability Prediction for Service-Oriented Systems”, in Proc. IEEE/ACM 32nd Int'l Conf. Software Engineering (ICSE'10), (**2010**), pp. 35-44.

Authors

Fardin Abdali Mohammadi is a PhD candidate of computer engineering at the Faculty of Engineering of the University of Isfahan (UI). He earned his M.Sc. and B.Sc. degrees from the Sharif University of Technology and Amirkabir University of Technology, respectively. His research interests are Semantic Web, Web Service Technology, Bioinformatics, and Artificial Intelligence. He writes two academic books that are used as course materials in some universities in Iran. He was lecturer at Lorestan University for three years. Also, he cooperate with some universities in Iran.



Naser Nematbakhsh received his B.Sc. degree of Science in Mathematics from Isfahan University, Iran in 1973 and Master of Science in Computer Science in 1978 from Worcester Polytechnic Institute, USA. He received the Ph.D. in Computer Engineering from University of Bradford, England in 1989. working as Assistant Professor in the Department of Computer Engineering, Isfahan University. His experienced areas include Software Engineering Methods, Service Oriented Computing and Software Reliability.

Mohammad Ali Nematbakhsh is an associate professor of computer engineering in the School of Engineering at the University of Isfahan. He received his BSc in electrical engineering from Louisiana Tech University in 1981 and his MSc and PhD degrees in electrical and computer engineering from the University of Arizona in 1983 and 1987, respectively. He had worked for Micro Advanced Co. and Toshiba Corporation for many years before joining The University of Isfahan. He has published more than 80 research papers, three U.S. registered patents and a database book that is widely used in universities. His main research interests include Semantics Web and Computer Networks Application.